

Original research

GPU-based real-time soft tissue deformation with cutting and haptic feedback

Hadrien Courtecuisse^{a,*}, Hoeryong Jung^{a,b}, Jérémie Allard^a, Christian Duriez^a,
Doo Yong Lee^b, Stéphane Cotin^a

^aSHAMAN Project, INRIA, France

^bRobotics and Simulation Laboratory, KAIST, Republic of Korea

ARTICLE INFO

Article history:

Available online 29 September 2010

Keywords:

Biomechanics

Soft tissue

Real-time

Finite element method

GPU

Haptic rendering

ABSTRACT

This article describes a series of contributions in the field of real-time simulation of soft tissue biomechanics. These contributions address various requirements for interactive simulation of complex surgical procedures. In particular, this article presents results in the areas of soft tissue deformation, contact modelling, simulation of cutting, and haptic rendering, which are all relevant to a variety of medical interventions. The contributions described in this article share a common underlying model of deformation and rely on GPU implementations to significantly improve computation times. This consistency in the modelling technique and computational approach ensures coherent results as well as efficient, robust and flexible solutions.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

During the past decade, a number of new minimally invasive surgical techniques have been introduced in an effort to reduce patient's pain, recovery time, and in some cases, operating time. One of the most important changes has been laparoscopic surgery, which brought new technologies into the operating room and created a distance between the surgeon and the patient. More recently, other minimally invasive techniques have been proposed, such as natural orifice transluminal endoscopic surgery, which can be considered as an evolution of laparoscopic surgery. As a new surgical technique, laparoscopy requires surgeons to acquire new skills, and adapt to changes from conventional open surgery (e.g. amplified tremor, diminished tactile sensation, loss of depth perception). This has been a motivation for a number of works in the field of surgery simulation, real-time deformable models, or haptic rendering (see for instance Marescaux et al. (1998), Picinbono et al. (2000), Brown et al. (2002), Forest et al. (2004), or Harders (2008)). As a result, it has been demonstrated that the use of computer-based simulators can lead to a more effective and systematic training, thus providing objective assessment of technical competence (Seymour et al., 2002). Other studies also show that the skills learned thanks to a simulation can be transferred into the operating room (see Grantcharov et al. (2004) for instance).

More recently, a new thrust has appeared with the development of medical imaging techniques which could make it possible to develop patient-specific simulations. Such simulations could be beneficial in certain situations, when the patient presents a rare pathology or when the best surgical strategy is unclear. In this case the simulation can be used as an efficient planning tool, however more realistic models of the patient's anatomy and behaviour are required. A final step concerns the use of augmented reality systems for image guided surgery, to improve the accuracy and limit the adverse effects of surgery. In order to register pre-operative data on the real organs (to visualize the targeted area while the procedure is progressing for instance) accurate, real-time biomechanical models are needed, but their interaction with medical devices also needs to be modelled. Such interactions not only involve tissue manipulation, but also tissue dissection.

In this context, the development of fast algorithms to compute the deformation, contact response, cutting and haptic feedback of soft tissues could enable a number of the aforementioned applications (see Fig. 1). More specifically, when considering requirements for realistic interactive simulations of medical procedures, several elements seem mandatory: anatomical models and tissue properties need be patient-specific and obtained without complex additional procedure; soft tissue behaviour needs to be realistic and demonstrate a predictive capability, yet it should be compatible with real-time computation; interactions with the surrounding anatomy and with medical devices need to involve advanced contact models that can be computed in real-time; the different types of dissection performed on soft tissues should be simulated; and finally realistic

* Corresponding author. Tel.: +33 689663283.

E-mail address: hadrien.courtecuisse@inria.fr (H. Courtecuisse).

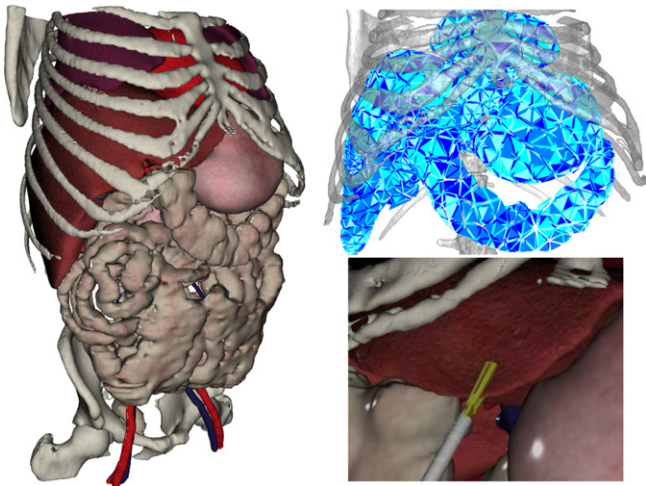


Fig. 1. GPU-based simulation of multiple anatomical structures reconstructed from a CT dataset. The simulation is based on a series of methods proposed in this article and involves implicit co-rotational FEM, frictional contacts, and cutting with haptic feedback.

visual and haptic feedback should be provided to create a higher level of immersion, in particular during training sessions.

1.1. Previous work

Among the numerous publications in the field of biomechanics, real-time deformable models, collision detection, contact modelling or haptics, few methods have been proposed to address all (or at least a majority) of the requirements listed above. Among the existing approaches which at least partially aim at this objective, we can cite methods based on spring-mass networks, methods based on linear elasticity, and explicit finite element models for non-linear materials.

Mass-spring networks: one of the most popular methods for real-time computation of the deformation of soft objects is based on spring-mass networks (see [Montgomery et al. \(2002\)](#) for instance). In addition, such an approach is well suited to benefit from GPU-based acceleration, as demonstrated by ([Sorensen et al., 2006](#)). Another obvious advantage of using spring-mass networks is their ability to handle topological changes at a reduced computational cost. Yet, although they are quite simple to implement and very fast to compute, spring-mass methods fail to properly characterize soft tissues deformation as they introduce artificial anisotropy through the choice of the mesh, and make it difficult to relate spring stiffness to material properties such as Young modulus. A recent work by [Delingette \(2008\)](#) has shown that such a relationship can be established in the case of hyperelastic constitutive laws. In this case, it is possible to compute springs stiffness on triangular and tetrahedral meshes relating to St Venant-Kirchhoff materials.

Linear elasticity with a finite element method: there are several instances in the literature of real-time simulation which rely on linear elasticity. Initial work (see [Bro-Nielsen and Cotin \(1996\)](#), [Cotin et al. \(1999\)](#) or [James and Pai \(1999\)](#)) made the assumption of small displacements and often relied on precomputed responses based on Finite Element Models (FEM) and related techniques. The main benefit of approaches based on pre-computation is the important speed up that can be obtained, allowing not only real-time deformations but also haptic feedback. However, two main issues arise from such approaches, the first one is the impact of cutting on the pre-computed response, the second one is the inherent limitation of the small strain assumption. Some solutions have been proposed to solve the first problem, using for instance numerical techniques to update a pre-inverted stiffness matrix

([Lee et al., 2005](#)). The small strain limitation is known to lead to incorrect results (in particular when rotations are applied). An elegant solution for this problem was introduced by [Felippa \(2000\)](#) and is known as the co-rotational method. Other methods have been proposed to extend the idea of pre-computation, but in the case of geometrically non-linear deformations ([Barbič and James, 2005](#); [Mahvash and Hayward, 2004](#)).

Explicit non-linear FEM: another strategy to deal with real-time computation of soft-tissue deformations when relying on a finite element approach is to base the computation on an explicit integration scheme, as proposed by [Taylor et al. \(2008\)](#) for instance. The main advantage is that the solving process only involves the mass matrix, which is diagonal if mass lumping is used. Thus, the equations of motion can be decoupled and each degree of freedom can be solved independently. The solving process is then very quick and its parallelization is quite straightforward ([Comas et al., 2008](#)). Explicit integration methods are particularly well suited for applications such as real-time non-rigid registration of brain shift during surgery ([Joldes et al., 2009b](#)). In this case, only the steady state of the deformation is sought and the mass can be artificially increased in order to deal with stiff materials ([Joldes et al., 2009a](#)).

However, our goal is different. We aim at simulating complex, user-controlled interactions between medical devices and anatomical structures that are often deformable. This leads to complex interactions which are unpredictable and discontinuous in time (non smooth contact problems for instance). Additionally, such simulations are dynamic by nature, and may involve haptic feedback and topological changes. Based on these requirements, the choice of an implicit integration scheme offers the best tradeoff between robustness, stability, convergence and computation time, in particular when combined with a GPU implementation. Although this choice leads to added difficulties compared to an explicit approach, we show in this paper that it can be at the center of a framework which addresses all the requirements of interactive simulations.

1.2. Many-cores architectures

In recent years, the computational hardware available in high-performance workstations shifted from increasingly efficient but complex sequential computational units, to smaller units, each not much faster than previous generations, but duplicated to be able to execute more threads in parallel. This evolution has taken place both in the design of CPUs and recent Graphics Processing Units (GPUs). The latest generation of GPUs contains hundreds of computation units (240 in NVIDIA Geforce 285 GTX, 1600 in ATI Radeon 5870). This radical architectural change has important consequences on the type of algorithms which are applicable in interactive simulations.

In terms of programming, general purpose computations on GPUs initially required the use of graphics-oriented libraries. Recently, the two major GPU vendors released general programming APIs, CUDA ([Nickolls et al., 2008](#)) and CTM ([Peercy et al., 2006](#)) which provide direct access to the underlying parallel processors of the GPU, as well as full instruction sets, such as double precision computations and write operations at arbitrary locations. In 2009, a multi-vendor standard, named OpenCL ([Munshi, 2008](#)), was released, with a programming model very similar to CUDA. In the following sections we present a series of algorithms which have been implemented in CUDA. It is relatively straightforward to also implement them in OpenCL if necessary.

1.3. Summary of the contributions

In this article we introduce a suite of methods which rely on a common underlying model to obtain a coherent answer to the

Download English Version:

<https://daneshyari.com/en/article/2070577>

Download Persian Version:

<https://daneshyari.com/article/2070577>

[Daneshyari.com](https://daneshyari.com)