A Levels-of-Precision Approach for Simulating Multiple Physics-based Soft Tissues

Daniele Fernandes e Silva[1], Iago Berndt[1], Rafael P. Torchelsen[2], Anderson Maciel[1]
dfsilva@inf.ufrgs.br, iago.berndt@inf.ufrgs.br
rafael.torchelsen@inf.ufpel.edu.br, amaciel@inf.ufrgs.br
[1]Instituto de Informatica - Universidade Federal do Rio Grande do Sul
[2]CDTec - Universidade Federal de Pelotas

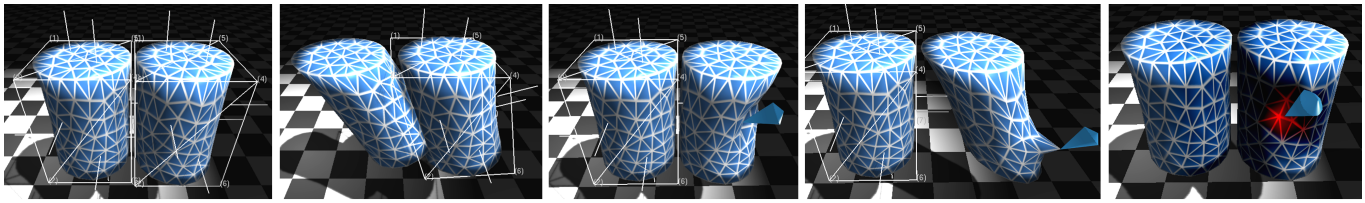

Fig. 1. Simple scenario illustrating important features for surgery simulation: (a) rest position; (b) collision interaction between two bodies; (c) poking and (d) picking interactions using a haptic cursor; and (e) heat diffusion. Presence of a bounding box indicates the use of a lower level-of-precision.

***Abstract*—Computer simulation of surgical environments is always oversimplified in terms of physical behavior due to the complexity of the tissues and interactions involved, which cannot be fully simulated in real time. To better manage this trade-off between efficiency and effectiveness, we present a hybrid and adaptive environment that combines a set of methods to achieve higher accuracy and performance. Our approach merges physics-based deformation methods (Finite Elements and Mass Spring) with a non-physical method (Green Coordinates) to approximate more coarsely the behavior when the focus of the interaction is away, and more precisely during direct interaction. We experimentally demonstrate that the computational complexity of the simulation with our method does not increase with the number of objects being simulated. With our approach, a virtual surgery environment with many dynamic organs can be computed at interactive rates for the first time.**

***Keywords*-computer animation; finite element method; mass-spring; green coordinates; minimally invasive surgery; physics-based animation; haptic feedback; heat diffusion**

I. Introduction

Minimally invasive surgery (MIS) includes a set of procedures which aim at maximum preservation of anatomy and minimal aggression to the organism [1], [2]. It is more complex to perform than conventional surgery, but it includes benefits, such as: reduced surgical scar, less pain, lower complication rates, quicker recovery, shorter hospital stay, and higher patient comfort. To obtain success with these procedures, students and professionals must receive specific training. Training involves refining motor skills to manipulate objects through long slender tools that are inserted through small portals open on the patient's body together with a camera. Besides, it involves the acquisition of knowledge about the sequence of steps to be carried out for each clinical indication. As surgeons perform MIS looking at a video monitor, virtual simulators are often used in many steps of the training before a trainee can perform real surgeries [3], [4].

Differently of animation movies, games and engineering simulations, virtual simulation for medical training and planning requires a finer degree of complexity in order to appear realistic while still being interactive. Major efforts have been made in the last decade to improve the quality and efficiency of the simulations [5]. However, this is a multifold problem where both appearance and behavior of living tissue and surgical tools must be modeled to the detail.

Typical algorithms used to render computer graphics in games and movies have been applied to organs with relative success. They are based on simplified equations defining light interaction with the models and artistic textures, where different materials can be configured for a surgery scenario. In the behavior side, physics-based algorithms are often used to provide deformation resulting from contacts between virtual instruments and soft tissues. As each organ has specific characteristics, different parameters of stiffness, elasticity, and viscosity must be carefully set to achieve a plausible behavior.

Moreover, surgery simulation must be interactive, and can greatly benefit from haptic feedback [6]. Haptic simulation, however, imposes severe requirements on the update frequencies (around 1000Hz) of the underlying model to be able to deliver a smooth and accurate feedback. Therefore, to find a compromise between accuracy and quality in surgery simulation one must direct the computational power available to where it provides more benefit. In terms of graphics, the current practice is to use methods similar to those used in

video-games, which can render at 30-60 frames per second (FPS). That is, in general, pleasing to the eye. But when it comes to behavior, the problem is more challenging. To ensure plausible deformations, the use of complex and computationally demanding physical models is necessary. And they must run hundreds of FPS to allow haptic interaction.

In this context, the current practice in commercial simulators is to simulate only one specific organ or even a portion of the organ that is more centrally related to each specific medical procedure. The portions of tissue that are simulated and allow interaction are usually modeled with either very simplified geometry based approaches, or with mass-spring-damper (MSD) systems, which are dynamic but not accurate, or with linear finite elements method (FEM), which is accurate but more time consuming [7]. This means that while the full computational power is spent with the simulation of one organ even when the surgeon is not touching it, all surrounding organs, nerves, blood vessels and other important anatomical landmarks are depicted as fixed textures, or not depicted at all. This limits the possibilities to vary the intercurrences during the medical simulation.

In this paper, we introduce a strategy to produce richer simulation scenarios without increasing the computational power required. This strategy is the main contribution of the paper. It consists in simulating different objects of the same environment with different levels of precision *on demand*, i.e., increasing the precision on the fly where it is required and reducing it in locations where it can be allowed. The assumption is that some tissues, in some moments, do not require a highly precise modeling, while others do. So, we propose a hybrid and adaptive environment where important tissues, i.e. the ones that require more precision, can be modeled by a more complex physical model, while the not so important tissues can be modeled by a faster and less precise method. The definition of what is important is dynamically defined and redefined according to the user interaction. We consider that tissues that do not suffer direct interaction are momentarily less important, and can be approximated with a faster method. This allows us to concentrate our computational resources on the organ or tissue that is being manipulated and thus has the attention of the user's eyes, while the peripheral tissues still undergo deformation but with less precision. The savings in computational power not only allow less powerful computers to run the application, but also contributes by enabling us to simulate a virtually unlimited number of objects.

To demonstrate the concept, we implemented and evaluated a hybrid and adaptive environment that simulates surgical procedures with many organs in real time. For completeness, our environment provides haptic feedback, and features common on surgeries, such as poking, picking and a heat transfer model.

II. Background and Related work

Advanced minimally invasive surgical techniques are more complex than traditional surgery. Due to that complexity, training MIS is very time and resource consuming. Training simulators brought a great help in developing and spreading the use of these techniques. There is a special program that focuses on those training: "The Fundamentals of Laparoscopic Surgery" (FLS). This educational program was developed by the Society of American Gastrointestinal and Endoscopic Surgeons (SAGES) for teaching and evaluating cognitive and technical skills, and surgical decision-making, in a scientifically accepted format.

The primary goal in training laboratory is to know and learn how to properly manipulate the equipment. In practice, the surgeon manipulates a three-dimensional equipment, inside an enclosure environment, visualizing a two-dimensional field through a display. Training is, therefore, necessary to familiarize this change of field of vision beyond specific materials.

According to the FLS program, the surgeon has first to study and develop skills on manual tasks (normally acquired in a box training). The goal is to manipulate the forceps and dissectors without direct visualization, using a mirror or a webcam to provide dissociation between hands and eyes [8], [9].

Virtual reality training has provided advantages over other techniques, such as box training, animals, or human cadavers. The computer simulators are able to model a full surgical procedure, calculate metrics to evaluate the training and score them, without the supervision of a specialist surgeon [10], [11].

Currently, several simulators are available as commercial products. However, they present several limitations, both on graphical and physical aspects of the simulation. Besides, a number of research works have been published focusing on physics-based methods for surgery simulation. Some aim specifically at the deformation problem [12], [13], others at the interaction with tissues [14], [15], and others at the visual and other aspects of organs and instruments [16]–[18]. We review some of these works in this section. The simulation is defined as a modulated system, where, for each task, the system uses different approaches to describe the visual and physical interaction. We present some research works which address the problems in a surgical simulator.

A. Deformable bodies

In a surgery, many organs and tissues are described differently based on their own characteristics. In order to simulate these properties, it is necessary to model such tissues using a physical method to ensure a realistic behavior.

Duan et al. aim at modeling the contact forces among deformable objects for tetrahedral meshes [19]. This ensures the preservation of volume and properties of the object. Using an implicit numerical integration scheme and appropriate constraints, a post procedure is performed to avoid the so-called "super-elastic" effect, which makes the system non realistic.

Sulaiman et al. presented an optimization of a physically-based model [13]. The configuration of the parameters of the model that governs the deformable object, i.e. the liver, allows a more realistic behavior. The mass-spring model used is optimized in order to adjust these parameters without interfering with the stability of the system. In conclusion, this liver-tissue model is more suitable for real-time interaction

with lower computational cost, being more accurate, realistic and acceptable to be used in the near future.

Morooka et al. approach the problem of navigation system, in surgery context, using a FEM based analysis [12]. Considering the tissue deformation by biomechanical behavior, their method describes the deformation by a real-time nonlinear FEM analysis using neural network. In addition, their approach uses several markers put on the surface of the tissue, which are used to generate the network and estimate the deformation based on the position of the markers.

We have discussed some recent approaches for body deformation. Note that each system works using a single method, which in turn brings advantages and drawbacks. Our method in the other hand, provides an environment that can take the advantages of each method, while making them to interact in a transparent way between each other.

B. Surgery interface and task training

Medical applications are usually very expensive, due to the devices used for user interaction. Recent researches aim at reducing costs by using alternative ways, such as accessible sensors. Queiros et al. propose a motion tracking system, that monitors the movement performed by the specialist and assists the user/trainee in the manipulation of the 6 DOFs of the instrument [16]. It uses a set of inexpensive sensors, such as accelerometer, gyroscope, magnetometer, and flex sensor, attached to specific laparoscopic instruments.

Park et al. approach the problem of device cost in a different way [17]. Their work uses the Microsoft Kinect 3D camera to detect the user's hand and arm. In this way, the application can correlate with the surgical instruments, designing an environment to describe a real MIS procedure.

In a suturing case, where the surgeon uses a specific tool for manipulating a string, this task presents a common problem, it models the suture line and simulate the knot by the procedure. Marshall et al. present a simulator system that assists residents in the development of skills, like manipulating surgical graspers and interacting with deformable tissue [18]. A set of algorithms is used to solve problems of the deformation method, collision detection, and haptic feedback.

Currently, there are a some surgical applications being commercialized on the market, such as LapMentor [20], LapSim [21], and daVinci [22]. All these private applications assist novice surgeons to develop and train their skills for a real situation.

C. Discussion

The major goal in medical applications is to achieve realistic appearance in real time. However, precise modeling methods imply high-computational cost. Surgery simulation carries the problems previously discussed, such as choosing the physically-based method to model the behavior of each organ, collision detection, interaction between tissues and with a external force (e.g., applied by an user), and haptic feedback to ensure tactile sense. Each of these features reduces the application performance, when it demands more and more realism. Thus, some features need to be put aside in order to obtain a desirable update rate.

In addition, haptic feedback requires high frequency update, which is not achievable in more complex applications. For this reason, most of the commercially available applications do not include this feature, using only stable models to simulate the behavior of different tissues from the human body.

III. Levels-of-precision deformation approach

Our approach minimizes the gap between accuracy and performance in surgical simulation. We present a hybrid and adaptive environment system capable of achieving high accuracy in real-time. Hybrid because it can use different deformation methods together, and adaptive because it can easily switch from one method to another.

Our proposal combines a physics-based deformation method, which is used for simulating tissues under direct interaction, and a non-physically based one, focused on performance, used when there is no direct interaction.

A. Overview

Taking a surgery scenario as an example, we use FEM for modeling tissues that are the main subjects of the procedure, and the MSD for modeling objects with less relevance on the simulation. These physics-based deformation methods, expressed by the theory of elasticity, ensure a realistic behavior to tissues deformation, one more accurate than another. When the organ is not under direct interaction, the simulation can hand over a little accuracy by using a non-physically-based method, represented by a generalization of barycentric coordinates, the Green Coordinates (GC). This method is used when no interaction is detected and the object is at rest, i.e. in an equilibrium status.

In our system, all objects start by being simulated using the fastest method: the GC. This ensures greater performance as it is less resource intensive (computationally). When user contact occurs on any object, the system triggers a state change, and this specific object assumes an accurate method, either MSD or FEM, depending on the importance of the tissue. Even if the user stops interacting with the body, our system will still use the more precise method until the deforming displacements converge to zero, reaching the equilibrium of the object (Fig. 2).

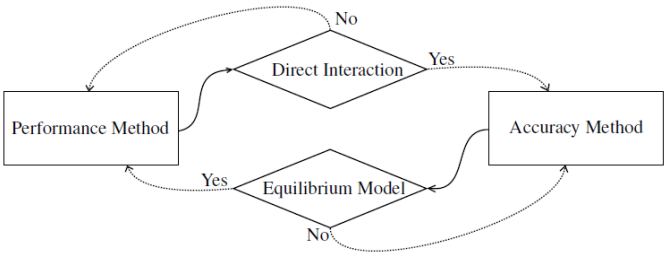

Fig. 2. Diagram of state changes. When a user interaction occurs, there is a context switch, making the system to use the more precise method until the model goes back to its equilibrium state, returning to use the faster method.

Our system could be adapted to use any deformation method. We opt by using FEM and MSD for high accuracy, and GC for high performance, due to their simplicity and popularity. We briefly describe them in the following subsections.

B. Finite Elements Method

FEM is a method that approximates the solution of boundary-valued problems, in the case of 3d solid modeling, it is usually used to solve the linear elasticity problem [23]. We use this method to solve the K part of a linear motion equation (Equation 1), where M, D and K are mass, damping and stiffness matrices, respectively, and f are external forces.

$$M\ddot{u} + D\dot{u} + Ku = f \tag{1}$$

Firstly, to create the K matrix, we model the domain Ω (which is subdivided in primitives) accordingly to the elasticity theory, using the minimum potential energy principle Π composed by energies which act over the domain (Equation 2). These energies are represented in matrix form as $\delta = Bu$ (strain) and $\sigma = C\delta$ (stress) [24].

$$\Pi(u) = \frac{1}{2}\iiint_{\Omega} \delta^T \sigma dx \tag{2}$$

For our tetrahedral mesh we formulate three displacement components $u(x, y, z)$, $v(x, y, z)$ and $w(x, y, z)$ defining the local coordinates x, y and z, respectively. These coordinates are described as a linear combination of the shape function N_i. For each node i, a shape function is computed as:

$$N_i(x, y, z) = a_i + b_i x_j + c_i y_j + d_i z_j \text{ for } j = 0, 1, 2, 3 \tag{3}$$

For each shape function we obtain 4 piecewise polynomials j, applying the properties given by Equation 4 [25].

$$N_i(x, y, z) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

After that, we can re-write the potential energy, for each subdivision of the domain e, as a local stiffness matrix K^e, based on the strain and stress energies (Equation 5).

$$K^e = B^{eT} C B^e V^e, \tag{5}$$

In the strain energy, the components are a Jacobian matrix B and a displacement vector u. And the stress energy, according with Hooke's Law, is defined by the strain vector and a symmetric material stiffness matrix C [24], [26], [27].

Thus, we assemble these local matrices to a global matrix ($K = \sum K^e$) and apply Dirichlet boundary conditions to solve a possible singularity problem, and our system can return a single solution. Finally, with the k part of the motion equation, we obtain the mass and damping matrices through the [27].

Although FEM is known by high accuracy, it is not advised when time is a concern. Thus, in a scenario with many tissues, only the organs with more relevance, i.e the ones that have more importance for the simulation, are modeled using FEM.

C. Mass-spring Model

MSD is described as an ordinary differential equation system which uses mathematical methods (explicit or implicit) to solve numerical approximations [28]–[32]. In this method, the object is discretized in nodal points represented as masses, each one with information of position, velocity, and acceleration values. The springs are the connections between neighbor nodal points, each spring also contains specific attributes, such as its stiffness and rest size value.

We solve the linear system using the classical Runge-Kutta technique, often referred as Runge-Kutta 4th order (RK4). RK4 presents a good trade-off between computation time and precision of the approximation. Moreover, with better approximations, we can use larger step sizes.

Each spring in the system follows the elasticity concept, using the Hookes law (Equation 6) to ensure an equilibrium point between the two masses it connects.

$$f = -kx \tag{6}$$

The Newtons Second Law (Equation 7) is used on each mass of the object for computing the deformation movement.

$$m_i \ddot{x}_i = -\gamma_i \dot{x}_i + \sum_j g_{ij} + f_i \tag{7}$$

where, m_i is the mass of the point i, and x_i is its position. The right side encodes the forces applied to the masses. The first term ($-\gamma_i \dot{x}_i$) is the damping force, which depends on the velocity of the point. The second term ($\sum_j g_{ij}$) is the contribution of the forces of the springs that are connected to the point i, and f_i are the external forces.

Although this method presents some problems such as difficulty of parametrization and instability, it is much faster than FEM. We intend to use MSD on less relevant structures in our scenario, without having huge impact on accuracy.

D. Green Coordinates

GC is a cage-based technique represented by affine combinations (Equation 8). Its main advantage is that it can perform deformation preserving the object shape [33].

$$p = F(p, C) = \sum \phi(p) v_i + \sum \psi_j(p) S_j n(t_j) \tag{8}$$

In Equation 8, p is each vertex of the object, and each one will be written up as a combination of the cage vertices (v_i). The shape-preserving property is represented with cage normals ($n(t_j)$) and a scaling factor (S_j) to describe deformation.

This method does not have any movement formulation to animate. Thus, to generate deformation on time, we use the cage that encloses the object modeled with a MSD method. Although the MSD be a physics-based time-consuming method, the complexity of this method is $O(n)$ and for our system, it means less computation cost, once it is used in a simple mesh.

There are a lot of generalized barycentric coordinates, such as the Mean Value Coordinates (MVC) [34], Harmonic Coordinates [35], and others [36], [37]. What motivates the

choice of GC was the straightforward extension of it to 3D space, and specially its shape preserving property [33]. As mentioned earlier, the model cage is defined as a rectangular prism based on the minimum and maximum values of x, y and z. Although we have achieved good approximations using this simple cage, we believe that generating a more refined cage could improve the accuracy of the approximations. Some methods are capable of performing automatic cage generation [38], [39], but it is still uncertain if it is worth to trade performance for precision in this case, where the GC intention is to be as fast as possible.

E. Additional Components

The above description shows that our environment is much more than choosing the method for deformation, it also detects interaction, changing the current model in real time. The deformation method used to simulate each tissue changes during the simulation depending on the organs (objects) that the user is interacting with. Therefore, it must be placed in the context of a complete simulator, with other components such as haptic interaction and collision detection for its complexity and implications to be fully understood. Also in the surgical context, the use of electrocautery is indispensable in most procedures, so we also need to simulate the same process for our surgical training system.

IV. Experiments

We have implemented in C++ all the deformation methods: MSD, FEM, and GC, and all the phantom interactions allowed in the system (such as collision, picking, and heating). The deformation computations are performed in a dedicated thread, which is asynchronous with the one of the environment rendering. This ensures that our system does not suffer from low FPS, but the models may have low update rates for high-cost deformation computations (in the case of high cost models using FEM).

We present two test scenarios, one using simple shapes, and another using more complex models. All the measures were taken on a 3.3 GHz i5 CPU with 8 GB of RAM. For each measure shown in the graphs, we have computed a robustness average (identifying and removing the outliers) over five evaluations.

A. Cylinders Scenario

In the first scenario (Fig. 3), we evaluate the deformation behavior of two cylinders, **Model A** and **Model B**, each one with 252 nodes, 336 triangles, and 939 tetrahedrons. These cylinders will be placed side by side, and we are going to perform evaluations on these objects simulated with different combinations of deformation methods.

For this scenario, the parameters for FEM model were *Yong's modulus = 600*, *Poisson's ratio = 0.0023*, *mass = 30*, *damping = 1.0*, and *time-step = 0.02*. For MSD model, the parameters were *stiffness = 700*, *mass = 120*, *damping = 0.4*, and *time-step = 0.02*. For GC approximation, its MSD parameters were *stiffness = 50*, *mass = 20*, *damping = 0.3*, and *time-step = 0.02*.

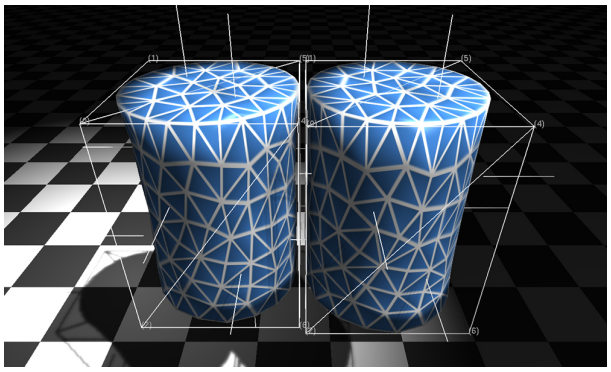

Fig. 3. The first scenario consists of two cylinders placed side by side. Each object is modeled using different combinations of deformation methods.

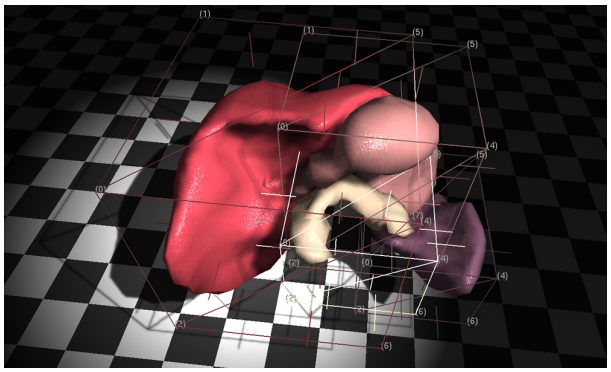

Fig. 4. The second scenario consists of multiple objects to simulate the behavior of real tissue organs. In this scenario we used shapes to describe the organ tissues, simulating a surgical environment.

B. Organs Scenario

In the second scenario (Fig. 4), we loaded four complex objects: a liver, a spleen, a stomach and a pancreas. Technical information, such as the number of nodes, can be found in Table I. Tetrahedral meshes were created using Tetgen from triangle meshes reconstructed from the Visible Human photographic dataset to simulate specific organs. All the physics parameters were empirically chosen to simulate an approximate behavior from real organs. The parameters for FEM were *Yong's modulus = 140*, *Poisson's ratio = 0.23*, *mass = 10*, *damping = 10*, and *time-step = 0.01*. For GC approximation, its MSD parameters were *stiffness = 50*, *mass = 20*, *damping = 0.3*, and *time-step = 0.02*. For this scenario, we have disabled the gravity of the environment, as it would hamper the expected behavior of the simulation due to the lack of fat and connective tissue (e.g. ligaments, muscles).

TABLE I
SECOND SCENARIO: SIMULATES A SURGERY ENVIRONMENT PROVIDING REALISTIC INTERACTION AND VISUALIZATION OF THE ORGANS.

Shape	Vertex	Tetrahedrons	Triangle surface
Liver	504	1578	1004
Spleen	218	661	432
Stomach	291	890	578
Pancreas	229	657	454

TABLE II
EFFICIENCY EVALUATION OF FIRST SCENARIO UNDER SIMPLE USER INTERACTION.

	Model A		Model B	
Method	FEM	GC	FEM	GC
Time (ms)	3.65	0.01	0.00	0.10
Method	FEM	GC	MSD	GC
Time (ms)	3.61	0.01	0.00	0.15
Method	MSD	GC	FEM	GC
Time (ms)	0.31	0.01	0.00	0.14
Method	MSD	GC	MSD	GC
Time (ms)	0.31	0.01	0.00	0.14

TABLE III
EFFICIENCY EVALUATION OF THE CYLINDERS SCENARIO UNDER USER INTERACTION THAT AFFECTS BOTH OBJECTS.

	Model A		Model B	
Method	FEM	GC	FEM	GC
Time (ms)	3.35	0.02	0.00	0.10
Method	FEM	GC	MSD	GC
Time (ms)	3.23	0.02	0.00	0.12
Method	MSD	GC	FEM	GC
Time (ms)	0.31	0.01	0.00	0.11
Method	MSD	GC	MSD	GC
Time (ms)	0.31	0.01	0.00	0.12

V. RESULTS

In this section we describe and discuss an assessment of our approach for efficiency and effectiveness. In the following figures, we will be showing graphs where the y-axis represents the time (in milliseconds) of the x^{th} iteration, represented by the x-axis. We are going to evaluate the times for computing each iteration.

A. Efficiency Evaluation

We measured the times to compute one step of simulation for each method in two situations. First, when we apply a force from user interaction, and then when an object collides with the other.

1) Force from user interaction on one object: We evaluate the behavior of each combination of deformation methods under user interaction. We recorded a simple interaction, where all models start, in their rest states, with GC. After that, the user interacts with Model A, pushing the object for some time, and then, release.

Table II shows time results for this trial. These times only account for measuring the computation of collision with the virtual scalpel and for the deformation method, not including the collision detection between objects or heat diffusion.

We notice that the times taken by the Model B is minimal when compared with Model A (we show each pair with the same scale factor). The steps seen in Model A graphs are due to the scalpel touching the object, forcing a state change from the faster method (GC) to the precise method (FEM/MSD).

2) User interaction causes objects to collide: We recorded an interaction with the Model A in a way that it collides with Model B, i.e., we perform an operation that affects both objects (one directly and the other indirectly).

Table III shows the time results for this trial using the same setups used on the previous subsection. In those times, we only considered the time for detecting the collision of the scalpel, and the time for computing the deformation. Later in this section, we present times taken to detect object collisions.

The most interesting observation here is that the computation load is the same between Table II and III, despite the fact that in Table III both objects are deforming.

3) Additional Evaluation: After seeing how our environment behaves when using different combinations of deformation methods for modeling simple objects, we present times for the more complex organs scenario. The parameter specification for the organs was done only for the FEM model, so we are only presenting times using this deformation method.

TABLE IV
EFFICIENCY EVALUATION OF SECOND SCENARIO (ORGANS) UNDER USER INTERACTION AFFECTING ALL OBJECTS.

	Method	Time (ms)
Liver	FEM	7.69
	GC	0.06
Spleen	FEM	0.00
	GC	0.06
Stomach	FEM	0.00
	GC	0.07
Pancreas	FEM	0.00
	GC	0.06

As the resting location of the organs causes them to touch, it is impracticable to perform a trial where an interaction does not result in a collision. Thus, this trial is based on the user interacting with the liver, and this action propagated to other organs. Table IV shows the times for computing the iterations of this scenario (notice that the y-axis scale is different for the liver). For larger meshes, the gap between the FEM and the GC becomes even higher, which makes unfeasible to keep modeling all the objects using FEM and justifies the use of GC whenever possible.

For comparison, we also performed the same test without GC. Table V shows the times for this case. Notice that the times for the spleen, stomach, and pancreas are more than 50 times lower when using GC.

B. Effectiveness Evaluation

It is widely accepted that it is unfeasible to find a parameter setting that will make an FEM and an MSD models behave in the exactly same way. Thus, our goal is not to compare each other. Instead, we evaluated how close our GC (MSD-based) approximates the full MSD and FEM simulations. As explained, the GC are only used in objects that are not under

TABLE V
EFFICIENCY EVALUATION OF SECOND SCENARIO (ORGANS) UNDER USER INTERACTION AFFECTING ALL OBJECTS **WITHOUT** GC

	Method	Time (ms)
Liver	FEM	16.41
Spleen	FEM	2.89
Stomach	FEM	5.29
Pancreas	FEM	3.21

direct interaction, so we are willing to tolerate some errors in those approximations.

In order to measure the effectiveness of our system, we have recorded the mean difference (based on all the nodes) of the object being modeled by a physical deformation method with the same object using GC, both under the same interaction. As mean squared error (MSE) and other error measures are only used for comparing two or more approximation methods (and we are just interested in evaluating how plausible a single approximation is), we thought this mean difference would be more useful. In order to obtain a glimpse of intuition about the errors, we took notes of the dimensions of the cylinders (after applying gravity): *diameter=1.24* and *height=1.72*.

Fig. 5 (left) shows the average and maximum errors for each iteration, using FEM as the ground truth, and GC as the approximation, while (right) shows these errors when using GC for approximating MSD. We decided to also record the maximum error, so when using a deformation like FEM (which is known for performing local deformations), we can evaluate the worst node approximation on each iteration. By considering the dimensions of the cylinders, we could say that the maximum error when approximating FEM was 4% of the size of the smallest dimension (width/depth), while when approximating MSD the maximum error was 8.1% of the smallest dimension. In our opinion, this is a low error compared to the performance improvement of using this approximation.

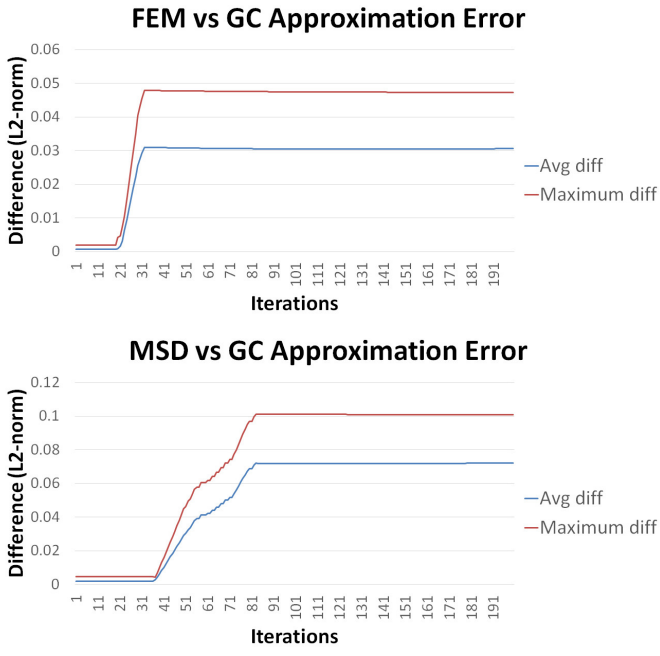

Fig. 5. Effectiveness evaluation of cylinders scenario. On the top, comparing the error when using GC to approximate FEM, and on the bottom the error when using GC to approximate MSD. The dimensions of the cylinders (after applying the gravity) are: *diameter=1.24* and *height=1.72*. We have recorded the average and maximum errors for each iteration.

C. Discussion

The implementation of all the presented features (collision, interaction, deformation computations, heating diffusion, and others) does not make use of any parallel optimization. As many of the calculations for solving these problems could be computed concurrently, we believe that the performance of our system could be greatly improved by parallelizing some steps.

The combination of a physical and a non-physical method allows a great performance vs. accuracy result. Although it directly affects the precision of the deformation, we managed this trade-off allowing our level-of-precision approach to decide when an accurate method is necessary.

The MSD parameters for the GC cage were chosen empirically. We struggled to find a correlation between the full deformation model (FEM/MSD) parameters and the ones used in the MSD-GC cage, but this has been proven to be a hard problem, even for simple objects such as cylinders.

VI. Conclusion

We presented a hybrid and adaptive environment to simulate mutiple deformable objects in real-time. Our method shrinks the gap between performance and accuracy by the combination of different (physical and non-physical) deformation methods. This provides a more complete scenario with the potential to increase the user perceived realism. The physically based methods are expressed by the theory of elasticity referred to as Hookes law, providing a high precision behavior on simulated objects. Furthermore, the non-physically-based method is represented by a generalization of barycentric coordinates, used to obtain high performance while maintaining a plausible deformation.

Our approach allows modeling each object of the environment according to its relevance to the simulation. Therefore, it allows the user to choose the deformation method (FEM or MSD) arbitrarily for each object. Furthermore, during the simulation, our environment approximates objects not under interaction using a faster method, saving computational resources. The collision detection is performed independently of the deformation method, but its outcome works differently depending on the method. While the collision affects the FEM and MSD modeled objects by applying computed displacements directly on the mesh vertices, for the case of GC, it computes the contribution of each displacement value for each cage vertex. This rough approximation can be further improved using a cage that better fits the object.

The surgery environment simulated by our system can help students and specialists, providing a safe and practical way to experiment complex surgical procedures. Our case studies do not cover a complete surgery simulation system. Nevertheless, they implement typical surgical interactions, such as picking, burning, and haptic feedback.

Acknowledgment

We gratefully acknowledge the partial financial support from CNPq through grants 449555/2014-3, 131679/2015-5 and 305071/2012-2, and Fapergs for PqG 2283-2551/14-8.

REFERENCES

[1] J. E. Wickham, "The new surgery." *BMJ*, vol. 295, no. 6613, pp. 1581–1582, 1987.

[2] R. Vecchio, B. V. MacFayden, and F. Palazzo, "History of laparoscopic surgery." *Panminerva Med*, vol. 42, no. 1, pp. 87–90, 2000.

[3] D. Stoyanov, M. ElHelw, B. Lo, A. Chung, F. Bello, and G.-Z. Yang, "Current issues of photorealistic rendering for virtual and augmented reality in minimally invasive surgery," in *Information Visualization, 2003. IV 2003. Proceedings. Seventh International Conference on*. IEEE, July 2003, pp. 350–358.

[4] S. De and A. Maciel, "Virtual reality-based surgical trainers," in *Textbook of Simulation: Skills and Team Training*, S. Tsuda, D. J. Scott, and D. Jones, Eds. Woodbury, CT: Cin-med, 2011, ch. 20, pp. 225–241.

[5] U. Meier, O. López, C. Monserrat, M. C. Juan, and M. Alcaniz, "Real-time deformable models for surgery simulation: a survey," *Computer methods and programs in biomedicine*, vol. 77, no. 3, pp. 183–197, 2005.

[6] C. Basdogan, S. De, J. Kim, M. Muniyandi, H. Kim, and M. Srinivasan, "Haptics in minimally invasive surgical simulation and training," *Computer Graphics and Applications, IEEE*, vol. 24, no. 2, pp. 56–64, March 2004.

[7] H. Delingette and N. Ayache, "Hepatic surgery simulation," *Commun. ACM*, vol. 48, no. 2, pp. 31–36, Feb. 2005.

[8] D. J. Scott, E. M. Ritter, S. T. Tesfay, A. Pimentel, Elisabeth A. andNagji, and G. M. Fried, "Certification pass rate of 100laparoscopic surgery skills after proficiency-based training," *Surgical Endoscopy*, vol. 22, pp. 1887–1893, 2008.

[9] A. Okrainec, L. Smith, and G. Azzie, "Surgical simulation in africa: the feasibility and impact of a 3-day fundamentals of laparoscopic surgery course," *Surgical Endoscopy*, vol. 23, no. 11, pp. 2493–2498, 2009.

[10] I. Choy and A. Okrainec, "Simulation in surgery: Perfecting the practice," *Surgical Clinics of North America*, vol. 90, no. 3, pp. 457 – 473, 2010, simulation and Surgical Competency.

[11] V. S. Arikatla, W. Ahn, G. Sankaranarayanan, and S. De, "Towards virtual fls: development of a peg transfer simulator," *The International Journal of Medical Robotics and Computer Assisted Surgery*, 2013.

[12] K. Morooka, Y. Nakasuka, R. Kurazume, X. Chen, T. Hasegawa, and M. Hashizume, "Navigation system with real-time finite element analysis for minimally invasive surgery," in *Proceedings of the 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2013, pp. 2996–2999.

[13] S. Sulaiman, T. Hui, A. Bade, R. Lee, and S. Tanalol, "Optimizing time step size in modeling liver deformation," in *Proceedings of the IEEE 3rd International Conference on System Engineering and Technology (ICSET)*, 2013, pp. 209–214.

[14] G. Cirio, M. Marchal, A. Le Gentil, and A. Le andcuyer, ""tap, squeeze and stir" the virtual world: Touching the different states of matter through 6dof haptic interaction," in *Proceedings of the IEEE Virtual Reality Conference (VR), 2011*, Singapore, march 2011, pp. 123 –126.

[15] Y.-J. Lim, J. Hu, C.-Y. Chang, and N. Tardella, "Soft tissue deformation and cutting simulation for the multimodal surgery training," in *Proceedings of the 19th IEEE International Symposium on Computer-Based Medical Systems. CBMS.*, 2006, pp. 635–640.

[16] S. Queiros, J. Vilaca, N. Rodrigues, S. Neves, P. Teixeira, and J. Correia-Pinto, "A laparoscopic surgery training interface," in *Proceedings of the IEEE 1st International Conference on Serious Games and Applications for Health (SeGAH)*, 2011, pp. 1–7.

[17] C. H. Park, K. Wilson, and A. Howard, "Examining the learning effects of a low-cost haptic-based virtual reality simulator on laparoscopic cholecystectomy," in *Proceedings of the IEEE 26th International Symposium on Computer-Based Medical Systems (CBMS)*, 2013, pp. 233–238.

[18] P. Marshall, S. Payandeh, and J. Dill, "Suturing for surface meshes," in *Proceedings of the IEEE Conference on Control Applications. CCA.*, 2005, pp. 31–36.

[19] Y. Duan, W. Huang, H. Chang, K. K. Toe, T. Yang, J. Zhou, J. Liu, S. K. Teo, C. W. Lim, Y. Su, C. K. Chui, and S. Chang, "Synchronous simulation for deformation of liver and gallbladder with stretch and compression compensation," in *Proceedings of Engineering in Medicine and Biology Society (EMBC), 35th Annual International Conference of the IEEE*, 2013, pp. 4941–4944.

[20] Simbionix, "Lap mentor," http://simbionix.com/simulators/lap-mentor/, 2014, lAP MentorTM 2014.

[21] S. Science, "Lapsim," http://www.surgical-science.com/lapsim-the-proven-training-system/, 2014, lapSim: The Proven Training System, 2014.

[22] I. Intuitive Surgical, "da vinci," http://www.intuitivesurgical.com/products/, 2014, daVinci 2014.

[23] S. Timoshenko and J. Goodier, *Theory of elasticity*, ser. McGraw-Hill classic textbook reissue series. McGraw-Hill, 1969.

[24] K. Bathe, *Finite Element Procedures in Engineering Analysis*, ser. Prentice-Hall Civil Engineering and Engineering Mechanics Series. New Jersey, United States: Prentice-Hall, 1982.

[25] M. Davis, *Numerical methods and modeling for chemical engineers*, ser. Wiley series in chemical engineering. Wiley, 1984.

[26] G. Yin, Y. Li, J. Zhang, and J. Ni, "Soft tissue modeling using tetrahedron finite element method in surgery simulation," in *Proceedings of the First IEEE International Conference on Information Science and Engineering*, ser. ICISE '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 3705–3708.

[27] M. Bro-Nielsen and S. Cotin, "Real-time volumetric deformable models for surgery simulation using finite elements and condensation," *Computer Graphics Forum*, vol. 15, no. 3, pp. 57–66, 1996.

[28] X. Provot, "Deformation constraints in a mass-spring model to describe rigid cloth behavior," in *In Graphics Interface*, 1996, pp. 147–154.

[29] S. F. F. Gibson and B. Mirtich, "A survey of deformable modeling in computer graphics," Mitsubishi Electric Research Laboratories, Cambridge, MA, USA, Tech. Rep., 1997.

[30] J. Georgii and R. Westermann, "Mass-spring systems on the gpu," *Simulation Modelling Practice and Theory*, vol. 13, no. 8, pp. 693–702, 2005.

[31] A. Selle, M. Lentine, and R. Fedkiw, "A mass spring model for hair simulation," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 64:1–64:11, Aug. 2008.

[32] J. Mesit, R. Guha, and W. Furlong, "Simulation of lung respiration function using soft body model," in *Proceedings of the Fourth UKSim European Symposium on Computer Modeling and Simulation (EMS)*, nov. 2010, pp. 102 –107.

[33] Y. Lipman, D. Levin, and D. Cohen-Or, "Green coordinates," *ACM Trans. Graph.*, vol. 27, no. 3, 2008.

[34] M. S. Floater, "Mean value coordinates," *Comput. Aided Geom. Des.*, vol. 20, no. 1, pp. 19–27, Mar. 2003.

[35] P. Joshi, M. Meyer, T. DeRose, B. Green, and T. Sanocki, "Harmonic coordinates for character articulation," *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007.

[36] M. Meyer, A. Barr, H. Lee, and M. Desbrun, "Generalized barycentric coordinates on irregular polygons," *J. Graph. Tools*, vol. 7, no. 1, pp. 13–22, Nov. 2002.

[37] T. Langer, A. Belyaev, and H.-P. Seidel, "Spherical barycentric coordinates," in *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, ser. SGP '06. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2006, pp. 81–88.

[38] C. Xian, H. Lin, and S. Gao, "Automatic generation of coarse bounding cages from dense meshes," in *Proceedings of the IEEE International Conference on Shape Modeling and Applications.*, June 2009, pp. 21–27.

[39] ——, "Automatic cage generation by improved obbs for mesh deformation," *The Visual Computer*, vol. 28, no. 1, pp. 21–33, 2012.