



HAL
open science

Actes Groupe de Travail Animation et Simulation (GTAS'07)

Florence Zara, B. Shariat

► **To cite this version:**

Florence Zara, B. Shariat. Actes Groupe de Travail Animation et Simulation (GTAS'07) . Florence Zara; Behzad Shariat. Groupe de Travail Animation et Simulation 2007 de l'AFIG (Association Française d'Informatique Graphique)., Jun 2007, Lyon, France. 2007. hal-01493226

HAL Id: hal-01493226

<https://hal.science/hal-01493226v1>

Submitted on 27 Mar 2017

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.

GTAS'07

Lyon, 25-26 juin 2007



**Actes des 14^{èmes} journées du
Groupe de Travail Animation et Simulation
de l'Association Française d'Informatique Graphique**

Actes

Groupe de Travail Animation et Simulation (GTAS'07)

Sous la direction de F. Zara, B. Shariat
(LIRIS, Université Lyon 1)

Association Française d'Informatique Graphique (AFIG)

Journées organisées et parrainées par :

Université Claude Bernard Lyon 1

Laboratoire d'InfoRmatique en Image et Systèmes d'information (LIRIS)

Équipe LIRIS-SAARA (Simulation, Analyse et Animation pour la Réalité Augmentée)

Association Française d'Informatique Graphique (AFiG)

Centre ETOILE

GdR Informatique Graphique (IG)

GdR STIC-Santé

EuroGraphics France



Organisateurs des journées :



Florence Zara
Pierre-Frédéric Villard (site web)
Behzad Shariat
Anne-Laure Didier
Alexis Guillaume
Sylvie Oudot (secrétariat)

Équipe LIRIS-SAARA
(Simulation, Analyse et Animation pour la Réalité Augmentée)

Table des matières

| | |
|--|-----------|
| Session 1 - Session Invités | 3 |
| <hr/> | |
| SOFA - An Open Source Framework for Medical Simulation FRANÇOIS FAURE | 5 |
| Projet européen ETOILE JOSEPH REMILLEUX | 7 |
| Session 2 - Session Médicale | 13 |
| <hr/> | |
| Modélisation des tissus mous dans leur environnement : un modèle discret pour l'aide aux gestes médico-chirurgicaux M. MARCHAL, E. PROMAYON, J. TROCCAZ | 15 |
| Résultats préliminaires et perspectives de la modélisation dynamique du pelvis féminin M.-E. BELLEMARE, N. PIRRÓ, O. DURIEUX, L. MARSAC, I. SIELEZNEFF, B. SASTRE | 17 |
| Breathing Thorax Simulation based on Pleura Physiology and Rib Kinematics A.-L. DIDIER, P.-F. VILLARD, J.-Y. BAYLE, M. BEUVE, B. SHARIAT | 29 |
| Suivi de mouvements respiratoires : survol et nouvelle approche B. VIGUIER, E. DESSERÉE, J.-M. MOREAU | 35 |
| Real-time and Markerless Full-Body Human Motion Capture B. MICHOD, E. GUILLOU, S. BOUAKAZ | 39 |
| La méthode HEML pour le calcul rapide et réaliste des déformations de matériaux hyper-élastiques en simulation de chirurgie F. GOULETTE | 51 |
| Squelettisation robuste de données volumiques pour la caractérisation et la quantification d'anévrismes S. HASSAN | 53 |
| Management de cancer à distance M. NAZEK, J.-P. AURAY | 61 |
| Simulation de radiologie interventionnelle S. COTTIN, J. DEQUIDT | 65 |
| Session 3 - Session Animation | 67 |
| <hr/> | |
| A New Step for Artificial Creatures N. LASSABE, H. LUGA, Y. DUTHEN | 69 |

| | |
|--|-----|
| On-line physically-base control of mass-interaction models | |
| A. ALLAOU, M. EVRARD, A. LUCIANI | 77 |
| Vers une uniformisation des animations de marche | |
| J. SALUT, C. SANZA, J.-P. JESSEL | 87 |
| Pedestrian Reactive Navigation for Crowd Simulation : a Predictive Approach | |
| S. PARIS, J. PETTRÉ, S. DONIKIAN | 93 |
| Animation de données volumiques à une résolution arbitraire utilisant des raideurs non-uniformes | |
| M. NESME, F. FAURE, Y. PAYAN | 103 |
| Implantation temps réel sur GPU d'un écran d'épingles dynamique | |
| K. SILLAM, M. EVRARD | 105 |

Session 1 - Session Invités

François Faure
(EVASION, INRIA Rhône-Alpes, Grenoble)

SOFA - An Open Source Framework for Medical Simulation

SOFA is a new open source framework primarily targeted at medical simulation research. Based on an advanced software architecture, it allows to (1)~create complex and evolving simulations by combining new algorithms with algorithms already included in SOFA; (2) modify most parameters of the simulation deformable behavior, surface representation, solver, constraints, collision algorithm, etc. by simply editing an XML file; (3) build complex models from simpler ones using a scene-graph description; (4) efficiently simulate the dynamics of interacting objects using abstract equation solvers; and (5) reuse and easily compare a variety of available methods. In this paper we highlight the key concepts of the SOFA architecture and illustrate its potential through a series of examples.

Joseph Remilleux
(Institut de Physique Nucléaire, Lyon)

Projet européen ETOILE

Introduction

L'élaboration du projet ETOILE a permis depuis 1999 de faire travailler ensemble, sur des programmes de recherche, des médecins, chercheurs et ingénieurs de disciplines aussi différentes que la radiothérapie, la radiobiologie, l'anatomie, l'économie de la santé, l'informatique, l'imagerie, la physique nucléaire et la physique des particules.

Ces équipes pluridisciplinaires constituent désormais un pôle important de recherche, d'innovation et de valorisation industrielle, avec des objectifs de recherche bien définis et avec encadrement de thèses et publications. Les programmes ont été initialement structurés par le réseau européen ENLIGHT. Ce pôle bénéficie du soutien humain et financier des unités de recherches régionales les plus concernées (unités rattachées aux universités de Lyon ou de Grenoble et associées soit au CNRS, soit à l'INSERM, soit à des services hospitaliers), de la création de trois postes d'enseignants chercheurs, et surtout du Contrat de Plan Etat/Région Rhône-Alpes "Hadronthérapie" qui a permis, depuis 2000, le financement des programmes de recherche sur trois volets : salaires (post-doctorants et ingénieurs), crédits d'équipements et crédits de fonctionnement. Il faut remarquer que les équipes ont de plus en plus de collaborations industrielles et des projets de valorisation (sources de création d'emplois), se tenant prêtes à accompagner la réalisation du centre ETOILE. D'importantes collaborations se développent entre les équipes de recherche Rhône-Alpes et celles de Caen et Paris, notamment en radiobiologie et en imagerie.

Une version nationale du projet de recherches est en cours d'élaboration et sera bientôt disponible.

Les thèmes

Les recherches sont regroupées en deux thèmes principaux d'expérimentation et de modélisation : la radiobiologie et l'imagerie. L'intégration des résultats scientifiques de ces deux thèmes, ainsi que leur transfert vers l'industrie, sont assurés par un troisième thème qui prépare la planimétrie des traitements en collaboration étroite avec les hôpitaux et les industriels.

Recherches en imagerie et modélisation

Ce programme concerne l'ensemble de la radiothérapie, seul le programme hadron-TEP est spécifiquement dédié à la thérapie par ions carbone.

Contexte et problématique

Le challenge du traitement par ions légers, comme celui des autres thérapies du cancer par rayonnements ionisants (radiothérapie, protonthérapie, curiethérapie) consiste à délivrer le maximum de dose à la tumeur tout en épargnant au mieux les tissus sains environnants. Ce principe est à la base de chacune des étapes de la thérapie et conditionne la qualité du traitement.

L'évolution de ces thérapies passe aujourd'hui par une utilisation de plus en plus importante d'images numériques biomédicales provenant de multiples modalités (CT, TEP, IRM...). Ces images sont utilisées avant, pendant et après chaque séance de traitement :

- avant : pour l'élaboration du plan de traitement (calcul de la balistique, détermination et délimitation des cibles),
- pendant : pour le contrôle du positionnement, le suivi des mouvements,
- après : pour le suivi a posteriori et l'adaptation éventuelle du plan.

L'objectif de ce programme est de conduire une recherche sur l'imagerie au service des thérapies par rayonnement ionisants pour optimiser le traitement en diminuant les incertitudes liées à la balistique et à la dosimétrie. Cela concerne les étapes suivantes :

- recalage, fusion et segmentation d'images de multiples modalités (TEP, CT, IRM, US...) pour le plan de traitement,
- contrôle du positionnement du patient,
- contrôle dosimétrique avec la TEP en ligne,
- prise en compte des tissus mous déformables (prostate, foie) ainsi que des localisations mobiles (mouvements respiratoires),
- adaptation du traitement au patient, à son évolution anatomique au cours des séances (amaigrissement, évolution rapide de la tumeur, ...).

Le programme est conduit à Lyon par 3 UMR (LIRIS, CREATIS et IPNL) et coordonné par le département de radiothérapie du Centre Léon Bérard. Cette communauté de médecins et chercheurs collabore depuis plusieurs années avec des équipes médicales belges (VUB), anglaises (Christie Hospital, Hull Hospital), allemandes (DKFZ), notamment dans le cadre du groupe de travail CONSENS, et plus récemment américaines (MGH à Boston). Les collaborations au niveau de l'instrumentation concernent principalement le CEA (Paris) et le CERN mais aussi le GANIL (Caen) et GSI (Darmstadt).

Contrôle de positionnement

L'objectif est d'automatiser le contrôle de positionnement à l'aide d'algorithmes de mise en correspondance (recalage) des images de contrôle avec des images de référence. Les images de contrôle sont : des images portales (Mega-Volt), des images radiographiques kilo-Volt (2D, 2D+t ou 3D), des vidéos ... Les images de références sont des images 3D CT, IRM et TEP, ainsi que les modélisations 4D associées. D'autre part, il faut évaluer, à partir des images portales, la dose reçue au cours d'une séance à l'aide de simulations. Enfin, le contrôle de traitement passe

également par l'utilisation de nouveaux imageurs 3D in-situ (Cone-Beam, le "Synergy" d'Elekta par exemple).

Ce programme se développe à Lyon au CLB en collaboration avec les UMR LIRIS et CREATIS.

Acquisitions, modélisation et simulation de la respiration

L'objectif est l'estimation et la modélisation des déformations du thorax lors de la respiration afin de contrôler la dosimétrie dans l'espace et le temps. Le principe de base est, dans un premier temps, d'asservir la respiration à l'aide d'un système, nommé ABC (Active Breath Control) permettant de bloquer la respiration du patient à un niveau déterminé, de façon à limiter les mouvements. On modélise ainsi la déformation du poumon à partir d'examens scanners thoraciques pris chez le même patient à des temps respiratoires différents. Des méthodes de recalage non-rigide sont développées permettant d'obtenir des champs 3D denses de vecteurs de déformation. Ces champs servent de bases à des modélisations géométriques (par éléments finis ou à l'aide de modèle masse-ressort) incluant des contraintes physiologiques. Des études en collaboration avec des médecins pneumologues servent à la validation des modèles.

Ce programme est placé sous la responsabilité du LIRIS-Lyon et du CLB-Lyon.

Dosimétrie dynamique et adaptative

Ce programme concerne l'étude de méthodes de calcul de la dosimétrie qui prennent en compte l'ensemble des informations relatives à la dynamique des organes (échelle de temps de l'ordre de la seconde). Un traitement adaptatif et en cours d'étude afin de corriger *a posteriori* la dosimétrie des mouvements internes enregistrés pendant la séance de traitement (échelle de temps de l'ordre du jour). Une première étape est le calcul de dose dynamique, tenant compte d'un mouvement du patient. L'objectif est d'adapter les marges au plus près des mouvements prévus, de façon personnalisée à chaque patient. À plus long terme, une adaptation dynamique de l'irradiation sera développée en fonction des mouvements observés à l'aide de différents capteurs (vidéo, radiographies dynamiques, spiromètres ...), corrélés à une modélisation 4D préalable.

Ce programme implique à Lyon le CLB en collaboration avec les trois UMR (LIRIS, CREATIS, IPNL).

Instrumentation TEP

Il s'agit de développer au sein de l'IPNL, sous la responsabilité de M. Boutemour, un prototype de caméra "hadronTEP" destinée à l'imagerie et à la dosimétrie par tomographie par émission de positons (TEP).

En effet l'imagerie et la micro-dosimétrie sont indispensables pour l'hadronthérapie et devront se faire en temps réel pendant les séances de traitement. Les caméras TEP actuelles sont destinées principalement au diagnostique chez l'homme, ou encore aux études bio-médicales sur le petit animal. Dans ces deux cas l'isotope émetteur

de positons est injecté à forte dose. Dans l'hadronthérapie les émetteurs sont les fragments des noyaux cibles et projectiles : leurs concentrations sont très faibles et ils décroissent rapidement avec le temps.

Une caméra TEP pour l'hadronthérapie devra donc avoir une sensibilité considérablement augmentée par rapport aux caméras actuelles (le seul prototype existant est celui de GSI mais il ne couvre qu'un angle solide de 9% de 4π avec une sensibilité ne dépassant pas 2,5 %).

Une caméra TEP *in-situ* et travaillant en temps réel constitue la solution idéale pour l'hadronthérapie. La distribution des émetteurs β^+ n'est malheureusement pas proportionnelle à la dose locale déposée. Cela rend non triviale l'extraction des paramètres d'importance clinique comme la portée des particules, la position latérale du champ d'irradiation et la dose locale réellement distribuée. Un calcul précis des distributions des émetteurs β^+ à partir du plan de traitement est nécessaire. Le calcul de ces distributions devra passer par une modélisation réaliste de tous les processus conduisant à un signal TEP. La fluence des particules est déterminée par le plan de traitement et elle devra être contrôlée namiquement par retour d'information de la TEP.

La comparaison en temps réel des distributions β^+ calculées avec celles mesurées par TEP sera à la base du contrôle dynamique du traitement et celui des doses distribuées. Finalement les images par TEP devront être superposées sur des images CT pour obtenir une corrélation avec l'information anatomique.

Le projet consiste donc à concevoir, simuler et réaliser un prototype de TEP évoluée à partir :

- des technologies développées et des résultats obtenus pour la TEP petit animal,
- des simulations déjà réalisées,
- des expérimentations avec des cibles appropriées et un faisceau d'ions carbone incident.

Le LSO est utilisé comme milieu scintillateur et deux configurations (en matrice et en bloc) sont en cours de test. Pour la chaîne d'acquisition du système, la lecture de la charge en sortie de chaque cellule du photomultiplicateur s'effectue à l'aide d'une électronique frontale auto-déclenchée. Une puce analogique est en cours de développement. Elle permettra de réduire à zéro le temps mort et de corriger l'hétérogénéité des gains des photomultiplicateurs multi-anodes. Une version R&D de cette même puce analogique est spécialement réalisée pour la lecture des APD (photo-diodes à avalanche). Ces dernières offrent une meilleure couverture de détection à coût plus faible que les photomultiplicateurs et peuvent fonctionner dans un environnement magnétique. Cependant, leur faible gain électronique nécessite le développement d'une électronique d'amplification bas bruit et stable. Cette électronique bas bruit permettra, dans le futur, de remplacer les photomultiplicateurs par des APD.

La construction du prototype par l'assemblage de têtes modulaires est en cours de tests sous faisceau à Caen (GANIL) et à Darmstadt (GSI). Les données brutes issues du système

d'acquisition seront intégrées dans un format permettant de stocker les informations concernant la machine, l'acquisition, les types de données et tous les paramètres nécessaires au traitement. La reconstruction des images devra faire appel à de nouvelles méthodes analytiques compte tenu de la géométrie particulière du détecteur.

A Lyon, le projet "hadronTEP" s'appuie sur le savoir-faire de 4 UMR (IPNL, LPCML, CREATIS, LIRIS) et du CERMEP, et sur l'expérience clinique des médecins du Centre Léon Bérard (CLB) et des CHU de Lyon et Grenoble. La complémentarité entre ces laboratoires permet d'assurer le développement du projet, depuis la chaîne d'acquisition jusqu'aux logiciels et l'exploitation clinique du système.

Ce programme se déroule en collaboration avec le LPSC-Grenoble et le DAPNIA-Saclay.

Traitement des images

Le savoir-faire du CREATIS à Lyon (en collaboration avec le CLB, le LIRIS et l'IPNL) est mis à contribution pour résoudre les nombreux problèmes de traitement des images intervenant dans le plan de traitement de l'hadronthérapie.

Il s'agira de gérer l'imagerie multi-modalités en assurant les recalages et la fusion des images. Les problèmes de segmentation seront traités notamment pour aider la délimitation des organes à risques.

Dans le cas particulier de l'hadron-TEP dédié il s'agira de générer les algorithmes de reconstruction et de segmentation TEP en ligne. La multidisciplinarité physique-informatique est au coeur du projet. L'objectif à plus long terme est d'intégrer et de combiner les dernières avancées en analyse et traitement d'images (recalage déformable 3D...) avec les logiciels les plus à la pointe de simulation physique (Geant4, Gate ...).

Une importante partie des recherches nécessite des développements informatiques haut-niveau (Unix, C++, programmation orientée objet, ...). Ces développements devront être assurés par un ingénieur de recherche afin d'assurer la pérennité, le développement et la maintenance des codes informatiques des chercheurs et d'aider au développement des simulations de physique.

Session 2 - Session Médicale

Titre :

Modélisation des tissus mous dans leur environnement : un modèle discret pour l'aide aux gestes médico-chirurgicaux

Auteurs :

Maud Marchal, Emmanuel Promayon, Jocelyne Troccaz
Laboratoire TIMC, Equipe GMCAO
Institut d'Ingénierie de l'Information de Santé
38706 LA TRONCHE Cedex

Résumé :

Dans le cadre de procédures médicales ou chirurgicales minimalement invasives, l'assistance aux cliniciens pour la planification et la réalisation de leurs gestes joue un rôle important dans la réussite des interventions. Dans ce contexte, la modélisation et la simulation de structures anatomiques de manière précise et réaliste représentent des outils intéressants pour la réalisation de plannings chirurgicaux. Nous proposons une méthode de modélisation permettant d'obtenir des simulations qui prennent en compte à la fois l'environnement naturel d'un organe donné et l'influence des instruments chirurgicaux. Le travail de modélisation est appliqué à la simulation de procédures médicales pour le diagnostic et le traitement du cancer de la prostate (biopsies et curiethérapies). Le modèle développé est un modèle physique discret qui permet notamment de simuler le comportement de tissus mous en interaction avec d'autres structures anatomiques. Une formulation basée sur un principe de mémoire de forme locale est proposée et permet de modéliser l'élasticité d'un matériau grâce à une mesure géométrique de la déformation locale. Une modélisation de l'insertion d'aiguilles dans des tissus mous complète le modèle de l'environnement anatomique. Afin de valider le comportement de la méthode de modélisation développée, une démarche expérimentale est proposée pour comparer les simulations obtenues avec d'une part différentes méthodes de modélisation des tissus mous telles que la méthode des éléments finis ou la méthode des masses-ressorts et d'autre part des données réelles provenant de fantômes et d'images médicales. La méthode a ensuite été appliquée à la simulation des deux procédures médicales étudiées afin de modéliser dynamiquement le comportement de la prostate dans son environnement anatomique et médical.

Résultats préliminaires et perspectives de la modélisation dynamique du pelvis féminin

M.-E. Bellemare, N. Pirró, O. Durieux, L. Marsac, I. Sielezneff, B. Sastre

Résumé

Les simulateurs chirurgicaux ont connu un développement important notamment depuis l'essor de la chirurgie cœlioscopique. Les principaux simulateurs de chirurgie digestive ont été développés pour les organes fermes tels que le foie et la rate. Les travaux portant sur les tissus mous et sur les organes pelviens sont rares. Le but de ce travail est de créer un modèle dynamique du pelvis féminin qui servira de base au développement d'un simulateur de chirurgie pelvienne patient-spécifique.

Un modèle spécifique, statique et dynamique du pelvis féminin est en effet réalisé à partir d'une imagerie volumique IRM et d'une modélisation physique des organes fondée sur des systèmes masses-ressorts.

Le modèle résultant est la base de l'élaboration d'un simulateur de chirurgie pelvienne patient-spécifique qui pourra être un outil pour la planification du geste chirurgical. Il s'intégrera alors dans l'ensemble des outils visant à améliorer la prise en charge des malades ayant un trouble de la statique pelvienne en permettant une évaluation fonctionnelle préopératoire du traitement chirurgical envisagé.

I. Introduction

Les principaux simulateurs de chirurgie digestive ont été développés pour des organes fermes comme le foie et la rate [1,2,3]. Les études relatives aux tissus mous comme les organes pelviens restent rares [4]. L'objectif principal de cette étude est la réalisation d'un modèle du pelvis féminin, la mise en exergue des contraintes et des difficultés de cette modélisation. La validation d'un modèle physiologique est la première étape avant l'élaboration d'un modèle développé à partir de malades présentant des troubles de la statique pelvienne. Les troubles représentés par un tel modèle pathologique pourraient être corrigés en simulant les modifications proposées par la chirurgie envisagée. Il serait ainsi possible de réaliser une évaluation fonctionnelle du résultat de la correction chirurgicale avant l'intervention.

Cet article présente une première étape vers l'objectif de la simulation du comportement des organes pelviens sous les contraintes de poussée. À partir d'une acquisition IRM, un modèle anatomique de la patiente est réalisé puis ce modèle est associé à un modèle physique qui permettra la simulation physiologique.

II. Intérêt clinique

L'imagerie tridimensionnelle des structures pelviennes a eu un développement relativement récent. Cette imagerie est utilisée pour étudier les rapports entre les organes pelviens au repos et lors des efforts [5], pour quantifier la masse musculaire du plancher pelvien [5-7] et pour identifier les modifications morphologiques des malades ayant un prolapsus pelvien [8], [9]. Le développement de modèles dynamiques animés en vue de simuler les déplacements des organes pelviens pourrait permettre une analyse dynamique de leur morphologie et de leurs rapports pendant l'effort de poussée. Cette imagerie tridimensionnelle permet une visualisation des différents organes. La modélisation statique et dynamique permettrait en plus une meilleure compréhension de la physiopathologie des troubles de la statique pelvienne. En effet, bien que des recherches significatives aient été menées [10], [11], la physiologie pelvipérinéale et les bases anatomiques des troubles de la statique sont mal connues et leur prise en charge reste un sujet de controverse [12] [13]. De plus, la simulation des modifications liées à l'âge (modification de la

courbure lombo-sacr e, descente p rin ale) ou la simulation des cons quences d'interventions chirurgicales (hysterectomie, correction de cystoc les...), permettrait l'identification des patientes qui pr sentent un risque  lev  des troubles de la statique pelvienne.

La prise en charge des malades ayant un trouble de la statique pelvienne repose sur la r ducation p rin ale, sur les traitements m dicamenteux ou sur de nombreux traitements chirurgicaux. Ces diff rents traitements pouvant  ventuellement  tre associ s. Toutefois, les r cidives sont fr quentes. Elles sont en partie dues   la difficult    mettre en rapport une symptomatologie avec un trouble anatomique et   une prise en charge m dicochirurgicale non optimale.

Diff rents mod les ont  t  utilis s pour la formation des praticiens et des autres professionnels m dicaux [15]. Cependant ces mod les sont en g n ral de simples simulations interactives permettant la manipulation de corps plut t rigides. Les simulateurs chirurgicaux mettant en sc ne des interventions de chirurgie co lioscopique sont peu nombreux du fait de la difficult  de la mod lisation des tissus mous. Les b n fices procur s par l'enseignement et l'entra nement   l'aide de simulateurs sont nombreux [16]. Les simulateurs de r alit  virtuelle permettent une  valuation objective de la performance psychomotrice et augmentent la dext rit  chirurgicale [17-19]. Ils autorisent la r p tition de t ches de base et fournissent des mesures objectives de la performance du geste co lioscopique. Le d veloppement de simulateurs patients sp cifiques permettrait la pratique r p t e de gestes complexes prenant en compte les variations anatomiques et les caract ristiques physiologiques sp cifiques de chaque malade. Cette simulation patient sp cifique n'est pas r alisable avec les simulateurs actuels.

L'objectif principal de cette  tude est de montrer la faisabilit  d'un simulateur patient sp cifique et de mettre en exergue les difficult s de la r alisation des organes pelviens. La simulation d velopp e concerne principalement les mouvements physiologiques des organes de la r gion pelvienne. En permettant une meilleure compr hension des pathologies, le simulateur pourrait servir des objectifs p dagogiques et au-del , dans la salle d'op ration, permettre au chirurgien de planifier des proc dures sp cifiques.

III. Construction du mod le g om trique

A. La segmentation

Le mod le g om trique est construit   partir d'une acquisition IRM d'une patiente,   ce titre il peut  tre consid r  patient-sp cifique. Cette modalit  est ici utilis e car elle est particuli rement discriminante pour les tissus mous. De plus, elle fournit   la fois une image volumique de la r gion d'int r t et une s quence dynamique acquise dans le plan m dian sagittal. La segmentation automatique des volumes IRM est encore un probl me ouvert dans un contexte d'organes de formes libres   faible contraste. Il est alors propos  d'utiliser une m thode semi-automatique qui implique l'op rateur dans le processus. L'outil de segmentation utilis  est bas  sur un logiciel libre et gratuit, ITK-SNAP[20], qui a  t  am lior  pour r pondre aux besoins sp cifiques des donn es disponibles. La segmentation repose sur une m thode de contours actifs impl ment s avec des "Level-Sets" [21]. Le contour actif est une surface $C(u, v, t)$ dont l' volution est pilot e par l' quation diff rentielle (1).

$$\frac{\partial C}{\partial t}(u, v, t) = F \cdot \vec{N} \quad (1)$$

o  $F = \alpha \eta + \beta \kappa$: $\beta \kappa$ (La force interne) est li e   la courbure gaussienne, et $\alpha \eta$ (La force externe) est li e aux contours des organes.

Le logiciel initial a  t  modifi  et compl t  avec un ensemble d'outils nouveaux. Un ou plusieurs germes sont plac s manuellement   l'int rieur des zones   segmenter dont les fronti res sont

définies par un seuillage manuel. La région ainsi limitée permet de contraindre la région d'évolution du contour actif (figure 1). Cette étape d'initialisation détermine la surface $C(u,v,0)$ comme l'union des surfaces des germes initiaux. La croissance des germes dépend à la fois de la pondération du terme d'expansion et de la pondération du terme de courbure (α, β). Ces paramètres sont modifiés en fonction du seuillage et de l'hétérogénéité des organes. L'utilisateur peut modifier la carte de pondération de façon à permettre la segmentation dans des régions hétérogènes. Un critère d'arrêt automatique reposant sur la variation du volume V des germes a été également ajouté au logiciel. Les variations de volume sont estimées toutes les 10 itérations (i) et l'arrêt est automatique lorsque le rapport V_{i+10}/V_i est inférieur à 1,01.

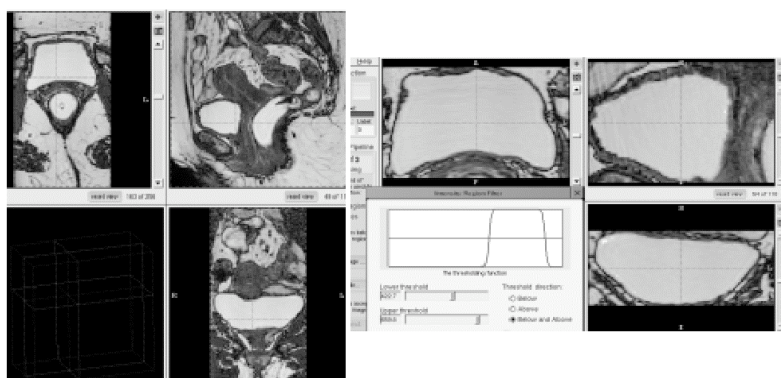


Figure 1. Segmentation d'un volume IRM à l'aide de l'interface du logiciel ITK-SNAP. À droite, une représentation de 3 coupes orthogonales; à gauche, l'étape de seuillage pour la vessie.

À la fin de l'étape de segmentation, les irrégularités géométriques sont automatiquement atténuées grâce à l'augmentation de la pondération de courbure (plus faible valeur de α) durant 40 itérations.

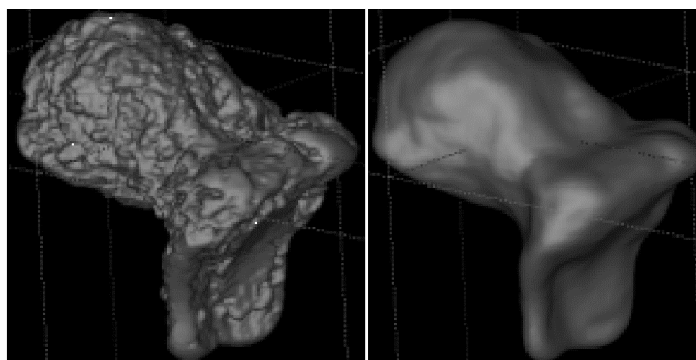


Figure 2. Résultat de la régularisation de la surface d'un utérus segmenté avec un critère géométrique local. À gauche le résultat obtenu avant l'étape de régularisation, à droite la surface lissée.

À partir de ces images, un maillage 3D de la surface des organes et des os de la région pelvienne est obtenu à l'aide d'un algorithme Marching-Cubes [22] et une méthode de rendu volumique permet la visualisation interactive du résultat. Pour réduire le temps de calculs de l'étape de déformation qui va suivre, le nombre des triangles est réduit. Cette réduction est obtenue par un algorithme de décimation appliqué aux triangles coplanaires et par la réduction du nombre de

faces des régions dont la courbure gaussienne est faible. Cette décimation permet une réduction du nombre de points sans induire une trop grande simplification géométrique.

B. Maillage volumique

Pour réaliser le modèle volumique, la surface 3D est remplie avec des tétraèdres à l'aide d'une méthode de Delaunay [23]. L'algorithme utilisé permet une vérification des tétraèdres créés de façon à leur assurer un volume minimum. Pour des ensembles de 1500 à 2000 points de surface, cette étape produit pour chacun des trois organes principaux des volumes contenant typiquement de l'ordre de 7000 à 8000 tétraèdres.

Les régions dont les niveaux de gris sont trop homogènes (ouraque, urètre, partie basse du vagin, périnée) ou qui sont en dehors de la zone d'acquisition (muscles élévateurs) sont difficiles à segmenter et seront donc représentés par des modèles géométriques 3D simples. Ces modèles sont alors positionnés en utilisant des amers adaptés dont les coordonnées sont disponibles en 3D dans le volume image initial. Le modèle géométrique ainsi obtenu et complété sert de support au modèle mécanique (figure 3).

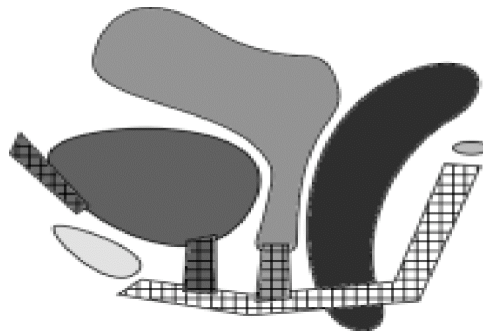


Figure 3. Représentation en coupe du modèle géométrique complet avec les modèles simples ajoutés manuellement (zones hachurées).

IV. Le modèle physique

A. Le réseau masses-ressorts.

Le modèle géométrique est associé à un modèle de déformations physiques de façon à simuler les déplacements physiologiques des organes. Les systèmes masses-ressorts reposent sur un modèle physique qui est largement et efficacement utilisé pour modéliser des organes mous [4], [24]. Le comportement physique de ce modèle est simple et bien compris. Le système masses-ressorts possède une structure simple et permet la mise en œuvre de diverses opérations, depuis des déformations importantes jusqu'à des changements de topologie. Le modèle physique consiste en un treillis de points masses connectés par des liens élastiques. Le réseau masses-ressorts est intégré au modèle géométrique précédemment décrit de sorte que les masses correspondent aux sommets alors que les ressorts correspondent aux arêtes du modèle géométrique. Le modèle associe un ressort et un amortisseur (modèle de Voigt) à chaque arête du modèle géométrique de façon à réduire l'influence des modes oscillatoires (figure 4).

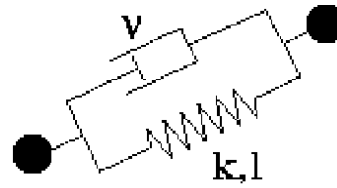


Figure 4. Le modèle masses-ressorts entre 2 sommets

Le modèle est alors géré par une équation classique (2): considérant un sommet, i , son vecteur de position, \mathbf{x} , sa masse, m , les forces de liaison $\mathbf{F}_{i \rightarrow j}$, les forces de collision \mathbf{F}_{ci} , et les forces externes \mathbf{F}_{ei} :

$$m\ddot{\mathbf{x}} = \sum_{j \in J} \vec{F}_{i \rightarrow j} + \vec{F}_{c_i} + \vec{F}_{e_i} \quad (2)$$

Deux sommets voisins i et j créent une force de liaison qui dépend de : leur distance d ; de la raideur du ressort, k ; de la longueur du ressort au repos, l ; de la différence de leurs vitesses \mathbf{V}_{ij} ; et du coefficient d'amortissement v .

Ce qui permet la mise en équation (3) suivante :

$$\vec{F}_{j \rightarrow i} = k \cdot (d - l) \frac{\vec{U}_{ij}}{d} + v \cdot \vec{V}_{ij} \quad (3)$$

Une fois que ces équations sont déterminées les interactions inter organes peuvent être calculées.

B Les collisions

À l'intérieur du modèle global, certains organes sont statiques comme les os (os iliaque et coccyx) et quelques muscles (muscle élévateur), *i.e.* ils ne pas concernés par les déformations et les mouvements. Les interactions entre organes peuvent créer des collisions ou des liens. Le calcul des forces de collisions repose sur un principe de pénalités [25]. Si deux objets A et B se recouvrent, pour chaque paire de tétraèdres concernés T_a et T_b où T_a appartient à A et T_b appartient à B , les forces répulsives sont appliquées. L'intensité de chacune des forces dépend de : la différence de vitesse, \mathbf{V}_{ab} ; du volume d'intersection $V_{int_{ab}}$; et des constantes réelles μ et λ . Pour simplifier les calculs, l'intersection des tétraèdres est remplacée par l'intersection de leurs boîtes englobantes.

$$\vec{F}_{b \rightarrow a} = \mu \cdot \vec{V}_{ab} + \lambda \cdot V_{int_{ab}} \frac{\vec{V}_{ab}}{\|\vec{V}_{ab}\|} \quad (4)$$

Ainsi un tétraèdre T_a est soumis aux forces de collision dues à chacun des tétraèdres T_b .

$$\vec{F}_{c_i} = \sum \vec{F}_{b \rightarrow a} \quad (5)$$

C La simulation

Le diagramme (figure 5) représente la boucle de simulation utilisée pour déterminer la position des sommets à chaque instant $t=k \cdot dt$, où dt est une constante de pas temporel. Chaque itération implique d'abord le calcul des différentes forces et ensuite l'intégration des équations. Les forces internes sont calculées de façon indépendante pour chacun des organes non osseux de façon à créer de plus petits systèmes. Le calcul des collisions est l'étape la plus longue à cause du nombre important de tétraèdres et de la complexité de l'algorithme ($O(n^2)$). Le calcul des forces externes est rapide parce que peu de points sont liés et que les forces de poussées sont connues.

L'intégration des équations est obtenue par une méthode classique basée sur un algorithme de Runge-Kutta à l'ordre 4, et implémentée avec la bibliothèque GSL [26].

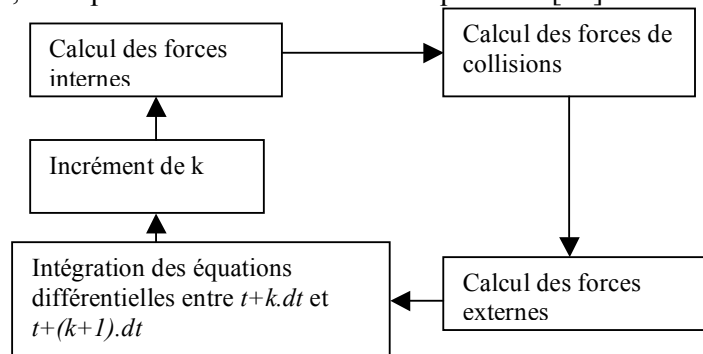


Figure 5. La boucle de simulation

Les forces externes concernent la force abdominale et la force diaphragmatique. Ces forces sont à l'origine du comportement qui doit être simulé. La force abdominale est appliquée sur les points situés sur la partie antérieure de l'utérus et de la vessie alors que la force diaphragmatique est appliquée sur les points situés sur les régions supérieures de ces organes. L'intensité de la force évolue selon un profil présenté à la figure 6 qui définit le comportement de la simulation.

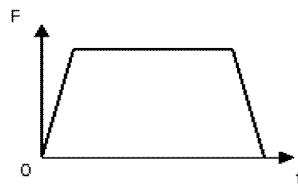


Figure 6. Représentation de l'évolution de l'intensité des forces pour une simulation

Les paramètres mécaniques des organes ne sont pas définis de façon évidente. L'approximation de l'isotropie est établie ce qui signifie que, pour chaque organe, la masse des sommets, la valeur des paramètres de dureté et d'amortissement sont les mêmes. La position de repos des ressorts correspond à la longueur de chaque arête du modèle géométrique au départ de la simulation. Les caractéristiques mécaniques et physiques des organes pelviens étaient inconnues et ont été approchées de façon empirique, l'utérus étant l'organe non osseux le plus rigide.

V. Discussion

A. Résultats

Un modèle complet du pelvis féminin a ainsi été généré en utilisant les images de haute qualité produites par l'IRM. La durée totale de l'examen d'acquisition est de l'ordre de 20 minutes dont environ 5 minutes reviennent à l'acquisition du volume initial. La segmentation complète et la correction des irrégularités géométriques prennent environ 5 heures. La segmentation de chacun des organes est une opération qui dure de 5 à 45 minutes. La segmentation de la vessie et des os est une opération facile alors que les segmentations du rectum et de l'utérus sont plus délicates à cause du faible contraste de ces organes dans l'image. Ces dernières structures nécessitent l'utilisation de plusieurs germes lors de l'initialisation de la segmentation. Pour ce qui concerne les organes non segmentés, la création des modèles supplémentaires et leur positionnement en 3D est une étape dont la durée est de l'ordre de 2 heures. Bien entendu, tous les organes peuvent être

visualisés en 3D avec des possibilités de rotation et d'agrandissement pour un contrôle par l'opérateur. La morphologie des organes segmentés est similaire aux données radiologiques et patient-spécifique.

Le modèle géométrique obtenu grâce à une segmentation efficace intègre toutes les caractéristiques morphologiques de la patiente et permet la visualisation d'un ou plusieurs organes en 3D. La visualisation tridimensionnelle fournit des avantages évidents pour l'appréhension de la complexité de la configuration des organes. Elle permet la détermination des principaux organes et du plancher pelviens, et pourrait fournir une représentation plus précise des relations entre les différentes structures pelviennes. Cela constituerait un nouvel outil pour la compréhension de l'anatomie pelvienne en remplaçant les atlas traditionnels, bi-dimensionnels et statiques. De plus, dans un contexte pédagogique un tel outil pourrait permettre de compléter l'usage des modèles cadavériques limités : par la dégénération des tissus; par leur coût et par la visibilité restreinte de cette zone profonde.

La décimation des triangles et la tétraédrisation du volume nécessitent 2 heures. Pour une simulation, le calcul des déformations dure de 4 à 5 heures.

Plus d'une centaine de simulations ont été réalisées dans le cadre de cette étude en modifiant un par un les paramètres physiques du modèle de façon à obtenir un résultat satisfaisant. Une estimation quantitative du comportement réaliste des organes pourrait permettre une meilleure, sinon automatique, paramétrisation des modèles des organes. Comme cela était attendu, les résultats sont excellents pour l'ensemble du modèle au repos avec une correspondance parfaite entre les images du modèle et celles acquises par IRM volumique. À la fin de la simulation qui correspond à l'effort maximal de poussée, la correspondance entre les images IRM et les images du modèle est satisfaisante pour les organes pelviens et excellente pour le plancher périnéal. En général, les variations de volume durant la simulation sont inférieures à 15% comme indiqué sur la figure 7. La conservation du volume est une des limites de la méthode choisie. Elle n'a pas encore été prise en compte et nécessiterait le développement d'algorithmes supplémentaires tels que ceux qui sont reportés par [27], [28]. Ainsi, lors de cette étude, la variation du volume de la vessie est assez importante. Cela pourrait expliquer l'écart entre les déplacements observés à l'IRM et ceux qui sont réalisés par le modèle. Une caractérisation plus précise des paramètres des organes ainsi qu'une meilleure conservation volumique devrait permettre une simulation plus précise des déplacements.

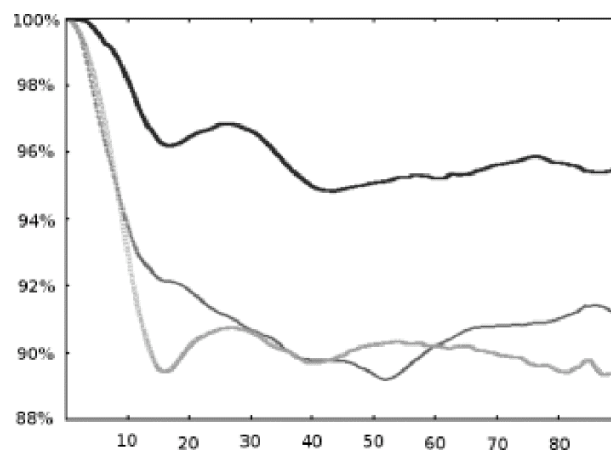


Figure 7. Une représentation des variations des volumes des trois organes principaux lors d'une simulation. Le rectum produit les plus faibles variations alors que l'utérus-vagin produit les variations les plus importantes.

B. Validation

Le d veloppement d'un simulateur r aliste bas  sur la physique des objets d formables requiert l'utilisation de donn es empiriques concernant le comportement des organes mous. De nombreuses proc dures reposant sur la compression, l'aspiration, l' tirement [29]-[33] ont r cemment  t  mises en place pour tester les propri t s m caniques des organes mous d'origines animale et humaine. Toutefois, il n'y a pas,   notre connaissance, de travaux concernant les propri t s m caniques in vivo sp cifiques aux organes pelviens. Tr s peu d' tudes pr sentent les propri t s  lastiques des tissus rencontr s lors des proc dures chirurgicales. Les travaux de Yamada [34] concernent des propri t s m caniques de la plupart des organes mais ils sont limit s   des  tudes cadav riques et particuli rement orient s vers les limites de compression des organes plut t que par leur propri t s  lastiques dans un contexte physiologique normal. Il est essentiel d'am liorer la connaissance des caract ristiques des organes pelviens de fa on   pouvoir reproduire des comportements biom caniques r alistes des organes. Il est bien connu qu'une difficult  majeure avec les mod les   base de masses-ressorts r side dans la d termination de param tres ayant une signification physique [35]. Dans le cadre de cette  tude, ces param tres ont  t  d finis de fa on empirique. Ils ont  t  modifi s un   un jusqu'  l'obtention de r sultats satisfaisants. Le r glage des param tres des ressorts pourrait  tre r alis  automatiquement par recuit simul  ou algorithme g n tique [36], mais l'essentiel de cette  tude r side dans la faisabilit  de la mod lisation et dans l'utilisation de l'IRM dynamique comme v rit  terrain. La caract risation des param tres physiques des organes pelviens est une  tape pr liminaire essentielle pour le d veloppement d'un simulateur r aliste de la correction des troubles de la statique pelvienne.

La validation du mod le pr sent  est essentiellement qualitative dans cette  tude pr liminaire. Elle a  t  obtenue par la mise en correspondance d'un plan m dian sagittal construit   partir du mod le et les images IRM acquises dans le plan m dian sagittal en mode dynamique (figure 8).



Figure 8. Comparaison entre le r sultat d'une simulation avec l'observation par examen IRM dynamique. Les points blancs correspondent   une coupe du mod le 3D par un plan m dian sagittal en fin de simulation (pouss e maximale). Le recalage entre les 2 coupes a  t  r alis  manuellement.

Cette validation peut être complétée par une évaluation quantitative en mesurant les coordonnées 3D d'amers spécifiques repérés à la fois dans le modèle et dans les images. La comparaison des mouvements de tels points permettra une meilleure évaluation à la fois statique et dynamique. Elle permettra également la mesure automatique de plusieurs paramètres cliniquement pertinents comme l'angle anorectal, l'angle cervico-urétral, la descente périnéale. Dans cette étude l'estimation des déplacements des organes n'a été réalisée ni dans le plan frontal ni dans le plan transversal. Seules quelques récentes études ont évalué ces déplacements durant les efforts de poussée [37]. En fait, la connaissance de ces déplacements reste mauvaise, mais ils semblent être limités même s'ils sont supposés être plus importants et moins symétriques pour les femmes âgées.

Cette étude préliminaire a permis la réalisation d'un modèle spécifique, statique et dynamique obtenu à partir d'une IRM d'une patiente donnée. L'acquisition IRM de la région du petit bassin nécessaire pour la segmentation des organes a été réalisée dans le cadre du bilan radiologique faisant partie de l'activité clinique sans grand impact sur cette dernière. L'acquisition pelvienne complète durant 10 à 15 minutes reste compatible avec la routine clinique. Cela permet d'accepter le temps supplémentaire que nécessite la segmentation des muscles élévateurs et du plancher pelvien. Les modélisations de ces muscles réalisées dans d'autres études à l'aide de méthodes différentes de segmentation pourraient encore améliorer la précision anatomique du modèle et rendre ainsi la modélisation dynamique plus réaliste.

La méthode de segmentation qui a été présentée repose sur une procédure semi-automatique qui permet au praticien de superviser le processus. Le temps d'apprentissage de l'utilisation du logiciel est relativement court et sera encore diminué par l'amélioration de son interface. Les temps de calculs pour la modélisation dynamique sont actuellement importants à cause de la gestion de plusieurs organes en même temps ce qui nécessite la détection des collisions. Cependant cette durée pourra être réduite par la simplification des modèles géométriques et par l'optimisation des algorithmes. De plus l'ensemble du processus décrit ici fonctionnait sur un ordinateur du type PC (Station Linux avec Pentium IV). L'utilisation de systèmes plus performants et le développement de nouveaux processeurs permettra l'utilisation d'algorithmes plus complets tout en diminuant les temps de calculs. La simulation temps réel permettra alors la naissance des simulateurs de réalité virtuelle.

La prochaine étape de ces travaux concerne d'abord la mise en place d'une méthodologie de validation quantitative basée sur les mesures et le recalage. Plus que la compréhension de la physiologie, la validation des paramètres physiques permettrait la mise en place de la base d'un simulateur de chirurgie pelvienne.

VI. Conclusion

L'amélioration des sciences de l'informatique et de l'imagerie médicale ont permis le développement de simulateurs chirurgicaux de plus en plus réalistes. Toutefois, la physiologie et la pathologie pelviennes sont des problèmes difficiles à modéliser à cause de la présence de plusieurs organes mobiles et déformables.

Cette étude a permis le développement d'un modèle spécifique de pelvis féminin qui simule l'effort de poussée bien que le prise en compte de tous les phénomènes de collisions impliquant des organes mous reste un problème complexe. L'objectif n'était pas orienté vers la vitesse d'exécution mais plutôt vers la possibilité de profiter d'une vérité terrain disponible à travers l'acquisition IRM dynamique en vue de régler les paramètres du modèle. La précision des

mod les des organes est assez  lev e apr s l' tape de segmentation. En r duisant cette pr cision des simulations plus rapides pourraient  tre produites selon les objectifs vis s par l'application. Dans le m me temps, le mod le physique pourrait  tre affin  mais au d triment du co t en temps de calcul [38], [39]. Ce compromis classique illustre le fait que la simulation chirurgicale peut favoriser un autre point de vue que la r solution maximale de visualisation.

La r alisation de ce mod le est l' tape pr liminaire pour le d veloppement d'un simulateur patient sp cifique des troubles de la statique pelvienne. Une  tude orient e vers la validation quantitative des comportements des organes est actuellement un travail en cours. Cependant les principaux r sultats sont encourageants et le choix d'amers consistants devrait permettre de fixer l'ensemble des param tres qui produira une simulation encore plus r aliste.

R f rences.

- [1] Villard C, Soler L, Gangi A (2005) "Radiofrequency ablation of hepatic tumors: simulation, planning, and contribution of virtual reality and haptics". *Comput Methods Biomech Biomed Engin* 8: 215-227
- [2] Lang H, Radtke A, Hindennach M, Schroeder T, Fruhauf NR, Malago M, Bourquain H, Peitgen HO, Oldhafer KJ, Broelsch CE (2005) "Impact of virtual tumor resection and computer-assisted risk analysis on operation planning and intraoperative strategy in major hepatic resection". *Arch Surg* 140: 629-638
- [3] Hirooka M, Iuchi H, Kurose K, Kumagi T, Horiike N, Onji M (2005) "Abdominal virtual ultrasonographic images reconstructed by multi-detector row helical computed tomography". *Eur J Radiol* 53: 312-317
- [4] Dornheim L, Tonniens KD, Dixon K (2005) "Automatic segmentation of the left ventricle in 3D SPECT data by registration with a dynamic anatomic model". *Med Image Comput Comput Assist Interv Int Conf Med Image Comput Comput Assist Interv* 8: 335-342
- [5] Hoyte L, Fielding JR, Versi E, Mamisch C, Kolvenbach C, Kikinis R (2001) "Variations in levator ani volume and geometry in women: the application of MR based 3D reconstruction in evaluating pelvic floor dysfunction". *Arch Esp Urol* 54: 532-539
- [6] Tunn R, Delancey JO, Howard D, Ashton-Miller JA, Quint LE (2003) "Anatomic variations in the levator ani muscle, endopelvic fascia, and urethra in nulliparas evaluated by magnetic resonance imaging". *Am. J. Obstet Gynecol* 188: 116-121
- [7] Fielding JR, Dumanli H, Schreyer AG, Okuda S, Gering DT, Zou KH, Kikinis R, Jolesz FA (2000) MR-based three-dimensional modeling of the normal pelvic floor in women: quantification of muscle mass. *Am J Roentgenol* 174: 657-660
- [8] Hoyte L, Schierlitz L, Zou K, Flesh G, Fielding JR (2001) Two- and 3-dimensional MRI comparison of levator ani structure, volume, and integrity in women with stress incontinence and prolapse. *Am J Obstet Gynecol* 185: 11-19
- [9] Singh K, Jakab M, Reid WM, Berger LA, Hoyte L (2003) "Three-dimensional magnetic resonance imaging assessment of levator ani morphologic features in different grades of prolapse". *Am J Obstet Gynecol* 188: 910-915
- [10] Madoff RD, Mellgren A (1999) "One hundred years of rectal prolapse surgery". *Dis Colon Rectum* 42: 441-450
- [11] Weber AM, Richter HE (2005) "Pelvic organ prolapse". *Obstet Gynecol* 106: 615-634
- [12] Maher C, Baessler K, Glazener CM, Adams EJ, Hagen S (2004) "Surgical management of pelvic organ prolapse in women". *Cochrane Database Syst Rev* 18: CD004014.
- [13] Purkayastha S, Tekkis P, Athanasiou T, Aziz O, Paraskevas P, Ziprin P, Darzi A (2005) A comparison of open vs. laparoscopic abdominal rectopexy for full-thickness rectal prolapse: a meta-analysis. *Dis Colon Rectum* 48: 1930-1940
- [14] Safir MH, Gousse AE, Rovner ES, Ginsberg DA, Raz S (1999) "4-Defect repair of grade 4 cystocele". *J Urol* 161: 587-594
- [15] Gerson LB, Van Dam J (2003) "A prospective randomized trial comparing a virtual reality simulator to bedside teaching for training in sigmoidoscopy". *Endoscopy* 2003 35: 569-575 Erratum in: *Endoscopy*. 2004 Feb;36(2):185.
- [16] Sz kely G, Brechb hler Ch, Dual J, R. Enzler, J. Hug, R. Hutter, N. Ironmonger, M. Kauer, V. Meier, P. Niederer, A. Rhomberg, P. Schmid, G. Schweitzer, M. Thaler, V. Vuskovic, G. Tr ster, U. Haller, M. Bajka (2000) "Virtual Reality-Based Simulation of Endoscopic Surgery". *Presence* 9: 310-333
- [17] Grantcharov TP, Kristiansen VB, Bendix J, Bardram L, Rosenberg J, Funch-Jensen P (2004) "Randomized clinical trial of virtual reality simulation for laparoscopic skills training". *Br J Surg* 91: 146-150
- [18] Youngblood PL, Srivastava S, Curet M, Heinrichs WL, Dev P, Wren SM (2005) "Comparison of training on two laparoscopic simulators and assessment of skills transfer to surgical performance". *J Am Coll Surg* 200: 546-551.
- [19] Seymour NE, Gallagher AG, Roman SA, O'Brien MK, Bansal VK, Andersen DK, Satava RM (2002) "Virtual reality training improves operating room performance: results of a randomized, double-blinded study". *Ann Surg* 236: 458-463
- [20] SNAP, ITK-SNAP Homepage, <http://www.itksnap.org/>
- [21] Sethian, James A. "Level Set Methods and Fast Marching Methods". Cambridge University Press. ISBN 0521645573, 1999.
- [22] Lorensen, William and Harvey E. Cline. "Marching Cubes: A High Resolution 3D Surface Construction Algorithm". *Computer Graphics (SIGGRAPH 87 Proceedings)* 21(4):163-170, 1997.
- [23] Shewchuk J. R., "Tetrahedral Mesh Generation by Delaunay Refinement". *Proceedings of the Fourteenth Annual Symposium on Computational Geometry (Minneapolis, Minnesota)*, pp 86-95, 1998.
- [24] Sorensen TS, Mosegaard J (2006) "Haptic feedback for the GPU-based surgical simulator". *Stud Health Technol Inform* 119: 523-528
- [25] M. Moore and J. Wilhems. "Collision detection and response for computer animation". *Computer Graphics*, August 1988.
- [26] GSL, GNU Scientific Library, <http://www.gnu.org/software/gsl/>
- [27] Y. Lee, D. Terzopoulos, K. Waters, "Realistic Modeling for Facial Animation", *Proc. SIGGRAPH* 1995: 55-61, Aug. 6--11, Los Angeles
- [28] E. Promayon, P. Baconnier, C. Puech. "Physically-Based Deformations Constrained in Displacements and Volume". *Computer Graphics Forum, Eurographics96*, 15(3):155-164, August 1996.
- [29] Brown JD, Rosen J, Kim YS, Chang L, Sinanan MN, Hannaford B (2003) "In-vivo and in-situ compressive properties of porcine abdominal soft tissues". *Stud Health Technol Inform.* 94: 26-32

- [30] Lim YJ, Jones DB, Singh TP, De S. (2006) "Measurement of the mechanical response of intra-abdominal organs of fresh human cadavers for use in surgical simulation". *Stud Health Technol Inform.* 119: 322-327
- [31] Carter FJ, Frank TG, Davies PJ, McLean D, Cuschieri A (2001) "A Measurement and modelling of the compliance of human and porcine organs". *Med. Image Anal.* 5: 231-236
- [32] Miller, K. Chinzei, G. Orssengo and P. Bednarz, (2000) Mechanical properties of brain tissue in vivo: experiment and computer simulation *J Biomech* 33: 1369-1376
- [33] Zheng Y, Mak AFT (1996) "An ultrasound indentation system for biomechanical properties assessment of soft tissues in vivo". *IEEE Trans Biomed Eng* 43 : 912-918
- [34] Yamada H (1970) "Strength of biomechanical materials". In: Evans, F.G., (Ed.), Williams and Wilkins, Baltimore.
- [35] A. Maciel, R. Boulic, D. Thalmann, "Deformable Tissue Parameterized by Properties of Real Biological Tissue". In. *International Symposium on Surgery Simulation and Soft Tissue Modeling, 2003, Juan-les-Pins, France: 74-87, Springer-Verlag, Berlin 2003*
- [36] Gérald Bianchi, Barbara Solenthaler, Gábor Székely, Matthias Hadders: "Simultaneous Topology and Stiffness Identification for Mass-Spring Models Based on FEM Reference Deformations." *MICCAI (2) 2004: 293-301*
- [37] Constantinou CE, Hvistendahl G, Ryhammer A, Nagel LL, Djurhuus JC (2002) Determining the displacement of the pelvic floor and pelvic organs during voluntary contractions using magnetic resonance imaging in younger and older women. *BJU International* 90: 408-414
- [38] Cooper, L. and S.C. Maddock (1997), "Preventing Collapse Within Mass-Spring-Damper Models of Deformable Objects", *Proc. The Fifth International Conference in Central Europe on Computer Graphics and Visualization'97 (Plzen, Czech Republic, February 1997): 70-78.*
- [39] Picinbono G, Delingette H, Ayache N: "Real-Time Large Displacement Elasticity for Surgery Simulation: Non-linear Tensor-Mass Model." *MICCAI 2000: 643-652*

Breathing Thorax Simulation based on Pleura Physiology and Rib Kinematics

Anne-Laure Didier¹, Pierre-Frédéric Villard¹, Jean-Yves Bayle²,
Michaël Beuve¹, Behzad Shariat¹

¹ Université de Lyon, Lyon, F-69003, France ; université Lyon 1, CNRS UMR5205,
LIRIS, Villeurbanne, F-69622, France

² Hôpital Louis Pradel, Lyon, France

{anne-laure.didier, pierre-frederic.villard}@liris.univ-lyon1.fr

Abstract

To monitor a lung mechanical model and then predict tumour motion we proposed a approach based on the pleura physiology. By comparing the predictions to landmarks set by medical experts, we observed better results with regards to the one obtained with approaches found in the literature. Beside, we focus on the rib cage kinematics, which play a significant role in the pleura outer-surface motion and therefore in the lung motion. We proposed a kinematic model of the rib cage based on the finite helical axis method and we show out interesting results.

Keywords— Finite Element Method, Dynamic Deformable Model, Thorax Physiology, Rib Cage Kinematics

1 Introduction

During a radiotherapy treatment, healthy tissues have to be preserved while the dose has to be concentrated on the tumour. These goals are especially difficult to achieve with pulmonary tumour due to patient's breathing. Hence, the knowledge of lung-tumor positions during any step of breathing cycle would dramatically improve lung cancer treatment.

We developed a lung numerical model to estimate such a motion. This model is based on the continuous mechanics laws, which are solved with the finite element method. Our results led to compute the displacement of every point inside the lung with only knowing lung surface motion.

Our strategy is now to monitor our simulation with lung external surface. Therefore, we will provide in a first section some elements lung physiological and lung environment. We will especially focus on the pleura, the rib cage and the diaphragm because they play a significant role on lung-surface motion. This information is essential to understand our approach, which is strongly based on medical reality. In a second section we will then explain how our model considers the pleura physiology and we will propose a clinical evaluation. Finally, a motion model of the

rib cage is presented in the last section.

2 Respiratory Physiology

Respiration is a complex mechanical process, which mobilises many elements we describe below.

2.1 The lung

Lungs take up most of the thorax volume. The left lung is smaller due to the space occupied by the heart. The two lungs are separated by the mediastinum, which contains the heart, important arteries, veins, trachea, oesophagus and lymph gland. During breathing, the lungs keep contact with the rib cage and the diaphragm through the pleura and follow any of their motions.

2.2 The pleura

The pleura is composed of two membranes: the first called parietal covers the chest wall, mediastinum and diaphragm while the second is called visceral and covers the outer surface of the lungs. The space in between, known as the pleural space, is filled with an incompressible pleural fluid. The surface tension of this fluid enables the contact between lung surface on one hand, and, the rib cage and the diaphragm on the other hand. This fluid lubricates the pleural surfaces and allows the lungs to easily slide against the chest wall when they expand. During inhalation, the motion of the diaphragm and of the rib cage induces a negative pressure (with regards to atmospheric pressure) in the pleural cavity leading to a tensile stress on the lungs.

2.3 Rib cage and intercostal muscles

The rib cage is composed of the sternum, twelve rib pairs, costal cartilage and twelve thoracic vertebrae. The first seven ribs, called true ribs, are linked directly to the sternum (by their own costal cartilage) while the height to ten ribs joined the seventh rib by their costal cartilage. The last two ribs, called false ribs, are not link to the sternum and have not costal cartilage. All ribs are connected to spine and are more or less inclined astern. The Thorax has

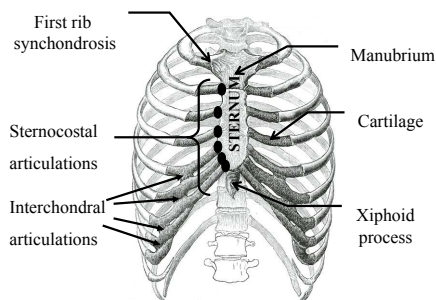


Figure 1. rib articulations

a lot of articulations and we will enlighten: 1- the sternocostal (between the cartilage and the sternum), 2-the interchondral (cartilage between themselves), and 3- the costovertebral (synovial joint between ribs and vertebrae).

We can assimilate the rib motion to a “bucket handle” movement mixed with a “pump handle” one. We can note that the upper ribs have a more pronounced “pump handle” movement than the lower ribs for which “bucket handle” is sharper. Rib motion is led by the intercostal muscles, which stand between the ribs. Internal intercostal muscles are only activated during forced exhalation (quiet exhalation is a passive process). External intercostal muscles (EIM) are relatively important in both quiet and forced inhalation. EIM insert into ribs 2-12 and are responsible for ribs elevation and Postero-Anterior thoracic diameter raising.

2.4 The diaphragm

The most important muscle of the respiration is the diaphragm. Our final aim is to include this muscle in our numerical model. It is a digastric muscle that separates the thoracic cavity from the abdominal one. It is made of two domes: the right comes up on the fourth intercostal space whereas the left does not go past the fifth one. The diaphragm is constituted by a peripheral part (muscular fibers) and by a central tendon. Its peripheral part is linked with the whole lower thoracic cavity perimeter and has three major insertions: lumbar, sternum and ribs.

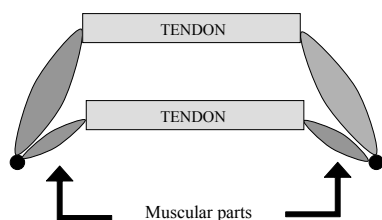


Figure 2. diaphragm motion simplification

During inspiration, the contraction of muscular fibers brings down the central tendon (figure 2). This lowering increases the vertical diameter of the thorax. There are many inspiration accessory muscles like scalenes or sternocleidomastoid but they do play a role only for high ventilation that is out of our application field.

3 Pleura Influence Modeling

In the literature, many methods were studied to model the behaviour of thorax system. But only two classes of methods focus on lungs: 1 - The discrete methods consist in discretising the organ into mass points and in applying interaction laws in between ([8], [12] and [9]) 2 - The continuous methods, consist in defining every mechanical quantities as continuous function inside the whole organ. Continuous approach seems more adapted to model lungs within the accuracy required by our radiotherapy context: they generally use continuous mechanics principles but few of them focus on respiration modelling. We note that these latter models do not take into account the pleura influence. For instance, in [3], boundary conditions are set by the displacements determined from a non-rigid registration. In [2], Brock *et al.* simulate lung inflation by imposing orthogonal displacements to its surface (Cf Figure 3, left). In terms of mechanics, this **OD** method (**OD** for **Orthogonal Displacements**) consists in assuming that inhalation is only produced by the negative pressure of the respiration muscles. Hence, each lung-surface point follows a displacement perpendicular to the lung surface. In [10], we defined boundary conditions that explicitly take into account the pleura influence as described in §2.2. Here, We propose to compare the results obtained for both approaches in order to show the importance of the pleura role.

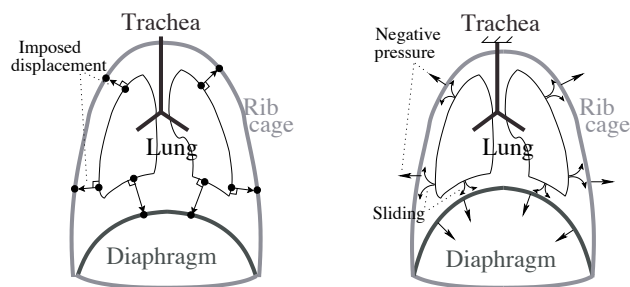


Figure 3. Boundary conditions apply to lungs: left: OD method and right: Our method

3.1 Our model

Our model is based on continuous mechanics laws. We consider lung tissue as homogeneous, isotropic, and its constitutive equation defined by a linear elastic behaviour.

Moreover, we assume that normal breathing can be approximated by a series of quasi-static states. Our boundary conditions are illustrated on Figure 3, right. Lungs are fixed near the trachea, and the pleura behaviour is simulated by applying contact conditions allowing lung surface to slide against pleura. The contact conditions model the negative intra-pleural pressure and the sliding surface models the lubricating fluid inside pleural cavity (Cf §2.2). At the same time, the inflation from an initial state to a final state is caused by a negative pressure applied to lung surface. This models the muscle actions.

The system with these three kinds of equations (constitutive, dynamics and boundary conditions) is solved with the finite element method. We use Code-Aster [1], a finite element software developed by the French Electricity Board (EDF) for the last 20 years. The required computing time is then 10 times more important than with the OD method. This is due to the non-linearity of the system (contact conditions). In principle the results obtained with our model should be in better agreement with reality because it includes a more realistic description of anatomy. However we verified this assumption by a clinical comparisons of both methods.

3.2 Clinical comparison

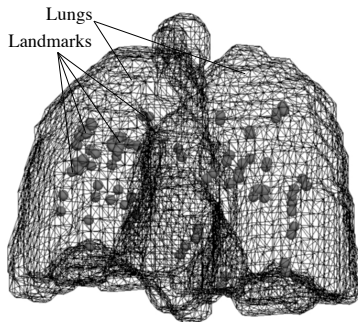


Figure 4. Example of landmark positions inside Patient 2's lungs. Each landmark has been determined by medical experts.

To compare both methods we defined a validation protocol. We considered two sequences of 4D CT scan (one for Patient1 and one for Patient2) acquired according to a protocol similar to the one described in [7]. On each sequence, we selected four levels of lung inflation. This allows to define three transitions that is to say three pairs of initial and final states. We assumed that real motion data could be estimated by a series of "landmarks" determined by three medical experts. Example of landmark positions could be seen on Figure 4.

We proceed now to the simulation of these three lung motions using first the OD method and then our own

method. Orthogonal displacements are defined on each mesh vertex and are computed with the ray-triangle intersection method presented in [6].

We defined for both models the same mesh, composed of 30000 elements, and the same computer, a Pentium 4, 2.40GH with 1GB memory. Finally, only the boundary conditions were different. We gather the simulation results on Table 1 for Patient 1 and Table 2 for Patient 2. For each selected points, we calculated the differences between the simulated displacements and the landmark displacements (reference values). Histograms of errors can be found on Table 1 and Table 2. OD average error amounts to 3.1mm for Patient 1 and to 6.16mm for Patient 2, while the same error with our method amounts to 2.14mm for patient 1 and to 5.16mm for patient 2.

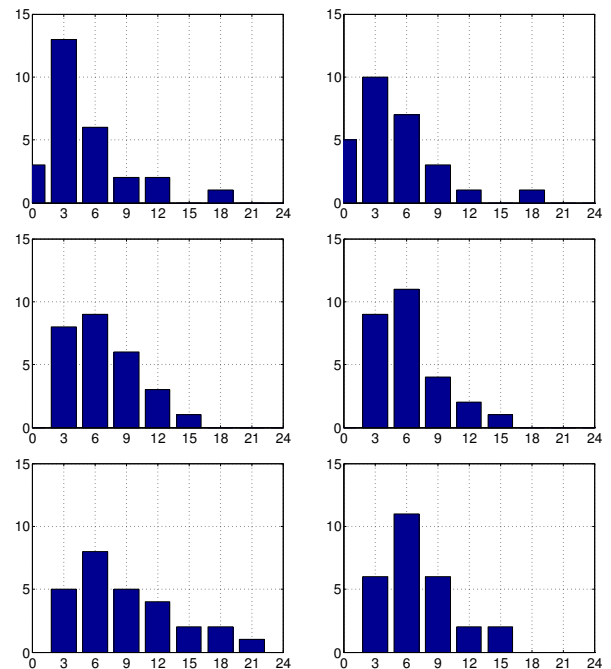


Table 1. Patient 1 histograms of displacement errors such as $error = |position_{simulation} - position_{experts}|$. X axis represents the intervals of displacement errors (in mm) and the number of such errors. From top to bottom the three levels of inflation and Left: OD method, right: Our method

3.3 Discussion on pleura influence

Tables 1 and 2 show that pleura influences lung behaviour. Indeed, not only it shows that simulations from both methods lead to different results but also that taking into account the pleura sliding gives results in better agree-

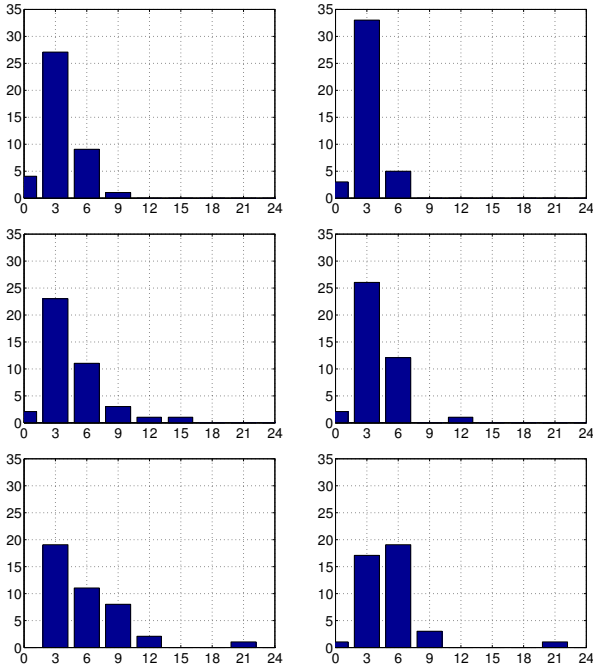


Table 2. Patient 2 histograms of displacement errors such as $error = |position_{simulation} - position_{experts}|$. X axis represents the intervals of displacement errors (in mm) and the number of such errors. From top to bottom the three levels of inflation and Left: OD method, right: Our method

ment with clinical data. For both patients, histograms with our method have more bins corresponding to small errors. The average error is also less important with our method than with the **OD** method. In particular, we note that the lateral displacement is underestimated with **OD** approach.

For both methods majority of displacement errors remains below $1cm$ which is better than actual routine method consisting in ignoring the displacements. Nevertheless, results could be improved. Mesh resolution is only $10cm^3$. Besides, landmarks were chosen by experts on strong heterogeneity area (carina, calcified nodules, culmen-lingula junction, division branch of pulmonary artery, apical pulmonary vein of upper lobe, etc), where displacement errors are expected more important because of the assumption of homogeneous mechanical properties. Finally, dynamic simulation of lung behaviour will be more accurate by taking pleura influence into account.

4 Control of the pleura outer-surface motions

As mentioned in section §2, the pleura and therefore the lung motion are monitored by the diaphragm and the rib cage. In this section, we propose a kinematics model of the rib cage. By the past, few authors took an interest in the rib kinematics like T.A.Wilson [11] who studied the displacement of human ribs and defined their rotation axis with a system of rib planes. To obtain rib shape and magnitude of displacements in **all** directions, we chose another method called the finite helical axis method.

4.1 Finite Helical Axis Method

The FHA method is often applied to describe a rigid body displacement between two successive states. The aim of the FHA method is to describe a solid movement by a rotation and a translation around and along an axis. The principles of this method are illustrated in the Figure 5, where X_i, Y_i, Z_i is a coordinate system linked to a solid at an initial state and X_f, Y_f, Z_f is the same coordinate system after transformation.

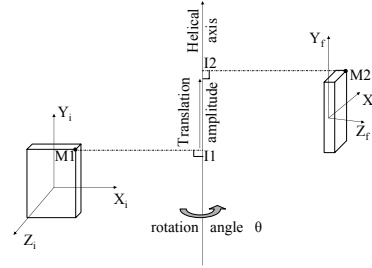


Figure 5. Finite helical axis method principle

To get the FHA parameters of ribs, we worked only on their solid parts (Cf section §2) and we used the relative rotation operator (Equation 1).

$$\begin{pmatrix} c\theta + kx^2(1 - c\theta) & kxky(1 - c\theta) - kzs\theta & kys\theta + kxkz(1 - c\theta) \\ kzs\theta + kxky(1 - c\theta) & c\theta + ky^2(1 - c\theta) & -kxs\theta + kykz(1 - c\theta) \\ -kys\theta + kxkz(1 - c\theta) & kxs\theta + kykz(1 - c\theta) & c\theta + kz^2(1 - c\theta) \end{pmatrix} = [a]_{ij} \quad (1)$$

where $c\theta = \cos \theta$, $s\theta = \sin \theta$, kx, ky, kz : axis direction

Chèze et al. [4] applied it, with markers glued on the skin, and noted that the inaccuracy in the measurement leads to errors in the position and direction of the FHA. Spoor and al. [5] showed that these measurement errors cause a result error inversely proportional to the rotation magnitude. They insisted on the fact that a high degree of measurement accuracy of marker coordinates is required for reasonably accurate of the direction and position of the

axis. In our study, we extracted from CT-scan anatomical point landmarks, which is unmistakably a more precise method than markers. To get the FHA parameters, the displacement and orientation changes of a coordinate system attached to the rib through respiration-cycle time are required. This system is built from three non-aligned rib points (one near the rib head, another on the middle and the last just before the cartilage part). These point positions are identified on the scans at each breathing level. So, we obtain the coordinate-system position attached to a rib at the initial state and at the final state. By applying the equation (1), we acquire the rotation angle, the FHA direction and orientation, the translation amplitude by using the relations described in [4].

The aim of our work is to predict, from the rotation parameters, the rib positions at any time. Therefore, we applied Rodrigues formula, which reads in the case of an helical displacement (Cf Figure 5):

$$M_1 \vec{M}_2 = 2\vec{k} \tan \frac{\theta}{2} \times \left(\frac{I_1 \vec{M}_1 + I_2 \vec{M}_2}{2} \right) + I_1 \vec{I}_2 \quad (2)$$

To verify the performance of our proposal, at the initial state (end of exhalation), we chose ten points placed on the seventh rib midline illustrated on Figure 6. We plotted them on the thorax which has been segmented from the initial scan. We applied our method and we obtained these point positions at the final state (Figure 7).

We proposed a linear interpolation of the transformation to predict the rib motion at any intermediate breathing states as follows:

$$\begin{aligned} \theta &= x\theta_{max} \\ t &= xT_{max} \end{aligned} \quad (3)$$

Where $x \in [0, 1]$ is the interpolation coefficient, θ_{max} and T_{max} are respectively the rotation angle and the translation amplitude between exhalation end and inhalation end. To evaluate the validity of our interpolation, we compute the rib points for $x=0.5$ (which is certainly the most penalizing for our algorithm) and compared them to the position of the seventh rib at the intermediate breathing level (Figure 8).

4.2 Firsts Results

We defined some reference-point positions at the end of exhalation as the rib midline (cf. Figure 6. The reference points were linked in order to enhance visualization.

We computed, by using the Rodrigues Formula (Equation 2), the positions of these reference points at the final state (end of inhalation). The results are illustrated on Figure 7.

We observe that the predicted final points match well with the rib midline. We observed this agreement whatever the visualization directions is.

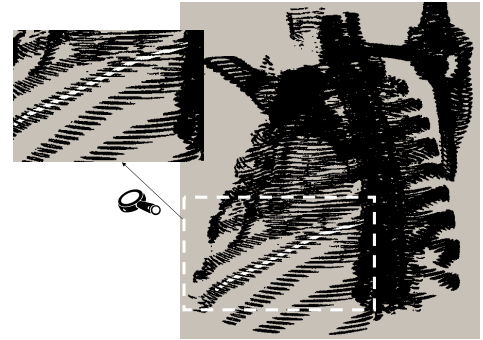


Figure 6. Profile of segmented thorax: definition of the reference points at the final state (in white)



Figure 7. Simulation of the reference points at the final state (in white)

We evaluated our interpolation method for the case (Equation 3) $x=0.5$ by comparing our predictions to the seventh rib on the segmented thorax at the intermediate breathing level.

Our computed final-points are very close to the rib midline points, so, we can conclude that the rib motion can be well interpolated with our method.

To evaluate the robustness of our method, we estimated the error generated by the uncertainty in setting the point coordinates of the coordinate systems attached to the rib.

As expected, the error in Figure 9 is more important than in Figure 7 but it is still acceptable and the data remain workable.

Conclusion

In this work we focused on the monitoring of lung mechanical models to simulate motions during breathing. We first showed that taking into account the pleura physiology gives predictions in better agreement with clinical data with regard to the results of mechanical model proposed in the literature. Since the external surface of pleura follows the rib cage, we secondly proposed a FHA model of rib cage kinematics that predicts the rib position at any time. The only required input is the manual registration



Figure 8. Interpolation method: simulation of reference points at the intermediate state (in white)



Figure 9. Simulation of the reference point positions at the final state (in white)

of 3 non-aligned rib points for one breathing state to another. The first results are promising. Beside we numerically evaluated the sensibility of our approach to the accuracy of the landmark-points setting and we concluded to a rather correct robustness of our method. Actually, the landmark points are chosen manually, but we plan to explore deformable-registration techniques to reduce man/machine interactions. The complete monitoring of the lung motion will be achieved after modeling of diaphragm motion with the help of additional data such as the patient air-breath rate.

5 Acknowledgements

We thank the French league against cancer for their financial support and all our partners: Léon Bérard Centre and ETOILE¹ Center for their support.

References

- [1] Code Aster. <http://www.code-aster.org/>.

¹<http://etoile.in2p3.fr>

- [2] K.K. Brock, M.B. Sharpe, L.A. Dawson, S.M. Kim, and D.A. Jaffray. Accuracy of finite element model-based multi-organ deformable image registration. *Medical physics*, 32(6):1647–59, June 2005.
- [3] Y. Chi, J. Liang, and D. Yan. A material sensitivity study on the accuracy of deformable organ registration using linear biomechanical models. *Medical physics*, 33(2):421–433, February 2006.
- [4] L. Chèze, B.J. Fregly, and J. Dimnet. Determination of joint functional axes from noisy marker data using the finite helical axis. *Human Movement Science*, 17:1–15, 1998.
- [5] C.W.Spoor. Explanation, verification and application of helical-axis error propagation formulas. *Human Movement Science*, pages 95–117, 1984.
- [6] T. Möller and B. Trumbore. Fast, minimum storage ray-triangle intersection. *journal of graphics tools*, 2(1):21–28, 1997.
- [7] T. Pan, T.Y. Lee, E. Rietzel, and G.T.Chen. 4D-CT imaging of a volume influenced by respiratory motion on multi-slice CT. *Medical physics*, 31(2):333–340, 2004.
- [8] A.P. Santhanam, C.M. Fidopiastis, F.G. Hamza-Lup, J.P. Rolland, and C.Imielinska. Physically-based deformation of high-resolution 3d lung models for augmented reality based medical visualization. In *MIC-CAI AMI-ARCS*, pages 21–32., 2004.
- [9] P.W. Segar, D.S. Lalush, and B.M.W. Tsui. Modeling respiratory mechanics in the mcatt and spline-based mcatt phantoms. *IEEE Transactions on Nuclear Science*, I(48):89–97, 2001.
- [10] P.-F. Villard, M. Beuve, B. Shariat, V. Baudet, and F. Jaillet. Simulation of lung behaviour with finite elements : Influence of bio-mechanical parameters. *IEEE, Mediviz, Conference on Information Visualization*, pages 9–14, 2005.
- [11] T. A. Wilson, K. Rehder, S. Kraymer, E. A. Hoffman, C. G. Whitney, and J. R. Rodarte. Geometry and respiratory displacement of human ribs. *Journal of Applied Physiology*, 62(5):1872–1877, 1987.
- [12] V.B. Zordan, B. Celly, B. Chiu, and P.C. DiLorenzo. Breathe easy : Model and control of simulated respiration for animation. *ACM SIGGRAPH Symposium on Computer Animation*, 2004.

Suivi de mouvements respiratoires: survol et nouvelle approche

Benoit VIGUIER, Élodie DESSERÉE, Jean-Michel MOREAU
LIRIS - SAARA

GTAS'07, juin 2007

Résumé

Nos recherches tentent d'apporter une solution aux problèmes de suivi de patient qui se posent dans le domaine médical. Alors que certaines équipes travaillent sur la simulation d'organes en mouvements lors du cycle respiratoire, nous ne disposons pas encore de dispositif de pilotage pour ces procédés. Parmi les méthodes de vision par ordinateur et d'acquisition de mouvements, nombreuses sont celles qui restent inadaptées au contexte médical que nous nous imposons. Nous effectuons un survol des limites existantes, et introduisons une nouvelle approche, en cours de soumission.

1 Contexte

Les domaines de l'informatique liés à la réalité augmentée prennent de plus en plus d'importance dans le domaine médical, que ce soit pour assister les gestes médicaux, simuler des comportements d'organes ou tout simplement mieux visualiser certaines données. Les contraintes d'utilisation sont pourtant très différentes de ce que pourraient être celles d'une autre application (vidéo ludique, immersion historique...); l'interactivité, la précision, la robustesse... autant de termes qui sont omniprésents dans toute construction d'un système dédié à un contexte médical.

Le traitement des tumeurs cancéreuses mobiles (organes situés dans le thorax) soulève une problématique importante : comment suivre le déplacement d'une tumeur en fonction du mouvement respiratoire? Certains travaux mettent en avant la possibilité de simuler le mouvement interne des organes du thorax dans le cadre du cancer du poumon [VBS*04], mais ceci nécessite alors un pilotage des modèles par un suivi des mouvements respiratoires du patient. Le suivi de patient est un problème général des traitements médicaux; pouvoir comparer la position d'un patient avec une position *référence*, savoir s'il est en train de bouger, si ce mouvement est *dangereux*, être capable d'extraire des données quantitatives de différentes natures (volume de la cage thoracique, fréquence du rythme respiratoire...). La réalité augmentée, par le biais de la vision par ordinateur, peut apporter une solution à ces différentes questions. Pour mettre en place un tel système en contexte médical, nous détaillons les conditions requises, les avantages et inconvénients des méthodes *classiques* existantes, et nous proposons une nouvelle approche répondant à ces besoins.

2 Acquisition de mouvements

L'acquisition de mouvements a suscité des recherches bien avant que l'on ne songe à une utilisation à des fins médicales; aussi toutes les solutions avancées ne sont pas adaptées au contexte. De plus, précisons que dans le cadre d'acquisition de mouvements respiratoires, on parlera plus de déformation de la cage thoracique que d'un mouvement de déplacement. Même si l'on n'exclut pas de vouloir suivre le déplacement du patient, les conditions d'un traitement médical imposent à ce dernier une certaine immobilité (coque de contention). Une certaine partie des applications d'acquisition de mouvements se destinent à de l'animation

de personnages (cinéma, jeux-vidéos...); les mouvements à acquérir sont donc d'une nature différente de ceux d'un patient allongé sur une table.

2.1 Capteurs

Certains dispositifs permettent de donner leur position (relative ou absolue) dans l'espace, en se fondant sur des technologies de capteurs d'inertie [Xse] ou capteurs magnétiques [ASMA00]. Les progrès réalisés dans ces domaines ont permis de réduire l'encombrement de ces derniers, et donc de les placer sur une personne en lui laissant une liberté de mouvements (systèmes sans-fils notamment). Bien que très efficaces pour l'animation de personnages, ces méthodes sont peu adaptées à une utilisation médicale; même si les capteurs sont *portables*, ils peuvent gêner le *naturel* des mouvements à acquérir. De plus, dans le cadre de l'acquisition des déformations thoraciques, la résolution d'acquisition pourrait impliquer un bien trop grand nombre de capteurs par rapport à la surface considérée. Enfin, il n'est pas toujours possible d'envisager la présence de dispositifs magnétiques dans une salle de traitement (radiothérapie par source ionisante par exemple), sans compter qu'il peut être nécessaire d'implanter des marqueurs à l'intérieur du corps du patient [ASMA00], ce que nous souhaitons éviter. D'autres méthodes proposent d'utiliser des *exo-squelettes* pour suivre le mouvement d'une articulation précise [Ani]; ceci ne semble pas applicable pour le suivi de mouvements respiratoires.

2.2 Vision par ordinateur

La vision par ordinateur propose de nombreuses méthodes d'acquisition de données volumétriques, voire de mouvements; elles sont en général moins invasives que les méthodes basées sur des capteurs. L'utilisation de marqueurs permet d'obtenir des résultats similaires à l'utilisation de capteurs; bien que les dispositifs employés soient moins complexes, les marqueurs physiques peuvent toujours être considérés comme gênant selon la nature des mouvements à observer. La nature des systèmes optiques utilisés permet de répartir les différentes méthodes de vision en deux classes. Lorsque des caméras seulement sont utilisées on parlera de méthode *passive*, en opposition aux méthodes *actives* utilisant des systèmes optiques permettant de projeter des motifs lumineux (vidéo-projecteur, laser...). Quel que soit le choix effectué, la difficulté des méthodes de vision réside dans le problème *d'appariement*, ou de *mise en correspondance*: comment déterminer les correspondances entre les pixels de deux systèmes optiques distincts? Un point de l'espace peut être reconstruit en 3D s'il est identifié sur les *écrans* d'au moins deux systèmes optiques. L'un des outils principaux dans cette tâche est la géométrie épipolaire [Fau93], qui fournit une contrainte nécessaire à l'appariement de deux points images.

2.2.1 Méthodes passives

L'utilisation de marqueurs est une approche passive en vision [Moc]; son principal avantage est de mettre certains points de l'espace en évidence pour effectuer les mises en correspondance de ceux-ci. En s'interdisant l'utilisation de marqueurs pour limiter l'invasivité du procédé, il faut détecter des détails dans l'image en se basant sur des caractéristiques de formes, de couleurs, et sur la géométrie épipolaire (exemple: identifier le coin d'une table, les carreaux d'un carrelage); ceci reste difficile sur une surface homogène comme un thorax.

Certains travaux utilisent différentes silhouettes obtenues à partir d'autant de caméras placées autour du sujet [MGB06]; ces vues partielles permettent de *sculpter* un volume dans l'espace. La précision du résultat est alors directement liée au nombre de caméras, et les mouvements acquis s'apparentent le plus souvent à des déplacements, et non à des déformations. Pour finir, l'utilisation d'un modèle théorique du sujet observé permet de réaliser des opérations de recalage pour faire coïncider ce qui est vu avec le modèle [DL05]. Ces méthodes nécessitent souvent un modèle différent pour chaque situation nouvelle, et la précision obtenue dépend de la complexité du modèle.

4 CONCLUSION

2.2.2 Méthodes actives

La projection de différents motifs à l'aide d'un vidéo-projecteur permet de créer des marqueurs virtuels sans gêner les mouvements du sujet. L'identification de détails est donc plus aisée, reste donc le problème de mise en correspondance de ces détails. Notons que celle-ci doit pouvoir être réalisée en une seule image (méthode *one-shot*) pour préserver l'interactivité de l'application ; certaines méthodes ne travaillent que sur la reconstruction d'objets statiques (œuvres d'art, maquettes...), il est alors possible de mettre en place des algorithmes procédant en plusieurs *passes* de motifs. Pour contourner cette dernière difficulté, différentes approches existent.

La projection de motifs colorés [SPB04] [ZCS02] permet d'identifier chaque motif en fonction de sa couleur propre, ainsi que de celle de ses *voisins*. Le principe est de projeter un ensemble de motifs simples, agencés de telle manière que le *sur-motif* constitué d'un ensemble de motifs voisins soit unique dans toute la scène [JPF05]. Bien qu'il soit possible de distinguer clairement des motifs distincts avec ces méthodes, elles nécessitent une certaine continuité de la surface observée ; l'utilisation des couleurs peut aussi être un obstacle si la peau du sujet observé n'est pas de couleur homogène (cicatrices, tâches de naissance...). Des travaux similaires se servent d'hypothèses de continuité sur la surface thoracique pour résoudre le problème d'appariement avec des méthodes d'interférométrie [FLB99] ; les occultations sont hélas un problème incontournable.

3 Nouvelle approche

En s'appuyant sur les difficultés rencontrées dans les méthodes actuelles, nos travaux ont eu pour but de proposer de nouveaux outils pour s'affranchir du plus grand nombre d'hypothèses sur la scène observée ; sous ces conditions, seule la géométrie épipolaire peut nous permettre de résoudre le problème d'appariement. En stéréo-vision (passive ou active), la contrainte épipolaire est rarement suffisante pour effectuer des mises en correspondance dans beaucoup de situations (c'est une condition nécessaire mais pas suffisante). Pour contourner cette difficulté, nous utilisons trois systèmes optiques : deux caméras, et un vidéo-projecteur projetant n points lumineux indiscernables les uns des autres. Leur couleur unique et leur forme simple (ponctuelle) permet de les détecter dans beaucoup plus de cas : surface presque tangente au rayon lumineux, couleur de peau non homogène, différents supports...

En modélisant la géométrie épipolaire de chaque point image (caméra ou vidéo-projecteur) à travers un *graphe d'appariement*, nous proposons un algorithme optimal, de complexité globale $O(n \log n)$, pour la résolution du problème d'appariement (n étant le nombre de points vidéo-projetés). La géométrie épipolaire à elle seule ne peut pas suffire à résoudre tous les problèmes d'appariements d'un point de vue théorique ; néanmoins nous montrons que toutes les mises en correspondance *non ambiguës* sont retrouvées par notre algorithme. Dans la pratique, l'implémentation actuelle de ces travaux permet l'acquisition d'environ 150 points sur un thorax à l'aide de deux caméras 640×480 et d'un vidéo-projecteur 1280×720 , à une fréquence d'environ 2 images par secondes.

La figure 1 illustre un exemple de résultats obtenus ; à gauche et droite respectivement ce que voient les caméras, au centre la reconstruction effectuée. Les différentes annotations sur les images des caméras correspondent à des codes d'appariement ajoutés sur l'image en post-traitement (ils ne sont pas présents sur la scène observée). On remarque que les points situés de part et d'autre du sujet ne sont vus que par une seule caméra ; la résolution progressive des contraintes épipolaires permet néanmoins d'en apparier un certain nombre.

4 Conclusion

Nos travaux proposent une alternative aux méthodes déjà existantes en lumière structurée ; ils permettent d'utiliser au maximum la seule géométrie épipolaire. L'utilisation de trois systèmes optiques permet de contourner les limitations existantes en stéréo-vision, tout en s'adaptant aux éventuelles occultations de points d'une caméra. L'utilisation d'un graphe nous permet de modéliser la géométrie épipolaire de la situation observée, et de résoudre efficacement le problème de mise en correspondance.

 RÉFÉRENCES

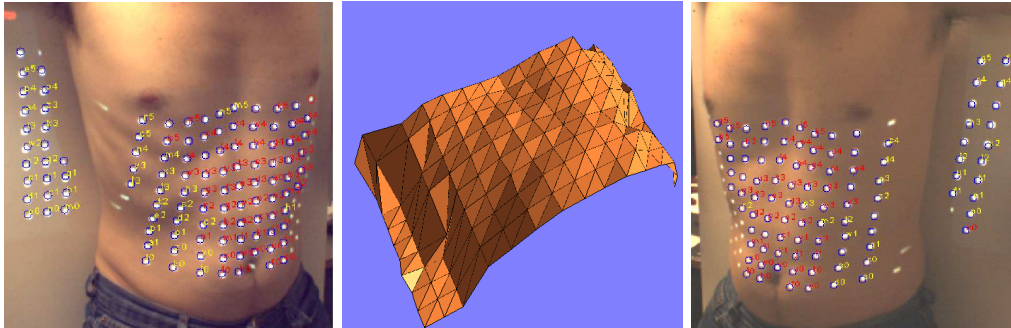


FIG. 1 – Exemple de reconstruction

Ces résultats nous encouragent à coupler la robustesse de nos travaux avec des méthodes *multi-résolution* déjà existantes [WNH05] pour améliorer l'échantillonnage de la surface observée. La robustesse de l'acquisition devrait permettre le pilotage interactif d'outils, développés au LIRIS, simulant les mouvements d'organes internes [VBS*04]. L'intégralité de ces travaux, modèles et preuves sont en cours de soumission à une revue.

5 Références

Références

- [Ani] ANIMAZOO : Gypsy motion capture system. <http://www.animazoo-europe.com>.
- [ASMA00] A. SCHWEIKARD G. GLOSSER M. B., MURPHY M., ADLER J. : Robotic motion compensation for respiratory movement during radiosurgery. *Computer Aided Surgery* 5 (2000), 263–277.
- [DL05] DESSERÉE E., LEGRAND L. : First Results of a Complete Marker-Free Methodology for Human Gait Analysis. In *IEEE EMBC 2005* (Sept. 2005).
- [Fau93] FAUGERAS O. D. : *Three-Dimensional Computer Vision : A Geometric Viewpoint*. MIT Press, 1993.
- [FLB99] F. LILLEY M. L., BURTON D. : Robust fringe analysis system for human body shape measurement. *Optical Engineering* 39, 1 (1999), 187–195.
- [JPF05] JORDI PAGÈS JOAQUIM SALVI C. C., FOREST J. : Optimised de bruijn patterns for one-shot shape acquisition. *Image and Vision Computing* 23, 8 (2005), 707–720.
- [MGB06] MICHOD B., GUILLOU E., BOUAKAZ S. : Shape From Silhouette : Towards a Solution for Partial Visibility Problem. In *Eurographics 2006* (Sept. 2006), C.Hansen D., (Ed.), Eurographics 2006 Short Papers Preceedings, pp. 13–16.
- [Moc] MOCAP.LT : Motion capture studio. <http://www.mocap.lt>.
- [SPB04] SALVI J., PAGES J., BATTLE J. : Pattern codification strategies in structured light systems, 2004.
- [VBS*04] VILLARD P.-F., BEUVE M., SHARIAT B., BAUDET V., JAILLET F. : Lung Mesh Generation to Simulate Breathing Motion with a Finite Element Method. In *Information Visualisation* (July 2004), IEEE Computer Society, pp. 194–199.
- [WNH05] WONG A. K. C., NIU P., HE X. : Fast acquisition of dense depth data by a new structured light scheme. *Comput. Vis. Image Underst.* 98, 3 (2005), 398–422.
- [Xse] XSSENS : Motion technologies, inertial motion capturing. <http://www.xsens.com>, moven project.
- [ZCS02] ZHANG L., CURLESS B., SEITZ S. M. : Rapid shape acquisition using color structured light and multi-pass dynamic programming. In *The 1st IEEE International Symposium on 3D Data Processing, Visualization, and Transmission* (June 2002), pp. 24–36.

Real-time and Markerless Full-Body Human Motion Capture

Brice Michoud¹, Erwan Guillou¹ and Saïda Bouakaz¹

forename.name@liris.cnrs.fr

Université Claude Bernard Lyon 1 - Liris
8 Boulevard Niels Bohr, 69622 Villeurbanne, France

Abstract. We propose an efficient real-time method for markerless 3D human motion capture that requires a single computer. Using input from at least three calibrated webcams, an extended "Shape from Silhouette" algorithm reconstructs the filmed person in real-time. Fast 3D shape and 3D skin parts analysis provide a robust and real-time system for human full-body tracking. Animation skeleton and simple morphological constraints ease the motion capture process. Thanks to fast and simple algorithms, and appliance to low-cost cameras, our system is well suited for home entertainment device. Results on long video sequences with fast and complex movements, demonstrate our approach robustness.

1 Introduction

Marker-free motion capture has long been studied in computer vision as classic and fundamental problems. While commercial products are already available for real-time marker-use, robust on-line marker-free systems remains an open issue because many real-time algorithms still lack robustness. While most popular techniques run on pc cluster, our system require at least three low-cost cameras (as webcams) and a single computer. In this paper we propose a fully automated human-machine interaction device for home entertainment (see Fig. 2(a)). Because interactions constraint, our system works in real-time (at least 30 fps), without markers (active or passive) or any particular sensors.

To tackle the marker-free motion capture problem, several techniques has been proposed (according to camera numbers and feature analysis). We only review some works related to our work. Among various types of approaches many methods work with a single camera [1, 2]. When the method is based on the object silhouette, they suffers from ambiguous response in case of local minimum as different positions can yield the same silhouette. Other methods use multiple cameras. Some of them work only on a 3D human shape analysis [3, 4]. These techniques provide good results when the 3D shape topology correspond to the filmed human topology *i.e.* each body parts is clearly identifiable in the estimated 3D shape. With self-occlusion cases or large contacts between limbs and body these techniques frequently fail. Caillette *et al.* [5] method involves shape and color clues. They link colored blobs to a kinematic model to track individual body parts. This technique requires contrasted clothing between each body parts for tracking, thus adding an usability constraint. Few methods provide real-time motion capture. Most of them run only with interactive frame rate (10 fps for [5]).

We therefore propose a fully automated system for practical real-time motion capture from at least three calibrated webcams. Our method runs at 30 fps because, it is based on simple heuristics, driven by shape and skin parts topology analysis, and temporal coherence.

Figure 1(a) outlines the two main stages of our method : 3D reconstruction and analysis of the 3D information. In the first section we present our work for real-time 3D reconstruction. We explain in the second section our method for real-time full-body pose estimation. Then, we discuss on the results obtained from real and complex data. Finally, we conclude about our contributions, and we present some perspectives for this work.

2 3D shape and skin parts estimation

We propose extensions for "Shape From Silhouette" (SFS) algorithms, which reconstruct in real-time 3D shape and 3D skin color parts of the person from calibrated cameras.

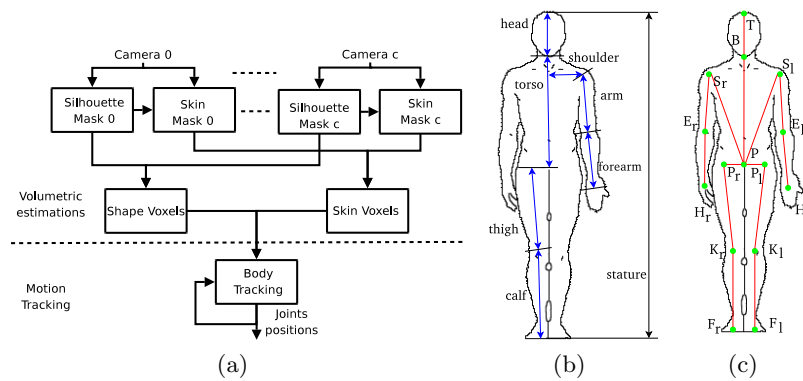


Fig. 1. (a) System overview: Reconstruction algorithms and Pose estimation algorithms. Body parts labeling (b) and joint naming (c).

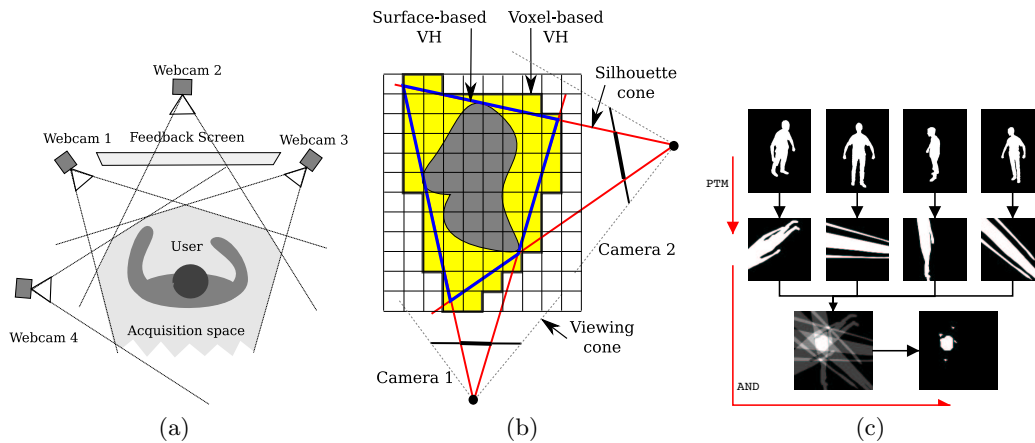


Fig. 2. (a) Interaction setup. (b) Object reconstruction by surface and volumetric approaches represented in 2D. SFS computation using "Projective Texture Mapping" method: First all silhouette masks are projected on a stack, logical AND is used to compute the projection intersection.

Currently only SFS methods compute in real-time 3D shape estimation of an object, from its silhouette images. Silhouette images are binary masks corresponding to captured images where 0 correspond to background, or 1 stands for the (interesting) feature of the object. The formalism of SFS was introduced by A. Laurentini [6]. By definition, an object lies inside the volume generated by back-projecting its silhouette through the camera center (called silhouette's cone). With multiple views of the same object at the same time, the intersection of all the silhouette's cones build a volume called "Visual Hull", which is guaranteed to contain the real object. There are mainly two ways to compute an object Visual Hull.

Surface-based approaches Surface-based approaches compute silhouette cone surface intersections (see Fig. 2(b)). First silhouettes are converted to polygons. Each edge is back-projected to form a 3D polygon. Then each 3D polygon is projected onto each other images, and intersected with each silhouette in 2D. The resulting polygons are assembled to form polyhedral shape estimation (see [7, 8]). Resulting Surface-based shape from silhouette is underlined Fig. 2(b). These approaches are not well suited to our application because of the complexity of the underlying geometric calculations that are not real-time on a single computer. Incomplete or corrupted surface models can be created, directly depending on polyhedron sharpness and silhouette noise.

Volumetric-based approaches Volumetric-based approaches [3, 9–11] generally estimate shape by processing a set of voxels. The object acquisition area is split up into a 3D grid of voxels (volume elements). Each voxel remains part of the estimated shape if its projection in all images lies in all silhouettes (see Fig. 2(b)). This volumetric approach is adapted for real-time pose estimation, due to its fast computation and robustness to noisy silhouettes.

We propose a new framework which computes a 3D volumetric shape and skin parts estimation on a single computer. After 3D Shape from silhouette estimation on GPU, we compute voxels visibility. Then all visible voxel which project themselves on skin masks, are then classified as skin voxels. First we explain camera calibration, silhouette segmentation and skin segmentation steps, which are input data for 3D estimations. Then GPU SFS implementation, voxel visibility computation, and skin voxel computation are presented in second part.

2.1 Input Data

First, webcams are calibrated using the method proposed by Zhang *et al.* [12] which is one of the most popular calibration algorithm. A Color calibration step is added to enforce coherency between the two webcams using the method proposed by N.Joshi [13].

Second step consists in silhouette segmentation (see [14] for silhouette segmentation algorithm comparative study). Then we assume that the background is static and the subject moves. We use the method proposed by [9]. In beginning we acquire images of background (without user). The user is then detected in the pixels whose value has changed. By hypothesis only one person is in the field of view of webcams, then it is represented only by one connex component. Due to webcam noise, we can have several connex parts, but the smallest are considered as noise.

Last step before voxels computation, we extract skin parts from silhouettes and color images. Normalized Look-up Table method [15] provides fast skin color segmentation. This segmentation is applied to each images limited to silhouette mask because skin color pixels outside to the silhouette correspond to background pixels.

2.2 GPU SFS implementation

Volumetric SFS is generally based on voxel projection: a voxel remains part of the estimated shape if it projects itself into each silhouette. To better fit a GPU implementation we choose the opposite: we project each silhouette into the 3D voxel grid as proposed in [9]: if a voxel is intersected by all the silhouettes projections, then it represents the original object. The classical N^3 voxel cube can be considered as a stack of N images of resolution $N \times N$. We stack the N image in screen parallel planes. For each camera view, silhouette masks are projected on each slice using the "projective texture mapping" technique [16]. Intersection of silhouettes projections on all slices provides voxel-based 3D shape. Intersection of silhouette mask projections on a single slice is underlined Fig 2(c). To save video bus bandwidth, computations for a voxel cube are made in the same frame buffer, which is tiled by all the N slices of resolution $N \times N$.

To estimate skin voxels, we compute each voxel visibility from each camera. The voxel visibility is based on Item Buffer method used in some voxels coloring algorithms [17]. An unique identifier is associated to each voxel (like color) and voxels are rendered on raster based frame buffers, corresponding to each cameras views. For each frame buffers, colors describes visible voxels and this enable bidirectional pixel to voxel mapping. If a voxel is skin consistent (*i.e.* it is mapped to skin mask pixels in all of its viewing camera) then it is classified as skin voxel. To improve visibility computation time, only surface voxels (*i.e.* voxels which have less than 26 neighbors) are tested.

To reduce computation time for pose estimation we propose to keep the visible voxels. Let $\mathcal{V}_{\text{skin}}$ be the the selected voxels form shape voxel set, $\mathcal{V}_{\text{skin}}$ be the skin consistent voxel set, and \mathcal{V}_{all} be their union.

Our implementation provides up to 100 reconstructions per second. As webcam acquisition is done at 30 fps, it allows us to save time for motion capture calculus, hence achieving our real-time goal.

3 Motion Capture

Motion capture is equivalent to determine the pose of the body. It may be seen as classifying each voxel to a body part. Joints labeling is presented in Figure 1(c). We propose a system based on simple and fast heuristics. Less accurate than the registration based methods, this approach nonetheless run at real-time. Robustness is increased by using a multi-modal scheme composed on both shape and skin parts analysis, temporal coherence and human anthropometric constraints.

Our system runs on two steps: initialization and tracking; both use the same algorithm with different initial conditions. The initialization step presented section 5 estimates anthropometric values, and the initial pose. Then using this information, the second step tracks joint positions (see section 4). The only assumption is that both hands and person's face are partially uncovered, that the torso is dressed, and that the clothing have a non-skin color. We present here some common notations that the reader can refer to:

L_x denotes the length of body part x (see Fig. 1(b)),

D_x its orientation and

R_x its radius (of sphere or cylinder).

J^n denotes the value of a quantity J (joint position, voxel set...) at frame n .

l and r indices denote respectively left and right side

\mathcal{V}_x denotes a set of voxel,

$\mathcal{E}_{\mathcal{V}_x}$ its inertia ellipsoid and

$Cog(\mathcal{V}_x)$ its gravity center.

$J(i)$ denotes the J quantity value at step i when dealing with iterative algorithms.

4 Body Parts Tracking

To achieve body parts tracking we assume to be known the previous body pose and anthropometric estimations. Using 3D shape estimation and 3D skin parts we track the human body parts in real-time. The tracking process works on active voxels \mathcal{V}_{act} . This set of voxels is initialized to all voxels \mathcal{V}_{all} and updated at each step by removing voxels used to estimate body parts. First we estimate head joints. Torso is connected to head, then we next track the torso. In the end we compute limb joints that are connected to torso.

4.1 Head Tracking

This step aims to find T^n and B^n , the positions of the top of the head and the connection point between head and neck at frame n .

Let \mathcal{V}_{face}^n be the face's voxels at the current frame. By hypothesis \mathcal{V}_{skin}^n contains face and hands voxels. Using Temporal coherency criteria \mathcal{V}_{face}^n is the nearest connex component of \mathcal{V}_{skin}^n from the previous set of face voxels \mathcal{V}_{face}^{n-1} .

The center of the head C^n is computed by fitting a sphere $S(i)$ in \mathcal{V}_{act}^n (see figure 3). $S(i)$ is defined by its center $C^n(i)$ and radius R_{head} .

head fitting algorithm $C^n(0)$ is initialized as the centroid of \mathcal{V}_{face}^n .

At step i of the algorithm, $C^n(i)$ is the centroid of the set $\mathcal{V}_{head}^n(i)$ of active voxels that lie into a sphere $S(i-1)$ defined by its center $C^n(i-1)$ and its radius R_{head} (see Fig. 3(a)).

The algorithm iterates until step k when the position of C^n stabilizes, *i.e.* the distance between $C^n(k-1)$ and $C^n(k)$ falls below a threshold ϵ_{head} .

head joints estimation Knowing C^n position, B^n (respectively T^n) is computed as the lower (resp. upper) intersection between $S(k)$ and the principal axis of $\mathcal{E}_{\mathcal{V}_{head}^n}$ (see Fig. 3(b)).

The back-to-front direction D_{b2f}^n is defined as the direction from C^n towards the centroid of \mathcal{V}_{face}^n (note that voxels from the back of the head are not in \mathcal{V}_{skin}). At this point, we remove from \mathcal{V}_{act}^n the set of elements that belongs to \mathcal{V}_{head}^n .

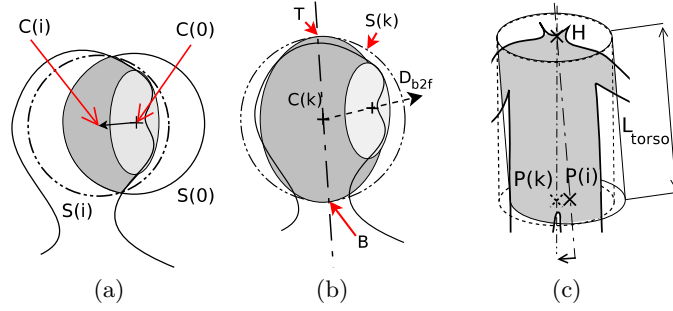


Fig. 3. (a) Sphere fitting (light gray denotes $\mathcal{V}_{\text{face}}^n$, dark gray denotes $\mathcal{V}_{\text{head}}^n(i)$), (b) joints estimation and (c) torso segmentation by cylinder fitting.

4.2 Torso Tracking

This step aims to find P^n the pelvis position, by fitting a cylinder in $\mathcal{V}_{\text{act}}^n$. Torso shape estimated by a cylinder provides simple and fast pelvis localization method. Let $\mathcal{V}_{\text{torso}}^n$ be the set of voxels that describes the torso, they are initialized using voxels $\mathcal{V}_{\text{act}}^n$. At step i , the algorithm estimates D_{torso}^n by fitting a cylinder $\mathcal{CYL}(i-1)$ in $\mathcal{V}_{\text{torso}}^n(i)$ (see Fig 3(c)). $\mathcal{CYL}(i)$ has a cap anchored at B^n , as radius R_{torso} , its length is L_{torso} and its axis is $D_{\text{torso}}^n(i)$.

Torso fitting algorithm $\mathcal{V}_{\text{torso}}^n(0)$ is initialized with $\mathcal{V}_{\text{act}}^n$ and the vector from B^n to P^{n-1} define $D_{\text{torso}}^n(0)$ initial value.

At step i , $\mathcal{V}_{\text{torso}}^n(i)$ is computed as the set of elements from $\mathcal{V}_{\text{torso}}^n(i-1)$ that lie in $\mathcal{CYL}(i-1)$. $D_{\text{torso}}^n(i)$ is then the principal axis of $\mathcal{E}_{\mathcal{V}_{\text{torso}}^n(i)}$ (see Fig. 3(c)).

The algorithm iterates until step k when the distance between the axis of $\mathcal{CYL}(k)$ and the centroid of $\mathcal{V}_{\text{torso}}^n(k)$ falls below a threshold ϵ_{torso} . P^n position is defined as the center of the lower cap of $\mathcal{CYL}(k)$

Global body orientation The top-down orientation D_{t2d}^n of the acquired subject is given by $P^n - B^n$. D_{b2f}^n was computed in 4.1. The left-to-right orientation D_{l2r}^n of the acquired subject is given by $D_{\text{l2r}}^n = D_{\text{t2d}}^n \times D_{\text{b2f}}^n$.

$\mathcal{V}_{\text{act}}^n$ is then updated by removing its elements that belongs to $\mathcal{V}_{\text{torso}}^n$.

4.3 Hands and forearms Tracking

We propose a simple and robust algorithm to compute the forearms joints positions. First we compute hands position from skin voxels. Helped by given anthropometric measurement of forearm length, we determine the elbows positions. Temporal coherence is used to compute their sides.

Let $\mathcal{V}_{\text{hand}}^n$ be the set of potential voxels of hands. $L_{\text{stat}}/2$ is a raising of arm length. $\mathcal{V}_{\text{hand}}^n$ is defined by the voxels of $\mathcal{V}_{\text{skin}}^n - \mathcal{V}_{\text{face}}^n$ that lie within a sphere defined by its center B^n and its radius $L_{\text{stat}}/2$. By hypothesis $\mathcal{V}_{\text{skin}}^n$ contains hands and face voxels. The different forearms configurations are:

Two distinct hands : $\mathcal{V}_{\text{hand}}^n$ contains several connex components. Let $\mathcal{V}_{\text{hand}0}^n$ and $\mathcal{V}_{\text{hand}1}^n$ be the two biggest, corresponding to the two hands with $H_x^n = \text{Cog}(\mathcal{V}_{\text{hand}x}^n)$ with $x \in [0, 1]$.

Forearms have constant length L_{farm} across time. The potential voxels for forearm $_x$ are the voxels from $\mathcal{V}_{\text{act}}^n$ which lies within a sphere of radius L_{farm} , centered in H_x^n . The connex component of these voxels which contains H_x^n represents the forearm $_x$. Let $\mathcal{V}_{\text{farm}x}^n$ be this connex component; there are two possible cases to identify elbow.

If forearms did not collide *i.e.* $\mathcal{V}_{\text{farm}0}^n \cap \mathcal{V}_{\text{farm}1}^n = \emptyset$, then we use the principal axis of $\mathcal{E}_{\mathcal{V}_{\text{farm}x}^n}$ and L_{farm} to compute the elbow position E_x^n . The sides are computed using temporal coherence

criteria: the side of the forearm_x is the same than the closest forearm computed at the previous frame.

Else forearms collide and $\mathcal{V}_{\text{farm}0}^n \cap \mathcal{V}_{\text{farm}1}^n \neq \emptyset$. First we identify the hand sides by the property of constant forearms length. H_x^n is right sided if

$$\|d(H_x^n, E_r^{n-1}) - L_{\text{farm}}\| < \|d(H_x^n, E_l^{n-1}) - L_{\text{farm}}\|$$

else H_x^n is left sided. The voxels v_i of $\mathcal{V}_{\text{farm}0}^n \cup \mathcal{V}_{\text{farm}1}^n$ are segmented in two parts $\mathcal{V}_{\text{farm}r}^n$ and $\mathcal{V}_{\text{farm}l}^n$ using point to line mapping algorithm (see 4.5). If v_i is more close to $[H_r^n E_r^{n-1}]$ than $[H_l^n E_l^{n-1}]$, v_i is added on $\mathcal{V}_{\text{farm}r}^n$. Else v_i is added on $\mathcal{V}_{\text{farm}l}^n$. Principal axis of $\mathcal{E}_{\mathcal{V}_{\text{farm}r}^n}$, $\mathcal{E}_{\mathcal{V}_{\text{farm}l}^n}$ and L_{farm} are used to compute E_r^n and E_l^n .

One hand or jointed hands : $\mathcal{V}_{\text{hand}}^n$ contains only one connex component and it corresponds to jointed hands or to only one hand (the other is not visible). We use the temporal coherence to disambiguate these two cases.

If H_r^{n-1} and H_l^{n-1} are close to $\mathcal{V}_{\text{hand}}^n$, then the hands are jointed and $H_r^n = H_l^n = \text{Cog}(\mathcal{V}_{\text{hand}}^n)$ and we compute $\mathcal{V}_{\text{farm}}^n$ as proposed previously. We segment $\mathcal{V}_{\text{farm}}^n$ in two parts $\mathcal{V}_{\text{farm}r}^n$ and $\mathcal{V}_{\text{farm}l}^n$ by the orthogonal plane to $[E_r^{n-1} E_l^{n-1}]$ containing H_l^n . Principal axis of $\mathcal{E}_{\mathcal{V}_{\text{farm}r}^n}$, $\mathcal{E}_{\mathcal{V}_{\text{farm}l}^n}$ and L_{farm} are used to compute E_r^n and E_l^n .

Else the closest hand H_x^{n-1} to $\mathcal{V}_{\text{hand}}^n$ is used to compute the side of H_x^n and $H_x^n = \text{Cog}(\mathcal{V}_{\text{hand}}^n)$. We compute $\mathcal{V}_{\text{farm}}^n$ as proposed previously and its principal axis of inertia is used to compute E_x^n .

No visible hand : $\mathcal{V}_{\text{hand}}^n$ is empty, then no hand is visible. We take back the positions computed at the $n - 1$ frame to the current frame.

In all case $\mathcal{V}_{\text{act}}^n$ is updated by removing its elements that belongs into forearm or hand.

4.4 Shoulders Tracking

We have estimated articulations positions of the head, the torso, the hands and the elbows. To finalize upper body tracking, we compute shoulders positions. As we argue that arms are in a sphere centered on bottom head, with a radius of $L_{\text{stat}}/2$, then voxels of $\mathcal{V}_{\text{act}}^n$ which are in this sphere, contain arms voxels and noise voxels. Let $\mathcal{V}_{\text{arms}}^n$ be the set these voxels.

Elbow is on one extremity of arm, then the second estimates shoulder. We know the current position of elbow, then we determine arm voxels. Let $\mathcal{V}_{\text{arm}x}^n$ (where x corresponds to the side) be the closest ¹ connex component of $\mathcal{V}_{\text{arms}}^n$ to E_x^n . Furthermore arm length L_{arm} is constant, then current shoulder position S_x^n for the x side is given by:

$$S_x^n = E_x^n + \frac{\text{Cog}(\mathcal{V}_{\text{arm}x}^n) - E_x^n}{|\text{Cog}(\mathcal{V}_{\text{arm}x}^n) - E_x^n|} L_{\text{arm}}$$

$\mathcal{V}_{\text{act}}^n$ is updated by removing its elements that belongs into each arm.

4.5 Legs Tracking

All body parts but the legs have been estimated, hence $\mathcal{V}_{\text{act}}^n$ contains only the legs voxels. Our leg joints extraction is inspired from "point to line mapping" process used to bind an animation skeleton on a 3D mesh [18]. The elements of $\mathcal{V}_{\text{act}}^n$ are split up into four sets $\mathcal{V}_{\text{thigh}l}^n$, $\mathcal{V}_{\text{calf}l}^n$, $\mathcal{V}_{\text{thigh}r}^n$ and $\mathcal{V}_{\text{calf}r}^n$ depending of their euclidean distance to segments $[P_l^{n-1}, K_l^{n-1}]$, $[K_l^{n-1}, F_l^{n-1}]$, $[P_r^{n-1}, K_r^{n-1}]$, and $[K_r^{n-1}, F_r^{n-1}]$ (see Fig. 4(a)). For the left/right side x , we compute the inertia ellipsoid $\mathcal{E}_{\mathcal{V}_{\text{calf}x}^n}$ (let Ex_0 and Ex_1 be its extrema points) and the inertia ellipsoid $\mathcal{E}_{\mathcal{V}_{\text{thigh}x}^n}$.

The knee is the intersection point of thigh and calf (Fig. 4(b)), hence the foot position F_x^n is given by the farthest extrema point of $\mathcal{E}_{\mathcal{V}_{\text{calf}x}^n}$ from the inertia ellipsoid of $\mathcal{V}_{\text{thigh}x}^n$ (let say it's Ex_1). Then knee is aligned on $[Ex_0 Ex_1]$, Ex_0 sided, at a L_{calf} distance of F_x^n . Hip position P_x^n is given by the farthest extrema point of $\mathcal{E}_{\mathcal{V}_{\text{thigh}x}^n}$ from the inertia ellipsoid of $\mathcal{V}_{\text{calf}x}^n$, corrected to be at a L_{thigh} distance of K_x^n .

¹ In term of euclidean distance

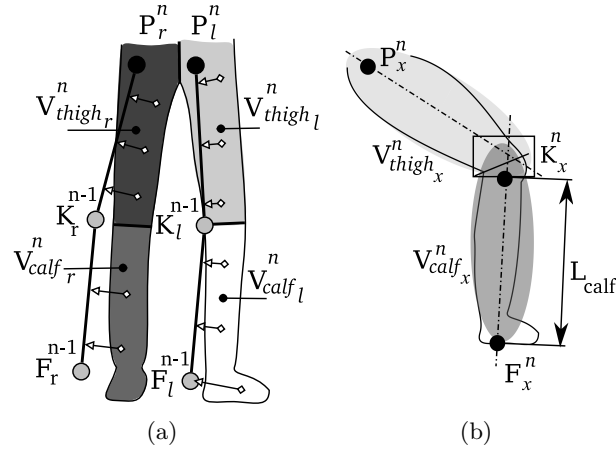


Fig. 4. (a) torso segmentation by cylinder fitting. (b) the "binding" step of legs tracking and (c) legs articulations estimation.

5 Body Parts Initialization

We present in this section our techniques to estimate the anthropometric measures and the initial body pose. The literature in connection with this step can be classified in three categories. In the first one [7], the dimensions and initial pose are manually specified. Second kind of methods need of an initialization pose like T-pose [19]. These methods are real-time. The third class is composed by fully automated methods [3] which are generally non real-time processes. Our approach is real-time and fully automated for any kind of movements as long as the filmed person is standing up, his/her hands are below the level of the head, and his/her feet are not joined. After anthropometric estimations, our method computes each body parts parameters sequentially with the tracking step ordering.

Anthropometric Measurements They correspond to lengths of each body parts[20]. We have estimated some anthropometric measures as average ratios of the human body length. Let L_{stat} be the acquired human body length, estimated as the maximum distance of foreground voxels to floor plane. Hence, knowing L_{stat} , guesses for anthropometric measures are given by these ratios:

$$\begin{aligned} R_{head} &\approx L_{stat}/16 & L_{torso} &\approx 3L_{stat}/8 & L_{calf} &\approx L_{stat}/4 \\ L_{farm} &\approx L_{stat}/6 & L_{arm} &\approx L_{stat}/6 & L_{thigh} &\approx L_{stat}/4 \end{aligned}$$

Like for tracking step, active set of voxels \mathcal{V}_{act} is initialized by all voxels \mathcal{V}_{all} .

Head Initialization This step aims to find T^0 and B^0 . From our initialization hypothesis, the face's voxels \mathcal{V}_{face}^0 of acquired subject are defined by the top most connex component among \mathcal{V}_{skin}^0 . Then Head Tracking algorithm (section 4.1) is applied to compute T^0 and B^0 , without estimation of the face position step. \mathcal{V}_{act}^0 is updated by removing elements that belongs to \mathcal{V}_{head}^0 .

Torso Initialization The torso fitting algorithm (section 4.2) is applied using \mathcal{V}_{act}^0 as initial value for $\mathcal{V}_{torso}^0(0)$. $D_{torso}^0(0)$ is initialized as the vector from N^0 toward the centroid of $\mathcal{E}_{\mathcal{V}_{act}^0(0)}$. Pelvis position P^0 , D_{t2d}^0 and D_{l2r}^0 are then computed. \mathcal{V}_{act}^0 is updated by removing the elements that belongs to \mathcal{V}_{torso}^0 .

Arms Initialization We initialize hands and forearms positions using the tracking algorithm presented Section 4.3. We have no previous arms position, then we can only compute forearms positions when there is two distinct forearms. Having this criteria verified, we can compute H_r^0 , H_l^0 , E_r^0 and E_l^0 . \mathcal{V}_{act}^0 is updated by removing its elements that belongs into forearms. Shoulders

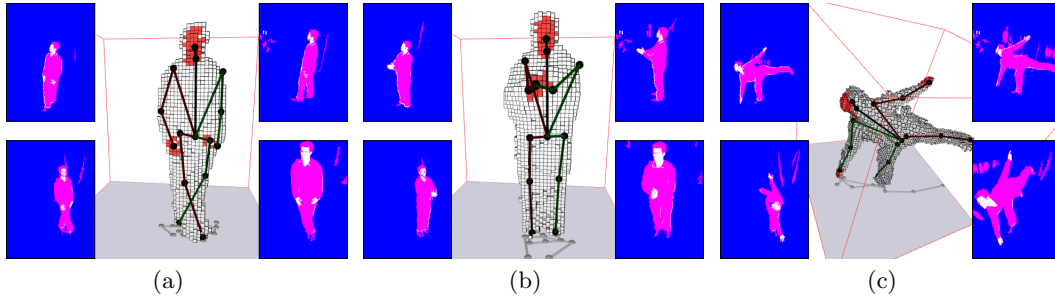


Fig. 5. (a) (b) and (c) underline results for challenging poses. The user recovered pose is presented as an animation skeleton having right sided parts in red and left sided parts in green. Shape voxels are presented in white and skin voxels in red.

positions S_r^0 and S_l^0 are initialized using directly the shoulder tracking algorithm presented part 4.4.

Legs Initialization Tracking algorithm outlined Section 4.5 need of legs previous positions. We simulate them by a coarse estimation of knees, feet and hips articulations, then we compute more precise position of the legs articulations using the legs tracking algorithm.

\mathcal{V}_{act}^0 contains the voxels that haven't been used for any other parts of the body. First we compute the set of connex components from elements of \mathcal{V}_{act}^0 having their height below $L_{stat}/8$. If there is less than 2 connex components, we assume that feet are joined and can't be distinguished. Otherwise we use the two major connex components \mathcal{V}_{footl}^0 and \mathcal{V}_{footr}^0 . Left and right assignation of voxel's set is done using the left-to-right vector D_{12r} . For the left/right side x , let v_x be the vector from P^0 to the centroid of \mathcal{V}_{footx}^0 . Knee and Foot joints are guesses using the following equations:

$$K_x^{-1} = P^0 + v_x \frac{L_{thigh}}{|v_x|} \text{ and } F_x^{-1} = P^0 + v_x \frac{L_{thigh} + L_{calf}}{|v_x|}$$

We estimate hips previous positions P_l^{-1} and P_r^{-1} as P^0 . Finally we compute F_r^0 , K_r^0 , F_l^0 and K_l^0 using the legs tracking algorithm.

6 Results

Figure 2(a) outlines the system configuration. The acquisition infrastructure is composed of four Phillips webcams (SPC900NC) connected to a single Pc (CPU: p4 3.2ghz,GPU: NVIDIA Quadro 3450). Webcams produce images of resolution 320×240 at 30fps.

Our method has been applied on different persons doing fast and challenging motions. Thanks to shape analysis and skin parts knowledge, our system is able to acquire the joint positions for a challenging pose outlined on the Figure 5(a). This pose is difficult because the 3D shape topology is not a human corresponding one. The temporal coherence is the success key for the pose presented Figure 5(b). This underlines the case of jointed hands (4.3) which is successfully recognized. A very difficult pose is underlines Fig. 5(c) which is successfully recovered by our system. Images of Figure 6 argue that our system works for large range movements acquisition.

Some results are included in the supplementary video clip ². This proves the robustness of our approach on long video sequences with rapid and complex movements.

Our current experimental implementation provides more than 30 poses tracking per second on a single computer, which is faster than the webcams acquisition frame rate. An optimized implementation can be usable for current generation of home entertainment computers. As our algorithm is based on 3D reconstruction, it is independent of the number of cameras used, but it depends on the voxel grid resolution. We reconstruct a voxel grid composed of 64^3 voxels in a $6m^3$ box. This resolution is sufficient for entertainment human-machine interfaces.

² <http://liris.cnrs.fr/brice.michoud/gtas07videos/>

Our motion capture system is based on a Shape-From-Silhouette algorithm. This algorithm computes an object 3D shape estimation from its silhouettes. The result directly depends on the silhouette segmentation quality, which is always an opened problem of the computer vision science. If the silhouette mask contains some noises like camera noise or object shadows, the volume reconstruction will be very noised. Thus the results of the motion capture will be worse. But our method is also based on a skin segmentation which is a more robust faced to camera noise. Then the hand and head articulations are more noise-resistant, than others articulations.

7 Conclusion

In this paper, we describe a new marker-free human motion capture system at least three webcams connected to a single computer. Fully automated and working under real-time constraint, the system is based on both a 3D shape analysis, human morphology constraints, and a 3D shape skin segmentation. Combining different 3D information, the approach is robust to self-occlusion and to coarse 3D shape approximation provided by voxel estimation sub-system. We are able to estimate the fifteen main human body joints, at more than 30 frames per second, which can be used for home entertainment applications.

The current system provides real-time motion capture for only one person. Current work aims to provide motion capture of multiple persons filmed together in the same area, even they are in contact. For home entertainment application, the major limitation is silhouette preprocessing, because the background cannot be guaranteed to be static at home. We work on a new segmentation algorithm based on statistical background model helped by optical flow algorithm.

References

1. Agarwal, A., Triggs, B.: Monocular human motion capture with a mixture of regressors. In: CVPR '05, IEEE Computer Society (2005) 72
2. Chen, Y., Lee, J., Parent, R., Machiraju, R.: Markerless monocular motion capture using image features and physical constraints. In: CGI '05: Proceedings of the Computer Graphics International 2005 (CGI'05), Washington, DC, USA, IEEE Computer Society (2005) 36–43
3. Mikic, I., Trivedi, M., Hunter, E., Cosman, P.: Human body model acquisition and tracking using voxel data. *Int. J. Comput. Vision* **53**(3) (2003) 199–223
4. Tangkuampien, T., Suter, D.: Human motion de-noising via greedy kernel principal component analysis filtering. In: ICPR (3), IEEE Computer Society (2006) 457–460
5. Caillette, F., Galata, A., Howard, T.: Real-Time 3-D Human Body Tracking using Variable Length Markov Models. In: Proceedings BMVC '05. Volume 1. (2005)
6. Laurentini, A.: The visual hull concept for silhouette-based image understanding. *IEEE Trans. Pattern Anal. Mach. Intell.* **16**(2) (1994) 150–162
7. Ménier, C., Boyer, E., Raffin, B.: 3d skeleton-based body pose recovery. In: Proceedings of the 3rd International Symposium on 3D Data Processing, Visualization and Transmission, Chapel Hill (USA). (2006)
8. Franco, J.S., Boyer, E.: Exact polyhedral visual hulls. In: Proceedings BMVC'03. (2003) 329–338 Norwich, UK.
9. Hasenfratz, J.M., Lapierre, M., Sillion, F.: A real-time system for full body interaction with virtual worlds. *Eurographics Symposium on Virtual Environments* (2004) 147–156
10. Cheung, K.M., Kanade, T., Bouguet, J.Y., Holler, M.: A real time system for robust 3d voxel reconstruction of human motions. In: Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '00). Volume 2. (2000) 714 – 720
11. Michoud, B., Guillou, E., Bouakaz, S.: Shape from silhouette: Towards a solution for partial visibility problem. In: Eurographics 2006 Short Papers Preceedings. (2006) 13–16
12. Zhang, Z.: Flexible camera calibration by viewing a plane from unknown orientations. In: ICCV. (1999) 666–673
13. Joshi, N.: Color calibration for arrays of inexpensive image sensorss. Technical report, Stanford University (2004)
14. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: Principles and practice of background maintenance. In: ICCV (1). (1999) 255–261

15. Vezhnevets, V., Sazonov, V., Andreeva, A.: A survey on pixel-based skin color detection techniques. In: Proceedings of Graphicon-2003. (2003)
16. Segal, M., Korobkin, C., van Widenfelt, R., Foran, J., Haeberli, P.: Fast shadows and lighting effects using texture mapping. In: Proceedings of SIGGRAPH. (1992)
17. Culbertson, W.B., Malzbender, T., Slabaugh, G.G.: Generalized voxel coloring. In: Workshop on Vision Algorithms. (1999) 100–115
18. Sun, W., Hilton, A., Smith, R., Illingworth, J.: Layered animation of captured data. *The Visual Computer* **17**(8) (2001) 457–474
19. Fua, P., Gruen, A., D'Apuzzo, N., Plankers, R.: Markerless Full Body Shape and Motion Capture from Video Sequences. In: Symposium on Close Range Imaging, International Society for Photogrammetry and Remote Sensing, Corfu, Greece. (2002)
20. Dreyfuss, H., Tilley, A.R.: *The Measure of Man and Woman: Human Factors in Design*. John Wiley & Sons (2001)

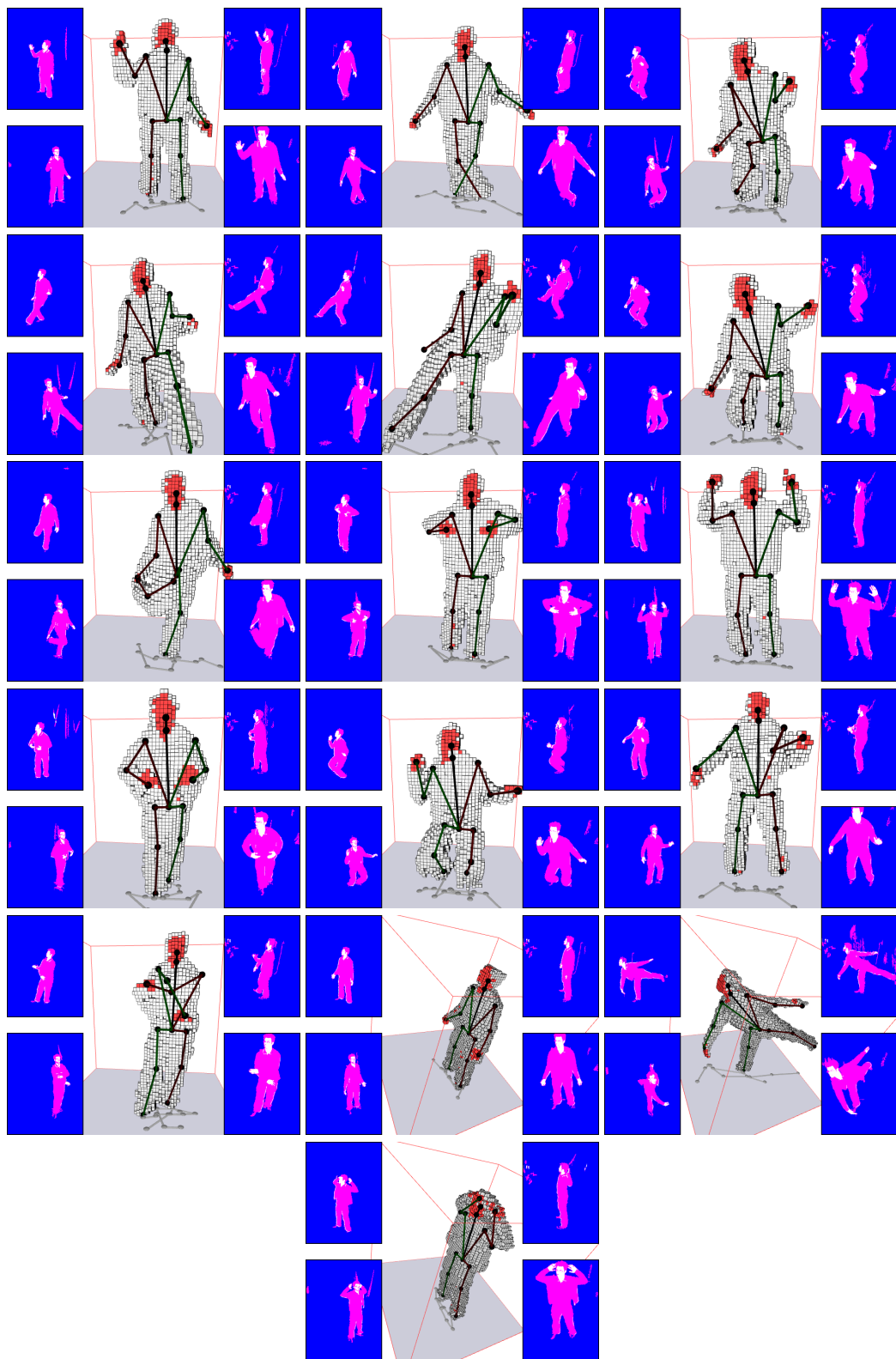


Fig. 6. Results for a wide range of movements

La méthode HEML pour le calcul rapide et réaliste des déformations de matériaux hyper-élastiques en simulation de chirurgie

François Goulette
CAOR, Mines Paris
E-mail : goulette@ensmp.fr

Résumé

Nous présentons dans cet exposé la méthode HEML, une méthode de calcul de déformations réalistes de matériaux déformables. Cette méthode permet entre autres de calculer des déformations de corps mous en simulation interactive de chirurgie. Elle est plus générique et plus rapide que les autres méthodes existantes (masse-ressort, masse-tenseur), et permet de calculer des déformations, y compris en grands déplacements et grandes déformations (c'est-à-dire hors du cadre de l'approximation linéaire), de matériaux représentés par leur loi de comportement hyper-élastique.

Présentation

Les simulateurs de chirurgie virtuelle suscitent beaucoup d'intérêt dans le monde de l'apprentissage de la chirurgie, où ils permettraient d'améliorer la qualité de la formation au geste des chirurgiens. L'une des difficultés techniques majeures actuelles est la possibilité de réaliser des algorithmes de déformation rapides et réalistes du comportement élastique des organes. Face à cette difficulté la plupart des simulateurs existants utilisent des modèles "Masse-Ressort", rapides mais au réalisme pauvre. Des méthodes temps réel ont été mises au point dans le cadre d'approximations de l'élasticité linéaire, ce qui n'est pas satisfaisant pour l'application visée. L'une des méthodes les plus avancées actuellement en termes de réalisme, et assez répandue pour cela, est la méthode "Masse-Tenseur". Cette méthode, dérivée de la méthode des éléments finis, se limite à la seule loi de comportement de Saint Venant-Kirchhoff, supporte les grandes déformations, mais est mal adaptée aux grands déplacements.

Nous présentons dans cet exposé une nouvelle méthode que nous appelons "HEML" (Hyper-Elastic Mass Links) [Gou06]. Cette méthode, particulièrement rapide, également dérivée de la méthode des éléments finis, est valable pour les grands déplacements et les grandes déformations. Par rapport à la méthode "Masse-Tenseur", pour le même type de loi constitutive (Saint Venant-Kirchhoff) nous avons calculé des gains de 40 à 200 fois plus vite ; d'autre part nous ne sommes pas limités à Saint Venant-Kirchhoff mais pouvons utiliser toute loi hyper-élastique, comme Néo-Hookeen, Mooney-Rivlin (avec des complexités calculatoires comparables). La validité a été testée par rapport à une solution analytique dans un cas de compression pour Saint Venant-Kirchhoff. Nous présenterons quelques exemples de démonstrations de déformations temps réel pour les lois de Saint Venant-Kirchhoff et Néo-Hooke.

Références

[Gou06] F. Goulette and S. Chendeb, "A Framework for Fast Computation of Hyper-Elastic Materials Deformations in Real-Time Simulation of Surgery". Proc. MICCAI Computational Biomechanics Workshop, Copenhague, October 2006.

Squelettisation robuste de données volumiques pour la caractérisation et la quantification d'Anévrismes

Résumé

Dans cet article, nous présenterons un nouvel algorithme pour trouver le squelette de l'arbre vasculaire. Et puis, nous nous servirons de ce squelette pour détecter automatiquement les anévrismes et leurs collets. Notre algorithme d'extraction de squelette est basé sur l'algorithme de Dijkstra, et nous proposerons une nouvelle méthode pour l'extraction des branches. Les résultats illustrent l'efficacité de notre algorithme et sa robustesse.

1 Introduction

Dans le cadre des maladies des vaisseaux sanguins, les médecins ont besoin de détecter les déformations : dépôts calcaires, étranglements, anévrismes. L'anévrisme est une dilatation localisée de la paroi d'une artère aboutissant à la formation d'une poche de taille variable, communiquant avec l'artère au moyen d'une zone rétrécie que l'on nomme **le collet**. Sa forme habituelle est celle d'un sac, son diamètre pouvant atteindre plusieurs centimètres. Les dimensions d'un anévrisme, sa forme et son collet jouent un rôle critique dans le choix de traitement. Soit l'embolisation de l'anévrisme à l'aide de *coils* (petits ressorts en platine), voir figure 1.

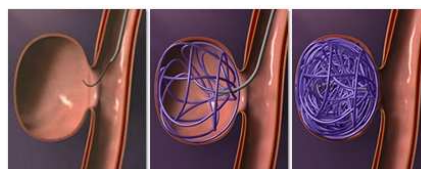


FIG. 1 – Embolisation

Soit l'intervention chirurgicale (trépanation et mise en place d'un clip au niveau du collet de l'anévrisme). Voir figure 2.

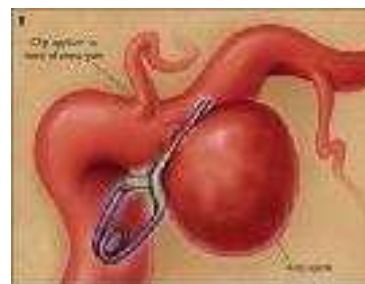


FIG. 2 – *clipping*

Pour aider les médecins, et à partir des images bien segmentées, on extrait le squelette de l'arbre vasculaire, et puis on détecte les anévrismes en trouvant les branches qui partent dans ces anévrismes. Ensuite, on donne une courbe fermée qui représente le collet de chaque anévrisme.

2 État de l'Art

Pour **détecter les anévrismes**, nous avons choisi d'extraire le squelette de l'arbre vasculaire. Alors, on est passé vers l'extraction d'une "ligne de centre". Une définition de la ligne de centre qui nous intéresse est : une ligne connexe, singulière qui traverse le vaisseau en s'approchant le mieux possible du centre de vaisseau.

On peut classer les méthodes existantes pour l'extraction de la ligne de centre dans les catégories suivantes :

1. Les méthodes manuelles :

Dans ces méthodes, l'utilisateur choisit manuellement les points clés, et l'algorithme fait l'interpolation entre ces points. Les résultats sont très subjectifs au choix de points, qui est une tâche fastidieuse et non précise.

2. Amincissement topologique :

L'idée principale derrière ces méthodes est l'épluchage successif du volume jusqu'à l'arrivée à la singularité. Il s'agit d'enlever les points dits simples, qui sont les points dont la suppression ne change pas la topologie. Ces points peuvent être supprimés en parallèle [4] ou en séquentiel [1]. Les résultats sont normalement des lignes de centre : homotopiques (par construction) avec l'objet, fines, et bien représentatives de l'objet (si les points simples sont bien caractérisés) mais pas forcément au centre.

3. Gradient zero-crossing.

Dans [3], les auteurs se servent du gradient et des valeurs propres de la matrice Hessienne pour interpoler les *zero-crossing* du vecteur de gradient des niveaux de gris dans la direction *cross-sectional*. Cette idée est valable surtout dans la géométrie continue (directement sur les images) et non pas discrète (sur les images segmentées composées de valeurs binaires).

4. Carte de distance :

Les algorithmes de *distance mapping*, consistent à associer à chaque voxel (pixel)

une distance. Cette distance peut être une distance à un point source comme dans l'algorithme de Dijkstra, ou une distance de chanfrein. Il s'agit ensuite de trouver la ligne de centre qui est la branche la plus longue dans l'arbre construit pour trouver ces distances. Ces méthodes sont rapides et donnent une ligne connexe et singulière, mais qui coupe les virages. Pour résoudre ce problème, les auteurs dans [6] trient les voxels selon l'inverse de leurs distances au bord, pour obliger la ligne à être plus au centre. Une idée très semblable est employée dans [2].

L'algorithme que nous avons développé tombe dans cette catégorie, et il est détaillé dans le dernier chapitre.

Nous avons choisi de travailler directement sur les voxels plutôt que sur une triangulation représentant la surface de l'objet, et calculée, par exemple, grâce à l'algorithme des *marching cubes* [5], car l'étude est plus simple et nous apporte suffisamment d'information.

3 Notre méthode

3.1 La ligne de centre

Dans [6], est expliqué un algorithme pour extraire une ligne de centre. L'algorithme est basé sur le calcul de l'arbre des plus courts chemins d'un voxel source à tous les autres voxels selon l'algorithme de Dijkstra avec une modification. La modification proposée est que, au lieu de trier les voxels selon leur distance à la source, les auteurs proposent de les trier selon l'inverse de leur distance à la surface. Cette modification sert à traiter en priorité les voxels les plus profonds, et ainsi éviter que la ligne de centre coupe les virages.

Dans la figure 3 on voit le *flowchart* qui explique comment construire l'arbre de Dijkstra.

Pour comprendre le *flowchart* notons que :

1. *DFS* correspond à la distance à la source.
2. *DFB* correspond à la distance à la surface.

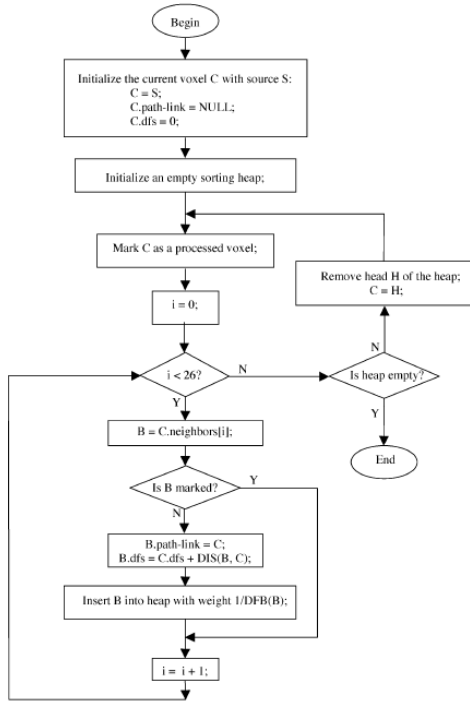


FIG. 3 – L’algorithme proposé dans [6]

3. $DIS(B, C)$ correspond à la distance Euclidienne entre les centres de deux voxels B et C.

Après la construction de l’arbre, on définit le voxel *final* comme le voxel le plus loin de la *source*.

Le voxel *source* est choisi au début de l’algorithme soit automatiquement soit par l’utilisateur. Alors, nous avons choisi de trouver ce voxel automatiquement. Il est calculé en prenant un voxel V au hasard, puis en prenant comme source le voxel le plus loin de V . De cette façon, on est sûr que la source est une extrémité de l’arbre. Ensuite on construit l’arbre.

Cependant, l’algorithme tel qu’il est décrit dans [6] est erroné et ne donne pas une ligne de voxels passant au centre des vaisseaux. En effet, il manque un test avant modifier un des voisins d’un voxel. Ce test est le suivant :

$$DFS[voxel] + DIST(voxel, voisin) < DFS[cevoisin]$$

Notre idée est que : au lieu de trier les voxels selon leurs profondeurs, nous proposons de changer la fonction de distance. C’est à dire, au lieu de calculer la distance euclidienne entre deux voxels, on calcule la fonction suivante :

$$d(F, Ch) = \frac{dist(F, Ch)}{1 + (DFB[F] + DFB[Ch])}$$

Comme la fonction de distance est proportionnelle à l’inverse des profondeurs des voxels, ceci garantit que les voxels plus profonds seront prioritaires, et cela garantit que la ligne de centre soit au centre (ne coupe pas les virages). En même temps, cette fonction est proportionnelle à la distance euclidienne entre les voxels, qui garantit que entre deux voxels qui ont le même profondeur, on va choisir le plus proche, contrairement à la méthode proposé dans [6], qui va tirer au sort entre les voxels de même profondeur.

3.2 Les branches

Les branches doivent satisfaire les deux qualités suivantes :

1. Au niveau de la jonction, la branche doit être connectée à celle d’avant (son père) d’une façon naturelle. C’est à dire, il ne doit pas commencer parallèlement à son père.

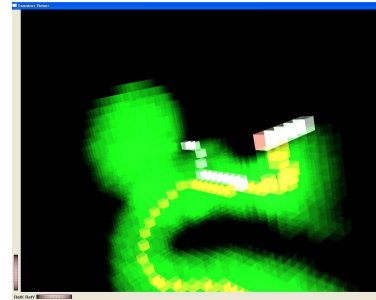


FIG. 4 – Les branches selon l’algorithme de [6]

2. Les branches doivent être extraites dans l’ordre décroissant de leurs longueurs.

Pour garantir la première qualité, c’est à dire, obtenir des branches de l’arbre de Dijkstra qui sont perpendiculaires à la ligne de centre, nous proposons l’algorithme suivant :

- Après l'extraction de la ligne de centre principale, on met les distances des voxels de cette ligne à la source à zéro.
- On reconstruit un arbre de Dijkstra (en utilisant toujours notre fonction de distance).
- On cherche le voxel le plus loin de la source.
- De ce voxel on descend dans l'arbre jusqu'à rencontrer un voxel de la ligne de centre.
- On répète ces étapes tant que la longueur de la branche est assez importante (par exemple : supérieure au diamètre de la ligne de centre principale).

Pour la deuxième qualité, qui est de trouver les bonnes extrémités pour les branches, nous avons choisi de garder notre fonction de distance pendant la construction de l'arbre pour être toujours au centre. Toutefois, notre mesure de distance est biaisée par la profondeur, et notre arbre ne fournit pas toujours le voxel le plus lointain au sens Euclidien. Alors, nous utilisons la distance euclidienne pour la recherche du voxel le plus loin de la source.

Dans la figure 5, on voit le squelette construit selon notre méthode.



FIG. 5 – Le résultat de notre algorithme

4 Détection des anévrismes

Pour détecter les anévrismes on propose de :

- Trouver les branches qui partent dans les anévrismes.
- Se servir de ces branches pour trouver précisément les voxels qui appartiennent aux anévrismes.

4.1 Trouver les branches des anévrismes

Si on voulait détecter les anévrismes non automatiquement, il suffirait d'extraire les branches successivement, en demandant au médecin de s'arrêter quand la branche de l'anévrisme est extraite. Mais, comme on insiste à la détection automatique des anévrismes, il fallait trouver le caractère qui différencie une branche d'un anévrisme d'une branche normale.

Regardant la forme arrondie d'un anévrisme, on peut dire que la distance de la surface de l'anévrisme à sa ligne de centre (la branche qui est dans l'anévrisme) varie plus que dans une branche normale. Alors, reposant sur cette idée, on a calculé la déviation de cette distance pour toutes les branches extraites. Si cette déviation est importante (supérieure à un seuil) on peut dire que la branche représente un anévrisme.

Puis, on regarde si la branche n'est-elle pas connecté à celle d'un anévrisme (il y a plusieurs branche dans un seul anévrisme), car il suffit, dans ce cas, de trouver la branche principale, et comme ça on est sûr que chaque branche trouvée représente un anévrisme séparé.

Mais parfois, et à cause de la segmentation, on pourrait extraire des branches non normales (déviation importante). Alors, il fallait trouver un caractère encore plus précis que la déviation. Si on regarde la distance des voxels de la surface de l'anévrisme à la branche comme une fonction dont l'abscisse et la distance de ces voxels au point de connexion entre cette branche et celle d'avant, on retrouve la forme illustrée dans la figure 6.

Nous n'avons pas appliqué cette idée à cause de manque de temps. Alors, pour bien trouver les branches des anévrismes et continuer/tester

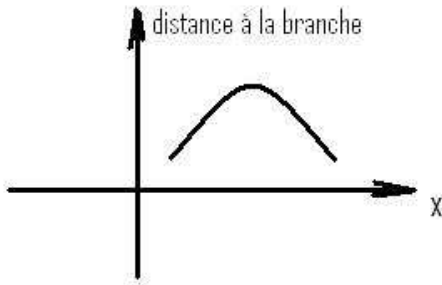


FIG. 6 – Fonction de distance

les étapes suivantes (détection des anévrismes et de leurs collets), nous l'avons fait manuellement, en demandant d'extraire deux branches (dans l'exemple, ceci garantit l'extraction de la branche normale et celle de l'anévrisme).

4.2 Trouver les voxels des anévrismes

Une fois la branche de l'anévrisme bien détectée, en déduire les voxels de l'anévrisme est relativement simple. Tous les fils des voxels de la branche de l'anévrisme, ainsi que leurs descendants, tombent dans l'anévrisme, sauf quelques voxels autour du point de jonction entre la branche de l'anévrisme et celle d'avant (son père).

Pour ne pas compter ces voxels, il suffit d'ajouter une condition aux voxels de la branche avant les considérer comme voxels de l'anévrisme. Cette condition est que la distance de voxel à la branche *père* de la branche de l'anévrisme, soit supérieure à la profondeur du voxel de jonction (entre la branche de l'anévrisme et son père). En effet, cette distance représente le diamètre du vaisseau. Cela nous donne le résultat dans la figure 7.

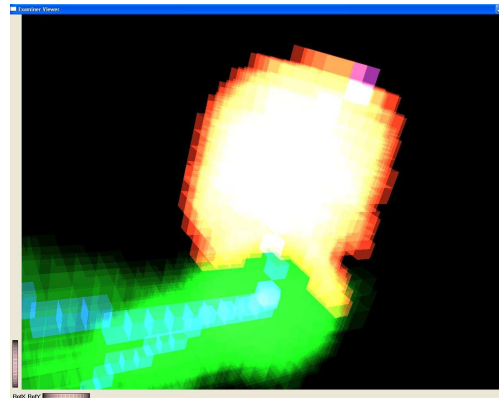


FIG. 7 – Les voxels de l'anévrisme

5 Détection du collet d'un anévrisme

5.1 Première estimation

Étant donné l'anévrisme, on peut définir les voxels du collet par les voxels qui ont au moins un voisin qui n'appartient pas à l'anévrisme. En appliquant cette définition on obtient le collet que l'on voit sur la figure 8.

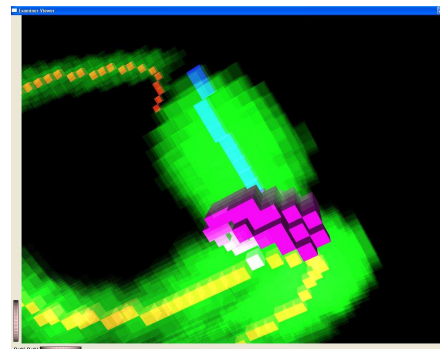


FIG. 8 – Première estimation du collet

Alors, on voit que la définition appliquée ne marche pas avec la définition médicale du collet (la zone rétrécie avec laquelle l'anévrisme communique avec l'artère). En effet, le collet que l'on voit sur la figure est plutôt la paroi du vaisseau qui existerait s'il n'y avait pas un anévrisme. Le dessin illustre la différence :



Mais, en même temps on peut profiter du collet trouvé pour trouver le bon, c'est à dire, parmi les voxels de ce collet trouver ceux qui forment la courbe fermée la plus courte possible. Donc il y a deux étapes : premièrement, trouver une courbe fermée, deuxièmement, optimiser cette courbe pour qu'elle soit la plus courte possible. Ces étapes sont détaillées dans les deux sections suivantes.

5.2 Deuxième estimation

La première étape pour trouver le collet tel qu'il est défini médicalement, et de trouver une courbe fermée parmi les voxels de la première estimation du collet (voir 5.1).

L'algorithme que je propose est basé sur l'algorithme de Dijkstra. Tout d'abord, je veux préciser qu'on va travailler sur les voxels de la surface de la première estimation du collet, je vais appeler ce groupe de voxels *Surface1* dans la suite. L'algorithme est le suivant :

- Construire l'arbre de Dijkstra en prenant n'importe quel voxel de *Surface1* comme source.
- Chercher le voxel le plus loin de la source *E1*, stocker les voxels de la branche de l'arbre qui part de *E1* jusqu'à la source dans une liste.
- Mettre la distance à la source des voxels de la liste et tous leurs descendants à 0.
- Chercher dans le voisinage de *E1*, le voxel *E2* le plus loin de la source.
- Ajouter à la liste les voxels de la branche qui part de *E2* vers la source.
- Les voxels de la liste forment une courbe fermée.

La figure 9 illustre le résultat.

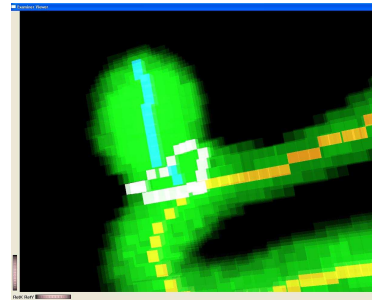


FIG. 9 – Courbe initiale

En effet, la définition de l'arbre Dijkstra (l'arbre de plus courts chemins à un point source), et le fait qu'on construit cet arbre pour un ensemble de voxel qui forment une surface fermée, garantit qu'il y aura deux "relativement longues" branches de l'arbre qui forment une courbe fermée. La troisième étape est pour garantir que la deuxième extrémité trouvée est dans l'autre moitié de l'arbre, et la quatrième étape est pour garantir la connexion entre les deux branches.

5.3 Optimisation

Une fois la courbe initiale trouvée, il suffit d'appliquer une optimisation pour trouver le collet. L'algorithme d'optimisation que je propose est le suivant :

- Commencer par n'importe quel voxel *V* de la courbe initiale, et garder ces deux successeur dans la courbe *S1, S2*.
- Chercher dans le voisinage de *V* et de *S2* (les voisins qui sont dans *Surface1*), un voxel *S1'* tel que :

$$Dist(V, S1') + Dist(S1', S2) < Dist(V, S1) + Dist(S1, S2)$$
- Si *S1'* est retrouvé, mettre : $S1 = S1'$
- Avancer : $V = S1$
- Répéter jusqu'à retourner au voxel de départ sans rien changer

6 Conclusions et travaux futurs

Nous avons présenté un algorithme robuste pour extraire le squelette d'un ensemble connexe de voxels. Nous avons appliqué cet algorithme sur l'arbre vasculaire pour détecter les anévrismes et leurs collets d'une façon automatique.

Il nous reste à faire plus de tests surtout au niveau de la détection des anévrismes et des collets, et améliorer notre méthode de détection pour la rendre plus robuste.

Dans le futur, nous envisagerons de continuer sur le sujet d'anévrismes, dans le sens de simuler le processus d'embolisation. Une telle simulation peut aider le médecin à estimer le résultat de la chirurgie.

Références

- [1] Gyi, Erich Sorantin, Emese Balogh, Attila Kuba, Csongor Halmi, Balázs Erdohelyi, and Klaus Hasegger. A sequential 3d thinning algorithm and its medical applications. In *IPMI '01 : Proceedings of the 17th International Conference on Information Processing in Medical Imaging*, pages 409–415, London, UK, 2001. Springer-Verlag.
- [2] Guangxiang Jiang and Lixu Gu. An automatic and fast centerline extraction algorithm for virtual colonoscopy. 2005.
- [3] K. Krissian, C.F. Westin, and R. Kikinis. Fast and automatic vessel centerline detection for mra. 2005.
- [4] Christophe Lohou and Gilles Bertrand. A new 3d 6-subiteration thinning algorithm based on p-simple points. In *DGCI '02 : Proceedings of the 10th International Conference on Discrete Geometry for Computer Imagery*, pages 102–113, London, UK, 2002. Springer-Verlag.
- [5] 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.
- [6] Ming Wan, Zhengrong Liang, Qi Ke, Lichan Hong, Ingmar Bitter, and Arie E. Kaufman. Automatic centerline extraction for virtual colonoscopy. *IEEE Trans. Med. Imaging*, 21(11) :1450–1460, 2002.

Management de cancer à distance :

Analyse les réseaux de cancer pour la modélisation de système de prise en charge des patients atteints de cancer hors de l'hôpital, en utilisant les nouvelles technologies d'information et de communication pour faciliter le management de cancer à distance.

Nazek MOHAMMAD*, Jean-Paul AURAY**

* (Chercheur en Thèse), LIRIS-CNRS Axe connaissance et système complexes, Equipe MA2D (Méthodes et Algorithmes pour l'Aide à la Décision), Laboratoire LIRIS, UMR 5205 du CNRS/INSA de Lyon/ Université Claude Bernard Lyon1/ Université Lumière Lyon2/ Ecole Centrale de Lyon

** Professeur Directeur d'équipe MA2D au sein du Laboratoire LIRIS, UMR 5205 du CNRS/ INSA de Lyon/ Université Claude Bernard Lyon1/ Université Lumière Lyon2/ Ecole Centrale de Lyon

Résumé:

L'objectif de ce travail consiste à modéliser d'un système d'information générique, dont l'objectif de développer des applications de télémédecine dans le cadre d'un réseau de cancer pour la prise en charge des patients atteints de cancer à domicile. Dans ce travail qui est en cours de réalisation, nous évaluerons et analyserons l'état actuel de la prise en charge des patients atteints de cancer hors de l'hôpital, pour le concevoir d'un système de prise en charge en utilisant les nouvelles technologies d'information et de communication pour faciliter le management de cancer à distance

Mots clés:

Télémédecine, hospitalisation à domicile, réseau de cancer, système d'information, oncologie, management de cancer, dossier médical.

Abstract:

The aim of this research is to design a generic information system to employ telemedicine applications for patient at home hospitalisation. To approach this aim, we are going to evaluate and analyse the actual situations of cancer patients at their homes by cancer health networks, and to design a specific model based on the new information and communication technologies to facilitate cancer health management at home.

Keywords:

Telemedicine, patient at home, cancer health network, information system, oncology, cancer health management, patient health record.

Introduction:

Dans nos jours, le développement des Technologies de l'Information et de la Communication (TIC) ouvre une large gamme d'applications dans les différents domaines de la vie, et plus particulièrement dans le domaine de la santé. Dans ce dernier, et la Télémédecine permet de lier les TIC et la médecine afin de fournir des soins de santé (diagnostique, traitement, suivi et prévention, etc.) pertinents aux patients. Cela est rendu possible par la collecte, l'organisation, le partage et l'échange d'informations médicales et d'images nécessaires pour la prise en charge d'une maladie donnée.

Nous reprenons la définition du professeur Lareng cité par Houdart (1): «La Télémédecine consiste en l'utilisation des moyens et des techniques qui permettent à distance la pratique médicale». Cette conception consiste à l'utilisation de téléphone, Internet, et d'autres moyens techniques afin de transmettre des données et des informations médicales (données médicales, images, traitement, etc.) entre les acteurs de soins.

Les études ont montré l'intérêt d'appliquer les méthodes de la télémédecine dans le domaine de cancer (2) soit en télé ou visio-conférence, dossier médical informatisé partagé, télédiagnostique, télé-éducation, télésurveillance, etc.

Dans ce travail de recherche nous allons adapter les applications de la télémédecine aux besoins des réseaux de cancer en France pour mieux gérer le suivi des patients atteints de cancer à leur domicile.

La présentation de la problématique

Les réseaux du cancer ont une nouvelle forme d'organisation des soins coordonnés de façon professionnelle par une structure dédiée ou la notion de coordination est fondamentale entre les professionnels de santé et les patients, pour la prise en charge des patients atteints du cancer. Cette maladie qui accompagne le patient pendant longue durée avec pathologie jusqu'à la rétablissement ou jusqu'à la fin de sa vie ; ici monte le problème de prise en charge de patient pendant cette longue durée, et suivre son état de santé dans et hors de l'hôpital. Et avec la croissance rapide de l'utilisation des nouvelles technologies de l'information et des communications (TIC) dans le secteur de la santé, Ce développement des structures de coordination fait suite à plusieurs expérimentations personnelles de soins en dehors de l'hôpital (ex : l'utilisation de téléphone, Internet) Pour gérer plus globalement la prise en charge des patients atteints de cancer.

Au cours de ces expérimentations on remarque les problématiques suivantes: (3)

- Un problème de prise en charge et suivre l'état de santé des patients atteints du cancer hors de l'hôpital (ex : en cas

- de chimiothérapie orale, ou après la sortie de l'hôpital, etc.).
- Problème de coordination entre les acteurs de réseau.
- Problème de perte des données médicales des patients au cours de transmission des données entre les acteurs des réseaux.
- Problème de qualité des données médicales concernant les patients atteints du cancer tout au long de sa pathologie et surtout hors de l'hôpital.
- L'utilisation des outils de la télémédecine (téléphone et l'internet) entre les professionnels de santé dans le réseau de façon personnelle pour faciliter les tâches de travail, avec un mauvais management et l'ignorance des modalités scientifiques optimales d'utiliser les applications de la télémédecine dans un réseau de cancer.

Ces différentes problématiques ont été décrites dans un fonctionnement au quotidien dans un réseau de cancer (réunion avec les responsables de réseau ONCORA, Rhône-Alpes, le 22 Février / 2007).

La question est:

Comment améliorer la prise en charge des patients atteints de cancer, et suivre leurs états de santé hors l'hôpital sans perte des données médicales, tout en réduisant les coûts, en utilisant les nouvelles technologies d'information et communication ?

Les objectifs

Notre objectif dans ce travail de recherche s'appuie sur l'évaluation et l'analyse de l'état actuel de la prise en charge des patients atteints de cancer hors de l'hôpital, pour le concevoir d'un système de prise en charge en utilisant les nouvelles technologies d'information et de communication pour faciliter le management de cancer à distance.

Nos objectifs sont:

- Evaluer les pratiques médicales dans un réseau de cancer.
- Evaluer la fréquence d'utilisation des nouvelles technologies d'information et communication dans un réseau de cancer: (la fréquence d'utiliser les appels téléphoniques et de l'Internet et d'autre moyens dans les pratiques médicales dans le réseau pour la prise en charge d'un patient atteint de cancer à distance).
- Analyser les coordinations, les acteurs, la structure de réseau.
- Analyser les flux de patient, et les flux d'informations médicales dans le réseau de cancer.
- Identifier les points faibles dans le système d'information médical (SIM) dans le réseau.
- Développement d'un modèle de système de prise en charge des patients atteints de cancer hors de l'hôpital, et la proposition des nouveaux moyens de gestion de capitalisation de l'information et de planifications de mise en œuvre, pour faciliter le management du cancer hors de l'hôpital.

Les hypothèses

Afin de répondre à l'ensemble de ces problématiques nous proposons de mettre en œuvre une approche globale de la conception de prise en charge des patients atteints de cancer hors de centre hospitalier.

On prendra un exemple réel dans notre étude, en analysant le Réseau ONCORA dans la région Rhône-Alpes.

Ce projet de recherche propose les étapes suivantes pour répondre à l'ensemble de ces problématiques :

- L'analyse de l'état de lieu actuel d'un réseau de cancer, et d'identifier et caractériser les flux actuels de trajectoire de patient (hôpital-domicile) tout au long de sa pathologie, ainsi que l'évaluation et le diagnostic des moyens mis en œuvre pour prendre en charge des patients atteints de cancer hors de l'hôpital.
Cette première étape nous permettra de bien comprendre la situation actuelle et les moyens de la prise en charge des patients atteints de cancer hors de l'hôpital, mais aussi mettre en évidence les points forts et faibles du réseau de cancer. (4), (5)
- Dans un second temps, évaluer les besoins et moyens nécessaires pour l'avenir en considérant les points de vue: politique, économique et social, besoins des patients atteints de cancer, attentes des praticiens, évolutions des techniques de l'information et de communication médicale.
Cette étape nous permettra d'identifier les besoins et moyens nécessaires pour faire face aux problématiques à partir des indicateurs dans le monde réel. (4), (5)
- Et à partir des résultats que nous trouverons dans les étapes précédentes, et avec l'analyse des résultats des autres expériences internationales en adoptant les nouvelles technologies d'information et de communication dans le secteur de la santé qu'on peut s'en servir pour résoudre les problèmes.
Cette étape nous permettra d'avoir une conception d'organisation, et une vision plus claire pour la proposition des nouveaux moyens pour la prise en charge des patients atteints du cancer à distance.
- Et après les étapes précédentes, on proposera de nouveaux moyens de gestion de capitalisation de l'information, ainsi que la spécification d'un système d'information qui servira à la mise en œuvre des outils d'aide à la décision

pour le pilotage de ces organisations, et pour améliorer la prise en charge des patients atteints de cancer hors de l'hôpital en utilisant les nouvelles technologies d'information et de communication. (6)

Les résultats attendus

Un système d'information médical optimal pour le pilotage des prises en charge des patients atteints de cancer hors de l'hôpital, en adoptant les nouvelles technologies d'information et de télécommunication pour faciliter le management du cancer.

L'application de ce système dans les réseaux de cancer réalisera les suivants:

- Maîtriser les flux de trajectoire de patient (hôpital-domicile). (7)
- Maîtriser les flux d'information et la perte des données médicales.
- Organiser la capture et le partage des informations médicales entre les acteurs au cours des pratiques médicales.
- Améliorer la qualité des données médicales des patients atteints du cancer tout au long de la pathologie (dans et hors de l'hôpital).
- Améliorer la coordination entre les acteurs de réseau. (9)
- Augmenter et améliorer l'accès au soin par la population (égalité d'accès aux soins de qualité). (8)
- Améliorer la qualité et la continuité de soins. (8), (11)
- Améliorer la prise en charge du patient atteint du cancer dans et hors de l'hôpital. (8), (9)
- Améliorer l'autonomie de patient hors de l'hôpital, et l'encourager à être un acteur de sa santé. (11)
- Réduire les coûts (coûts d'hospitalisation, coûts de déplacement de patient).(10), (12)

La méthodologie suivie

Pour qu'on soit capable de trouver les éléments nécessaires de notre étude on suivra la méthodologie suivant:

- **Un cas réel à étudier (case study):** on analysant un réseau de cancer (ici nous prenons le Réseau ONCORA dans la région Rhône-Alpes), Ainsi que l'analyse des rapports et des documentations de la Fédération Nationale des Centres de Lutte Contre le Cancer (FNCLCC), et l'analyse d'autres documents et études liées à nos problématiques dans les organismes français : cette approche nous permettra de bien analyser l'état actuel et réel des pratiques médicales en utilisant la (TIC) dans un réseau de cancer. Cela va nous aider à mieux comprendre la situation actuelle, et à identifier les besoins d'un réseau de cancer. Selon Markus (1983), la case study peut se faire par:
 - Interviews avec plus des personnes utilisateurs de TIC dans le réseau de cancer (cancérologue, médecine de ville, patient, proche de patient, infirmière, informaticien, administrateur, etc.)
 - Analyse des documents autour de la prise en charge des patients atteints de cancer hors de centre hospitalier de cancer (rapports annuels, rapports organisationnels, documents structurels)
- **Un revue des documents, rapports ,et des études internationales** pour la prise en charge des patients hors de l'hôpital, et l'utilisation des nouvelles technologies d'information et de communication en oncologie, en passant par les aspects psychologie, éthique, sociologique, économique, et légalité, etc.

Bibliographies:

- 1) Hervé, C., Gaillard. M., Bontemps. A. 1996. l'accès aux soins à l'aide de la télémedecine: enjeux étiques. Informatique et Santé: Paris
- 2) Ricke, J., Beartelink, H. 2000. Telemedicine an its impact on cancer management. European Journal of Cancer
- 3) Mohammad, N. 2006. Compte rendu de réunion dans le réseau ONCORA.
- 4) Schult, P. 2004. Complex IT Project Management: 16 steps to success. New York: CRC Press
- 5) Gingrich, G. 2003. Managing IT Government, Business & Communities. USA: IRM Press
- 6) Kenneth, R. Baine. 2004. Integrated IT Project Management: a model-centric Approach. Boston: Artech House
- 7) Surbone, A., Kagawa-Snger, M. 2007. The illness trajectory of elderly cancer patients across cultures: SIOG position paper. Annals of Oncology
- 8) Larcher, B., Arisi, E., Berloff, F. 2003. Analysis of user-satisfaction with the use of a teleconsultation system in oncology. Med. Inform
- 9) Stafors, J., Holm-Sjogren, L., Schwieler, A. 2003. satisfaction with telemedicine presentation at a multidisciplinary tumour meeting among patients with head and neck cancer. Journal of Telemedicine and Telecare
- 10) Braford, W D., Kleit, A N., Krousel-Wood, M A., Fichard, M Fe. 2004. Willingness to pay for telemedicine assessed by the double-bounded dichotomous choice method. Chareston, USA: Journal of Telemedicine and Telecare
- 11) Harris, L., Dresser, C., L.Kreps, G. 2003. E-Health as Dialogue: Communication and Quality of Cancer Care
- 12) Lethiais, V. 2002. TIC et réorganisation spatiale des activités économiques: Développements théoriques et application aux services de santé. Rapport Scientifique

Stéphane Cottin, Jérémie Dequidt
(LIFL, INRIA, Lille)

Simulation de radiologie interventionnelle

With the recent FDA's decision to mandate training as a prerequisite for carotid stenting, a new emphasis has been brought to the simulation of interventional neuroradiology procedures. The development of a realistic computer-based training system would provide procedural and skills training in an integrated educational curriculum which will permit learning to occur safely, without putting patients at risk. Effective training requires a high level of realism, which has not been reached yet by existing systems. To meet this goal, we have created a prototype system utilizing new approaches to synthetic angiography, virtual catheter/guidewire modeling, and flow dynamics. Anatomical models of cerebrovascular anatomy are derived from CTA datasets using semi-automatic segmentation algorithms. Interactivity is provided through the use of actual catheters and guidewires tracked via a dedicated haptic interface. Designed to incorporate an educational curriculum providing a variety of case scenarios, the current system delivers high fidelity simulation while based on a cost-effective design.

Session 3 - Session Animation

A New Step for Artificial Creatures

Nicolas Lassabe
IRIT/UT1 UMR 5505
2 Rue du doyen Gabriel Marty
31042 Toulouse cedex 9
FRANCE
Email: lassabe@irit.fr

Hervé Luga
IRIT/UT1 UMR 5505
2 Rue du doyen Gabriel Marty
31042 Toulouse cedex 9
FRANCE
Email: luga@univ-tlse1.fr

Yves Duthen
IRIT/UT1 UMR 5505
2 Rue du doyen Gabriel Marty
31042 Toulouse cedex 9
FRANCE
Email: yves.duthen@univ-tlse1.fr

Abstract—In this article, we present our research on the evolution of morphology and behavior of complex creatures in virtual environments. We propose to study the evolution of creatures faced with different situations from crawling and walking to more complex activities like climbing and skating. Our creatures use solid 3D blocks and graphals like Karl Sims's creatures and a new type of controller based on classifier system. The results constitute a new step toward creatures adapted to more complex environments.

I. INTRODUCTION

In this article we propose a new step in virtual creature evolution. In recent years, many articles on evolving creatures have been released [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12]. They often focus on the morphology and behavior of one or two creatures but do not take into account the complexity of their environment. In contrast, some simulations focus on the interactions of population dynamics but deal only with much simpler creatures [13], [14]. Our approach tries to merge both approaches to evolve complete complex creatures in complex environment. After a brief discussion on previous approaches, we will present our system and focus on a new type of controller based on classifier systems. We will then show how evolved creatures are able to cope with various situations caused by a complex environment.

The present work is organised as follows: in the next section will present the works about evolving creatures. In the third section of this article we will present a brief description of our creatures. We will particularly emphasize our proposition to use a classifier system as a tool to model complex activation function. We will then present a set of benchmark tests to evaluate our architecture and we will describe the results. At the end we discuss how various fitness functions have an influence on the phylogeny of creatures and on various ideas for future studies. In the results section, we present the best evolving creatures from our experiments. The study of emerged strategies to resolve the various problems is promising and opens new perspectives. We discuss the influence of fitness on the symmetry and modularity of creatures. To conclude, we present some perspectives of the project and the interest that a complex ecosystem environment presents.

II. RELATED WORK

There are many existing solutions for representing morphologies and controllers. One of the first works was published

by A. Fukunaga, J. Marks and J.T. Ngo [15]. Karl Sims's creatures certainly remain the most evolved [8], [9]. He has achieved evolutions bound by very few constraints and obtained a large diversity of creatures giving birth to crawling and swimming behaviors. Thereafter many works reproduced his results with different approaches for the representation of morphologies and controllers, such as the L-System creatures of Hornby [3] and Lipson's Golem project [5]. Their goals were to optimize 3D creatures in relation to a preset fitness inside a physics engine and to reproduce them in the real world. Another approach to obtain more complex behavior is to optimize a set of 2D creatures in a virtual pool by means of natural selection [13], [14]. Each creature has an amount of energy and tries to survive looking for food in the environment. Dealing with simpler creatures, this approach allows to simulate many creatures at one time. Komosinski did the same experiment in a 3D environment [4]. Recently, the EvolGL project [16] proposed to simulate a virtual world with worms. The evolving worms belonged to herbivorous, carnivorous and omnivorous species and developed different strategies of survival.

These works distinguish two distinct ways to optimize complex behavior evolving creatures: with a preset fitness or by means of natural selection. The preset fitness approaches continue to be improved with recent results, like evolving flying creatures [7] and evolving creatures for fighting [6]. In these works, Miconi reproduced the Karl Sims's creatures and improved the co-evolution [17], [18], [12]. One promising perspective for the morphology of creatures could be artificial embryogeny to have a more complex representation of creatures [19]. However, much could be improved to offer better results in more complex environment [20], [21], [22]. The works based on natural selection are becoming achievable and will certainly be interesting to have more adaptive solutions [16].

III. ENVIRONMENT AND ARCHITECTURE

A. Morphology

In present work, our creatures use a morphology based on *graphals* similar to those from Karl Sims's ones [8], [9]. The genotype represents the creature's development. For each creature we define a root node. Each node contains the information of one block. Blocks are represented by solid 3D

cubes. The data of cubes are their x , y and z sizes, color and the link to the parent block if it is not the root block. The link between two nodes will be represented by a type of joint in the phenotype. A block could be linked with itself, this providing the possibility of repeating the expression of a node. The modularity is very visible in our results. We choose not to explicitly represent the symmetry in our creatures as in Karl Sims to study how the evolution could make it appear.

This morphology is not static but could be extended because we want to keep the ability of generating various lifeform inside our futur ecosystem. It provides us with good results for our current simulations (Fig. 12) but we plan in the near future to broaden the types of joints between blocks and types of blocks defining the creatures's morphologies.

B. Controller

The creature uses a classifier system [23], [24] as a controller. The controller needs to activate the articulations correctly (junction between the blocs). The requested signal to do this needs to be continuous and must vary in time according to the incoming information issued from the sensors. This signal is generally a sine wave [4] or a composition of periodical functions but it is also possible to compose a signal from some preset patterns (Fig. 1). In this latter approach, the main problem is to conserve the continuity of the signal, while selecting the next adapted pattern. We start by defining a data base of one thousand random generated patterns. We then define one main composition rule: two patterns could be appended to compose a part of the control signal only if the following conditions are satisfied:

- Patterns are defined between $[-1,1]$ and the same holds for their minimum and maximum.
- If P_1 and P_2 are two patterns and P_2 is appended to P_1 , it is necessary that:

$$(P_1(1) - P_2(-1)) < \varepsilon$$

where $\varepsilon = 0.01$

The classifiers control the selection of patterns. The entries of the classifiers, which define the incoming information from the environment, are the creature sensors. According to the sensors's state, a set of rules pointing to patterns in the data base is triggered. The composed signal is used by the effectors controlling the creatures's muscles. If the resulting action is satisfactory, the rules used to generate that action are rewarded, otherwise they are penalized. The advantages of this method lie in the fact that it is easy to enrich the used data base and obtain more complex signals than those obtained by simple primitive signal composition.

Another approach could be to use mathematical functions (which may be generated by the use of Genetic Programming [25] for example) to replace the pattern. However, there are too many possibilities and we do not exploit all of them here.

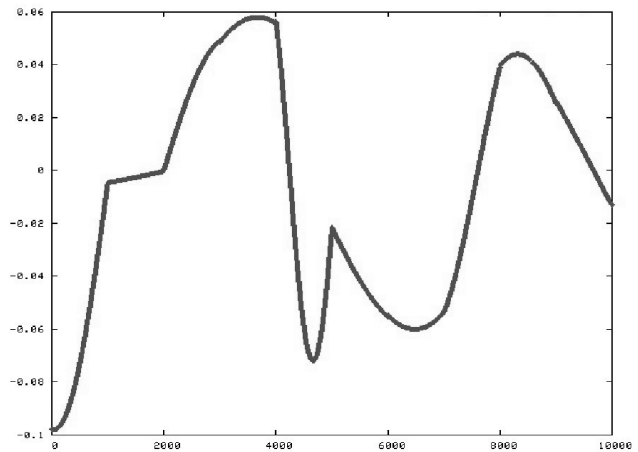


Fig. 1. Signal composed by a classifier system for a controller (x axis: steps, y axis: velocity).

C. Genetic operator

1) *Mutation*: The mutations are operated on the graph, hence if a mutation is operated on a node recursively reproduced, this mutation will affect all the blocks issued from this node. This allows our structure to exhibit modular properties such a pattern repetition.

The possible mutations on the graph are:

- A new block could be added randomly to the graph.
- A block could be deleted randomly from the graph. It is necessary to check whether or not this operation is possible. This operation cannot be applied to the root block. A part of the subgraph could be deleted by this mutation if blocks are connected to the deleted block.
- A mutation could slightly and randomly move one of the axes x , y or z of the parent block link between two connected blocks.
- A mutation could slightly and randomly move one of the axes x , y or z of the link between two connected blocks.
- A mutation could slightly and randomly center one of the axes x , y or z of the between one block and its parent.
- A mutation could slightly and randomly center one of the axes x , y or z of the link one parent and one of these blocks.
- The size x or y or z of a randomly selected block can be randomly changed.
- The number of iterations of a random node can be randomly changed.
- The number of iterations of a random node can be increased by one.
- The number of iterations of a random node can be decreased by one.
- The axis of a random joint can be changed.

2) *Crossover*: We use the *graphs* method as in Karl Sims's works [8], [9] to reproduce two creatures. That is, we randomly select one node in the graph of the first creature parent and another one in the second parent. To finish, we permute the

subgraphs issued from the selected nodes. This approach is simple and gives us good results. We plan to implement the other method by Karl Sims, *crossover*, the *picking* method of Miconi [18] in the next step of our work and compare the results to see the influence of the different methods of crossing over.

3) *Genetic algorithm*: The population of the simulation is initialized to one hundred individuals. Each creature's genotype is randomly created according to a maximum number of blocks. For a typical simulation, this number is generally set to ten. This number could quickly increase by mutation of the number of iterations on each node provided by the modular definition of our genotype. Each individual's performance is then evaluated in the simulated environment during 40000 steps (40s). The fitness used to evaluate the creatures will be described later for each experiment. For the selection process, we use tournament selection between three individuals. The rank of reproduction is seventy percent and that rank of mutation is ten percent. The evolution stops when the desired fitness is attained.

D. Breve

The implementation of a physics engine is a difficult task requiring a great deal of work. Our choice was therefore to use Breve [26], an engine based on ODE (Open Dynamic Engine). This environment is well suited for artificial life simulation because it provides a set of high level primitives which suit the simulation of articulated bodies. Breve includes low level physics dealing with torques and strength computation, gravity, collision detection but also several higher level property computations. For example, Breve already implements a lot of useful functionalities such as the quantity of light perceived by an object that we plan to use in future experiments. Our program is written in C++ and inserted in Breve as plugins. In this way, our program is not dependent on Breve and can be used as a library with another physics engine.

IV. RESULTS

We here show the ability of our system to evolve creatures in different situations. In general, in our experiments, the fitness tries to maximize the distance traveled by a creature. All the creatures are recorded on videos¹.

A. Walking on a flat floor

The first experiment is the traditional walk. This original experiment is similar to the work of Karl Sims [8], who presented the first one. Many works try to reproduce it. Its goal is to optimize both the morphology and the controller of a creature which has to walk on a flat floor environment.

In this experiment, our fitness F is not the distance traveled d but the average speed during the time t of evaluation of the creature. $F = d/t$

One of the best results that we get is a tripod (Fig. 2.a). This creature is not symmetric: it has only three blocks for

¹A video with the best creatures is on the first authors's home page: <http://www.irit.fr/~Nicolas.Lassabe/videos/alife2007.mov>

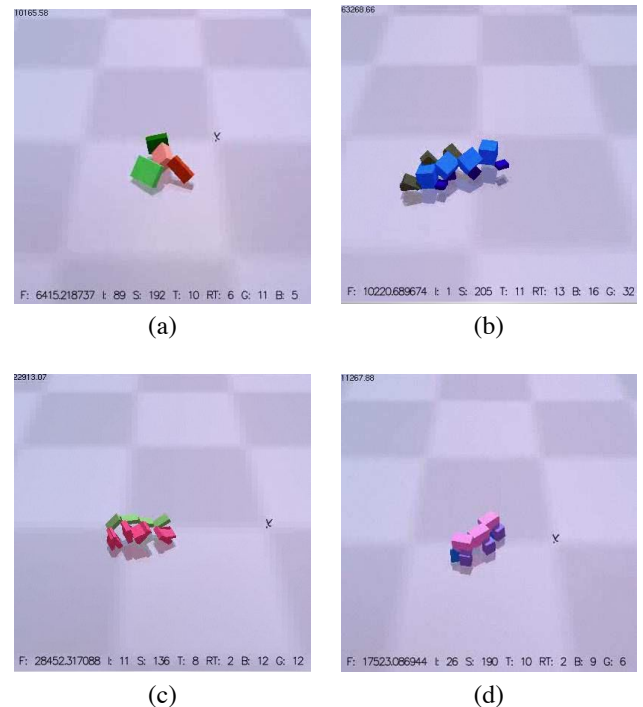


Fig. 2. (a) Evolving creature walking like a tripod. (b) Evolving creature walking like a crab. (c)(d) Creatures with a modular body

legs and one for the body and is not modular (no repeated block). Various other creatures used interesting strategies to move like a crab or have kinds of legs (Figs. 2.(b-d)) or rolling (Fig. 3.(a-b)).

B. Walking in a specific direction

In the experiment, we keep the same simulation and parameters but we change the fitness function. In this case, we indicate a specific direction for the creature, for example, to follow the x axis. The new fitness is $F = |x|/t$

One might imagine the results would be the same. In fact, the creatures are optimized to go in this direction and this is manifested by a perfect symmetry of the best creatures (Fig. 4). An interesting feature our creatures's morphology of creatures is that they are composed of a modular body. This modularity gives them a great efficiency. Among the best creatures are kinds of worms with some legs allows than to walk with long steps (Fig. 4(b-c)).

C. Thought trenches

In these experiments, we isolate the creature on a small cube. Some other cubes are aligned with the first one separated by trenches. The creature's goal is to join the last cube. Many strategies are possible like jumping over trenches. Or, if the creature is tall enough; it could crawl to the next cube.

1) *Interest of the experiment*: The interest of this experiment is that this situation often appears in reality and is a difficult task to resolve. It is easy to improve the difficulty of this task changing the width of the trenches. It is possible to alternate the positions and heights of cubes.

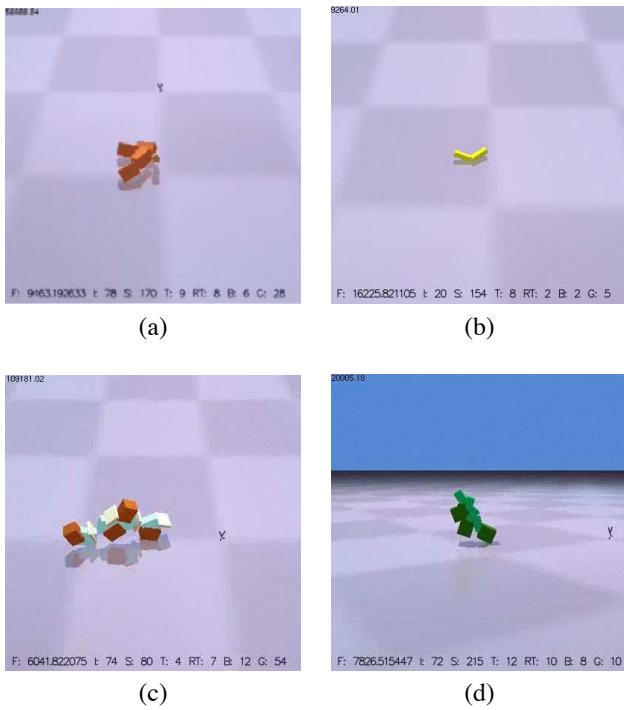


Fig. 3. (a)(b) Evolving creatures rolling. (c) Look like a robot (d) Creature with a human profil

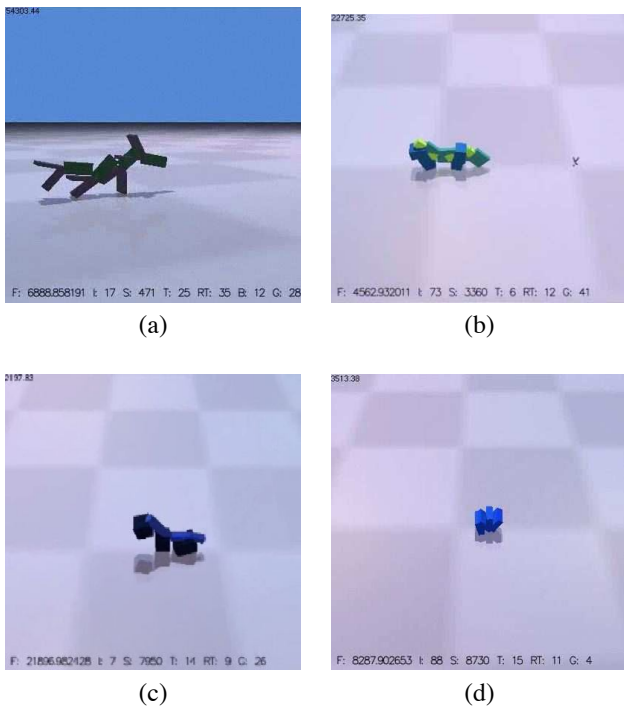


Fig. 4. (a) Evolving symmetric creature walking in a specific direction are symmetric. (b)(c) Modular evolving symmetric creatures with kinds of Legs. (d) Creature Rolling

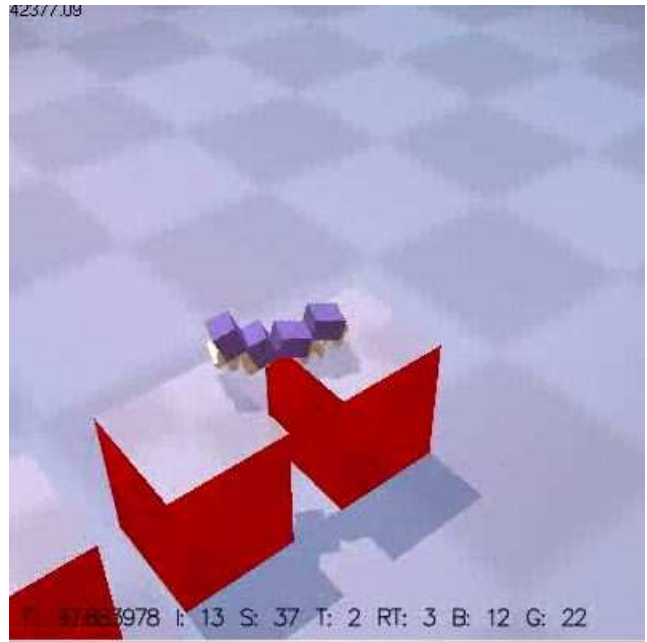


Fig. 5. Evolving creature walking cube by cube

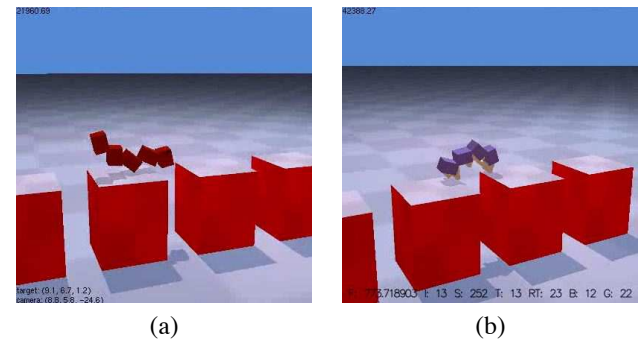


Fig. 6. Two evolving creatures on cubes.

2) *Experiment:* In the first simulation, the trenches are not very large, just a quarter the width of the cube. The fitness is defined to attract the creatures on the last cube. If the creature falls down the fitness is divided by ten, to push the creature to go in the direction of the next cube. It should not be possible for a creature to mount on top of a cube. The best creature uses its legs to cross the holes (Fig. 5.(b)).

D. Stair Climbing

The environment of the creature is often a flat floor. To increase the difficulty, we try to see if an evolution could generate a creature able to climb a stair. The difficulty could be increased by the height of steps. In the first test, we set up the height of the steps to half the average size of creatures and the width as the average width of the creature. The fitness is defined to indicate the direction to the top of the stair and optimizes by speed to which the creatures go.

1) *Interest of the experiment:* Stairs are everywhere in daily life, and can be of several different types. It is a difficulty for a

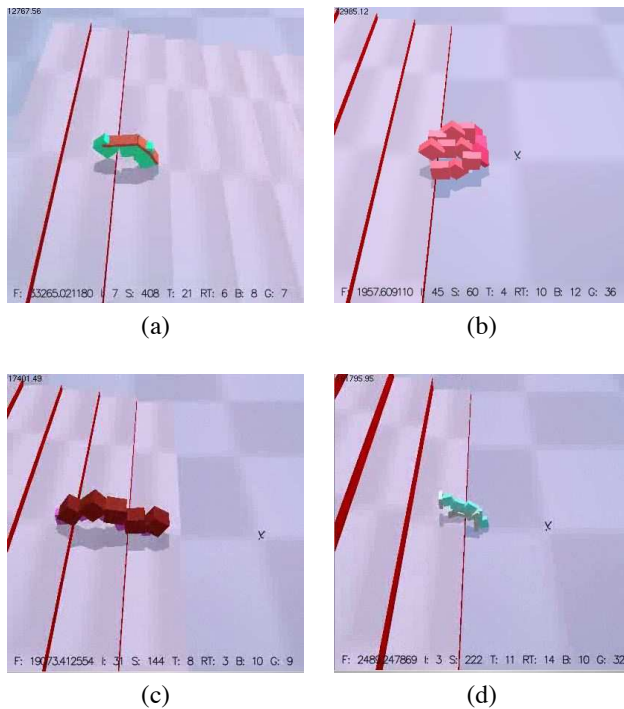


Fig. 7. (a) Evolving creature jumping step by step. (b) Evolving creature climbing a stair. The body of the creature is modular with optimized legs to quickly climb the stair. (c) Looks like a worm, (d) looks like a hand.

robot. Using evolving creatures to see the different strategies to climb a stair is an interesting approach. The morphology, the balance and the synchronisation of movements are difficult to design to work on any stair [27].

2) *Experiment*: The best creature can climb to the top of a stair quickly. The strategies employed are different. Some creatures which move like a worm are able to climb almost two steps at a time (Fig. 7.(c)). Others are perfectly designed to jump step-by-step (Fig. 7.(a)). Once again, the bodies of the best creatures are modular. Moreover the creatures are also symmetric (Fig. 8). If a creature has a good potential to climb but is not totally symmetric, the evolution will optimize the creature in this way. To improve the difficulty of the experiment, we increase the height of the step and run a new evolution. The best creature uses a new strategy to climb the step using flat legs (Fig. 7.(d)).

E. Walking terrain

The world is rarely perfect, so, in this experiment, we design a terrain with hills and dales. The amplitude of these heights is ten times greater than the average size of the creatures. The fitness is the distance traveled divided by the time of the simulation as in the walk experiment: $F = d/t$. Our creatures have more difficulty in being effective in this type of environment. This is because the environment is unpredictable. The best results are less efficient than the creatures on the floor but still have good performance (Fig. 9).

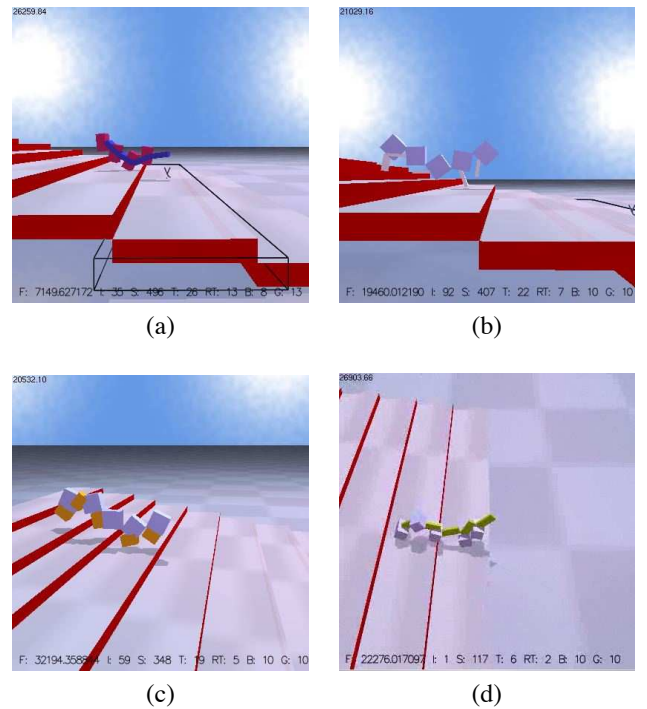


Fig. 8. Evolving creatures climbing a stair. The body of each creature is modular with optimized legs to climb the stair quickly.

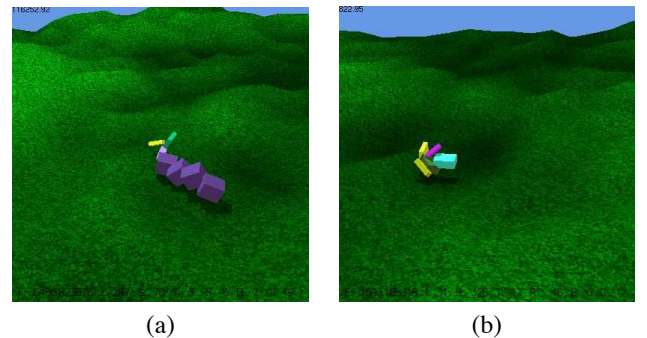


Fig. 9. Evolving creatures crawling on a terrain.

F. Skating

The interaction within the environment is an important issue in improving the complexity of evolving creatures. In general, we underestimate the capacity and the power of adaptivity of evolving creatures. Many experiments need to be tried to find out their real limit and their potentialities. In this experiment, we define a kind of skateboard composed of four wheels and one large board. The goal is to see creatures could use it whether to move in some direction. For this, the creatures fall down close to the skateboard. The creature and the skateboard are two different objects and do not have any connection between them. The function fitness is defined to maximize the distance covered with skate ds .

$$F = d + ds - (d - ds)$$

Many strategies shows employed: some creatures jump on

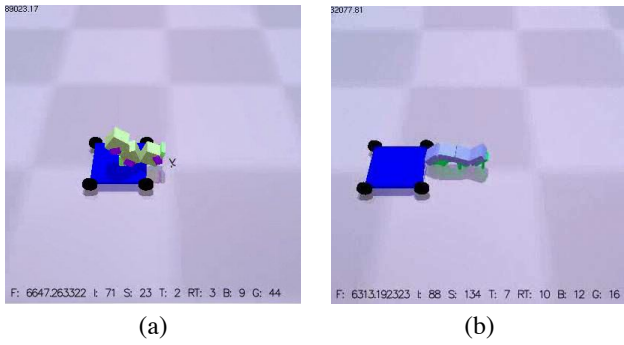


Fig. 10. (a) Evolving creature skating by jumping on the skate. (b) Evolving creature pushing the skate.

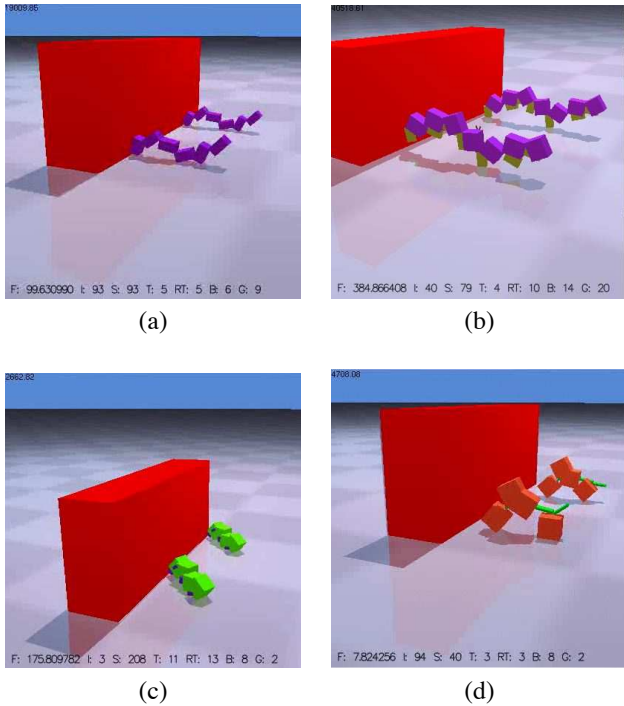


Fig. 11. (a)(b) Evolution of creatures few generations later. (c) Evolving creatures pushing a cube. (d) Cooperation between evolving creature.

the skate to slide with it (Fig. 10.(a)), others push it and run behind it (Fig. 10.(b)) and others go up and push from the back to move. These results show the ability of the evolving creatures to be adapting to the environment.

G. Cooperation

In the last experiment, the goal is to push a block in the right direction. The block is defined to be too big for only one creature to push. For this experiment, one creature and his clone are initialized for each simulation in the environment. The fitness is to maximize the distance traveled by the wall dw (center of gravity) in one direction (axis x): $F = dw$. In this experiment, the best creatures are bigger to push the wall (Fig. 11) and give every good results. In the future, we plan to improve this experiment, using two different creatures.



Fig. 12. The project of our ecosystem

V. DISCUSSION

These results prove the possibility of using a new kind of controller based on classifier systems to control evolving creatures in various activities. The next step is to emerge our system inside our virtual ecosystem with plants and various lifeforms. We have already started this implementation [28] (Fig. 12).

A. Implementation

For the moment, the lifeforms and most of the plants have been implemented. An evolution of the creatures with a preset fitness presented in this article give some good results. For the final simulation, we will use Breve network implementation to distribute all the calculations on different machines. The complete simulation will require a great deal of computation and will not be computed on a single machine. Every lifeform is implemented in C++ in different plugins and these plugins are interfaced with Breve. We try to maintain a maximal flexibility to add different lifeforms in order to get the broadest, most various and interesting results possible.

B. Why ecosystem?

The realization of an ecosystem can allow the generation of lifeforms able to adapt to a dynamic environment, the specification of species and the emergence of life cycles. The interest of simulating such virtual ecosystems are several. In the field of biology, it could lead to a better understanding of the evolution and extinctions of species as well as of some mechanisms of life [29]. The evolution of morphology and behavior in relation to a specific environment has an interest for the design of robots and their controllers [30], [5],

[31]. A complex environment should generate adapted survival strategies during evolutions cycles.

Interesting questions are whether different evolution scenarios can emerge from an identical initial environment and whether the environment complexity is a relevant criterion on the emergence of different survival strategies [20], [21], [22]. If that were the case, how does one evaluate complexity? What are the elements necessary for the emergence of complex life forms and behaviors? Is the use of a complex physics one of these elements or can biological complexity be supported by simpler physical models? What is the importance of the interactions? These are open questions, which may be answered through experimentation with a complex artificial ecosystems.

VI. CONCLUSION

We have shown that it is becoming possible to evolve virtual creatures in complex environment that provide us with physical simulation but also with some kinds of interactions between the creatures and the simulated world. The implementation of such ecosystems becomes essential to encourage the development of complex adaptation strategies and to provide more complex behaviors, which were not possible to obtain in simple worlds. It will also allow regrouping evolving simulations of several species to cope with pilogeny dynamics (appearance and extinction of new species) inside one application (i.e., one world) and therefore, allow observing the relations between the evolved species. For that purpose we propose a new approach which uses classifier systems as controllers as an alternative to the use of neural networks. Several experiment were analyzed which exhibit a strong link between the fitness function and the morphology of the evolved creatures in terms of symmetry and modularity.

VII. PERSPECTIVES

The next step for our project will be to propose some paths to improve the interaction within the environment, and also to confront the evolved creatures with more complex situations to exhibit their adaptation abilities. Introducing an evolving language between the creatures could be interesting and would allow a significant increase in the number of evolving strategies. It could also be very valuable to change our morphogenesis definition allowing genotypic re-use. This could lead us to an evolving system based on the artificial embryology paradigm where a compact genetic code could lead to the growth of complex structures. Another possibility for the controller could be to compose patterns by genetic programming as we do in generic composition patterns [32].

REFERENCES

- [1] J. Bongard and C. Paul, "Investigating morphological symmetry and locomotive efficiency using virtual embodied evolution," in *From Animals to Animats: The Sixth International Conference on the Simulation of Adaptive Behaviour*, J.-A. M. et al., Ed., 2000.
- [2] P. Eggenberger, "Evolving morphologies of simulated 3D organisms based on differential gene expression," in *Proceedings of the Fourth European Conference on Artificial Life, ECAL97*, P. Husbands and I. Harvey, Eds. MIT Press, 1997, pp. 205–213.
- [3] G. Hornby and J. B. Pollack, "Evolving l-systems to generate virtual creatures." *Computers & Graphics*, vol. 25, no. 6, pp. 1041–1048, 2001.
- [4] M. Komosinski, "The world of framsticks: Simulation, evolution, interaction," in *VW '00: Proceedings of the Second International Conference on Virtual Worlds*. London, UK: Springer-Verlag, 2000, pp. 214–224.
- [5] J. B. Pollack, G. S. Hornby, H. Lipson, and P. Funes, "Computer creativity in the automatic design of robots," *Nature*, vol. 406, 2000.
- [6] M. J. T. O'Kellym and K. Hsiao, "Evolving simulated mutually perceptive creatures for combat." in *Proc. of Artificial Life IX, ed.*, I. T. Bedau M., Husbands P. and W. R. A., Eds., 2004, pp. p111–118.
- [7] S.-J. K. Yoon-Sik Shim and C.-H. Kim, "Evolving flying creatures with path following behaviour." in *Proceedings of 9th International Conference on the Simulation and Synthesis of Living Systems/ (ALIFE IX)*. Artificial Life, September 2004.
- [8] K. Sims, "Evolving 3d morphology and behavior by competition," in *Artificial Life IV*, Artificial Life. MIT Press, 1994, pp. pp. 28–39.
- [9] —, "Evolving virtual creatures," in *Siggraph'94*, 1994, pp. pp. 15–22.
- [10] T. Taylor and C. Massey, "Recent developments in the evolution of morphologies and controllers for physically simulated creatures," 2000.
- [11] J. Teo and H. A. Abbass, "Multiobjectivity and complexity in embodied cognition," *IEEE Trans. Evolutionary Computation*, vol. 9, no. 4, pp. 337–360, 2005.
- [12] T. Miconi and A. Channon, "The n-strikes-out algorithm: A steady-state algorithm for coevolution," in *the IEEE Congress on Evolutionary Computation*, 2006.
- [13] J. Ventrella, "Designing emergence in animated artificial life worlds," in *VW '98: Proceedings of the First International Conference on Virtual Worlds*. London, UK: Springer-Verlag, 1998, pp. 143–155.
- [14] —, "Attractiveness vs. efficiency (how mate preference affects location in the evolution of artificial swimming organisms)," in *ALIFE: Proceedings of the sixth international conference on Artificial life*. Cambridge, MA, USA: MIT Press, 1998, pp. 178–186.
- [15] A. Fukunaga, J. Marks, , and J. Ngo, "Automatic control of physically realistic animated figures using evolutionary programming," in *Proc. of Third Annual Conference on Evolutionary Programming*, 1994, pp. 76–83.
- [16] S. Garcia Carbajal, M. B. Moran, and F. G. Martinez, "EvolGL: life in a pond," in *Artificial Life XI Ninth International Conference on the Simulation and Synthesis of Living Systems*, J. Pollack, M. Bedau, P. Husbands, T. Ikegami, and R. A. Watson, Eds. Boston, Massachusetts: The MIT Press, 12-15 Sept. 2004, pp. 75–80.
- [17] T. Miconi and A. Channon, "A virtual creatures model for studies in artificial evolution," 2005.
- [18] —, "An improved system for artificial creatures evolution," Cambridge, MA, USA, 2006.
- [19] K. O. Stanley and R. Miikkulainen, "A taxonomy for artificial embryogeny," *Artif. Life*, vol. 9, no. 2, pp. 93–130, 2003.
- [20] C. Adami, C. Ofria, and T. C. Collier, "Evolution of biological complexity," *PROC.NAT.ACAD.SCI*, vol. 97, p. 4463, 2000.
- [21] A. K. Seth and G. M. Edelman, "Environment and behavior influence the complexity of evolved neural networks," *Adaptive Behavior*, vol. 12, no. 1, pp. 5–20, 2004.
- [22] A. K. Seth, "The evolution of complexity and the value of variability," in *ALIFE: Proceedings of the sixth international conference on Artificial life*. Cambridge, MA, USA: MIT Press, 1998, pp. 209–218.
- [23] J. H. Holland and J. S. Reitman, "Cognitive systems based on adaptive algorithms," *SIGART Bull.*, no. 63, pp. 49–49, 1978.
- [24] H. Olivier, S. Stephane, B. Alain, and L. Herve, "Generic classifiers system and learning behaviours in virtual worlds," in *CW '04: Proceedings of the 2004 International Conference on Cyberworlds (CW'04)*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 113–120.
- [25] J. R. Koza, "Genetic programming," in *Encyclopedia of Computer Science and Technology*, J. G. Williams and A. Kent, Eds., vol. 39. Marcel-Dekker, 1998, pp. 29–43.
- [26] J. Klein, "Breve: a 3d environment for the simulation of decentralized systems and artificial life," in *ICAL 2003: Proceedings of the eighth international conference on Artificial life*. Cambridge, MA, USA: MIT Press, 2003, pp. 329–334.
- [27] M. M. Dalvand and M. M. Moghadam, "Stair climber smart mobile robot (msrox)," *Auton. Robots*, vol. 20, no. 1, pp. 3–14, 2006.
- [28] N. Lassabe, H. Luga, and Y. Duthen, "Evolving creatures in virtual ecosystems," in *16th International Conference on Artificial Reality and Telexistence*, to appear.
- [29] G. Abramson, "Ecological model of extinctions," p. 785, 1997.

- [30] D. Floreano and F. Mondada, "Automatic creation of an autonomous agent: Genetic evolution of a neural-network driven robot," in *Proceedings of the Conference on Simulation of Adaptive Behavior*, 1994.
- [31] S. Nolfi and D. Floreano, "Synthesis of autonomous robots through evolution," 2002.
- [32] N. Lassabe, S. Sanchez, H. Luga, and Y. Duthen, "Genetically programmed strategies for chess endgame," in *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*. New York, NY, USA: ACM Press, 2006, pp. 831–838.

On-line physically-based control of mass-interaction models

Ali Allaoui
ICA Laboratory
Ali.Allaoui@imag.fr
www-acroe.imag.fr

Matthieu Evrard
ICA Laboratory
Matthieu.Evrard@imag.fr
www-acroe.imag.fr

Annie Luciani
ICA Laboratory
Annie.Luciani@imag.fr
www-acroe.imag.fr

Abstract

Physical modeling allows producing a large set of motions for computer-based animation. Among the various proposed formalisms to design such physically-based models, the mass-interaction formalism is quite interesting because of its modularity, the simplicity of each module, and its generic aspect that leads to produce behaviors as different as deformable and non-deformable objects, smoke, pastes, liquids, crowds... A specific difficulty is to deal with phenomena that are highly non-linear by nature. The point of this article is to present a way of dealing with such phenomena. We describe a novel tool inside the mass-interaction formalism that permits to physically control such dynamic changes in the most generic way. It consists in introducing within the mass-interaction formalism a generic module which controls on-line a parameter of any physical module, mass or interaction. Once this tool is provided, it makes the design of a controller component possible by means of the same mass-interaction formalism. The article then tries to illustrate the methodology needed to design such controller by presenting two complementary cases in which the physical parameters are changing according to the dynamic states of the system. These examples are (1) the “stricture effect”, in which a fracture of a material occurs when stretches change its elasticity, and (2) a simple human gesture, a repeated jump, which requires to modify in a cyclic manner the tonicity of the muscle.

Keywords: physics-based animation, motion control, deformable models, mass-interaction, simulation, user-friendly interface, animation formalism.

1 Introduction

Physical modeling has become a widely used means to design animation. The mass-interaction system, that is a generalization of the usual mass-spring implementations, is able to produce a wide variety of different behaviors [1][2][3][4]. Very soon, it has been improved to obtain complex evolving phenomena by introducing an on-line control of some physical parameters, such as the rest length [5][6][7] or the stiffness of springs [8]. The work proposed here aims at generalizing these approaches by proposing a control structure constituted by three components:

- (1) A generic module for the on-line control of physical parameters of a 3D model, compatible with the mass-interaction formalism
- (2) A 1D mass-interaction network representing the controller disposal which provides scalar to (1)
- (3) Interaction between the 3D model and (2) that provided information on the 3D model state to (2).

This approach can be compared to other implementations of controls of physically based modules, as shown in section 2. Section 3 deals with our implementation of the generic modules that has been done inside MIMESIS, a user-friendly modeling interface. Then, section 4 presents the global scheme of the proposed

controller component. The use of this component is illustrated by two complementary modeling cases in which the physical parameters are changing according the dynamic states of the system: (1) the “stricture effect”, in which a fracture of a material occurs when stretches change its elasticity, and (2) a simple human gesture, a repeated jump, which requires to modify in a cyclic manner the tonicity of the muscle.

2 State of the Art

Physical modeling is a generative way of producing motion, contrary to descriptive modeling like key-framing for example. Let us call physical parameters the following ones: stiffness, viscosity, thresholds in distance and velocity, static and dynamic coefficients of friction, inertia. Consequently, positions, velocities, angles, forces, torques, as in physics are not parameters but variables that are inputs or outputs of the physical models.

When dealing with the control of physically-based models, we can distinguish two main directions: (1) one aiming at acting on the variables, for example forces, positions, velocity fields as done in [9,10]; (2) a second acting on the physical parameters of the 3D objects of the scene. In this paper, we focus on this second type of control.

A first approach to control the physical parameters of a physically-based model is described in figure 1.

A block of control is designed to modify the physical parameters of the 3D object. This technique has been used to modify the matter state changing by Terzopoulos and al. in [6]. It has been also used to obtain autonomous motor behaviors as in pioneer works of Miller [5], in which a muscle is modeled by a spring which rest length is modulated.

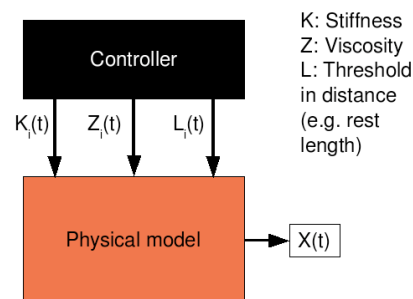


Figure 1: On-line control that do not take into account the state of the physical model.

It has been also used in [11] in which a high level behavioral layer is added, as described in figure 2, to select parameter evolution functions according to explicit behavioral states in conjunction with the 3D object variables.

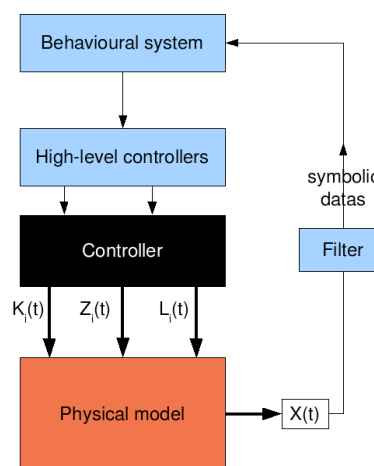


Figure 2: High level control and behavioral system

Another approach is the so-called “controller synthesis” that consists in producing controllers that satisfy the goals specified by the animator. The general scheme of such approach, based on a “generate and test” strategy, is described in figure 3. It has been used by [12] and [13] to control mass-spring models.

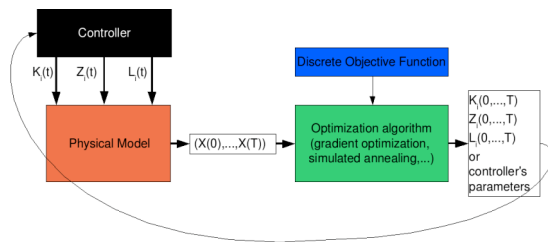


Figure 3: Controller synthesis

Another implementation of the paradigm of “controller synthesis” has been developed by Van de Panne and Fiume in [14] and [15], in sensors-actuators networks where the controller is a neural network.

Differently than in the previous approaches represented in figures 2 and 3, this approach implements a proportional-derivative servo-controller that aligns some data of the 3D object to the desired values computed by the neural network.

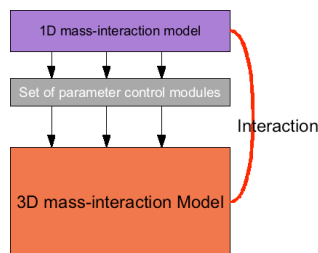


Figure 4: On-line physically-based control of physical parameters

We propose to design a controller system in two layers, as shown in figure 4:

(1) The controller itself: it is a physically-based mass-interaction system. Differently than all the previous approaches, this allows the user to build it so that it takes part in the animation process. The controller is in physical interaction with the 3D object. It reacts on line to the dynamical states of the 3D objects in their physical environments. This physical interaction is also designed by the user.

(2) A set of modules that modify the physical parameters of the 3D objects from the data provided by the controller.

So doing, we implement what can be called non-linear parameters modifications as it is in complex non-linear phenomena like Van der Pol oscillators for example [16].

3 The Parameter Control Module (PCM) in MIMESIS software

MIMESIS is a software based on mass-interaction modeling ([17]). In MIMESIS, the mass-interaction formalism is the core of the creation process at hand. Basically, MIMESIS is an interface that allows end-users to design mass-interaction networks as proposed by Luciani & al.([2]) in a user-friendly environment. It is therefore able to manage a lot of modules, edit initial conditions and physical parameters, simulate the designed networks, import and export evolution functions, coat the obtained movement, etc.

Until now, MIMESIS only has support for two categories of modules, the MATs and the LIAs (figure 5, up). A MAT module is a module that takes forces as input and returns a position. It is a kind of material element whose main parameter is its mass (or inertia). A LIA module connects two MATs. It represents the interaction between both MATs. This interaction can be a simple linear damped spring but also a more complex non-linear interaction (cohesion, plasticity, friction, etc.). Physical parameters of LIAs are stiffness, viscosity, thresholds on distance, static and dynamic friction factors.

This article introduces a new module called PCM (parameter control module). It allows connecting a MAT or a LIA parameter to the position of a MAT, i.e. the position of this MAT will give the value of the parameter. Unlike MAT and LIA, it is a directed module: it takes a position from the MAT, eventually applies a mathematical function on it (linear or exponential in the current implementation), and sets the parameter value to this scalar. As a parameter is a scalar value, the MAT connected to this parameter through our PCM module is a 1D MAT, i.e. a MAT that evolves in a 3D space or more basically whose algorithm return a 1D position (see figure 5, down).

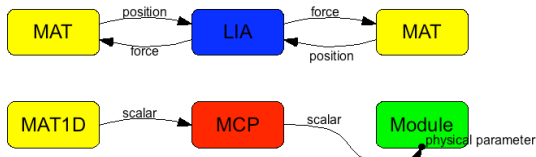


Figure 5: Modules in MIMESIS

4 Global scheme of the controller component

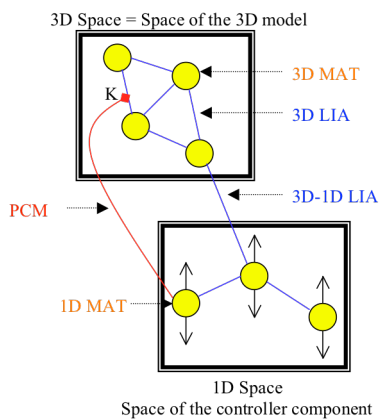


Figure 6: Space of the 3D model and space of the controller

Designing a mass-interaction controller component consists in:

- (1) Building a one-dimensional mass-interaction network. The low frame in figure 6 represents the 1D space in which this network evolves.
- (2) Connecting it to the output of a 3D model (figure 6, upper frame) to acquire state of the system. This is done by connecting the 1D MAT of the controller component to 3D MAT of the 3D models through 3D-1D LIA.
- (3) Connecting it to the dynamic parameters of this model through the PCM (figure 6, link in red).

Though such simple modules are easy to create and manipulate because of their modular nature, designing a low-level controller using mass-interaction requires a learning phase. Finding the critical parameters, the ranges within they evolve, mastering the generated non-linear dynamic is obviously a task that cannot be done without practice. In order to illustrate the way of thinking while

designing such kind of models this article presents two complementary basic illustrative examples in the two following sections.

5 Modeling a elementary stricture phenomena

5.1 The phenomena and the model controlled

A fracture with stricture effects like in ductile materials occurs when you stretch an elastic rubber till it tears apart (figure 7).

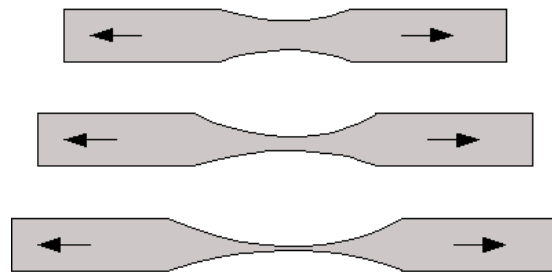


Figure 7: Stricture phenomena

As another example, we model a phenomenon that is close to a water drop falling from a tap. The tap will be a fixed MAT (called SOL), and the drop a moving MAT (called MAS), submitted to gravity and linked to the tap by a cohesion interaction (called COH3). This interaction is a piecewise linear interaction approaching the Van der Waals interaction (see figure 8).

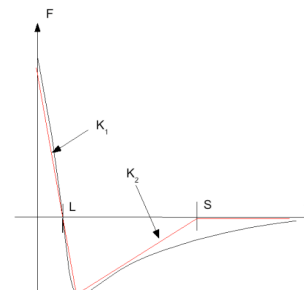


Figure 8: Cohesion interaction

Figure 9 presents a functional view of the model, explicating the role of parameters L and S of COH3.

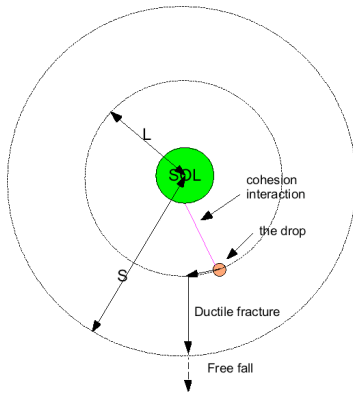


Figure 9: The model controlled

5.2 The controller

Ductile fracture will occur because of the weakening of the COH3 interaction. This weakening is triggered by parameters modification through our PCM when the drop comes down. The controller described in figure 10 does this.

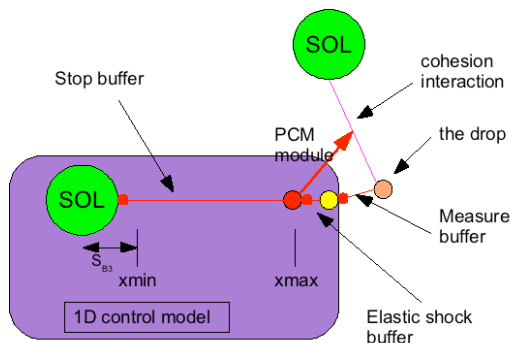


Figure 10: Stricture model and its control

The buffer interactions that can be seen in figure 10 are simple non-linear interactions, that model contact through a visco-elasticity activated when elongation falls under a specified threshold (figure 11).

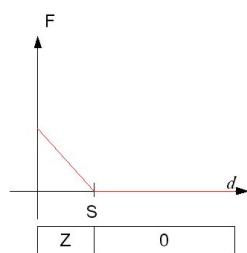


Figure 11: The buffer interaction

When the drop comes down, it hits a first mass (in yellow) thanks to the measure buffer. This is how the controller is connected to the 3D model. The information retrieved is the time when the drop comes down. Then this yellow mass hits in a perfect elastic shock the red mass. This last mass, travels from x_{max} to x_{min} , stopped by the stop buffer, a very viscous buffer interaction. The controlling 1D mass-interaction network is composed of the yellow and the red masses plus the SOL module and their interactions.

Finally, the position of the red mass controls the cohesion interaction via the PCM.

5.3 Two types of parameter control

We have experimented two ways of controlling the cohesion interaction (COH3), as described in figure 12.a) and 12.b).

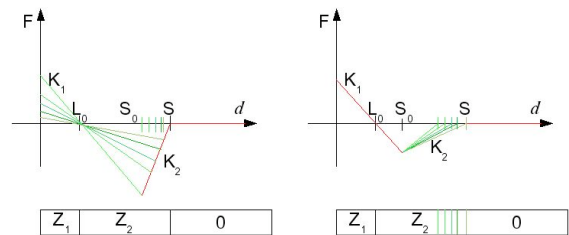


Figure 12.a): control of the parameter K_1 of COH3

Figure 12.b): control of the parameter S of COH3 together with K_2

In a) K_1 is controlled but the other parameters (K_2 , L and S) remain unchanged. Since S_0 is an implicit parameter of COH3 for the user who only manipulates K_1 , K_2 , L , S , Z_1 and Z_2 , this parameter is implicitly changed by the PCM.

In b), K_2 is controlled together with S , so that S_0 remains unchanged. These two controls are very different in theory: the first one changes the matter of the object without changing its size, the second one changes the matter and the size together.

5.4 Tuning the controller model

In order to obtain the stricture phenomena, we must determine which parameter is critical, i.e. which parameter determines whether the drop falls or not, and its dynamic during the ductile fracture. Once identified, we will choose the

range within which the controlled parameters evolve, and their dynamic of evolution.

This critical value is F_{rmax} , the value taken by the interaction function at S_0 and that is the minimum of the interaction function. Actually, F_{rmax} corresponds to the maximum force applied to maintain the cohesion. Consequently, the fracture will occur when this force is less than the weight of the drop plus its motion quantity.

Control a) makes that force decrease with K_1 , so we choose $K_{1,min}$ in order to have $F_{rmax,min}$ close to the weight of the drop and $K_{1,max}$ big enough to maintain the cohesion before the beginning of the fracture. We use a linear PCM module in order to convert $[x_{min}, x_{max}]$ to $[K_{1,min}, K_{1,max}]$. To prevent a premature fall, we adjust the motion quantity of the drop by slowing down the red mass. We do this by playing on the ambient viscosity, which is a parameter of the mass.

Control b) doesn't change this force, so it is already too low to counter just the drop's weight, and the control effect is just to slow down an unavoidable fall. The control must begin just after the drop has come down, unless the drop directly enters in free fall. We tune this by adjusting the threshold of the elastic shock buffer (figure 10), the bigger it is, the sooner the control will begin.

Figure 15 shows the result coated with Povray.

6 Modeling an elementary human motor gesture: jumps

6.1 The phenomenon and the model

We based our approach on a passive mass-interaction model of a jump introduced by Hsieh in [18]. In this model the muscles of the leg are modeled as a unique spring connecting the foot and the pelvis (figure 13).

Since it is a passive model, the rest length is not changed during the simulation, the jump is caused by the initial velocity of the pelvis. We wanted to make this model active in order to have a jumper who will jump repeatedly: he

starts a new jump as soon as he has landed.

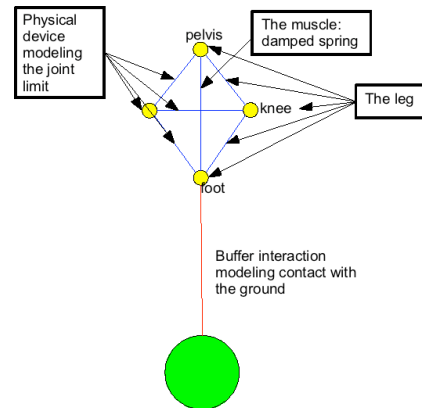


Figure 13: 3D model of a jumping leg

6.2 The controller

To do this we have modulated the rest length of the muscle by means of a mass-interaction controller. Figure 14 describes how the controller works.

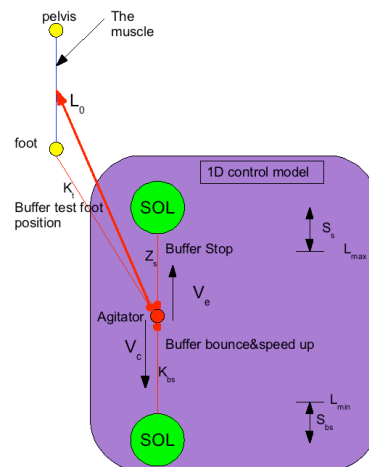


Figure 14: Repeated jump model and its control

The key element is once again the buffer interactions, which maintain the agitator between L_{min} and L_{max} and connects the controller to the foot, in order to know when the jumper lands.

The jumper prepares a jump by contracting his muscle, i.e. by decreasing the muscle rest length from L_{max} to L_{min} . Thus, the first jump is initiated by the initial velocity of the

agitator, $V_c(0)$. Then he jumps by extending his muscle quickly, i.e. by increasing the muscle rest length from L_{\min} to L_{\max} . This is done by making the agitator bounce and speed up when he reaches L_{\min} value, thanks to a well-tuned elastic buffer. When he reaches L_{\max} , the agitator is stopped by a highly viscous buffer to maintain the legs in extension during the jump. Finally, when the jumper hits the ground, he also hits the agitator, so that another jump is initiated.

6.3 Tuning the controller model

At first, we noticed that mainly 3 phenomenological parameters are significant at each cycle: the height of the jump H_j , the velocity of contraction V_c , and the velocity of extension V_e .

Moreover, these 3 parameters are influencing each other, from a cycle $n-1$ to the next cycle n :

- $H_j(n)=f(V_e(n),V_c(n))$
- $V_c(n)=g(H_j(n-1))$
- $V_e(n)=h(V_c(n))$

Fortunately, we have tools to tune these functions. They are the parameters of the buffer interactions. First, the stop buffer is just a viscous interaction, tuned to stop the agitator the closest to L_{\max} . Then, the buffer "test foot position" is the softest possible to detach the agitator from the influence of the stop buffer, so that V_c is the smallest possible. Finally, the buffer "bounce&speed up" takes advantage of the defaults of discretization, defaults that can be usefully mastered thanks to the work of F.Thill in [19] who makes the correlation between stiffness values and the gain or loss of energy. If we succeed to keep V_c in a certain range, then we will have V_e big enough to make the jumper jumps and small enough to make him jump not too high (unless V_c will exit from the desired range). Using this method, we have found parameters that make the jumper jumps about 10 times before it goes out of the desired range.

Figure 16 shows one cycle.

7 Conclusion

Generating highly non-linear movements is always a tricky task in modeling. However, to obtain a certain degree of complexity, non-linearities must be taken into account. Indeed, a lot of natural well-known phenomena are themselves deeply non-linear, as those in which there are changes in the physical states of the matter (for example, fluid - solid transitions), modulation of physical parameters according to the states of the system (for example, modulation of the elasticity according to the elongation of the spring), or in active systems such as human motricity. After the huge developments in computer animation, we assumed that it is a way to reach a new step in the complexity and the expressiveness of synthetic motions.

Two questions have been risen in the work presented here: first, how to give access to the user in the designing of such type of control? And second how implement it? We proposed to introduce a specific module for controlling physical parameters in a mass-interaction generic system and we set up methods to design controllers that are also physically-based and consequently in closed dynamic interaction with the objects to be controlled. So doing, we initiated the process of implementation of inmost closed-loop control systems, as it is necessary in complex non-linear systems. This method corresponds really to a shift in the way of thinking about the control process: from explicit control to implicit dynamic control. We experimented this approach in two complementary examples: the modulation of elasticity and the modification of the tonicity of a simple muscle-like object as it occurs in an active system.

The modules and models developed in this work have been implemented for the first time, within a user-friendly interface, which allow end-users to design mass-interaction models

for animation. Consequently, they are available for any user who wants to understand the method and so doing, to become able to use it to design his own physically-based models including dynamic parameters control. We assume that it could be the best "practical" way to introduce them freely in the domain of complex dynamic modeling in which computer animation is entering.

References

- [1] G.S.P. Miller, A. Pearce. Globular dynamics : a connected particle system for animating viscous fluids. *Computers & Graphics*, vol. 23(3):169-178, Elsevier, 1989.
- [2] A. Luciani, S. Jimenez, J.-L. Florens, C. Cadoz, O. Raoult. Computational physics: a modeler simulator for animated physical objects. *Proceedings of the European Computer Graphics Conference and Exhibition*. Eurographics'91, pp 425-436, Vienna, Austria, Elsevier Ed, Sep. 1991.
- [3] D. Tonnesen. Modeling liquids and solids using thermal particles. *Graphics Interface'91*, 255-262, 1991.
- [4] X. Provot. Deformation constraints in a mass-spring model to describe rigid cloth behaviour. *Proceedings of Graphics Interface 1995*, Graphic Interface'95, pp 147-154, Quebec City, Canada, (May 1995).
- [5] G.S.P Miller. The motion dynamics of snakes and worms. *Computer Graphics*, vol. 22, SIGGRAPH'88, 169-178, Aug 1988.
- [6] P. Boulenguez, A. Luciani. A Physical Particle Model of Morphogenesis. WSCG'03 Plzen (Czech Republic).
- [7] Sodaplay java applet - <http://www.sodaplay.com>
- [8] D. Terzopoulos, J. Platt, K. Fleisher. Heating and melting deformable models : from goop to glop. *Graphics Interface'89*, 219-226, 1989.
- [9] R. Barzel, A. H. Barr. A modeling system based on dynamic constraints. In *Proceedings of SIGGRAPH 1988*, ACM SIGGRAPH, 1988, pp. 179-188.
- [10] A. Witkin, M. Kass. Spacetime constraints. *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, p.159-168, June 1988
- [11] X. Tu and D. Terzopoulos, Artificial Fishes: Physics, Locomotion, Perception, Behavior. In *Computer Graphics*, volume 28, pages 43–49. SIGGRAPH'94, ACM Press (New York, USA), August 1994.
- [12] R. Grzeszczuk and D. Terzopoulos. Automated learning of muscle-actuated locomotion through control abstraction. In S. G. Mair and R. Cook, editors, *Automated learning of muscle-actuated locomotion through control abstraction*, pages 63–70, Los Angeles, California, August 1995. SIGGRAPH'95, ACM Press (New York, USA).
- [13] R. Grzeszczuk, D. Terzopoulos, and G. Hinton. Neuroanimator: fast neural network emulation and control of physics-based models. In S. Cunningham, W. Bransford, and M. F. Cohen, editors, *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 9–20, Orlando, Florida, July 1998. SIGGRAPH'98, ACM Press (New York, USA).
- [14] M. van de Panne, E. Fiume. Sensor-actuator networks. In M. C. Whitton, editor, *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 335–341, Anaheim, California, August 1993. SIGGRAPH'93, ACM Press (New York, USA).
- [15] M. van de Panne, R. Kim, and E. Fiume. Virtual Wind-up Toys for Animation. *Graphics Interface*, pp. 208--315, 1994.
- [16] B. van der Pol, J. van der Mark. Frequency demultiplication. *Nature*, 120, 363-364, (1927)
- [17] M. Evrard, A. Luciani, N. Castagné. MIMESIS : Interactive Interface for Mass-Interaction Modeling, In *Proceedings of CASA 2006*, Geneva, July 2006, Nadia Magnenat-Thalmann & al. editors. pp177-186.

- [18] C. Hsieh, A. Luciani. Generating Dance Verbs and Assisting Computer Choreography. Proceedings of MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia, pp 774—782. 2005.
- [19] F. Thil. Simulations de collisions et application à la synthèse d'objets sonores et percussifs. Rapport de stage de DEA, ACROE-ICA Grenoble 2003

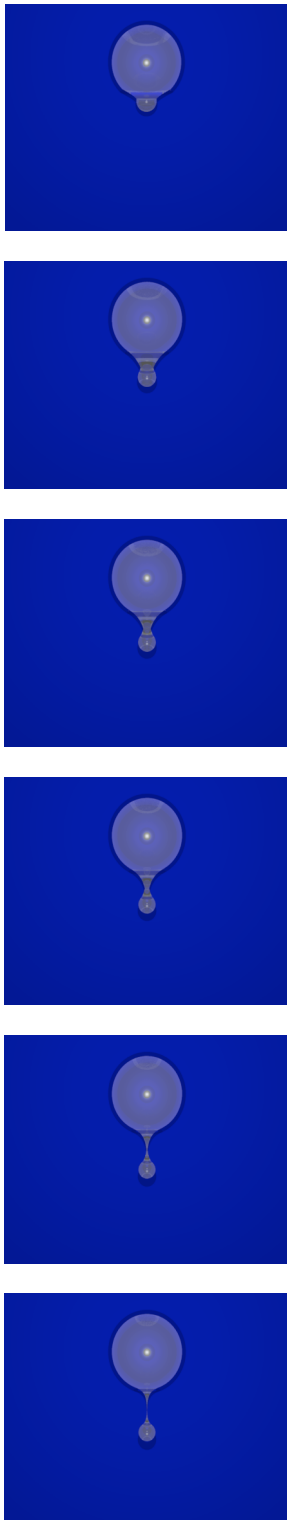


Figure 15: Different steps in the stricture effect

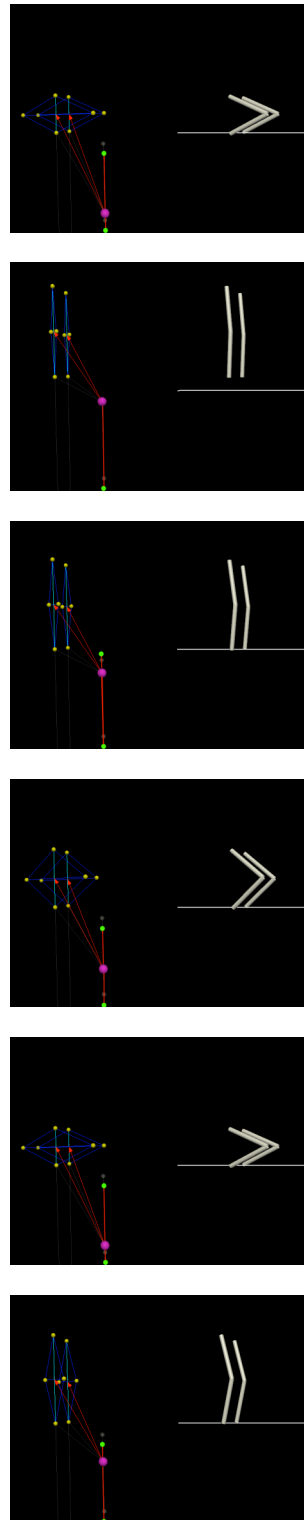


Figure 16: Different steps in the jump. Left parts: the mass-interaction network. Right parts: the movement coated with basic shapes

Vers une uniformisation des animations de marche

Jérôme Salut, IRIT
Université Paul Sabatier
Toulouse III
salut@irit.fr

Jean-Pierre Jessel, IRIT
Université Paul Sabatier
Toulouse III
jessel@irit.fr

Cédric Sanza, IRIT
Université Paul Sabatier
Toulouse III
sanza@irit.fr

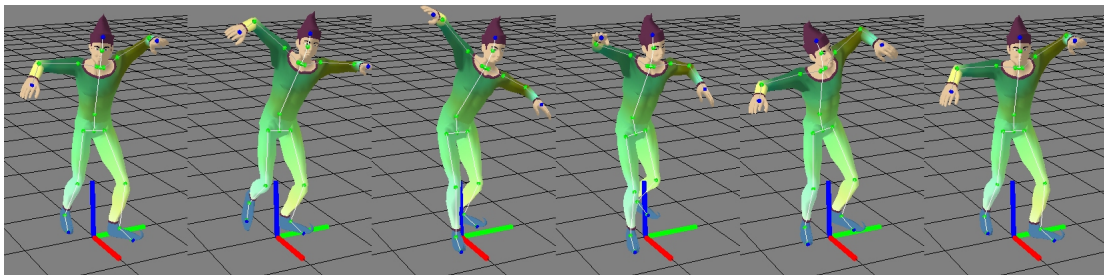


FIG. 1 – Une animation cyclique

Résumé

Les équipements de capture de mouvement permettent aujourd'hui de produire de nombreuses animations pour apporter plus de réalisme à des environnements de simulation, au travers, par exemple, d'humains virtuels. Le cas le plus classique dans ce genre d'animation est le cas de la marche humaine. Compte tenu de l'hétérogénéité des données de capture de mouvement (MoCap), nous proposons un modèle de transformations à appliquer pour faciliter l'utilisation des animations a posteriori. Dans un premier temps, nous déterminerons les paramètres que nous proposons de fixer avant d'effectuer une capture de mouvement. Puis, nous traiterons les différents critères à uniformiser sur une animation, et les résultats d'application de cette uniformisation dans nos travaux.

Mots clés : Capture de mouvement, Animation, Mouvement de marche, Uniformisation, Contrôle du mouvement

1 Introduction

Les animations que l'on obtient après une séance de capture de mouvement ne sont en général exploitables qu'après une phase de traitement, voir de nettoyage des données pour suppri-

mer la présence éventuelle de bruits lors de l'enregistrement. La conversion vers des formats évolués apporte une plus grande souplesse de manipulation mais ne corrige malheureusement pas la grande disparité des données. En effet, une animation représentant le même mouvement peut être enregistrée de différentes manières, selon le matériel utilisé et les conditions opératoires.

Dans le contexte de l'animation de groupes de personnages, nous proposons donc ici un ensemble de critères permettant une uniformisation des mouvements de marche, en vue d'effectuer par la suite le moins de post-traitement en cas de réutilisation d'une animation dans un environnement différent. Cela peut être aussi utile pour la création d'une bibliothèque de mouvement, où l'uniformité des captures de mouvement est indispensable pour toutes les méthodes d'édition et de génération utilisées en aval.

La plupart de ces critères peuvent s'appliquer à d'autres types d'animation que la marche, mais nous nous limiterons à ce cas d'étude, car c'est un parmi les plus intéressants que l'on rencontre dans le domaine de l'animation de personnage, utilisé dans la simulation d'environnements virtuels.

Les objectifs sont donc multiples. Nous souhaitons, en premier lieu, apporter une simplicité de traitement des animations en sortie directe après capture de mouvement, que ce soit pour une utilisation hors ligne ou en ligne, avec, par exemple, une

animation d'un avatar contrôlé par l'utilisateur. Dans un second temps, nous souhaitons maximiser la ré-utilisabilité des animations capturées, pour produire, par exemple, des bibliothèques de mouvements [3] [8] [14], mais aussi pour favoriser la réutilisation d'algorithme précédemment écrit pour une solution différente.

2 Travaux précédents

L'uniformisation des mouvements passe en premier lieu par l'uniformisation des représentations. La représentation peut-être directement inspirée du format de stockage de l'animation. Cela facilite directement l'inter-opérabilité entre plateforme. Par exemple, la norme H-ANIM¹ définit la représentation d'un personnage 3D au travers d'objets et de différents niveaux de granularité. L'implémentation la plus usitée de cette norme se fait au moyen de VRML.

R Kulpa, F Multon, et B Arnaldi [9] proposent une représentation normalisée, indépendante de la morphologie du personnage. Cette technique implique l'utilisation de cinématique inverse [15, 6, 12, 4] pour la reconstruction avant le rendu final, mais facilite grandement les manipulations temps-réel sur le squelette.

Charles Rose, Michael F. Cohen et Bobby Bodenheimer [14] exposent la construction d'une bibliothèque de mouvement par l'établissement de verbes et d'adverbes. L'interpolation entre les différentes animation de la bibliothèque se fait au travers d'une représentation par des B-Splines, même s'il est précisé que l'utilisation d'une décomposition de Fourier ou d'ondelettes est possible.

Dans les travaux de Shawkat Riyaz [1], l'accent est mis sur le développement d'une plate-forme orienté jeux, avec notamment la création de bibliothèque de mouvement par séparation des animations des différentes parties du squelette [5].

Nous ne proposerons donc pas une nouvelle représentation d'une animation mais un ensemble de critères à respecter, et les méthodes associées pour y parvenir, pour faciliter le traitement d'une animation de marche, notamment pour l'animation de groupe de personnages.

3 Critères d'uniformisation

3.1 Fréquence d'échantillonnage

La fréquence d'échantillonnage traduit le nombre de trames enregistrées par seconde. Cela conditionne notamment le volume des données capturées. De plus, une fréquence trop élevée

¹<http://www.h-anim.org>

alourdit les possibilités de traitement en temps réel. Il est préférable de fixer la fréquence d'échantillonnage selon le contexte d'utilisation de l'animation, car un ré-échantillonnage n'est pas souhaitable dans certains cas.

Pour des données provenant d'un matériel de capture magnétique, les fréquences d'échantillonnage sont en général de 33hz ou 66Hz, ce qui convient en général pour la plupart des utilisations. Si la vitesse de l'animation réelle est conservée, il est intéressant de garder la fréquence initiale.

Concernant les données provenant de système de capture optique, il n'est pas rare d'obtenir des fréquences de l'ordre de 200Hz ou plus, ce qui est un avantage certain dans des applications d'analyse de mouvements

Quel que soit le matériel utilisé, il faut donc choisir une fréquence d'échantillonnage appropriée pour l'utilisation (en ligne ou hors ligne) avant d'effectuer les captures de mouvements. Le sous-échantillonnage par décimation des données peut cependant être envisagé sans perte de précision.

3.2 Échelle de référence

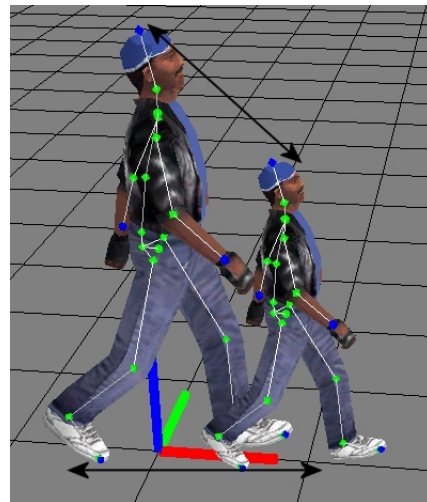


FIG. 2 – Changement d'échelle

Pour conserver les différences entre les captures, il est utile de fixer une échelle de référence. Par exemple, les captures de mouvement d'un enfant et d'un adulte, en phase de marche, ne donnent pas le même résultat visuel une fois appliqué sur un squelette de même taille. Il est donc important de choisir un individu de référence, et de réaliser les captures de mouvements pour respecter ces différences de proportions. Si cela n'est pas réalisable, il faut au moins annoter la morphologie de l'individu ayant servi d'acteur pour la capture. Le *retargeting* permet de gommer ce problème, mais ajoute une étape de traitement

supplémentaire.

Le position d'une animation par rapport à l'échelle de référence peut se faire par un simple facteur numérique indiquant la réduction ou l'augmentation du mouvement. L'adaptation se fait alors simplement par re-calcul du squelette et de ses données d'animations. Par exemple, pour des données provenant d'un fichier BVH², l'animation est représentée par un couple de valeurs de rotations et de translations. Le simple changement d'échelle des données de translation et du squelette permet a posteriori de recadrer l'animation sur une échelle de référence.

Dans le cas d'une animation associée à un personnage habillé par skinning³, il faudra penser à appliquer les changements sur l'enveloppe du personnage, comme sur la figure 2.

3.3 Sens et direction

La plupart des mouvements de marche que nous enregistrons sont des marches rectilignes. On adapte ensuite la position et l'orientation des pieds en temps réel selon la direction que prend le personnage. Partant de ce constat, il est intéressant de faire évoluer l'acteur sur un des axes du repère choisi pour la capture. Physiquement, cela permet de lui faire suivre une ligne droite, et facilite par la suite les modifications du sens global de la marche. Cependant, si ces conditions opératoires ne sont pas réunies pour des raisons techniques ou physiques, il est nécessaire de post-traiter les animations obtenues.

Dans un premier temps, nous nous attacherons à déterminer la correction de direction à appliquer, et dans un second temps, la façon d'appliquer cette correction.

3.3.1 Analyse de la correction

Le principe général est de calculer la droite de régression du nuage de points constitué par les positions de la racine du squelette au sol au cours du temps. On utilise la méthode des moindres carrés, exposées en annexe A, pour déterminer un angle α représentant le sens de la trajectoire globale de l'animation.

3.3.2 Application de la correction

L'application de la correction consister à intégrer l'angle α , précédemment déterminé, dans les données de rotations de la racine du squelette du personnage. Si ces dernières sont exprimées par des angles d'Euler, la transformation peut ne pas être directe.

Dans le même exemple de données provenant d'un

fichier BVH, les valeurs des rotations sur les trois axes X , Y et Z sont inter-dépendantes, et sont exprimées sous forme d'une rotation $R = YXZ(x)$. Nous voulons ajouter le paramètre α sur la rotation de l'axe Y du repère avant application des rotations. Ici, les trois rotations sont empilées et donc dépendantes des transformations précédentes. Nous voulons donc obtenir :

$$\begin{aligned} - R' &= YXZA(x) = Y_2X_2Z_2(x) \\ - A &= \text{rot}(\alpha)_Y \end{aligned}$$

La nouvelle rotation R' doit s'exprimer sous forme de seulement trois rotations. On obtient alors :

$$\begin{aligned} X_2 &= \sin^{-1}(\sin(A) \cos(X) \sin(Z) + \sin(X) \cos(A)) \\ Y_2 &= \tan^{-1}\left(\frac{\sin(A) * E + \cos(A) \sin(Y) \cos(X)}{-\sin(A) * F + \cos(A) \cos(Y) \cos(X)}\right) \\ Z_2 &= \tan^{-1}\left(\frac{\cos(A) \cos(X) \sin(Z) - \sin(A) \sin(X)}{\cos(X) \cos(Z)}\right) \end{aligned}$$

$$\text{avec } E = \cos(Y) \cos(Z) - \sin(Y) \sin(X) \sin(Z)$$

$$\text{et } F = \sin(Y) \cos(Z) + \cos(Y) \sin(X) \sin(Z)$$

Après application de la correction, nous obtenons un changement global de direction du personnage, illustré en figure 3.

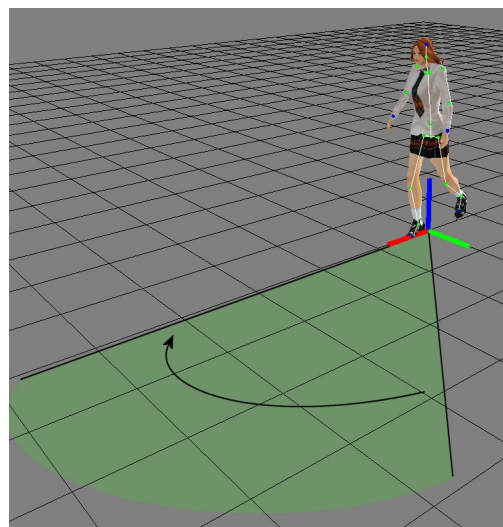


FIG. 3 – Correction de la direction d'une animation

²<http://www.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/BVH.html>

³Le skinning permet d'adapter une modèle 3D sur un squelette animé [7] [13] [10].

3.4 Centrage des données

3.4.1 Principe

Ce critère est optionnel, et son adoption doit se faire selon l'application voulue. Centrer les données revient à fixer globalement la position de la racine sur les deux axes formant le plan d'évolution de l'acteur. Ceci permet d'obtenir une animation de marche sans déplacement absolu dans le repère de l'animation. On peut alors coupler facilement cette animation avec un module de simulation comportementale, qui est, en général, le seul maître des décisions du personnage virtuel (sa vitesse, sa direction...). Cela permet alors un simple rendu graphique de l'animation à la position du personnage dans l'espace. Cependant, ceci n'exempte pas d'adaptations en temps réel, sur la position des pieds, pour s'adapter à la variation de la vitesse par exemple.

Si l'on se contente de fixer la position de la racine, le résultat n'est pas satisfaisant, car cela gomme un aspect de la marche. En effet, certaines personnes peuvent avoir une démarche féminine, où la position de la racine a justement tendance à effectuer des oscillations autour d'un des axes du repère. On parle donc de fixer *globalement* la position de la racine.

3.4.2 Réalisation

Nous voulons donc fixer globalement la position de la racine autour de sa position initiale. Le terme *globalement* s'exprime par la prise en compte des déplacements latéraux par rapport à la droite de régression du nuage de points des positions de la racine du personnage.

La méthode pour déterminer la droite de régression est la même que dans la section 3.3.1. Nous obtenons une droite de la forme $y = b_1x + b_0$, soit $b_1x - y + b_0 = 0$.

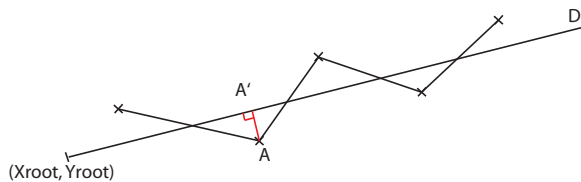


FIG. 4 – Déplacements latéraux

Soit $D : ax + by + c = 0$ la droite de régression. Soit A la position de la racine au cours du temps avant correction, et (x_A, y_A) les coordonnées au sol de A . Soit A' le projeté orthogonal de A sur D . Les données du problème sont exposées sur la figure 4. On a alors :

$$x_{A'} = \frac{b^2x_A - aby_A - ac}{a^2 + b^2}$$

$$y_{A'} = \frac{-abx_A + a^2y_A - bc}{a^2 + b^2}$$

Par identification de la droite D et de la droite de régression précédemment calculée, on obtient les nouvelles coordonnées de la racine à chaque instant :

$$x_{new} = X_{rootInitial} + x_A - x_{A'}$$

$$y_{new} = Y_{rootInitial} + y_A - y_{A'}$$

où $(x_A - x_{A'}, y_A - y_{A'})$ représente le déplacement latéral de la racine par rapport à D , et $(X_{rootInitial}, Y_{rootInitial})$ la position initiale de la racine.

3.5 Cycle d'animation

Une animation cyclique comporte un motif de mouvement répété. Dans le cas de l'animation de marche, il s'agit en général d'un seul cycle de marche, c'est à dire la pose d'un pied, le déplacement de l'autre, jusqu'au retour du premier pied en position initiale. Cette position initiale peut-être quelconque, mais, en général, nous la prenons à un instant caractéristique, comme, par exemple, la pose d'un des deux pieds au sol. Cela permet donc de jouer en boucle la même animation (fig 1). Il est rare d'obtenir des captures de mouvement cyclique sans traitement.

La première phase du traitement est donc la détermination du cycle à choisir dans l'animation complète. Une capture de mouvement de marche sur un acteur humain ne se rapproche d'un cycle qu'après une phase de mise en marche du mouvement : le sujet commence à marcher et atteint son allure régulière. Il faut donc écarter du traitement cette première phase de l'animation. Ensuite, la détection d'un cycle complet se fait par l'analyse du mouvement des pieds du personnage. Selon la disposition des capteurs qui ont servi à la capture de mouvement, on étudiera l'ensemble représentant le talon et le pied de l'acteur. Nous utilisons alors une méthode de seuillage combinant l'analyse des positions et des accélérations de l'ensemble mentionné ci-dessus, pour déterminer les instants de l'animation correspondant à la pose de chaque pied au sol. Vous trouverez une étude plus complète des phases de la marche dans [2] et [11]. Une fois un cycle de marche extrait, il suffit d'effectuer une interpolation entre les dernières trames de l'animation et les premières pour obtenir une animation cyclique. Le mélange s'effectue linéairement en fonction de la longueur totale du cycle. On veillera cependant à produire des

données cohérentes lors du mélange des données de rotation de l'animation, dans le cas d'utilisation d'angles d'Euler.

4 Résultats, discussion

Les transformations décrites dans cet article sont celles utilisées dans nos travaux en cours sur le clonage de mouvement (figure 5), et sont intégrées dans une chaîne de traitement automatique, ce qui permet d'obtenir des animations directement utilisables à partir de données brutes de captures de mouvements, pour l'animation de foules. L'idée principale est de limiter le plus possible l'intervention humaine sur des tâches qui pourraient être automatisées. Par exemple, la détection des phases d'une animation en vue de la rendre cyclique est un travail long et coûteux lorsqu'il est réalisé manuellement. Ici, la seule intervention humaine se situe dans l'identification des zones du squelette servant au support de la marche au personnage. Cependant, cette tâche pourrait être automatisée facilement, en détectant les zones du squelette qui servent au support de la marche (les terminaisons du squelette de plus basse altitude par exemple). Nous n'avons pas implémenté cette possibilité pour garder un contrôle manuel sur la sélection du nombre des articulations à traiter lors d'un rendu en-ligne.

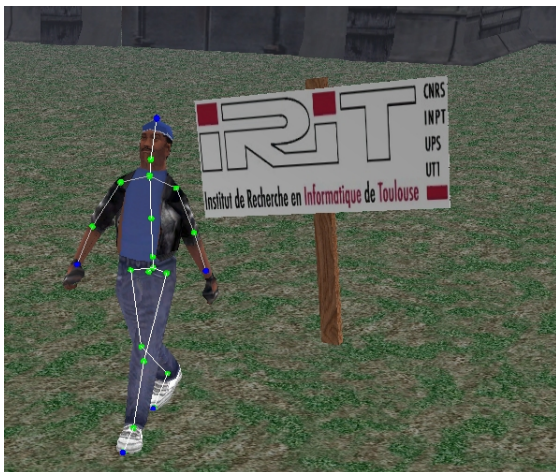


FIG. 5 – Utilisation d'une animation de marche produite, avec adaptation temps-réel dans un environnement virtuel

Nous utilisons actuellement des données provenant d'un système de capture magnétique, mais aussi de capture optique. L'utilisation de données de capture optique entraîne cependant des restrictions. La notion de hiérarchie de squelette, que l'on trouve dans la plupart des formats de capture de mouvement, est, dans ce cas, perdue. Chaque capteur est indépendant des autres, à moins d'une in-

tervention pendant la capture ou a posteriori, pour garantir l'obtention d'un squelette hiérarchique de capteurs. De plus, chaque capteur, ne dispose que d'informations de translations seulement. Cependant, cela peut-être compensé par l'accumulation de trois capteurs optiques, minimum, pour chaque capteur, habituellement utilisé et comportant une information de rotation.

Dans cet article, nous nous sommes restreints au traitement d'animations de marche simples. Cependant, ces traitements ont été testés sur des animations plus complexes, comme, par exemple, sur des démarches chaloupées et des mouvements de demi-tours, où le calcul d'une droite de régression n'a pas forcément de sens. Dans le cas d'une animation fortement chaloupée, le résultat du traitement est cohérent tant que la trajectoire est globalement rectiligne. Pour une animation de marche où la trajectoire s'assimile à un demi-cercle, le calcul d'une droite de régression linéaire ne représente pas la trajectoire du personnage, et le calcul des écarts relatifs du personnage par rapport à cette droite n'a pas de sens. Dans ce cas, la détection du sens et de la direction globale de la trajectoire n'est pas non plus probante. Cependant, on pourrait envisager de calculer une courbe de régression de degré supérieur à 1 (degré 2 ou 3), mais cela revient à extraire, au final, une animation de marche quasiment rectiligne d'une animation à trajectoire courbe. Nous préférons dans ce cas utiliser des animations de marche qui rentrent dans le cadre de notre étude, et effectuer, par la suite, des adaptations temps-réel sur le placement des pieds, qui, de toute façon, sont nécessaires. Seul le traitement d'animation de course, où le sujet n'a pas forcément un pied au sol en permanence, pose problème dans la détection automatique des phases de la marche.

L'application des traitements en chaîne est très rapide et réalisée de façon automatique l'un après l'autre. Le temps de réalisation d'une animation, utilisable dans une application en ligne, à partir de données de capture filtrées reste inférieur à 0.5 secondes. Ceci reste négligeable par rapport au coût d'une réalisation manuelle des traitements.

5 Conclusion, travaux futurs

Dans cet article nous proposons un modèle d'uniformisation d'animation de marche. Les animations produites doivent cependant être adaptées en temps réel pour permettre de refléter les caractéristiques du personnage (vitesse, accélération...).

Nous envisageons d'étendre le domaine d'application des traitements mis en oeuvre (type des animations traitées), et d'améliorer leur robustesse, tout en gardant un haut niveau d'automatisation.

Nous espérons aussi automatiser les phases de traitement initiale et manuelle, dans le cas de données provenant de captures optiques.

A Annexe : Calcul d'une droite de régression

Posons $Z = b_0 + b_1 * X$ la droite de régression. Soit X_i et Y_i les positions au sol de la racine au cours du temps, et n le nombre de points du nuage. Le problème est représenté sur la figure 6. On a alors :

$$b_0 = \frac{\sum_{i=1}^n y_i \sum_{i=1}^n x_i^2 - \sum_{i=1}^n x_i \sum_{i=1}^n x_i y_i}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2}$$

$$b_1 = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2}$$

L'angle de déviation α par rapport à l'axe X de référence est alors

$$\alpha = \tan^{-1}(b_1)$$

Attention toutefois, la fonction \tan^{-1} n'étant pas bijective, il faut effectuer une correction de α en tenant compte du début et de la fin de la trajectoire du personnage, pour déterminer le sens de déplacement en plus de la direction représentée par α .

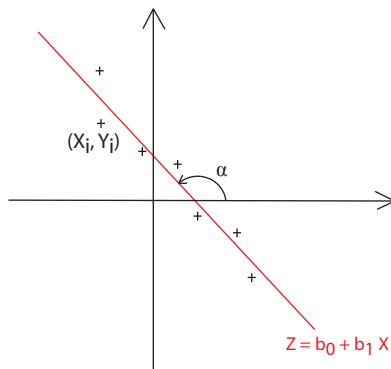


FIG. 6 – Droite de régression linéaire d'un nuage de points

Références

[1] Shawkat Riyaz Assrafally. Using motion capture data in computer games. Master's thesis, Department of Computer Science, University of Sheffield, Regent Court.
[2] Brice Fernandez. Analyse de la marche humaine par ondelettes. Master's thesis, INRIA Rhône-Alpes, 2005.

[3] M. Gleicher, H. Shin, L. Kovar, and A. Jepsen. Snap-together motion : Assembling runtime animation, 2003.
[4] Keith Grochow, Steven L. Martin, Aaron Hertzmann, and Zoran Popovic. Style-based inverse kinematics. *ACM Trans. Graph.*, 23(3) :522–531, 2004.
[5] Rachel Heckc, Lucas Kovarc, and Michael Gleicher. Splicing upper-body actions with locomotion. In *Computer Graphics Forum, Volume 25, Issue 3*, pages 459–466, 2006. Computer Graphics Forum.
[6] Charles F. Rose III, Peter-Pike J. Sloan, and Michael F. Cohen. Artist-directed inverse-kinematics using radial basis function interpolation. *Comput. Graph. Forum*, 20(3), 2001.
[7] Doug L. James and Christopher D. Twigg. Skinning mesh animations. In *SIGGRAPH '05 : ACM SIGGRAPH 2005 Papers*, pages 399–407, New York, NY, USA, 2005. ACM Press.
[8] Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. In *SIGGRAPH '02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, volume 21, pages 473–482, New York, NY, USA, July 2002. ACM Press.
[9] R Kulpa, F Multon, and B Arnaldi. Morphology-independent representation of motions for interactive human-like animation. *Computer Graphics Forum*, 24(3) :343–352, 2005.
[10] J. P. Lewis, Matt Cordner, and Nickson Fong. Pose space deformation : a unified approach to shape interpolation and skeleton-driven deformation. In *SIGGRAPH '00 : Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 165–172, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.
[11] Eric Menou. *Adaptation de mouvements par optimisation contrainte pour l'animation des personnages de synthèse*. PhD thesis, Université Paul Sabatier, Toulouse 3, 2005.
[12] M. Meredith and S. Maddock. Individualised character motion using weighted real-time inverse kinematics. In *Proc. GAME-ON*, 2004.
[13] A. Mohr, L. Tokheim, and M. Gleicher. Direct manipulation of interactive character skins, 2003.
[14] Charles Rose, Michael F. Cohen, and Bobby Bodenheimer. Verbs and adverbs : Multidimensional motion interpolation. *IEEE Comput. Graph. Appl.*, 18(5) :32–40, 1998.
[15] Chris Welman. Inverse kinematics and geometric constraints for articulated figure manipulation. Master's thesis, March 1993.

Pedestrian Reactive Navigation for Crowd Simulation: a Predictive Approach

Sébastien Paris

Julien Pettré

Stéphane Donikian

IRISA, Campus de Beaulieu, F-35042 Rennes, FRANCE
{sebastien-guillaume.paris, julien.petre, donikian}@irisa.fr

Abstract

This paper addresses the problem of virtual pedestrian autonomous navigation for crowd simulation. It describes a method for solving interactions between pedestrians and avoiding inter-collisions. Our approach is agent-based and predictive: each agent perceives surrounding agents and extrapolates their trajectory in order to react to potential collisions. We aim at obtaining realistic results, thus the proposed model is calibrated from experimental motion capture data. Our method is shown to be valid and solves major drawbacks compared to previous approaches such as oscillations due to a lack of anticipation. We first describe the mathematical representation used in our model, we then detail its implementation, and finally, its calibration and validation from real data.

1. Introduction

This paper addresses the problem of virtual pedestrian autonomous navigation for crowd simulation. One crucial aspect of this problem is to solve interactions between pedestrians during locomotion, which means avoiding inter-collisions. Simulating interactions between pedestrians is a difficult problem because its complexity grows rapidly with respect to population density. Also, obtaining realistic results is challenging: humans are used to observe navigating pedestrians in the real life and immediately detect artifacts in simulations. We present a reactive navigation technique for application in the domains of architecture, security, space ergonomics, and also the entertainment industry. We expect natural crowd motion emerging from a realistic microscopic pedestrian simulation.

Our solution for solving interactions between pedestrians is predictive and agent-based. Inputs are the definition of an environment, the current state and the destination of each pedestrian - destination is a desired direction derived from a navigation plan. The method first checks on future interactions between pedestrians: the evolution of pedestrians' position is predicted from an extrapolation of their current state. When needed, a long term avoidance motion is computed by taking into account these predictions. Our resulting microscopic pedestrian simulation model is calibrated and

validated using motion capture data. Data are acquired according to two successive protocols. First, we measure interactions between two participants and use the resulting data to calibrate our model. In a second stage, we push the number of participants to the limits of our motion capture system abilities and let them navigate among obstacles, allowing us to compare the measured data with our simulation results.

Our first contribution is to solve major drawbacks in previous microscopic approaches, such as oscillations and jams. We believe these drawbacks were due, firstly, to the lack of anticipation, and secondly, to the simplicity of the reaction computation technique from the observed situation. Our second contribution is to propose a motion capture-based calibration of the model and validation of our results. Validation is generally done using hand-processed video sequences. Motion capture data are more accurate than those derived from video sequences: this allows us to decompose precisely in time and space how humans react to potential collision with others. Particularly, we could extract a criterion to detect the need for a reaction and compute adequate corrections to the trajectory.

Section 2 states our contributions with comparison to previous approaches. Section 3 describes our method to solve interactions between pedestrians from a technical point of view: firstly, how it is integrated into a crowd simulator, sec-

only, how pedestrians perceive their surrounding environment, and thirdly, how a reactive motion is computed. Section 4 describes the protocols and the results of the experiments used to calibrate the model parameters and to validate the approach. We then present some simulation results in usual architectural configurations. Finally, we conclude and provide perspectives to our work.

2. Related Work

One of the most important skills of a human being is her ability to navigate inside her environment. Even if this navigation task is one of the most basic behavior in real life, it is not yet solved correctly in a virtual world. First to allow people to navigate, they should be able to perceive their environment but not only in a geometric way. Studies in psychology and urbanism have shown that visibility and topology are also important in the navigation task. A structured and informed environment has to be used for path planning and reactive navigation of virtual humans in real time. The simplest task, for a pedestrian walking in a street, consists in minimizing possible interactions, which mean avoiding static and dynamic obstacles. Goffman [Gof71] describes techniques used by pedestrians to avoid bumping into each other. The social link between strangers is characterized by silence and indifference and to perform that, different behaviors are used. The first technique called externalization concerns the way that people are constantly making others aware of their intentions in order to minimize the interaction. Pedestrians selectively gather externalized information from other people by a second technique called scanning. The third technique, called the minimization of adjustment, expresses that people adjust their trajectory several meters before the conflict to make it perceptible early by others with the objective to reduce interaction and avoid coordination. Goffman introduces the notion of an oval security region whose front distance corresponds to an anticipation area depending on the pedestrian speed, while the width is the accepted gap to pass beside a person or an obstacle or to follow a wall. He also defines the law of minimal change meaning that a pedestrian will try in its journey to reduce the amount and the amplitude of turns. These studies illustrate the importance of prediction and anticipation in the navigation task.

It is known that in crowd motions, pedestrian flows walking in opposite direction generate their splitting to create dynamically some bands of pedestrians walking in the same direction. When the density of pedestrians becomes very high, it is possible to approximate the overall behavior of the crowd by using the laws of fluid evolution [TCP06]. In panic situations, pedestrians wish to move more quickly than usual and, forgetting all social rules, accept to be in a physical contact with their neighbors. Due to this physical interaction situation, they are developing a mimetic behavior consisting in reproducing the behavior of preceding characters in the flow.

For Yamori [Yam98], this is obligatory followed by the notion of regulation, learned as a normative element by people living together inside the same macro structure or institution. However, as emphasized by Musse et al. [MT01], some small groups can be the motor and modify the behavior of biggest units such as a crowd, playing the role of the core group. Boles [Bol81] has observed the existence of a band structure inside a crowd of pedestrians moving on a sidewalk and explain this by its optimal configuration to regulate opposite flows. One of the most crucial problems to be solved, as pointed by Yamori, concerns the relation between microscopic and macroscopic structures and behaviors inside the crowd. The goal is to explain how an individual entity is constrained by the institution and on the other hand how the community impacts on the individual behavior along time. Yamori focuses his research in the formation of macroscopic band structures and postulate that such kind of macroscopic structure requires a critical density of population to emerge from the set of individual behaviors.

Macroscopic simulation has been historically the first approach to be studied to simulate the pedestrian displacement, due to its low calculation cost. In this approach the pedestrian is not treated on its own but as a component of a more macroscopic element [Hen71, PM78]. These macroscopic models are often used for animation purpose, like by Sung et al. [SGC04], to provide a globally convincing crowd motion. Another approach called microscopic simulation consists in handling the individual navigation of all moving entities. In that case, a system allowing dynamic collision avoidance is necessary to achieve consistency and realism. Several approaches can be distinguished such as particle and flocking systems. Particle systems are based on physical laws describing attractive and repulsive forces that can be associated to obstacles and moving entities. Forces applied to an entity are summed to calculate its new motion direction and speed [HFV00, BMdOB03, LKF05]. This model assimilates the displacement of an entity in the case of a high density to the motion of a particle inside a restricted area. I. Peschl [Pes71] justifies the use of this model in the case of an emergency situation with a high density of population. Particle based models allow the generation of a macroscopically plausible behavior in case of a high density, but they do not take into account anticipation, perception, or social rules. Moreover, close inspection of individual trajectories show some oscillations and unrealistic behaviors such as backward motion of the last people repulsed by the preceding one in a queue and many change of orientation along the path due to the interaction with other moving entities and static obstacles. Another drawback of this approach is its requirement of a small time-step for convergence purpose. Flocks are rule-based systems defining the behavior of an entity according to the behavior of the nearest entities [Rey00, BLA02]. It is well adapted for the collective motion of a group of animals following a leader but less for the variety of behaviors that can be observed in a sparsely populated crowd of humans.

Loscos et al. [LMM03] use a fine regular grid to handle reactive navigation and to store information about pedestrian movements enabling the emergence of flows of pedestrians. In the same way, Shao et al. [ST05] use a quadtree map for the path planning and a fine regular grid for obstacle avoidance.

Lakoba et al. [LKF05] argue that two points are necessary to improve the existing models: add decision-making capabilities and compare simulation results against measured data on pedestrian dynamics. Instead of classical models of crowd simulation based on fluid dynamics or particle systems which are only valid in very dense crowds, S. Goldenstein et al. [GKM*01] have proposed a multi-layer approach to model the behavior of crowd participants. We are also working on a multilevel model of each human allowing us to simultaneously take into account attraction/repulsion mechanisms such as in particle systems, dynamic computation of the neighborhood for sparse crowds, the management of social rules, path planning and activity planning. In this paper, we are focusing on the reactive navigation model and on the use of experimental data to validate the approach and calibrate the model.

3. Prediction and Resolution of Interactions

3.1. Principle

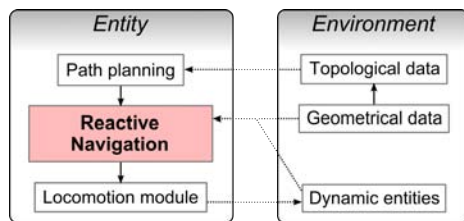


Figure 1: The simulation architecture.

The problem of reactive pedestrian navigation is part of the crowd simulation problem, and our method is included in a global architecture as shown in Figure 1. More details on the environment management and on dedicated path planning techniques we developed can be found in the literature [PDB05,PDB06]. The reactive navigation role is to steer entities in a realistic manner with respect to two possibly conflicting inputs: the goal of the considered pedestrian resulting from the path planning stage, and the current state of the environment, especially the presence of other pedestrians. The output we search for are updated speed and orientation allowing the pedestrian to avoid any static or moving obstacle while satisfying constraints of realism.

Our approach to this problem is a predictive one. For each entity, at desired rates, we search for a solution-move satisfying constraints and guaranteed to remain valid for a desired time window (at least the period at which reactive navigation is invoked). The key-idea is to model the environment

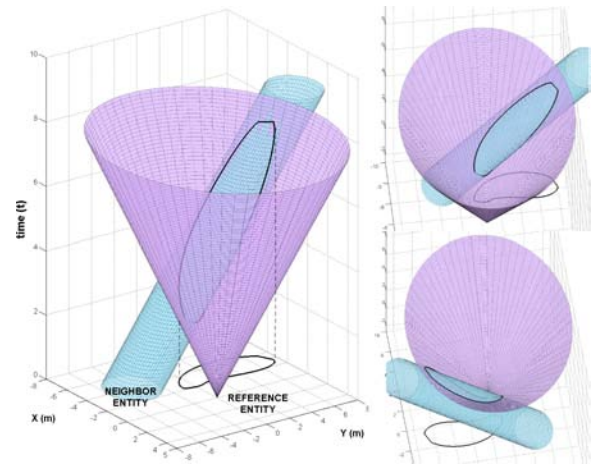


Figure 2: Modeling the interaction between a reference entity and a neighbor entity in the (x,y,t) -space. The predicted trajectory for the neighbor entity is the blue cylinder, whereas the reachable space for the reference entity is the violet cone. Their intersection delimited in black is a future collision area (also projected in the plane).

as shown in Figure 2 in the (x,y,t) -space, with (x,y) the horizontal plane and t the simulation time. We explore the reachable space of the reference entity in any direction and for a range of speed values, and search for possible collisions with neighboring entities. Figure 2 illustrates such an exploration for a given reference entity speed value. As any move direction is envisioned, the reachable space is then represented in the (x,y,t) -space as a cone whose opening angle depends on the considered speed value. Neighboring entities are then taken into account: they are represented as circles – whose radii are the sum of both the reference and the considered entity radii – moving along a predictive trajectory, computed from the current position, speed and orientation of the neighboring entity. Thus, the neighboring entity is modeled in the (x,y,t) -space as an elliptic cylinder. Consequently, the intersection of the cone and the cylinder delimits a collision area that the reference entity shall avoid. The difficulty of the problem is brought to mind looking at Figure 3, where different shapes of the collision area are displayed according to various solution speed values. Additionally, the figure does not represent the possible presence of several neighboring entities and of static obstacles which obviously increase the problem complexity drastically. As a result, we choose to base our solution on a discrete-time expression of the same modeling in order to avoid the problem complexity.

Three main steps, detailed in the next sections, allow us to compute the best speed and orientation for the reference entity:

1. Neighboring dynamic entities are first taken into account. From this, we deduce some sets of speed and orientation

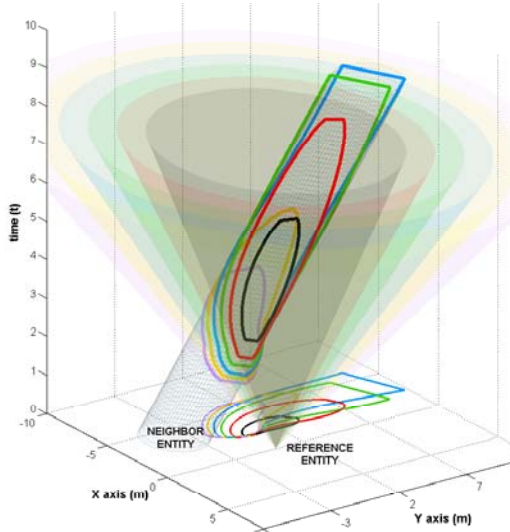


Figure 3: We explore the reachable space for the reference entity for a given range of speed. The opening angle of the cone changes accordingly in the (x,y,t) -space, as well as the intersection with the predicted trajectory of the neighbor entity (each color corresponds to a different cone opening angle, and thus to a different reference speed value).

ranges that allow collision free motion for a future time window.

2. In the same manner, static obstacles are considered. We deduce new valid sets of speed and orientation ranges.
3. Previous valid solution ranges are merged, scored and compared. The best one is returned as the solution.

3.2. Dynamic entities

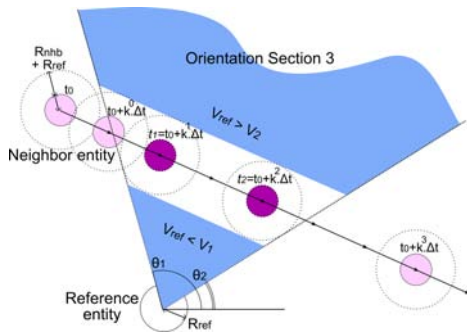


Figure 4: Example of reachable space sectioning for a given time-interval. The resulting orientation section is characterized by several parameters: $t_1, t_2, \theta_1, \theta_2, V_1, V_2$.

The objective of this first step is to compute a set of valid speed and orientation ranges for a given reference entity

E_{ref} , with respect to the presence of neighbor moving entities. We describe our method for a single neighbor entity E_{nhb} , whilst the case of several ones is detailed Section 3.4.

Time discretization. We consider the environment state at $t = t_0$ and propose to search the E_{ref} reachable area for potential collision as explained previously. For that, we consider successively adjacent time-intervals having different durations: $[0, k^0 \Delta t]$, $[k^0 \Delta t, k^1 \Delta t]$, $[k^1 \Delta t, k^2 \Delta t]$, $[k^2 \Delta t, k^3 \Delta t]$, etc. The $\Delta t > 0$ parameter defines the precision of the discretization, smaller being the best, and should correspond to the time needed by the entity to make one move. The $k > 1$ parameter is used to make the discretization non-uniform over the anticipated time, greater being the best, allowing the anticipation to be more precise in the near future than in the distant one. We use $\Delta t = 1$ and $k = 2$ in our model.

Reachable space sectioning. For each time-interval, we predict the E_{nhb} trajectory as a linear one and deduce the E_{ref} orientation range (orientation section) potentially leading to a collision with E_{nhb} as illustrated in Figure 4. We finally get as many sections as the number of time-intervals, each representing an orientation range for E_{ref} . The time-interval $[t_1 t_2]$ used to compute each section is stored.

Critical speeds computation. For each orientation section, we compute the critical speeds V_1 and V_2 defined as follows: V_1 is the maximal speed allowed to avoid a collision by passing behind E_{nhb} ; V_2 is the required minimal speed to avoid a collision by passing before (in front of) E_{nhb} . V_1 and V_2 have analytical expression, as solution to the following equation:

$$V_1 = \min_{t=t_1}^{t_2} \left(\left(\left\| \overrightarrow{P_R P_n(t)} \right\| - R \right) / t \right)$$

$$V_2 = \max_{t=t_1}^{t_2} \left(\left(\left\| \overrightarrow{P_R P_n(t)} \right\| + R \right) / t \right)$$

with $P_n(t) = P_n + \vec{v}_n t$ and where \vec{v}_n is the E_{nhb} speed vector, P_R and P_n are respectively the positions of E_{ref} and E_{nhb} at $t = t_0$, and finally R is the sum of the bounding circles radii of the considered entities, eventually increased by a security factor to avoid strict contact cases.

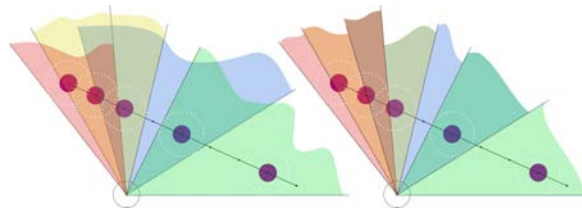


Figure 5: The sectioning results in overlapping orientation ranges (left image). The merge of overlapping sections is easily done by subdividing orientation ranges in order to get adjacent sections (right image). The new characteristics of each subdivision are directly deduced.

Merging several sections. By iterating previous computation for each time-interval, we get a set of orientation sections with their own characteristics (V_1 , V_2 , time-interval $[t_1 t_2]$). A problem is to merge overlapping sections. Sections are finally subdivided in order to get a set of adjacent sections. Each new subsection characteristics are computed as follow: $V_{1new} = \min(V_{1i}, V_{1j})$, $V_{2new} = \max(V_{2i}, V_{2j})$, $t_{1new} = \min(t_{1i}, t_{1j})$ and $t_{2new} = \min(t_{2i}, t_{2j})$ with i and j indexing the two merged orientation sections. Note that if three or more sections overlap, this process can be reiterated successively considering pairs of sections until all are merged.

3.3. Static entities

The second step of the reactive navigation module consists of considering static obstacles. We handle static obstacles in approximately the same way than dynamic entities, but the problem is obviously simpler. In our environment database, obstacles borders are modeled as line segments. Let us consider the case of a single line segment in the vicinity of E_{ref} . Our first objective is to subdivide S_{obst} as shown in Figure 6. For that, we first compute P_0 the nearest point from E_{ref} belonging to S_{obst} . We define points P_1 and P'_1 if existing so that the length $P_0P_1 = P_0P'_1 = v_{ref}\Delta t$ where v_{ref} is E_{ref} speed, and Δt defined in the previous Section. Then, we define the point P_2 so that $P_0P_2 = k^1 v_{ref}\Delta t$, P_3 so that $P_0P_3 = k^2 v_{ref}\Delta t$, and finally P_4 so that $P_0P_4 = k^3 v_{ref}\Delta t$. This set of points is arbitrary, however, it allows us to evaluate the constrained speeds toward the obstacle with a more accurate precision near to E_{ref} .

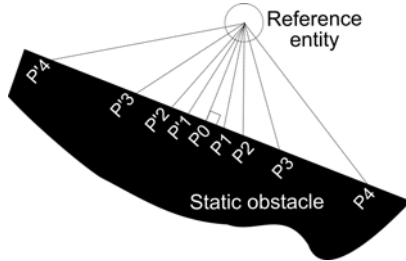


Figure 6: Sections computation for static obstacles

A set of adjacent orientation sections are computed as shown in Figure 6. We then compute the characteristics of each section in a same way than done previously with dynamic entities. However, V_1 is computed differently:

$$V_{1i} = \|\overrightarrow{P_i E_{ref}}\| / t_{2i}$$

where i indexes the considered orientation section. V_1 is thus the maximal speed at which E_{ref} can walk within the considered time-interval (depending on the considered section) without colliding the obstacle, analogically to the previous dynamic entity case. V_2 is here meaningless, and we set $V_2 = +\infty$.

3.4. Solving interactions

The third and final step of our reactive module consists of extracting a solution-move for the reference entity. The extraction is done in three successive steps. First, each orientation section previously computed is weighted using a cost function depending on sections characteristics. Second, superposed sections (related to different neighboring entities or obstacles) are merged (see Section 3.2) while accumulating costs. Third, the best section is used to compute the new speed and orientation.

Orientation section cost. The reference entity has to choose the best next speed and orientation according to the environment state and its goal. The function cost reflects the best choice among several criteria:

- Valid speed ranges (according to V_1 and V_2 of each section) must be close to the entity desired speed V_{des} and in its range of achievable speeds $[0, V_{max}]$.
- Orientation section limits $[\theta_1; \theta_2]$ must be as close as possible to the desired orientation θ_{des}
- Required accelerations to reach the new speed and orientations must be as limited as possible (limiting strictly them is not desirable because real humans are capable of important accelerations).
- The closer the section time-interval is in the future, the more confident we are in its cost.

The cost associated to speed variations and distance to desired speed is computed as follows:

$$C_{decel} = \begin{cases} 0 & \text{if } V_{des} \leq V_1 \\ 1 - \frac{V_1}{V_{des}} & \end{cases} \quad C_{accel} = \begin{cases} 0 & \text{if } V_{des} \geq V_2 \\ \frac{V_2 - V_{des}}{V_{max} - V_{des}} & \end{cases}$$

$$C_{speed} = \alpha \cdot \min(C_{accel}, C_{decel})$$

where $\alpha \in]0; 1[$ allows us to set a trade-off between speed changes and orientation changes. The cost associated to orientation changes (deviation) is computed as follows:

$$C_{dev} = (1 - \alpha) \frac{1 - \cos(\theta)}{2}$$

where θ is the minimum difference between the desired orientation and the orientation section limit angles. Note that $0 \leq C_{speed} + C_{dev} \leq 1$. According to t_1 the lowest bound to the time-interval of the concerned section, we finally compute a prediction confidence cost:

$$C_{pred} = 1 - \frac{t_1}{T + \beta} \quad \text{with } 0 \leq C_{pred} \leq 1$$

where T is the maximal considered time for the prediction, and $\beta \in [0; +\infty]$ is a user-fixed parameter allowing us to get more or less confidence in predictions (this will change the pedestrian adaptation-time before a potential collision). The total cost of a given section is then:

$$C_{total} = (C_{speed} + C_{dev}) \cdot C_{pred} \cdot C_{add}$$

where $0 \leq C_{add} \leq +\infty$ is an additive cost that can be introduced to take into account external factors, e.g., walking

close to another entity, preferential deviation to the left or the right, etc. Neutral value is $C_{add} = 1$.

Merging costs. Note, at this point, we have as many sectionings (whole sets of orientation sections) as the number of entities and static obstacles in the vicinity of E_{ref} . For each section of each set, we now have a cost. In exactly the same manner than presented in Section 3.2, we merge all the weighted sections. The cost of each subsection thus created is the sum of all the sections that were superposed and split to create it.

The fittest subsection has the lowest cost. In the corresponding range of valid speeds and orientations, we compute the closest to the ones desired by the reference entity, which is the final output to our reactive navigation module.

3.5. Discussion

Visibility of neighbor entities. As explained previously, our model predicts neighbor entities trajectories in order to decide on the best reaction to avoid them when necessary. We demonstrate in the next Section that real humans act in the same way, however, it is obvious they only do so for the humans they could visually perceive. In order to get a realistic reactive navigation, a perception field must be simulated. Our model distinguish two cases. When a neighbor entity is not seen because it is occluded by an obstacle, it is filtered out of the selection. As a result, an occluded neighbor entity has strictly no influence in the result. A limitation is the case were two pedestrians invisible one to the other converge toward the same place (e.g., at a street corner): they stop abruptly when finally perceiving one another, whereas a real human would anticipate this possibility and walks more carefully. When a neighbor entity is not seen because it is not in the field of view of the reference entity, we introduce it in the model, but having a null speed. Indeed, we consider that real humans feel someone is behind, but are unable to predict any trajectory. This also avoids backward motions provoking a collision with entities behind.

Connection with a locomotion animation module. As seen in Figure 1, output of our reactive navigation module is connected to a locomotion module. Our method may lead to important speed or orientation changes from which a realistic animation must be computed. In order to get as realistic animations as possible, we synchronize the animation module and the reactive navigation module, so that changes occur at the feet-land instants (left or right). Also, the locomotion module smooths variations itself.

4. Model Calibration and Validation

The previous Section described the technical basis of our model: resulting trajectories mainly depend on the parameters used in the cost functions (α, β). In order to get realistic

behaviors, a calibration of the model is required. In previous works, such a calibration is achieved by analyzing video captures of real crowd motions. However, automatic analysis techniques are not always applicable due to lighting conditions, and analysis by hand requires great effort. Moreover, pedestrians goals are generally unknown in video sequences, which may prevent strict comparisons between real data and simulations. For these reasons, we prefer to use a motion capture system to collect our reference data, with protocols defining the goals of each participant.

In a first experiment, we measure the interaction phenomenon in the following situation: two pedestrians achieve navigation tasks in an empty environment, we force them to have more or less interacting trajectories and observe adaptations to avoid contact. Results allow us to demonstrate the need for prediction in a realistic reactive navigation model and to calibrate some crucial factors. Secondly, we attempt to reproduce some typical crowd navigation situations at a microscopic level, such as corridor following, gate crossing, X-crossing, etc. in order to validate the model behavior in more complex scenes. For that, we captured as many participants as possible executing navigation tasks in an environment made of obstacles.

4.1. A protocol for model calibration: interactions between two pedestrians

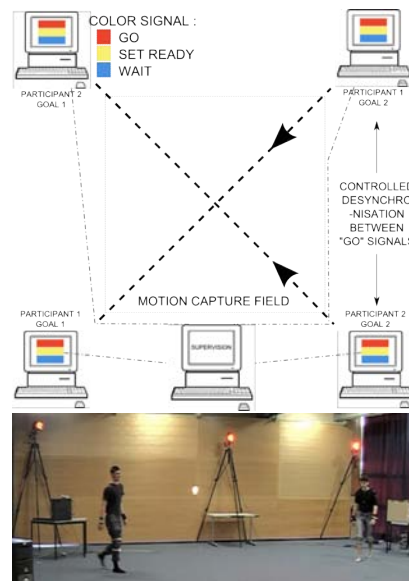


Figure 7: Four computers displaying signals to participants controlling their start-time and goal. A motion capture system retrieves resulting interactions.

The first experiment protocol allows us to control the interaction of two participants walking in an obstacle-free area, as illustrated in Figure 7. By interaction, we mean the

required trajectory adaptations made by each participant in order to avoid a collision. The objective of the experiment is to qualify the avoidance strategies developed by the participants given the conditions imposed by the protocol, and to quantify the trajectory corrections made in terms of velocity and orientation changes. For that, we place four computers at the corners of a square area visible for a motion capture system. Each participant must go from a given computer to the diagonally opposite one. We get a temporal control on the experiment by transmitting a start signal to participants using computer displays. All computers are synchronized in order to precisely control some delays between the start time of each participant, and to provoke more or less important interactions between them. We deduce the conditions for an interaction to occur or not. Participants always see each other but only perceive their own start signal. Finally, participants are equipped with 34 markers to get full body motion capture data. A total of 145 interactions were captured, 6 participants where involved, we placed computers in order to form $\pi/2$ or $\pi/3$ angles between trajectories.

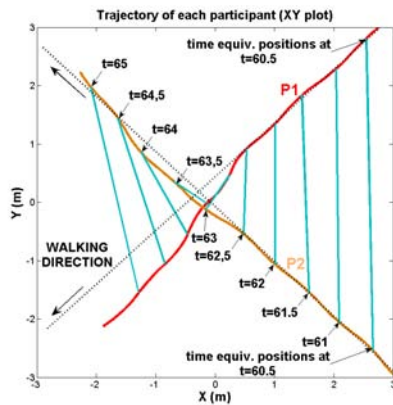


Figure 8: Horizontal trajectory of 2 interacting participants.

We detail our analysis method over a specific case whose results are shown in Figures 8 and 9. Here, as it can be seen in Figure 8 representing the horizontal trajectory of participants $P1$ and $P2$, a strong interaction occurred between experiment times 60s to 66s: trajectories are conspicuously deformed in order to avoid collision. We joined the respective positions of $P1$ and $P2$ at equivalent times (each half a second) in order to provide a temporal indication of events. A first look at the results leads one to think that participants reacted late ($t=62.5s$), just before collision: $P2$ passes before $P1$, $P1$ decelerates and turns to the left while $P2$ turns a little to the left to facilitate the passage.

But motion capture data allow a more precise analysis and the real intentions of participants appear clearly in Figure 9: top and bottom plots are the speed and orientation of each participant (respectively red and blue plots). In dashed horizontal lines, we represent some mean reference values for

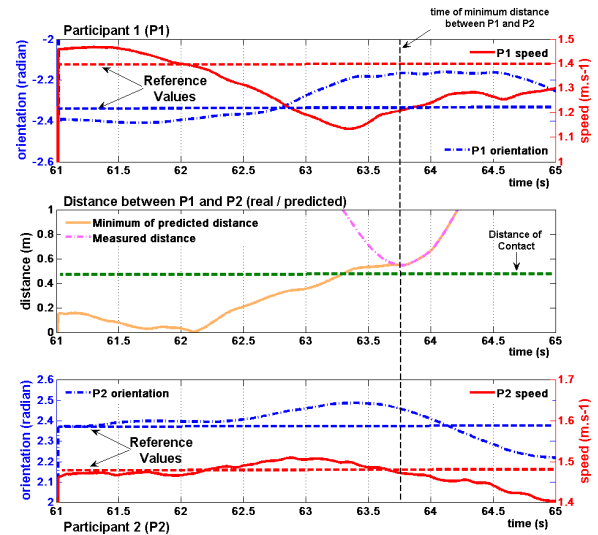


Figure 9: Orientation and speed variations of participants $P1$ (top) and $P2$ (bottom). Comparing the predicted and measured distances between participants allows to detect the time of adaptation to avoid a collision (center).

speed and orientation, measured during experiments where participants execute identical navigation tasks alone. We observe that two successive corrections finally compose this interaction:

- $t \in [61; 62s]$: $P1$ has a higher speed and deviates to the right compared to mean reference values. This reveals her first intention to pass before $P2$, who has a normal behavior. But $P2$ naturally walks faster (looking at the reference mean values) than $P1$ and this first strategy fails.
- $t \in [62; 62.5s]$: $P1$ decelerates.
- $t \in [62.5; 64s]$: A combined reaction is now clearly visible: $P1$ increases its deceleration and deviates to the left, which will allow her to pass behind $P2$, who facilitates the success of this new strategy by accelerating and deviating to the left.
- $t > 64s$: After the time where distance between $P1$ and $P2$ is minimal, participants achieve their goal and no particular interaction is observable.

Previous analysis showed that corrections may appear early in the experiments ($P1$ for $t = 61$ to $62s$), and differences between reference and measured values may be minor and hardly detectable. We introduce a criterion to both qualify automatically situations of interactions in the different experiments and answer a crucial question: are corrections made by participants pertinent (i.e., does collision occurs if no adaptation is made)? This will allow us to conclude on the accuracy of participants in evaluating potential collisions. Figure 9 illustrates our criteria: the red curve is the measured distance between $P1$ and $P2$. The violet one is $MinDist_{pred}$

defined as follow:

$$Pred_{1,2}(t,u) = Pos_{1,2}(t) + (u-t)\vec{v}_{1,2}(t)$$

with t is the experiment time, $u \in [t; \infty]$ a parameter, $Pred_1(t,u)$ and $Pred_2(t,u)$ are respectively the predicted positions of $P1$, $P2$ in the future time u , according to their current position $Pos_1(t)$, $Pos_2(t)$ and speed vector $\vec{v}_1(t)$, $\vec{v}_2(t)$.

$$InterDistance_{pred}(t,u) = \|\overrightarrow{Pred_1(t,u)Pos_2(t,u)}\|$$

is the evolution of the distance between $P1$ and $P2$ according to the previously predicted positions.

$$MinDist_{pred}(t) = \min_{u=t}^{\infty} (InterDistance_{pred}(t,u))$$

The plots reveal that the anticipation of $P1$ was accurate. Despite her initial faster speed and deviation to the right, she would not have been able to avoid $P2$ with a minimum predicted inter-distance below $0.2m$. Her change of strategy is revealed when $MinDist(t)$ is null at $t = 62.1s$ (from "passing in front of $P2$ " to "passing behind $P2$ ").

By analyzing the whole set of experiments our first conclusions are:

- Reactions are observable for $MinDist_{pred}(t) < 0.5m$, which allows automatic distinction between cases where interaction occurs or not.
- Anticipation is up to several seconds before a potential collision. Our experimental conditions did not allowed us to determine an upper bound to anticipation time because of the size of the motion capture field.
- Reaction is a combination of speed and orientation adaptation. Deviations are bounded whereas decelerations can lead to a complete stop for one of the participants. This occurs especially when participants modify both their trajectory so that $MinDist_{pred}(t)$ remains below $0.5m$ (conflicting corrections). The closer the collision is in time, the more speed adaptation is preponderant.
- Interaction is an accurate phenomenon. If no collision is predicted by our criterion, no reaction appears in data. At the other extreme, participants detect interaction situations early.

We calibrate our model to synthesize interacting pedestrians trajectories in a realistic manner. Anticipation is set to 8 seconds. In order to find a trade-off between performance and precision, time is discretized in a non-uniform manner ($1s$ steps in the near future, up to $4s$ for the last period). Minimal and maximal velocities to avoid a collision are computed for a set of walking directions, allowing combination of speed and orientation variations to avoid other pedestrians. We tuned the cost function to fit experimental data in a pragmatic manner. Statistical analyses are still on-going in order to calibrate the model automatically from data. Moreover, we noticed individual factors and deeper analyses should allow us to determine a variety of individual profiles to calibrate our model.

4.2. A protocol for model validation: capturing microscopic crowd phenomenon

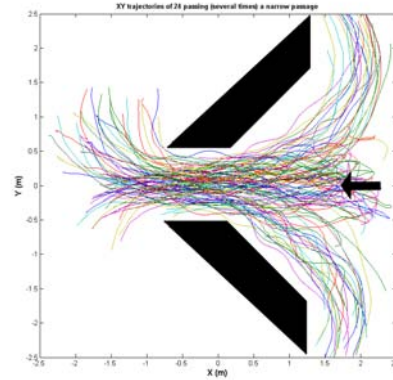


Figure 10: A microscopic crowd phenomenon: crossing a narrow passage with funnel shape.

In a second experiment, we capture 24 participants - each equipped with 5 markers placed on the chest - navigating in an area where we place static obstacles. The objective is to reproduce some frequently encountered situations where crowd flows meet, and to check for realistic emerging crowd behaviors in simulations (such as lane formations in corridors where opposite pedestrians flows meet). Figure 10 illustrates an example of obstacles setup, superposed with the set of recorded horizontal trajectories. Each participant crosses the narrow passage from the right to the left, and must circle the obstacle to reenter in the area from the right. We tested several setups: X-crossing with several flow directions repartitions, corridors with single or opposite flows, gate crossing, etc. We can draw first conclusions from comparisons between simulations and real data. Characteristic phenomena could emerge from our model as observed in real data such as lane formation in corridors where two opposite flows meet, pedestrians going in same directions at X-crossing tend to group in order to facilitate their passing-by, high speed changes occur where flows meet with visibility limited by obstacles, etc. Thus, we validate qualitatively the basis of our model. However, quantitatively, simulation and real data still need to be accurately compared, and we are currently performing a deeper analysis.

4.3. Discussion

From real data, we can discuss the limitations of our model. During the calibration step, we found a balance between the cost of direction changes and speed changes (α parameter of the cost function Section 3.4) that allowed us to have similar results in simulation and experiments with identical conditions. However, we could not find a calibration that satisfies all situations. Simulation results were still valid (no collision, and no dead-locks), but realism was decreasing in some

specific situations. We think an always-valid calibration is impossible to find for two reasons. First, our anticipation time, T , is a constant parameter, whereas real humans act in a more reactive way where population is dense. Second, time-interval duration step, Δt , is also a constant parameter during the sectioning step (see Section 3.2). In other words, our model precision is constant and should depend on the situation. We believe that these two model parameters should be adapted according to local population density (at least) in order to improve the realism of the results.

5. Results

Performance We have run a variety of situations reproducing the ones we motion captured, as shown in the accompanying video. Implementation is in C++ and simulations ran on a Pentium M 1.6GHz, 1GB memory, with an Nvidia 4200Go 64MB graphics card.

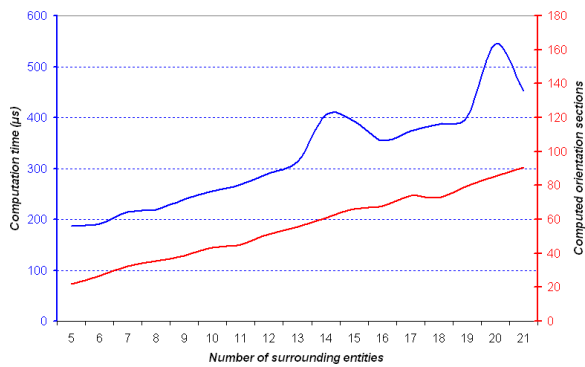


Figure 11: Solution Performance. The blue curve plots the computation time required for solving one interaction according to the number of neighbor entities taken into account. The red curve plots the number of orientation sections created in order to compute the solution. From these plots, it is possible to compute global performances with respect to the density of people.

The performance provided in Figure 11 does not correspond to the computation time functions of the total number of simulated entities, but to the computation time for one entity according to the density of people. In fact, this density of people implicitly defines the number of neighbor entities to take into account for the computation. Moreover, the plot limit of 21 surrounding entities has not been specified, but is a result of our benchmarks using by far more entities in different situations (an average of 100 moving agents, plus the walls). Then, the computation time for one entity, between 200 and 500 μs , may appear high compared to previous approaches, but is relatively stable for growing entities numbers. In addition, our model is only refreshed at each foot step, corresponding to an average rate of $1 - 2Hz$. Thereby, taking the worst cases, our model can easily handle 1,000 entities in real time, and its complexity is scalable and could

be reduced to simulate more entities by decreasing the anticipation time. However, as previously mentioned, this model is integrated in a whole virtual human architecture, managing other tasks like path planning, rational behavior, animation, and so on. Based on our experiments, we are able to fully simulate and animate approximately 150 entities in real time. Moreover, the performance depends on the complexity of the environment, and on the density of the crowd, that is why it is difficult to provide representative performance considering the number of influent parameters.

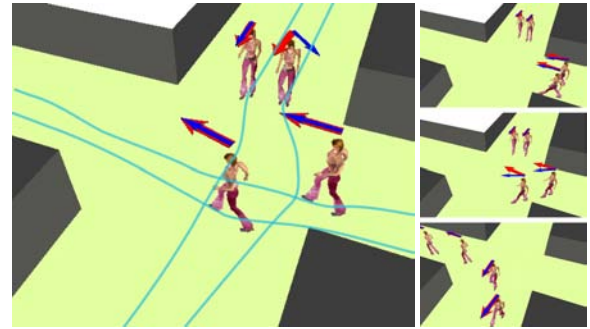


Figure 12: Detailed view of a unit interaction case. Red arrows are desired directions θ_{des} while blue ones result from the reactive navigation. Virtual humans adapt both their speed and orientation in quite an optimal and realistic manner to avoid each other (see video for animation).

Realism As mentioned Section 4.2, we could compare our simulations with real data, however, a quantitative validation of our model is still in progress. Our first results are promising. We solved several drawbacks observed in previous approaches. First, anticipated reactions in low density areas improve results realism (Figure 12): in such cases, conflicting states (typically, close entities face to face) never appear in simulations. Indeed, thanks to a sufficiently stable state of the environment, model predictions remain valid for a large enough interval to always avoid any close interaction. Second, in dense population situations, pedestrians do not have jerky trajectories with successive and contradictory reorientations or turn-backs. In this case, next-future predictions are accurate enough to avoid such unrealistic behaviors and dead-locks. However, distant time predictions appear to be useless because of the constantly changing situation.

6. Conclusion and Perspectives

We presented a novel approach to solve interactions between virtual pedestrians in the context of realistic crowd simulations. Our first contribution is to introduce long-time predictions in our model when accounting for other pedestrians moving around. Such a prediction allowed us to solve major drawbacks of previous approaches to this known difficult problem. Our second contribution is to propose experiments

to validate and calibrate our approach using a motion capture system. Compared to video-streams analysis, we reach a higher level of precision that allowed us to obtain a fine analysis of events occurring during real humans interactions. We also captured crowd behaviors at a microscopic level in order to validate our simulation results. Work is still in progress concerning this last point.

Our results are promising and we can identify our model limitations to get even more realistic results: this determines our future work directions. First, we want to dynamically adapt the model parameters to the variations of environment context. Especially, the prediction time-ranges and precision must fit the local population density. Second, the model does not account for social factors. We want our model to be able to consider couples of pedestrians or larger groups navigating among other pedestrians: such groups shall remain as gathered as possible during their navigation. We believe our implementation capable of supporting such evolutions easily, and work is underway. Finally, connections to the animation module must be enhanced. A feedback must emanate from the locomotion module in order to score the proposed reactions, purely in terms of realism of motion. This additive cost would help in taking realistic decisions. Recent results in animation evaluation techniques is an inspiration source to reach such a goal.

7. Acknowledgement

This work is funded by the French Research Agency ANR Project ANR05RNTL02501 *SIMULEM*, in collaboration with the French railroad company *SNCF*. We also thank members of our team who provided the animation module used in this work, *MKM* [KMA05], and who helped us during the motion capture experiments.

References

- [BLA02] BAYAZIT O. B., LIEN J.-M., AMATO N. M.: Roadmap-based flocking for complex environments. In *Pacific Conf. on Comp. Graphics and Applications* (2002), pp. 104–113.
- [BMdOB03] BRAUN A., MUSSE S., DE OLIVEIRA L., BODMANN B.: Modeling individual behaviors in crowd simulation. In *16th Int. Conf. on Computer Animation and Social Agents* (New Brunswick, NJ, USA, May 2003), IEEE, pp. 143–148.
- [Bol81] BOLES W.: Planning pedestrian environments: A computer simulation model. *Man-Environment Systems* 11 (1981), 41–56.
- [GKM*01] GOLDENSTEIN S., KARAVELAS M., METAXAS D., GUIBAS L., AARON E., GOSWAMI A.: Scalable nonlinear dynamical systems for agent steering and crowd simulation. *Computers and Graphics* 25, 6 (2001), 983–998.
- [Gof71] GOFFMAN E.: *Relations in public : microstudies of the public order*. New York : Basic books, 1971.
- [Hen71] HENDERSON L. F.: The statistics of crowd fluids. *Nature* 229 (1971), 381–383.
- [HFV00] HELBING D., FARKAS I., VICSEK T.: Simulating dynamical features of escape panic. *Nature* 407 (2000), 487–490.
- [KMA05] KULPA R., MULTON F., ARNALDI B.: Specific representation of motions for interactive animation with several characters. *Computer Graphics Forum* 24, 3 (2005), 343–352.
- [LKF05] LAKOBA T. I., KAUP D. J., FINKELSTEIN N. M.: Modifications of the helbing-molnár-farkas-vicsek social force model for pedestrian evolution. *Simulation* 81, 5 (2005), 339–352.
- [LMM03] LOSCOS C., MARCHAL D., MEYER A.: Intuitive behavior in dense urban environments using local laws. In *Theory and Practice of Computer Graphics* (2003).
- [MT01] MUSSE S. R., THALMANN D.: Hierarchical model for real time simulation of virtual human crowds. *IEEE Transaction on Visualization and Computer Graphics* 7, 2 (jun 2001).
- [PDB05] PARIS S., DONIKIAN S., BONVALET N.: Towards more realistic and efficient virtual environment description and usage. In *First International Workshop on Crowd Simulation (V-Crowds'05)* (2005), VRlab, EPFL.
- [PDB06] PARIS S., DONIKIAN S., BONVALET N.: Environmental abstraction and path planning techniques for realistic crowd simulation. *Computer Animation and Virtual Worlds* (2006), 325–335.
- [Pes71] PESCHL I.: Passage capacity of door openings in panic situations. *BAUN* 26, 2 (1971), 62–67.
- [PM78] PREDTECHENSKII V. M., MILINSKII A. I.: *Planning for foot traffic flow in buildings*, National Bureau of Standards ed. Amerind Publishing CO Pvt Ltd, New Dehli, India, 1978.
- [Rey00] REYNOLDS C. W.: Interaction with groups of autonomous characters. In *Game Developers Conference* (2000).
- [SGC04] SUNG M., GLEICHER M., CHENNEY S.: Scalable behaviors for crowd simulation. *Computer Graphics Forum, Eurographics'04* (2004).
- [ST05] SHAO W., TERZOPOULOS D.: Autonomous pedestrians. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2005), ACM Press, pp. 19–28.
- [TCP06] TREUILLE A., COOPER S., POPOVIĆ Z.: Continuum crowds. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers* (New York, NY, USA, 2006), ACM Press, pp. 1160–1168.
- [Yam98] YAMORI K.: Going with the flow: Micro-macro dynamics in the macrobehavioral patterns of pedestrian crowds. *Psychological Review* 105, 3 (1998), 530–557.

Animation de données volumiques à une résolution arbitraire utilisant des raideurs non-uniformes

Matthieu Nesme, François Faure & Yohan Payan

June 7, 2007

Abstract

Mettre au point des simulateurs d'entraînement aux gestes chirurgicaux exige de disposer d'une méthode d'animation de corps déformables rapide et autorisant un rendu détaillé. Nous présentons comment animer et afficher efficacement des données volumiques issues de scanners. Alors que la plupart des méthodes utilisent classiquement une représentation volumique exacte de l'objet à simuler pour calculer la mécanique des milieux continus, nos éléments finis sont des hexaèdres qui décomposent le volume englobant de l'objet à une résolution arbitraire. Ceci réduit la complexité et les limitations inhérentes à l'utilisation d'un maillage tétraédrique, et résulte en un maillage régulier bien conditionné. Ces voxels peuvent être obtenus automatiquement à partir d'une surface ou directement à partir de données scanner segmentées, ce qui est très utile lorsque l'on veut travailler sur des modèles spécifiques aux patients. Mécaniquement parlant, la principale différence se trouve au niveau des éléments englobant la surface de l'objet, car ceux-ci sont plus ou moins vides et certains de leurs noeuds sont à l'extérieur de l'objet. Les propriétés physiques, masses et raideurs, de ces éléments sont choisies telles que la distribution de la matière soit prise en compte de façon à modéliser du mieux que possible le comportement à toutes résolutions. Afin d'obtenir une animation rapide et stable, un calcul co-rotationnel des déformations est utilisé. Les résultats de cette méthode sont comparés à des valeurs théoriques. Enfin, nous montrons comment il est possible d'effectuer un rendu volumique rapide en se basant efficacement sur la structure de voxels utilisée pour la mécanique. Cette approche permet la simulation interactive de formes arbitrairement complexes et détaillées.

Implantation temps réel sur GPU d'un écran d'épingles dynamique

Kevin Sillam, Matthieu Evrard, Annie Luciani

ICA laboratory – INPG 46 avenue Félix Viallet 38000 Grenoble, France

Mail : prénom.nom@imag.fr

Résumé

Cet article présente une implantation temps réel sur processeur graphique de la méthode dite de l'écran d'épingles dynamique inspiré de la méthode de gravure d'Alexandre Alexeïeff. Cette méthode initialement, proposée par [HL02] et fondée sur un formalisme masses-interactions, était trop coûteuse pour pouvoir s'insérer dans une application interactive. Cet article résout ce frein majeur à l'exploration des possibilités dynamiques de la méthode. Il présente alors un certain nombre d'applications obtenues en temps-réel : (1) la visualisation de modèles particuliers simulés sur une machine amont en temps réel ou en temps différé, (2) l'utilisation de l'écran d'épingles dynamique comme modèle physique pour des scènes 3D complexes, (3) une gravure directe par l'intermédiaire de la souris.

Mots clés: modélisation physique, visualisation de modèles particuliers, implémentation sur GPU, écran d'épingles, phixel.

1 Introduction

Au cœur du processus de gravure, se trouve une relation entre deux corps physiques, l'un, dur, que l'on peut nommer « marqueur » vient modifier l'état de l'autre, appelé alors « support ».

La roche sédimentaire comme fossile sera un support durable pour rendre compte de la sépulture d'un animal préhistorique. La surface de l'eau au contraire sera un support plus volubile de la mémoire d'un galet qui a ricoché. Quoi qu'il en soit, le support est toujours une mémoire, une trace, du passage d'un marqueur, de sa dynamique, de l'action

qu'il a eu sur le support, que cette mémoire soit ou non éphémère.

Un certain nombre de films d'animation ont utilisé la gravure comme procédé central de création des images. La gravure permet en effet à l'artiste d'obtenir des formes aux contours flous, insaisissables.

Parmi la multitude de ces œuvres, on peut retenir l'approche originale d'Alexandre Alexeïeff [AAA96]. Ce graveur de formation met en place au début des années 30, un support de gravure fait d'une multitude d'épingles coulissant perpendiculairement à un plan de fixation. Alexeïeff vient alors graver ce support en déplaçant les épingles à l'aide d'objets divers faisant office de marqueurs. Une source de lumière éclaire cet « écran d'épingles » et fait apparaître des blancs dans les zones où les épingles ont l'altitude la plus élevée et des noirs dans les zones où elles sont les plus enfoncées. La succession des photos, prises de cet écran dans des configurations d'altitudes modifiées au fur et à mesure par le graveur, crée artificiellement le mouvement lors de leur projection séquencée.

Habibi et al. [HL02] partent de l'idée d'Alexeïeff pour résoudre une partie du problème lié à la visualisation de modèles particuliers. Des particules en interaction dont le mouvement est simulé produisent des comportements riches et variés. En revanche, une difficulté majeure inhérente à ce type de formalisme pour la production d'images animées réside dans le fait qu'il est difficile de produire des séquences d'images animées par le rendu direct des masses ponctuelles en mouvement, les masses ponctuelles n'ayant pas de spatialité. D'une manière générale, il est donc nécessaire de développer des méthodes qui étendent la spatialité de ces masses ponctuelles pour compléter la chaîne de production d'images animées par modèle physique particulière. L'idée d'Habibi et al. est

de venir marquer un écran d'épingles simulé lui-même par modèle physique masses-interactions par les masses en mouvement d'un modèle physique amont. L'écran d'épingles dynamique d'Habibi et al. a donc cela de différent de celui d'Alexeïeff que l'ordinateur permet de générer un mouvement pour les épingles, leur conférant un comportement dynamique.

Les auteurs obtiennent des images réellement convaincantes de fumées, de sables, d'eau et même de solides [HL02]. En revanche la méthode est, à l'époque relativement coûteuse en temps de calcul et ne permet pas d'être intégrée à une application interactive.

L'objet de cet article est d'utiliser les « shaders », qui sont des programmes compilant et s'exécutant sur les cartes graphiques modernes, de profiter des parallélisations de calculs qu'ils offrent, pour réimplanter la méthode d'écran d'épingles dynamique décrite par Habibi et al. et la faire tourner en temps réel.

La section 2 rappelle brièvement la méthode d'écran d'épingles dynamique de Habibi et al. La section 3 fait un état de l'art des implantations de modèles physiques particulières sur GPU. La section 4 présente l'implantation que nous avons réalisée. La section 5 relate l'incorporation de cette méthode dans une chaîne de simulation incluant les modèles particuliers amont simulés en temps réel. La section 6 décrit les performances obtenues. La section 7 propose une interface pour les utilisateurs. Finalement, la section 8 expose les résultats qui ouvrent eux-mêmes sur des applications nouvelles de l'écran d'épingles.

2 L'écran d'épingles dynamique

Pour plus de précision sur la méthode de l'écran d'épingles dynamique, on pourra se référer aux articles [HL02]. Il est cependant à noter que l'écran d'épingles est un réseau CORDIS masses-interactions comme décrit dans [LJF*91].

En entrée de l'écran d'épingles, il y a un ensemble de trajectoires de points que l'on nomme « marqueurs ». Ces marqueurs vont venir marquer l'écran, c'est-à-dire qu'ils vont agir dessus, créer une trace, sans que l'écran

agisse sur eux en retour. Ainsi les trajectoires des marqueurs peuvent être fournies par n'importe quel processus (simulation, entrée par capteur) sans que celui-ci soit nécessairement opéré en temps réel synchrone avec la simulation de l'écran d'épingles. Les trajectoires peuvent donc être notamment fournies par une simulation amont hors-ligne d'un ensemble de particules dont la trace a été conservée dans un fichier.

L'écran d'épingles dynamique est un ensemble d'éléments physiques nommés « phyxels » (cf. figure 1). Ces phyxels, sortes d'épingles dynamiques, sont des algorithmes dont la sortie est une valeur scalaire qui correspond à la « hauteur » de l'épingle. Plus précisément un phyxel sera composé de deux masses, dont une fixe, en interaction unidimensionnelle. L'interaction la plus simple sera l'interaction linéaire visco-élastique, mais des interactions plus complexes seront envisagées dans cet article.

Les phyxels peuvent aussi interagir entre eux. Ainsi chaque masse mobile de chaque phyxel pourra être en interaction avec les masses des phyxels voisins. Ce faisant, on permet ainsi une diffusion spatiale de la trace opérée par les marqueurs. L'ensemble des travaux présentés dans cet article s'est restreint à l'utilisation d'un écran d'épingles 2D, c'est-à-dire à un tableau 2D de phyxels. Un phyxel possède donc au maximum 8 voisins dans cette configuration.

Les marqueurs eux-mêmes sont en interaction visco-élastique unidirectionnelle (l'épingle n'agit pas sur le marqueur), seuillée en distance, avec chaque masse mobile de chaque phyxel. C'est cette interaction qui permet aux marqueurs de laisser une trace sur l'écran.

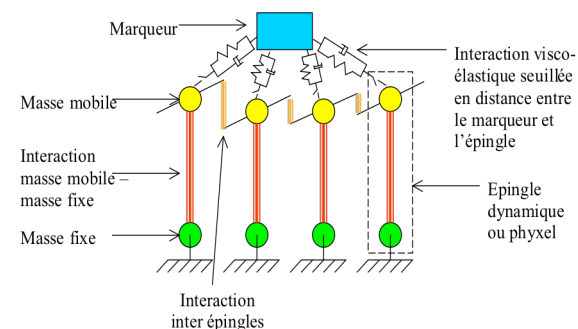


Figure 1 : le modèle masses-interactions de l'écran d'épingles dynamique

En sortie de l'écran d'épingles, on obtient bien un ensemble de valeurs scalaires correspondant aux hauteurs des épingles (distance masse fixe - masse mobile au sein d'un phixel). Ce tableau de scalaire peut alors être utilisé en entrée d'une méthode de rendu. La méthode la plus simple proposée par [HL02], est le contrôle de la couleur d'un ou plusieurs pixels de l'écran par la combinaison d'une ou plusieurs hauteurs en sortie d'un ou plusieurs phixel. On peut imaginer que les hauteurs contrôlent de la même manière des points de contrôle de surfaces dans un espace 3D ou encore des orientations de normales. Plusieurs méthodes de rendu seront présentées dans la suite de cet article et étendent les résultats présentés par [HL02].

3 Implantation de modèles physiques particuliers sur GPU

La programmation sur processeurs graphiques (GPU), non pas seulement pour des applications de rendu de scène 3D, mais pour tout type de calcul numérique, a connu un véritable succès ces dernières années. Ces types d'unités de calculs ont une architecture différente des processeurs présents sur les cartes mères (CPU). La différence fondamentale avec ces derniers est la parallélisation des calculs. Pour une revue de l'utilisation actuelle des GPU que peut en faire la communauté de l'informatique graphique, on pourra se reporter à [OLG*05].

Le principe général de programmation sur ce genre d'architecture est de stocker les variables de la simulation dans une structure de donnée propriétaire comme, par exemple, une texture qui était initialement conçue pour stocker les composantes RVBA des textures à afficher. Un programme appelé « shader » se compile et s'exécute sur une partie de la carte graphique. Il exécute la même tâche en parallèle sur les données stockées dans la structure de donnée en fonction de certains paramètres.

Plusieurs travaux récents se sont intéressés à l'implantation de systèmes de particules plongées dans des champs [KKK*05], en collision avec des objets rigides [KLR04] ou en interaction entre elles [GW05, LS06] sur des GPU en suivant ce principe de programmation. Nos travaux s'inscrivent dans

cette direction, mais l'application est sensiblement différente.

Si les variables stockées dans les textures étaient des positions 3D de particules ou de points de contrôle de formes rendues directement, donc des valeurs vectorielles correspondant directement à un élément géométrique de la scène, les valeurs scalaires que nous produisons avec notre écran d'épingles dynamique ne peuvent correspondre directement à un élément de l'espace 3D et contrôlent donc un processus de rendu qui peut aller du contrôle direct de la couleur d'un pixel à une chaîne d'affichage plus complexe (en contrôlant, par exemple, les orientations des normales d'une surface servant à déterminer le rendu de leur éclairage). Les interactions implantées dans les articles cités étaient de simples interactions linéaires de type ressort-frottement. Par conséquent, une autre différence notable avec les travaux ayant fait intervenir des masses en interaction est que des interactions plus complexes sont susceptibles d'intervenir au sein du modèle de notre écran d'épingles dynamique comme le propose déjà Habibi dans sa thèse [H97].

4 Implantation de l'écran d'épingles dynamique sur GPU

4.1 □ Encapsulation des variables de la simulation

Les variables et certains paramètres de la simulation sont encapsulés dans des structures de données de type textures. Le GPU effectue ensuite des calculs sur ces textures.

Pour un écran d'épingles de $m \times n$ phixels, une texture de taille $n \times m$ est créée sur la carte graphique (nous utilisons une texture rectangulaire pour avoir des coordonnées non normalisées). Chaque épingle est donc repérée par sa position en x et y sur la texture. Pour chaque épingle, la texture contient la position au temps t (composante Rouge), la position au temps $t-1$ (composante Verte). La composante bleue est conservée pour stocker certains paramètres des interactions amenés à évoluer.

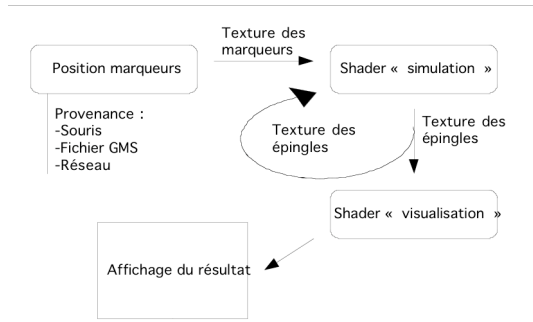


Figure 2 : Schéma général de l'architecture logicielle de la simulation de l'écran d'épingles dynamique sur GPU

4.2 □ Déroulement de l'algorithme

L'algorithme est une boucle composée de 3 passes distinctes (cf. figure 2):

Pour chaque boucle, il faut :

- Exécuter une passe « lecture » au cours de laquelle la position courante des marqueurs est inscrite dans une texture spécifique (texture « marqueur »).
- Lancer l'exécution de la passe « simulation » qui va inscrire dans la texture appelée texture « épingle » les nouvelles hauteurs des épingles en fonctions des différentes interactions.
- Effectuer une passe « rendu » qui va rendre à l'écran une image contrôlée par les valeurs scalaires de hauteur des épingles (sorties des phyxels).

4.2.1 La passe « Lecture »

Une texture est créée pour faire passer la position des marqueurs. Pour k marqueurs, la texture créée sera de taille $k \times 1$, les deux premières composantes codant la position, les deux dernières étant conservées pour d'autres informations, comme une raideur ou un coefficient d'aplatissement [cf. HL02].

Dans notre implantation, les trajectoires des marqueurs peuvent provenir de 3 sources différentes :

- Un fichier de trajectoires produit éventuellement par une simulation hors-ligne amont. Le format utilisé est le format de codage de données de mouvement GMS [LEC*06].
- Un flot de données provenant d'un réseau. Une simulation amont peut alors être

effectuée en temps-réel synchrone (voir section 5).

- La position d'un capteur comme la souris d'ordinateur.

Les trajectoires des marqueurs peuvent donc être parfaitement simulées par un modèle amont de type système de particules ou réseau masses-interactions. Cette simulation peut être temporellement décorrélée de la simulation de l'écran d'épingles dynamique puisque celui-ci n'agit pas sur les marqueurs. En revanche, si l'on souhaite que la simulation de l'écran d'épingles soit effectuée en temps réel, il convient que la lecture des trajectoires ne soit pas une phase bloquante. Ainsi la latence de l'accès matériel et de l'accès logiciel doit s'inscrire strictement dans une phase de la boucle de simulation.

4.2.2 La passe simulation

La passe « simulation » prend en entrée la position de tous les marqueurs, la hauteur de toutes les épingles au temps t et $t-1$ (hauteur de la masse mobile par rapport à la masse fixe), ainsi que tous les paramètres de l'écran d'épingles. Elle inscrit dans la texture en sortie (utilisation d'un frame buffer object), la nouvelle position des épingles après calcul, ainsi que leurs anciennes positions et les paramètres modifiables de l'interaction.

La simulation peut se décomposer elle-même en 4 phases :

- **Calcul de la force appliquée sur chaque épingle par les marqueurs**

Chaque marqueur interagit avec chaque épingle de son voisinage. L'interaction est une interaction visco-élastique seuillée en distance et unidirectionnelle [LJF*91].

- **Calcul de la force appliquée sur chaque épingle par son voisinage**

La masse mobile de chaque épingle est reliée par une interaction à certaines de ses voisines. Nous avons implanté des interactions avec (V4) les 4 voisins (les deux horizontales et les deux verticales) ainsi qu'avec (V8) les 8.

- **Calcul de la force appliquée sur chaque épingle par son interaction avec le sol**

La masse mobile de l'épingle est elle-même en interaction avec la masse fixe de la même épingle.

- **Détermination de la nouvelle hauteur des épingles**

La nouvelle hauteur de l'épingle est déterminée à partir de la sommation de toutes ces forces F_n

Ainsi, la nouvelle hauteur de l'épingle est donnée par :

$$Z_{n+1} = 2.Z_n - Z_{n-1} + F_n(Tech^2/m)$$

Avec m : la masse de l'épingle, Z_n et Z_{n-1} les hauteurs au temps t et $t-1$, et $Tech$ la période d'échantillonnage du mouvement.

Ces calculs sont effectués par le shader « simulation » pour chaque épingle. Les valeurs trouvées et stockées (hauteur au temps t , $t-1$, longueur à vide) sont stockées dans la texture « épingles » et envoyées par la suite à la passe « rendu ».

4.2.3 La passe « rendu »

Nous effectuons une deuxième passe sur le GPU, pour permettre le rendu des scalaires (hauteurs) fournies par chaque phyxel (épingles). Son but est de passer du phyxel au pixel par un procédé plus ou moins complexe. Nous avons expérimenté 3 méthodes différentes de rendu de l'écran d'épingles dynamique:

Attribution de couleur simple :

La couleur est uniquement attribuée en fonction de la hauteur. On transforme 1 scalaire en 3 composantes (RVB). Cela permet une visualisation simple, mais assez esthétique. On peut nuancer la couleur par la hauteur de l'épingle correspondante. Par exemple, on peut définir le blanc pour une hauteur minimale, et le noir pour une hauteur maximale : on aura toutes les nuances des gris en visualisation.

Le Bump – Mapping :

Cette technique consiste à émuler un relief à partir d'une « normal map » et d'une position de source d'éclairage. Nous obtenons cet effet

en créant une « normal map » par un « shader » en fonction de la dérivée de la hauteur des épingles (considération du voisinage). Il nous suffit alors d'éclaircir ou d'assombrir la lumière diffuse renvoyée par chaque pixel, pour obtenir un simulacre de relief. Cela permet une visualisation en fausse 3D mais ajoute un réel aspect de profondeur à l'observation.

Ajout d'une dimension :

Il peut être intéressant de visualiser l'écran d'épingles 2D que nous avons implanté dans un espace 3D. Ce type de rendu sera utilisé notamment pour le calibrage et pour une meilleure compréhension du phénomène dynamique.

Nous utilisons un « geometry shader » pour découper notre surface en un grand nombre de sommets, puis nous modifions la hauteur de ces sommets (ou vertex) en fonction de la hauteur de l'épingle lui correspondant. Nous pouvons grâce à cela, définir une surface variable en 3 dimensions réelles.

4.2.4 Initialisation

L'initialisation du système est effectuée en attribuant à la hauteur des épingles la longueur à vide de l'interaction avec la masse fixe. Cela permet de :

- s'assurer que toutes les épingles débutent à la même hauteur
- faire un premier test sur la stabilité de l'écran d'épingles (la hauteur des épingles ne varient pas avant une perturbation extérieur).
- préparer l'écran d'épingles à interagir avec les marqueurs. Cette initialisation est effectuée à chaque fois que l'utilisateur veut remettre l'écran d'épingles en position initiale (reset).

4.2.5 Codage des variables

La première implantation présentait des bruits dus à la quantification des données. En effet, la hauteur des épingles était codée sur 8 bits et réduite entre 0 et 1 (caractéristique d'OpenGL). Le bruit de quantification était alors très grand. Il convient donc de choisir une quantification bien supérieure pour résoudre ce problème.

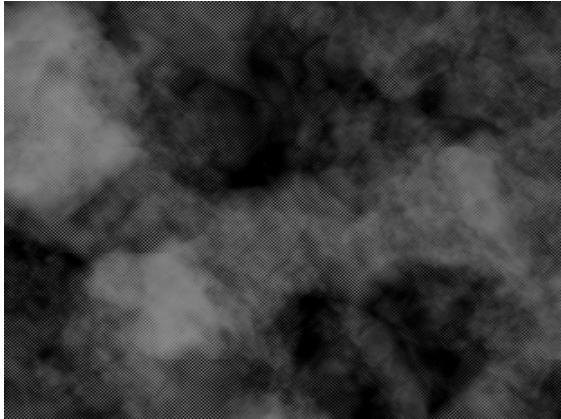


Figure 3 : Comportement de l'écran d'épingles à l'équilibre pour un mauvais codage des Hauteurs (ici 8 bits)

5 L'écran d'épingles dynamique dans une chaîne de simulation temps réel

Une simulation est effectuée sur une machine amont. Elle va, en sortie, délivrer la position de tous les marqueurs envoyés vers l'écran d'épingles. La question est de faire passer ces informations, le plus vite possible, de la machine amont à l'écran d'épingles implanté sur une autre machine.

Nous utilisons pour cela, le protocole UDP sur un réseau Ethernet. Une trame est constituée pour encapsuler les données et assurer la cohérence des informations à la réception. Cette trame est constituée de la sorte :

| | | | | | | | | | |
|-----------|-----------------|--------------|----------|---------|---------|----------|---------|---------|-----|
| Fanion | Nombre de masse | Numéro Trame | N° masse | pos X | pos Y | N° masse | pos X | pos Y | ... |
| FFFF.FFFF | 3 | 245 | 0 | 12,5 | 3,3 | 1 | -2,5 | -3,8 | ... |
| 32 bits | 32 bits | 32 bits | 32 bits | 32 bits | 32 bits | 32 bits | 32 bits | 32 bits | |

x Nombre masses

Elle permet à la machine réceptrice de se resynchroniser sur une nouvelle trame grâce au fanion. La lecture est donc non-bloquante, ce qui autorise le déroulement des deux simulations en parallèle. La communication est asynchrone dans le sens où seule une nouvelle trame va contraindre la simulation de l'écran d'épingles. Le nombre de masse permet de changer la simulation amont, et de faire passer l'information à l'écran d'épingles pour qu'il se réadapte au nouveau nombre de marqueurs. Le numéro de trame permet de repérer les frames sautées. Le numéro de masse est là pour assurer qu'il n'y ait pas de décalage majeur des indices. Si une trame est jugée incorrecte,

l'utilisateur est prévenu et la frame correspondante est sautée.

6 Performances obtenues

La simulation de l'écran d'épingles dynamique a été réalisée sur un GPU GTX 8800 (768 Mo) branchée sur un PC avec CPU Dual Core II 2,4 GHz.

Les performances dépendent énormément du nombre de marqueurs. Nous les présentons ici pour une résolution de 1100 x 1100 épingles. Le tableau suivant a été obtenu dans le cas où les marqueurs proviennent d'une simulation amont différée (fichier GMS).

| | | | | | | | |
|---------------------|-----|-----|-----|-----|-----|-----|-----|
| Nombre de marqueurs | 1 | 5 | 10 | 20 | 50 | 100 | 300 |
| Fréquence | 693 | 510 | 358 | 223 | 105 | 56 | 20 |

Dans le cas de la chaîne temps réel (section5), il y a plusieurs limites à la vitesse du système :

- La vitesse à laquelle tourne la simulation amont
- La vitesse de la communication (dans notre cas, Ethernet à 100 Mbits/s, en protocole UDP)
- La vitesse de simulation de l'écran d'épingles

L'écran d'épingles étant destiné à la visualisation, nous pouvons tolérer une latence non visible, mais nous devons respecter une cadence cohérente avec la chaîne amont. Par rapport à nos performances, cela devient un problème pour un nombre de marqueurs importants (supérieur à 200) tant au niveau de la communication Ethernet que de la simulation de l'écran d'épingles.

Le tableau suivant présente les performances obtenues lors d'une simulation effectuée sur une autre machine envoyant les marqueurs à l'écran d'épingles (1100 x 1100 épingles) par le réseau. Pour respecter une certaine limite maximale dans la latence, des frames (une boucle de simulation) sont sautées en cas de surcharge.

| | | | | | | | |
|---------------------|-----|-----|-----|-----|-----|-----|-----|
| Nombre de marqueurs | 1 | 5 | 10 | 20 | 50 | 100 | 300 |
| Fréquence | 690 | 508 | 352 | 220 | 104 | 55 | 23 |

7 Interface

Nous avons porté aux usages notre écran d'épingles dynamique grâce à une interface graphique développée en QT4.

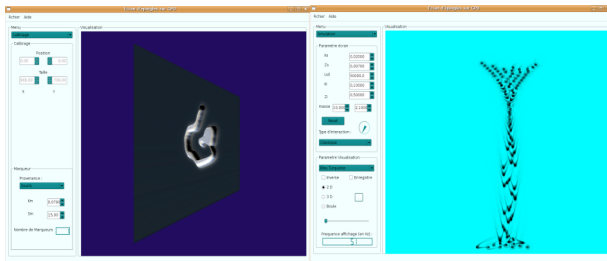


Figure 4 : *Interface utilisateur. A gauche : le mode calibration. A droite : le mode simulation.*

L'interface se décompose en deux modes distincts qui modifient les options accessibles : un mode « calibration » et un mode « simulation » (cf. figure 4).

7.1 □ *Le mode calibration*

Il permet de paramétrer l'écran d'épingles avant la simulation, c'est-à-dire de régler sa taille et sa position en fonction des marqueurs. L'utilisateur pourra le redimensionner pour le centrer sur la partie qui l'intéresse. Dans ce mode, la simulation n'est pas en cours, l'écran d'épingles est représenté par une surface semi-transparente et les marqueurs par des boules de diamètre de la taille de leurs rayons d'interactions. L'utilisateur a accès aux paramètres relatifs aux marqueurs (le seuil en distance de l'interaction, sa raideur, leur provenance...) ainsi qu'aux paramètres relatifs à la position et la taille de l'écran d'épingles dans l'espace.

7.2 □ *Le mode simulation*

Une fois les paramètres précédents réglés, l'utilisateur peut passer en mode simulation. Ce mode va lui permettre d'observer le comportement de l'écran d'épingles en temps réel. Il a ici accès à tous les paramètres des interactions, ainsi qu'un choix de plusieurs types d'interactions (ressort, viscosité, modification des longueurs à vide et d'autres...). De plus, l'interface permet de choisir parmi plusieurs types de rendu (ceux qui sont présents dans la section 8). Il peut ensuite jouer sur les couleurs et les textures.

Une fonction d'enregistrement image par image de la scène et d'exportation au format TGA lui est aussi proposée. Un certain nombre de comportements globaux (interactions et paramètres donnés) sont préenregistrés, et l'utilisateur peut enregistrer et charger ses propres modèles à l'aide du menu. De plus l'utilisateur a le choix de visualiser la surface de l'écran d'épingles en 2D ou en 3D. Dans le cas de la 3D, il peut se déplacer et observer la surface (creusée, déformée par la hauteur des épingles). Dans le cadre du plaquage de texture sur une surface déformable, seule la sphère est actuellement proposée.

8 Résultats

8.1 □ *Visualisation de modèles particuliers*

Nous avons testé les capacités maximales en temps réel de notre application. Le modèle choisit, pour réaliser ce projet, est le même que celui décrit dans [HL02]. Il s'agit de visualiser un modèle masses-interactions de fumée proposé par [HLV*96]. Le modèle de la fumée est simulé sur une station amont (PC biprocesseur) en temps réel. La fumée est composée de 300 masses. Nous nous trouvons donc bien dans une situation de fonctionnement limite pour l'écran d'épingles qui tourne alors à une vingtaine de Hertz. La figure 5 montre les résultats obtenus :

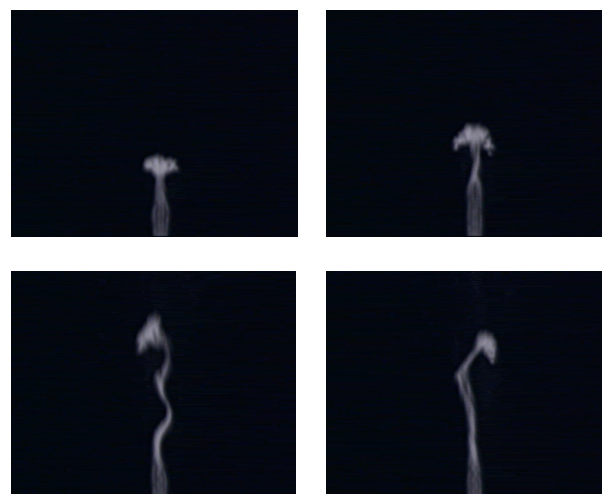


Figure 5 : *visualisation grâce à l'écran d'épingles dynamique d'un modèle masses-interactions de fumée à 300 masses*

8.2 □ Rendu « Bump mapping »

Dans les deux exemples suivants, l'écran d'épingles vient perturber les normales d'une surface pour créer, grâce au calcul de l'éclairage, un simulacre d'effet de relief.

8.2.1 Les sillons des bateaux

L'écran d'épingles est utilisé pour rendre les propagations sur la surface d'un lac (cf. figure 6). Il faut pour cela des paramètres qui permettent une propagation rapide (peu de viscosité et une forte raideur sur le voisinage). Les marqueurs sont placés au niveau de la ligne d'eau des bateaux (4 marqueurs par bateaux). La réfraction est simulée en appliquant une texture de fond. La réflexion est obtenue en effectuant une première passe projetant la scène à l'envers sur la surface de l'eau. La texture obtenue est ensuite stockée pour être envoyée au rendu final. Une « normal map » est générée en fonction de la hauteur des épingles. Elle permet d'aller lire à un autre endroit de la texture, pour simuler les déformations du reflet en fonction des vagues.



Figure 6 : Vue de haut des sillons provoqués par 3 bateaux.

8.2.2 Les traces des roues dans le sable

On dispose un marqueur sur chaque roue d'un véhicule. L'écran d'épingles va simuler le comportement d'un sol meuble sur lequel le véhicule se déplace. (cf figure 7). Pour cela, nous allons utiliser l'interaction plastique proposée par [CLH96]. Cette interaction présente un comportement d'hystérésis en modifiant elle-même sa longueur au repos lorsqu'elle dépasse un seuil d'élasticité, ce qui

permet de créer des traces permanentes. Les épingles ne sont pas en interaction entre elles car aucun effet de propagation n'est désiré.



Figure 7 : Véhicule sur sol meuble.

8.3 □ Déformations de surface

Dans l'ensemble de ces exemples, l'écran d'épingles contrôle la déformation d'une surface. Cette surface n'est pas nécessairement un plan, nous avons par exemple, déformé une sphère comme le montre la figure 8 en contrôlant les rayons des sommets.

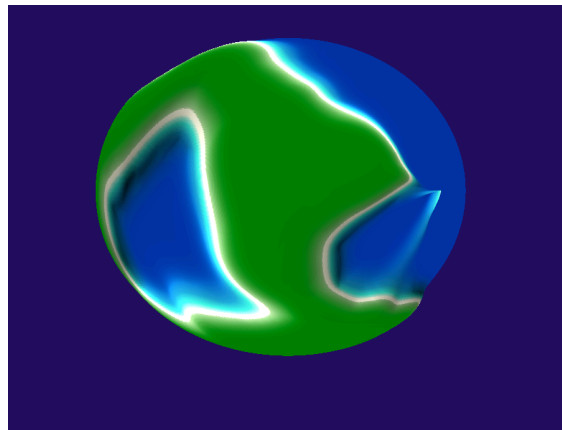


Figure 8 : Une sphère déformée par l'écran d'épingles.

En utilisant des interactions dont le comportement présente des irréversibilités, il est possible de créer des traces permanentes. On peut alors modéliser des phénomènes de morphogenèse comme la formation de profils de terrains (figure 9, 10) ou encore des fossilisations (figure 11,12).

Dans le cas des profils de terrains, l'utilisateur vient, à l'aide de la souris, « provoquer un

séisme» qui se propage. On obtient la configuration irréversible de l'écran d'épingles en supprimant toute élasticité entre la masse fixe et la masse mobile composant chaque épingle. Dans la figure 9, une carte d'altitude est appliquée à la surface déformée pour mieux rendre les différences de niveaux.

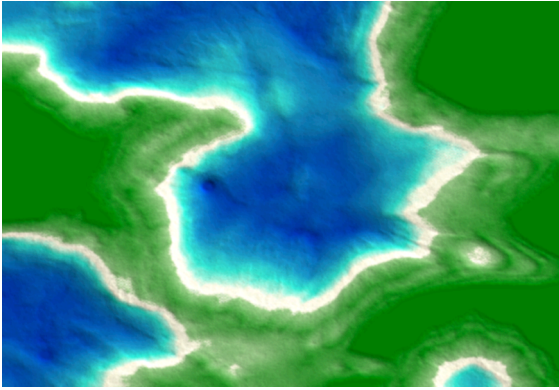


Figure 9 : Génération de terrain par propagation.

On peut aussi utiliser l'interaction plastique invoquée au paragraphe 8.2.2 pour modéliser des profils de terrains sablonneux (Figure 10). Dans ce cas, l'utilisateur vient lui-même « creuser » le sable :

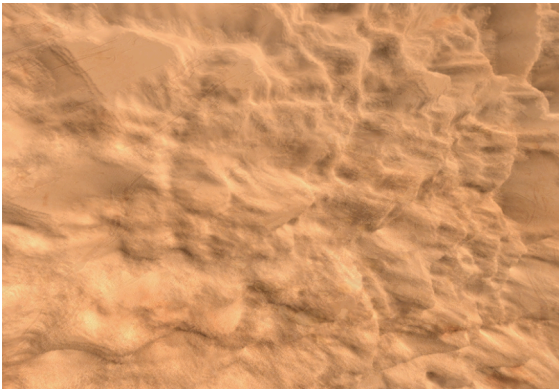


Figure 10 : Profil de terrain sablonneux.

Dans les figures 11 et 12, des sédimentations ont été réalisées par la même propagation que pour la génération de terrains, mais la propagation est beaucoup moins amortie (on joue sur une viscosité plus ou moins forte entre les épingles voisines). Une trace est ensuite effectuée par l'utilisateur dans le matériau résultant de cette sédimentation à l'aide de la souris. La permanence de la trace est rendu encore possible par une interaction hystérétique entre la masse fixe et la masse mobile de chaque épingle.

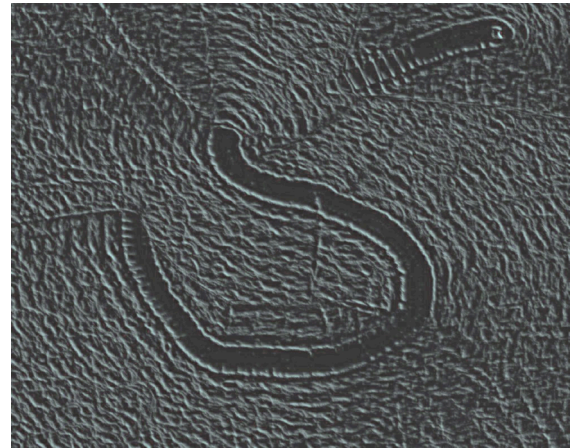


Figure 11 : Fossilisation obtenue grâce à la trace laissée par l'utilisateur.



Figure 12 : Une autre trace fossile dans un matériau qui semble plus dur.

9 Conclusion et perspectives

L'écran d'épingles peut être considéré comme une méthode d'extension spatiale de modèles particuliers, au même titre que les surfaces implicites. C'est en fait un générateur de surfaces implicites dynamiques puisqu'il génère un champ qui évolue dans le temps. Il peut aussi être vu comme un modèle physique en soit qui permet de modéliser un processus de gravure d'un genre nouveau.

Cet article a proposé une implantation temps réel de la méthode de l'écran d'épingles dynamique sur processeur graphique. On peut considérer que l'application arrive à ses limites pour un écran de 1100 par 1100 épingles et un nombre de marqueurs de l'ordre de 300.

L'implémentation a permis d'obtenir en temps réel des images qui ne pouvaient absolument pas l'être au moment où la méthode a été définie, vers 2002. Le temps réel a permis de l'incorporer à une interface interactive et

d'explorer de nouvelles possibilités notamment la modélisation en temps réel de profils de terrains ou de fossiles.

Plusieurs perspectives sont envisageables. L'une consiste à effectuer le travail théorique et l'implantation de la généralisation de l'écran d'épingles à un écran 3D (tableau 3D de phyxels). Il sera, de même, intéressant de connecter un système à retour d'effort à un certain niveau de la chaîne temps réel pour que l'utilisateur retrouve pleinement les sensations d'un graveur avec, en plus, une extension à des possibilités de comportements dynamiques nouveaux voire imaginaires qu'offre la méthode de l'écran d'épingles dynamique.

10 Remerciements

Nous tenons à remercier Mathias Paulin de l'Institut de Recherche en Informatique de Toulouse pour ses précieux conseils concernant la programmation sur cartes graphiques. Merci aussi à Julien Castet (ICA-INPG) et Jean-Loup Florens (ACROE) pour leur participation à la simulation temps réel de modèles de fumées, et le transfert des données de mouvement par réseau Ethernet.

11 Bibliographie

- [AAA96] Hommage à Alexandre Alexeïeff et Claire Parker, édité par l'A.A.A (atelier d'animation d'Annecy), 1996.
- [CL06] L. Chang, D. Liu. Deformable Object Simulation in Virtual Environment. Dans *Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications* ISBN:1-59593-324-7, pp 327-330.
- [CLH96] B. Chanclou, A. Luciani, A. Habibi. Physical Models of Loose Soils Dynamically Marked by a Moving Object. Dans *Proceedings of the Computer Animation*. Juin 1996, Genève, Suisse, pp 27-35.
- [GW05] J. Georgii, R. Westermann. Mass-Spring Systems on the GPU. Dans *Simulation Practice and Theory 2005*, Elsevier Science Juillet 2005
- [H97] A. Habibi. Visualisation d'objets très déformables. Relations Mouvement-Forme-Image. Thèse de l'INPG, Grenoble, 1997.
- [HL02] A. Habibi, A. Luciani. Dynamic Particle Coating. Dans *IEEE Transactions on Visualisation and Computer Graphics*, vol. 8, no. 4, Octobre-December 2002, pp 383-394.
- [HLV*96] A. Habibi, A. Luciani, A. Vapillon. A Physically-Based Model for the Simulation of Reactive Turbulent Object. « Winter School on Computer Graphics », 1996, pp 113-122.
- [KKK*05] J. Krüger, P. Kipfer, P. Kondratieva, R. Westermann. A Particle System for Interactive Visualisation of 3D Flows. Dans *IEEE transactions on visualization and computer graphics*, ISSN: 1077-2626/2005 Nov-Dec,11(6):744-56
- [KLR04] A. Kolb, L. Latta, C. Rezk-Salama. Hardware-based Simulation and Collision Detection for large Particle Systems. Dans *Graphics Hardware 2004*. Août 2004, pp 123-132.
- [LEC*06] A. Luciani, M. Evrard, N. Castagné, D. Couroussé, J.-L. Florens, C. Cadoz. A basic gesture and motion format for virtual reality multisensory applications. *Proceedings of the GRAPP conference*, February 2006.
- [LJF*91] A. Luciani, S. Jimenez, J.-L. Florens, C. Cadoz, O. Raoult. Computational physics: a modeler simulator for animated physical objects. *Proceedings of the European Computer Graphics Conference and Exhibition*. Eurographics'91, pp 425-436, Vienna, Austria, Elsevier Ed, Sep. 1991.
- [OLG*05] J.D. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Krüger, A. Lefohn, T. J. Purcell. A Survey of General-Purpose Computation on Graphics Hardware. Dans *Proceedings of Eurographics 2005, State of the Art Reports*. Dublin, Irlande, 29 août - 2 septembre, pp 21-51.

