Real-Time Visual Tracking of Deformable Objects in Robot-Assisted Surgery

Ibai Leizea ▪ *IK4-Ideko*

Ainitze Mendizabal ▪ *CEIT-IK4*

Hugo Alvarez ▪ *Vicomtech-IK4*

Iker Aguinaga, Diego Borro, and Emilio Sanchez ▪ *CEIT-IK4 and University of Navarra*

One of the most challenging problems in robot-assisted surgical systems is to provide surgical realism at interactive simulation rates. The proposed visual tracking system can track and register object deformations in real time using a physically based formulation, despite the occlusions produced by the robotic system itself.

The use of robotic devices in medicine and, more specifically, in the surgical field entails a remarkable improvement in the delivery of healthcare services. Today, robotic devices and computer-assisted technologies can offer guidance, diagnosis, verification, and general assistance during surgical interventions.[1]

However, the use of robotic devices in surgery creates new challenges that must be overcome to ensure successful deployment. A robot acting on a tissue exerts a force and produces deformations, but provides no direct feedback to the surgeon. Thus, we must track organ deformation to provide assistive technologies, such as augmented reality (AR). In addition to providing surgeons with visual feedback of the tissue deformations, such systems can also help surgeons locate the position of a malignancy to be removed from an organ. In most cases, however, it is not feasible to use sensors to track the position of the patient's organs. Thus, alternative approaches such as computer vision techniques are required to help solve this problem.

In practice, the robot's very presence produces occlusions that limit how much information can be obtained using only computer vision. Combining computer vision with a suitable physical simulation can help reduce the missing information. However, surgical simulations necessitate a high level of accuracy. As opposed to the deformable models used in video games and animations, the purpose of soft tissue models in medical simulation is to model realistically the behavior of biological tissues. Consequently, tissue deformation simulations should be controlled using real material parameters, and these parameters should be obtained from biomechanics experiments instead of intuitively adjusted parameters.

To further complicate the implementation of robotic guidance and surgery assistance systems, computational performance must also be taken into account in order to provide surgeons with timely visual feedback. Therefore, it is critical for such applications to obtain a compromise between accuracy and computational cost. As a consequence, one of the most challenging problems in robot surgical assistance systems is to provide surgical realism at interactive simulation rates.[2]

 Published by the IEEE Computer Society

We propose a novel visual tracking method based on physical simulation. This procedure can obtain the deformation produced by a robotic system on patient tissues, thus gathering the basic information needed to provide surgeons with assistance and guidance using visual or haptic feedback. Our approach is divided into two main phases. The first deals with handling robotic surgical arm movements, and the second utilizes a vision system to compute the deformations created in the first step. This vision system consists of an RGB-D camera that returns color and depth information. Because this data is incomplete (due to occlusions and areas where the sensor cannot capture data) and contains noise, we use a physical simulation module to reconstruct the complete object deformation. We feed the input into a nonlinear finite element method (FEM) formulation to simulate the physical behavior of the tissue under deformation (via its volume and the missing surfaces).

The proposed method has been integrated into a surgical module within the framework of a robotic surgery system.[3] To evaluate the proposed system's level of accuracy and computation requirements, we tested it using three different objects in ex-vivo conditions. We then compared the deformations obtained from the experiments with the theoretical results obtained by finite-element analysis obtained using the Abaqus software package. The results provide an accurate visual representation of the deformed solid.

Tracking Deformable 3D Objects in Surgical Environments

Recovering the 3D structure of a nonrigid object is a complex task in the computer vision field. Thus, the majority of approaches tackling this problem divide the process into two steps: image registration and shape inference. The first is related to solving the visual part of the problem, whereas the latter is responsible for deducing the shape's new structure.

Feature-based vision and pixel-based or direct methods are the most significant and commonly used techniques to solve the image registration problem. Feature-based methods focus on detecting visual cues (also called *features*) on the image and establishing correspondences with a reference image to solve the registration problem.[4] However, these approaches are primarily focused on planar surfaces or textured surfaces such as t-shirts.[4,5] Direct methods use intensity differences between two images to calculate the correspondences,[6,7] but these approaches require good initialization because drift problems may arise.

The shape inference stage uses visual information from the image registration process to compute the new 3D structure. Among the various solutions available, some rely on optimization techniques,[8] such as second-order cone programming (SOCP), and others use physics-based systems to determine the correct physical behavior. The FEM has been widely used for surgical simulations[9] because it incorporates real physical material parameters, and therefore, it models tissue properties more accurately than other approaches, such as mass-spring models.

The FEM requires the definition and modeling of the material behavior to be simulated. Currently, there is no agreement in the literature regarding how to define the most suitable material models required for surgical simulation. A key question in modeling tissue behavior for simulation is the complexity level of the model required for a certain application. More complex models can better simulate the detailed behavior of the tissue, but at high computational cost. For surgical simulation of biological tissues, models must be simple enough to solve a range of problems, but complete enough to realistically describe the behavior under a variety of load conditions.[10]

Linear FEM models are commonly used to model deformable materials, mainly because the equations remain simple and the computation can be optimized. Several researchers have used linear elasticity to model brain deformations, for example.[11] However, simulations built upon linear elastic models can be applied only to small deformations, and most surgical procedures involve organs being subjected to large ones.

The deformation of most biological materials under large strains can be described by the theory of nonlinear elasticity (such as hyperelastic models). Hyperelastic models are commonly used to simulate brain tissue,[12] liver tissue,[13] and skin[14]; for laparoscopic simulation[15]; or to simulate soft tissues in general.[16] Soft tissues can also be simulated with viscoelastic models. The literature includes uses of linear viscoelastic models,[17] quasi-linear viscoelastic models,[18] and even more complex nonlinear viscoelastic models.[19] Nevertheless, estimating parameter values for the nonlinear viscoelastic models can be complex and computationally expensive.

Although the nonlinearity and viscoelasticity of soft tissues are widely known, nonlinear viscoelastic models are not always the best choice for simulation. Zizchen Liu and Lynne Bilston concluded that a complex constitutive model may be required if the accuracy of the obtained deformation is critical, whereas in other applications, where accuracy is less important than computational efficiency, a simpler constitutive model may be more appropriate.[20] Adam Wittek and his colleagues recommended using a linear elastic constitutive model and a geometrically nonlinear FEM formulation to simulate brain tissue in the case of craniotomy-induced shift because it saves computational time and causes no loss of accuracy compared with a hyperviscoelastic model.[21]

Some recent works have presented robotic systems complemented by vision techniques in surgery. For example, Ghassan Hamarneh and his colleagues developed preoperative surgical planning, intraoperative image registration, and AR visualization for image-guided tumor identification.[22] Their work aims to help surgeons determine the tumor localization and resection margins. They focused on kidney cancer cases with robot-assisted partial nephrectomy performed with a da Vinci surgical robot (Intuitive Surgical). They built a biomechanical model of the kidney tissue and tumor with a FEM using a corotational tetrahedral formulation and a Eulerian implicit solver. However, their work is more focused on surface reconstruction than the 3D deformation of an object.

In another example, Nazim Haouchine and his colleagues developed an image-guided biomechanical model that captures the complex deformations the liver undergoes during surgery.[9] The system works with laparoscopic stereo cameras and uses the FEM to capture the object's behavior and compute the deformation. The FEM is based on a corotational formulation with a stress-strain relationship taken from the literature. However, the camera's position should be static, and in some cases, the simulation algorithms are not executed in real time. Haouchine and his colleagues also proposed a real-time method to register the nonlinear elastic model deformations using the image points acquired from a monocular camera.[23] That solution is based on an orthographic projection, which is easier to compute than a perspective projection. In the latter case, they used a Saint-Venant-Kirchhoff model for FEM formulation.

In both the Haouchine cases,[9,23] tracking is based on features. However, in environments such as medicine or industry, it is sometimes difficult to perform a tracking based only on visual cues (often essential for a vision system), either because of the lack of textured areas and models or the environmental conditions (such as blood in medicine).

Unlike existing solutions, the tracking algorithm we propose does not require features. That means that our work avoids the use of formulations that are not usually robust for textureless surfaces or objects. This makes our approach robust against illumination changes and adaptable for tracking any type of object (in terms of its geometrical shape). Furthermore, our method avoids the ambiguity that can be caused by orthographic projections. Other advantages of the presented method are that it is modular and the camera is not required to be static.

Proposed Method

This section presents a complete framework for registering deformations of nonrigid objects when a surgical robot is applying a force. The goal is to provide surgeons with extra visual feedback. The selected robot-assistant module has been developed as a prototype for a cooperative robotic platform aimed at assisting in surgery for lumbar transpedicular fixation.[3]

The vision module consists of a RGB-D camera that obtains color and depth data. As we explained earlier, this information is incomplete and noisy, so it provides insufficient visual feedback. We use the acquired raw information as the input to a FEM physical model to obtain a correct physical behavior. In this sense, the model represented as a triangle mesh is converted to a tetrahedral mesh to adjust it to the FEM formulation. This triangle mesh, in turn, is captured through a scanning process performed with a 3D Sense Scanner. The phases of the physical model simulation require specific types of models (different geometry mesh and material properties of the bodies). Figure 1 shows the three different models that were used in our experimentation process: a sponge, a porcine kidney, and a calf brain.

To obtain a compromise between the reconstruction computation time and accuracy, we model deformable objects as Saint-Venant-Kirchhoff material within a nonlinear FEM formulation. This material model appears to be an ideal compromise because it can handle nonlinear deformations, is rotationally invariant, and is simpler than other nonlinear models. The materials are defined by the Young modulus, Poisson's ratio, and density. To define the mechanical properties of the materials with a realistic physical behavior, we performed simple shear tests using a rotational rheometer.

Figure 2 illustrates the system configuration. Because we focus only on deducing the organ deformations, we compute the camera pose (camera tracking) using a marker-based system. This provides a level of accuracy high enough to ignore the error.

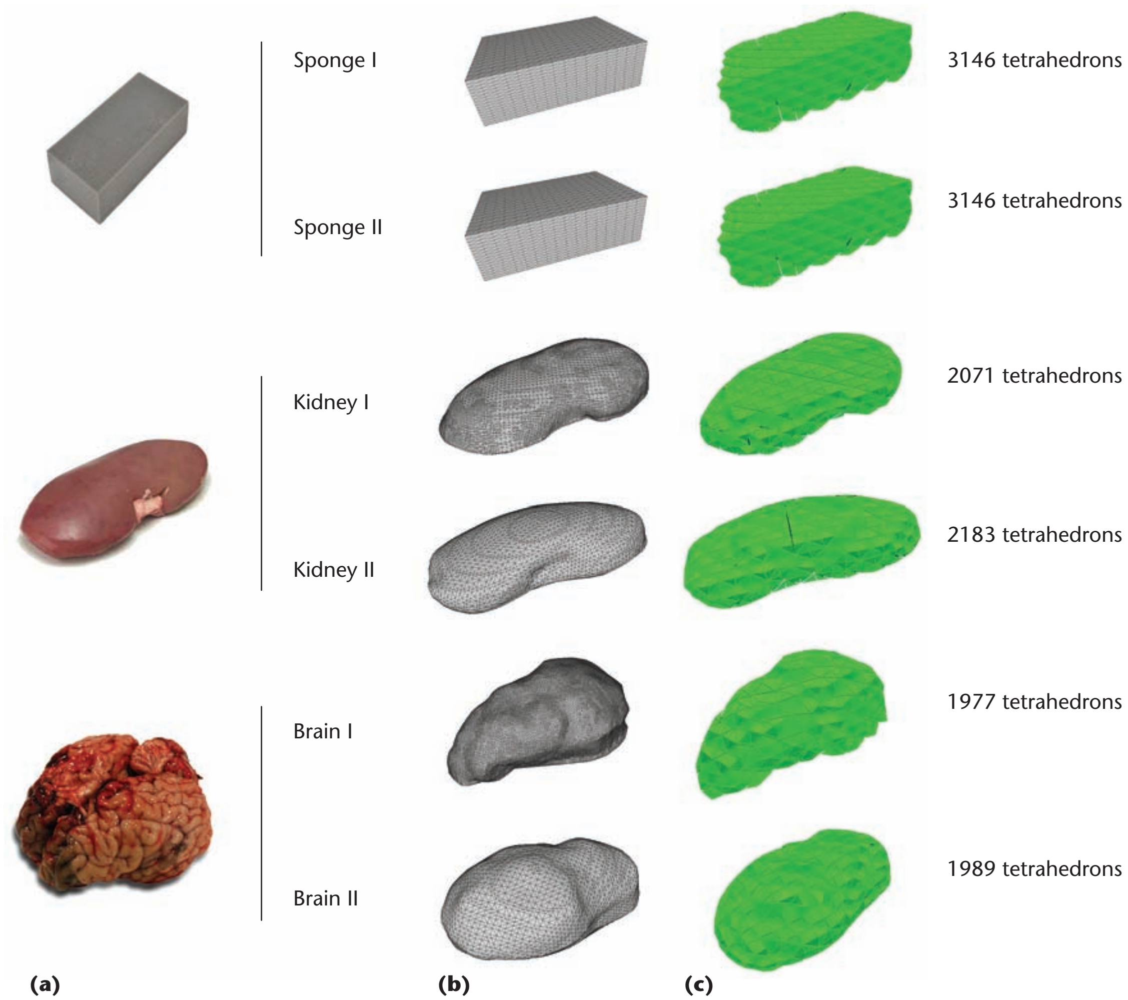

Figure 1. Transformation from triangle mesh to tetrahedral mesh. For the finite element method (FEM) formulation, (a) the model represented as (b) a triangle mesh is converted to (c) a tetrahedral mesh.

The surgical assistant we used consists of a commercial PA10-7C robotic arm (Mitsubishi Heavy Industries) with an open control architecture. The open architectures means that, using a generic programming language, it is possible to independently develop the control algorithms to be implemented in the robot control PC. The arm is a seven degree of freedom (DoF) open chain serial manipulator, and all of its revolute joints have a well-defined rotation axis. Its maximum load capacity is 98 N, and it can reach a distance of 1.03 m when fully extended. The robot has a force/torque sensor, the Mini40 (ATI Industrial Automation), that records the force performed by the robot on the soft tissue.

Figure 3 shows the robotic arm used as an indenter and in its exploded view. The end of the robotic arm consists of a force/torque sensor, a grip, a large indenter, and second smaller indenter.

Object Deformation

The main challenge of the nonrigid problem is to determine the mesh's transformation after deformation. Therefore, the object deformation phase is responsible for performing the deformable registration of the nonrigid model. As Figure 4 shows, this involves two main steps: preprocessing and mesh registration. The *preprocessing* is an offline process that consists of defining the proper parametrization of the physical model. The *mesh registration* is an online execution that calculates the correspondences between the model and the input

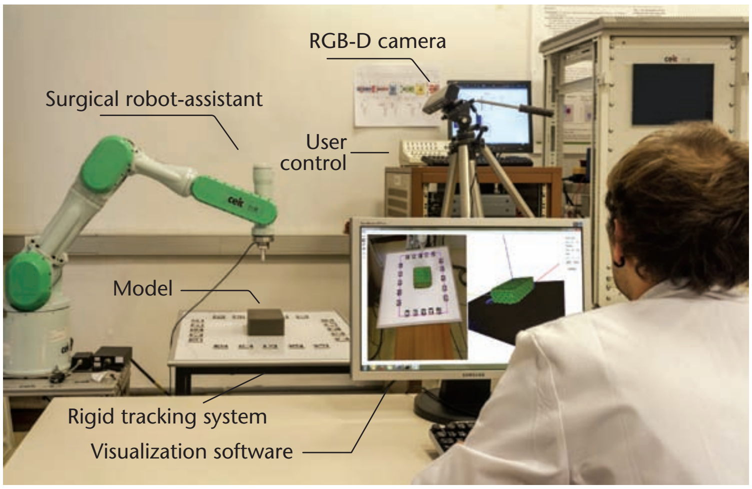

Figure 2. System configuration. The RGB-D camera gathers color and depth data as the surgical robot assistant makes indentation in the tissue samples.

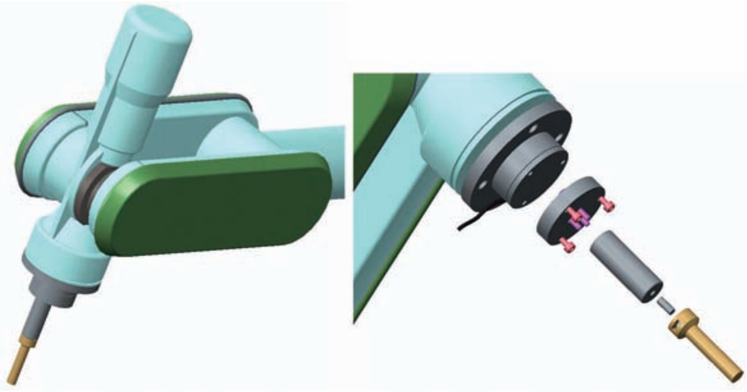

Figure 3. Robotic arm. These representations show (a) the arm used as an indenter and (b) an exploded view of its tip.

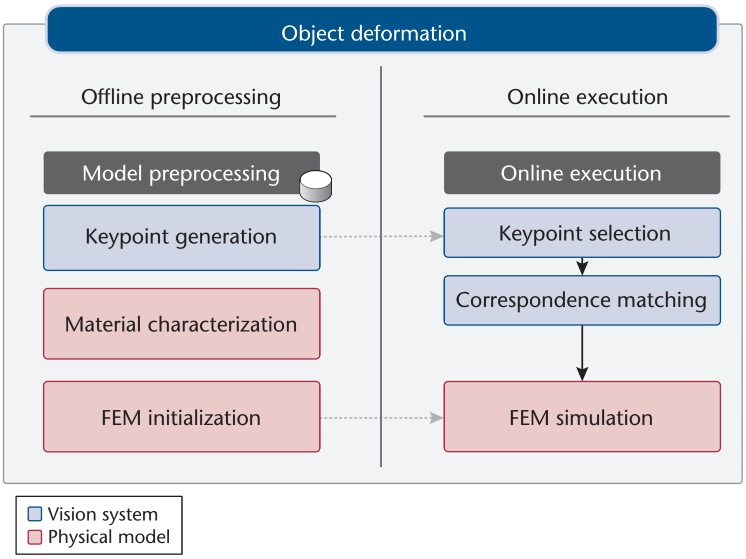

Figure 4. Overview of the proposed object deformation method. The preprocessing process occurs offline, whereas the mesh registration phase executes online.

point cloud acquired from the RGB-D camera to subsequently obtain the physical registration with the FEM formulation. Both processes are divided into two main phases, related to the visual part and the physical model, respectively.

Model Preprocessing

The offline model preprocessing phase, which is essential to the online mesh registration phase, is executed only once for each model. To complete this preprocessing task, it is necessary to generate information from the visual part of the system as well as the physical parametrization of the model. The first is responsible for generating a set of control points to determine the correspondences between the raw information acquired from the camera and the model's 3D mesh (keypoint generation). The physical model procedure in turn is divided into two main parts: the description of the physical formulation based on the FEM initialization and the range tests performed to define the material properties (material characterization).

Keypoint Generation. The goal of this phase is to define a set of control points that relate to the input point cloud acquired in each camera frame and the 3D mesh of tetrahedrons. Those control points, or *keypoints*, are in fact the vertices that are on the surface of the tetrahedral mesh that correspond to a set of uniformly distributed points on the surface. In this way, these keypoints serve to relate the vertices of the mesh to the raw point cloud in the online phase.

Moreover, this phase serves as a filter that removes the keypoints that could affect the correct functioning of the deformation process. This means that the keypoints that are relatively close (based on a threshold) to the object's bounding box are discarded to avoid noisy movements along the corners.

Material Characterization. To obtain the mechanical properties of the deformable objects, we tested a sample of each material in a parallel-plate rheometer (Anton Paar Physica, MRC 301). The rotational rheometer characterizes the material under shear loads, which are common loads for soft tissues during surgical procedures.[24]

Figure 5 shows a schematic representation of the experimental setup. The top plate of the rheometer is lowered until it contacts the sample's upper surface.

The measured strain γ is not constant along the sample because it is a function of the plate's radius R, the gap H, and the deflection angle φ:

$$\gamma[1] = \frac{\varphi[\mathrm{rad}]R[\mathrm{mm}]}{H[\mathrm{mm}]}.$$

Therefore, the maximum deformation and maximum shear rate occur at the edge of the plate, and the reported data is related to this position.

Amplitude sweeps are performed in cylindrical samples, with a diameter of 25 mm and thickness of 2–4 mm. An amplitude sweep is an oscillatory test performed at variable strain amplitudes, keeping frequency at a constant value. As long as strain amplitudes remain within the limits of the linear viscoelastic range, the values of the dynamic properties remain steady and the material shows reversible-viscoelastic behavior. This means the nonrigid object deforms elastically and will return to its original shape when the applied stress is removed. However, at amplitudes higher than this limit, some fraction of the deformation will be permanent and nonreversible; in some cases, the sample's structure could even be destroyed completely.

Thus, the mechanical properties of the tissues are defined within a linear viscoelastic range. We obtain the shear modulus G, imposing a strain ramp ranging from 0.001 to 100 percent, at 1 Hz. The value of G is within linear viscoelastic range, and we can select Poisson's ratio v for each. We also assume linear elasticity, isotropy, and homogeneity of the material. Therefore, the Young modulus E is obtained as $E = 2G(1 + v)$.

Density ρ is considered constant within the tissue sample and is determined by measuring its mass and volume. Table 1 shows the values of the mechanical properties established for each material.

FEM Initialization. The system uses a nonlinear total Lagrangian explicit finite element formulation using a tetrahedral mesh. This kind of formulation is well suited to surgery simulators because it provides a compact and efficient implementation. The formulation can easily handle nonlinear material and large deformations. Additionally, the method does not require the computation of a stiffness matrix, which allows an easy adaptation to topological changes.

In this case, for simplicity and computational efficiency, we implemented a Saint-Venant-Kirchhoff material model using the properties for the linear material model we described earlier.

This formulation allows a direct computation of the elastic forces acting on each node when the model is deformed.

In a Lagrangian approach, the dynamic analysis is performed by tracking the material particles forming a body. In particular, two sets of coordinates can be defined: the spatial coordinate $\mathbf{x}$ represents the position of one particle in the deformed state, and material coordinate $\mathbf{X}$ represents its original position. In a total Lagrangian representation, all forces, deformations, and material stresses are expressed in the material coordinate system $\mathbf{X}$. We iterate per element and compute the deformation gradient tensor $\mathbf{F}$:

$$\mathbf{F}(\mathbf{X}) = \frac{\partial \mathbf{x}(\mathbf{X}, t)}{\partial \mathbf{X}}.$$

In the case of tetrahedral elements, $\mathbf{F}$ can be expressed as

$$\mathbf{F} = [\mathbf{x}_2 - \mathbf{x}_1, \mathbf{x}_3 - \mathbf{x}_1, \mathbf{x}_4 - \mathbf{x}_1][\mathbf{X}_2 - \mathbf{X}_1, \mathbf{X}_3 - \mathbf{X}_1, \mathbf{X}_4 - \mathbf{X}_1]^{-1}.$$

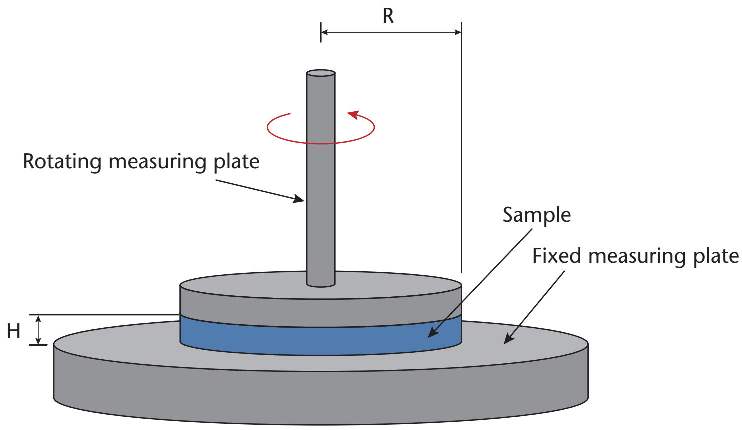

Figure 5. Parallel plate rheometer. The top plate of the rheometer is lowered until it contacts the sample's upper surface.

Table 1. Mechanical properties of the tested materials.

Property	Sponge	Kidney	Brain
Density, ρ (kg/m^3)	14.3	1,000	1,000
Young modulus, E (Pa)	76,261	1,500	1,085
Poisson's ratio, v	0.30	0.45	0.48

The Cauchy stress tensor σ is related to the material model by the strain energy function W as follows:

$$\sigma = \frac{2}{\sqrt{I_3}}\left[\frac{\partial W}{\partial I_1}\mathbf{B} + I_1\frac{\partial W}{\partial I_2}\mathbf{B} - \frac{\partial W}{\partial I_2}\mathbf{BB}\right] + 2\sqrt{I_3}\frac{\partial W}{\partial I_3}I,$$

where $\mathbf{B}$ is the left Cauchy-Green deformation tensor and I_i is an invariant of $\mathbf{B}$. The left Cauchy-Green deformation tensor is related to a deformation gradient tensor as $\mathbf{B} = \mathbf{FF}^T$.

Invariants of $\mathbf{B}$ are defined as follows:

$$I_1 = \mathrm{tr}B$$
$$I_2 = \frac{1}{2}\left[(\mathrm{tr}B)^2 - \mathrm{tr}B^2\right]$$
$$I_3 = J^2 = (\det B)^2 \quad .$$

The Cauchy stress tensor σ is defined with regard to the body's current configuration. Using a total Lagrangian formulation, we can define the stress tensor with regard to the object's initial configuration. The first Piola-Kirchhoff stress tensor $\mathbf{P}$ relates the element's deformation and the mechanical stress in the material, expressed in material coordinates: $\mathbf{P} = J\sigma\mathbf{F}^{-T}$.

When the constitutive model is given as a first Piola-Kirchhoff stress $\mathbf{P}$, an element's contribution to the finite-element force on one of its nodes x_a is given as[25]

$$\mathbf{f}_a^e = \int_{\Omega_m} \mathbf{P}\frac{\partial N_a^T}{\partial \mathbf{X}} d\mathbf{X} = \int_{\Omega_\xi i} P\frac{\partial N_a^T}{\partial \mathbf{X}} d\left|\frac{\partial \mathbf{X}}{\partial \xi}\right|\xi,$$

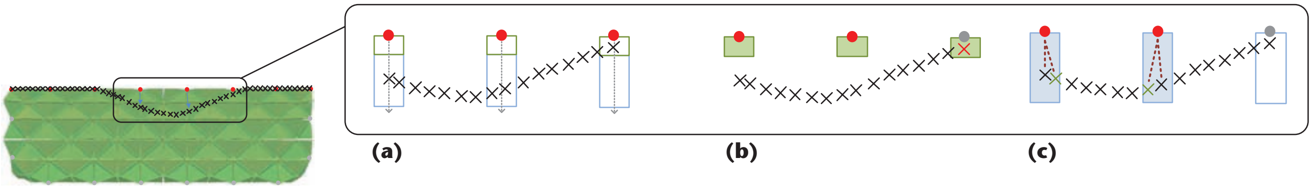

Figure 6. Correspondence matching. (a) Once the search areas are defined, two oriented bounding box (OBB) tests are applied in order to relate the keypoints to the point cloud: (b) OBB test 1 and (c) OBB test 2.

where Ω_m and Ω_ξ represent the volume of the element in the global framework and an ideal system, respectively, based on isoparametric coordinates; N_a represents the interpolation function for node a, and $\mathbf{X}$ is a particle's material coordinates.

The stress tensor $\mathbf{P}$ is constant within a tetrahedron and the integrals can be easily computed, leading to a force generated by each element in each node a equivalent to

$$\mathbf{f}_a^e = \frac{1}{3}\mathbf{P}\sum_{i \neq a} A_i \mathbf{N}_i ,$$

where $A_i\mathbf{N}_i$ is the area weighted normals of the faces of the tetrahedron incident in node a in the original position, which can be precomputed.

The total elastic force acting upon a node a is the sum of the contributions of each element sharing node a:

$$\mathbf{f}_a = \sum \mathbf{f}_a^e .$$

After the value of the elastic forces is updated, we can iterate through each node of the mesh to compute next position and velocity. The simulation is performed with a precomputed time step (Δt) that guarantees the stability of the Euler semi-implicit integrator employed:

$$\mathbf{v}_a(t+\Delta t) = \mathbf{v}_a(t) + \frac{\mathbf{f}_a(t)}{m_a}\Delta t$$
$$\mathbf{x}_a(t+\Delta t) = \mathbf{x}_a(t) + \mathbf{v}_a(t+\Delta t)\Delta t ,$$

where m_a is the mass of the node a. The mass matrix is considered diagonal. (See earlier work for additional details about this method.[25]) During the initialization, the first term is computed and stored.

Mesh Registration

Mesh registration is executed for every frame. For this purpose, we exploit the information from the RGB-D camera. The camera's color information is used to project the 3D deformed mesh in the image, and the depth information helps us obtain the point cloud. Furthermore, a multiple marker tracking system returns the camera's accurate position and orientation for each frame.

The camera's position and orientation, known as the *camera pose*, let us estimate the correspondences between the offline keypoints and the current point cloud captured by the RGB-D camera. In addition, these matches serve as the input displacements to the physical module, which is responsible for calculating the model's new shape. This stage is divided into a visualization step (keypoint selection and correspondence matching) and the mesh physical simulation module (FEM simulation).

Keypoint Selection. In addition to the keypoints discarded in the preprocessing step, the system must account for all the keypoints that are not visible during the online execution due to the camera's point of view. Therefore, all the keypoints that are not visible must be discarded, whether or not deformation is being applied at that point. We apply an occlusion query test to select the visible keypoints. This procedure consists of computing the normal vector of the keypoint (the normal of the triangle it belongs to) and calculating the difference (in degrees) with respect to the vector that represents the camera's point of view.

Correspondence Matching. The main goal of this step is to find the associations, also called *correspondences*, between the visible keypoints and the input raw point cloud. The procedure consists of an intelligent scale search (see Figure 6a). This procedure divides the problem of finding the correspondences into two main steps. First, we discard the keypoints that do not have any deformation. In the second step, we make associations between the keypoint and a sample of the input point cloud.

These two steps involve two search areas for all keypoints. These areas, represented by two oriented bounding boxes (OBBs), differ from the height value. The values for these boxes are fixed according to the measures of the global bounding box and are calculated offline. In order to find the deepest point, we define the orientation according to the normal of each keypoint.

The first step, which we call OBB test 1, corresponds to the search at the smallest OBB (see Figure 6b). The main goal of this test is to determine if there is any point inside a small box around its position. If there is, the keypoint will be discarded because there is no deformation for it or because the deformation is so small that it can be disregarded for the global deformation. The second step, defined as OBB test 2, finds a correspondence over the second box (see Figure 6c) to provide the most distant point.

Both searches are carried out using an efficient octree search based on a recursive search of the boxes (implementation provided by the PCL library). This search lets us obtain an octree representation for a given input point cloud and, consequently, perform fast intersection tests.

FEM Simulation. Once the nodes of the simulation mesh have been matched to the deformed surface, they are used to control a dynamic simulation of the solid's deformation. During this simulation, the nodes are displaced from their original position to the distance detected using the vision system. The simulation runs until it achieves the final configuration—that is, until it represents the deformed state of the solid. The simulation is performed with a precomputed time step that guarantees the stability of the semi-implicit Euler integrator that is employed.

Experiments

To evaluate the performance (error estimation and computational time) and adaptability of the proposed system for different kinds of models, we developed a set of experiments using the robotic arm we described earlier. Specifically, we used two cylindrical indenters of different sizes (see Figure 3) to deform the tissue: one has a diameter of 15 mm and a 45 mm length, and the smaller one has a diameter of 8 mm and a 36 mm length.

The hardware setup consisted of an Intel Core 2-Quad Q9550 at 2.83 GHz and 4 Gbytes of RAM equipped with a Kinect Xbox 360. We used sponge, porcine kidney, and calf brain models to evaluate the framework's performance. The sponge has a simple geometric shape that contains a homogeneous texture (it is textureless), while the brain and kidney provided alternative geometric shapes with differing textures. The texture of the surface was not used for recognition in any case. We used the same sponge for each experiment, but we used two different samples for both the brain and kidney models. The aim was to use the same material parametrization and make indentations in different areas to determine if our system's behavior was the same. For the brain and kidney categories, we applied a force in different areas for each sample by testing the deformation with different tools. Specifically, we used the smaller indenter for the brain II and kidney II samples. For the sponge model, we applied a force in two different areas for the sponge I and sponge II samples. Figure 1 depicts this categorization as well as the number of tetrahedrons of each model, which is a determining factor in the experimental results.

Table 2. Error values (in mm) between the models obtained using the online FEM and Abaqus simulations.

Model	Mean error	Standard deviation	Max error
Sponge I	0.85	2.20	37.68
Sponge II	0.72	1.88	32.89
Kidney I	1.28	2.23	14.97
Kidney II	0.40	0.68	6.42
Brain I	1.09	1.10	7.24
Brain II	1.08	1.37	10.36

Accuracy Level

We used two different techniques to validate the online FEM formulation's level of accuracy. These consist of comparing the results obtained with the online FEM simulation with those from a simulation of the same experiment using Abaqus software and with the 3D reconstruction obtained through a 3D scanner.

Abaqus. The experiments were simulated in Abaqus 6.13. The mechanical properties of both tools were considered rigid enough compared with the indented tissues (ρ = 7850 kg/m^3, E =3000000 Pa, ν = 0.2). We modeled the deformable objects with the same mesh as the one used in the online FEM simulation. Table 1 lists the properties for each material. The curve of the displacement versus time recorded by the robot in each experiment was imposed on the tool in the Abaqus simulation. The simulations were defined in a dynamic, implicit step using the Quasi-Newton solution technique.

We then compared the deformed meshes produced by the Abaqus and online FEM simulations and determined the error between the meshes by calculating the point-to-point Euclidean distance. Table 2 shows these error values for each material, and Figure 7 illustrates the visual results of this comparison.

Scanner. As we explained earlier, for these experiements, a force was applied to the models by the robotic arm, and the deformation was computed using the online FEM formulation. Simultaneously, we scanned the models using the Cubify 3D Sense Scanner during the deformation. The reconstruction of this scanner data served as reference for the error estimation—that is, it was used as the ground truth. According to the manufacturer's

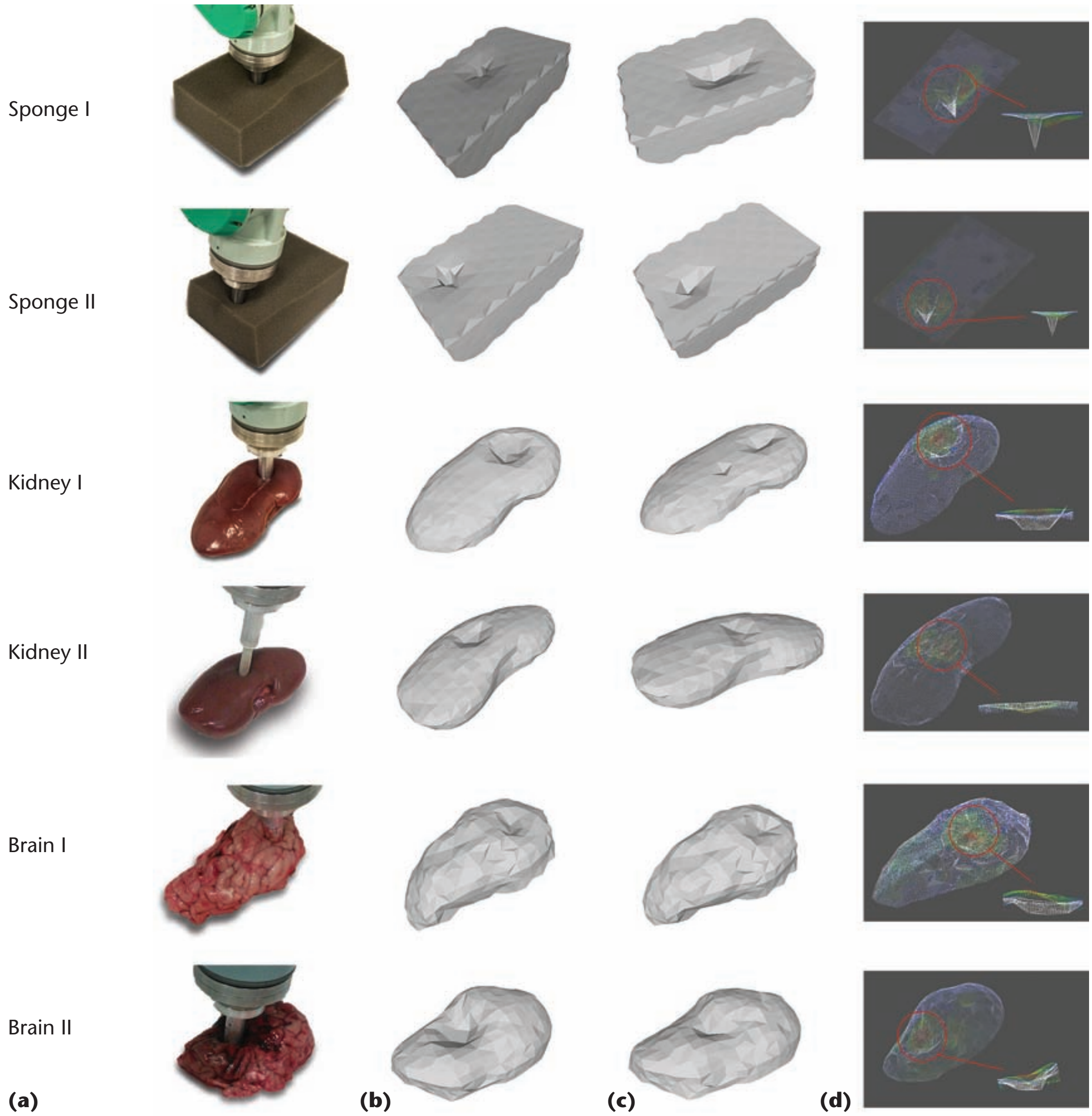

Figure 7. Comparison between the FEM formulation and Abaqus simulations for different models. (a) For each model, we show (b) the reconstructed mesh with Abaqus, (c) FEM returned mesh, and (d) a color map that represents the error between the solutions (extracted from CloudCompare software).

Table 3. Error values (in mm) between the online FEM simulation and scanner mesh.

Model	Mean error	Standard deviation	Max error
Sponge I	4.39	2.96	13.08
Sponge II	2.89	2.20	8.14
Kidney I	1.46	0.78	4.87
Kidney II	1.33	0.86	4.00
Brain I	2.26	2.00	11.68
Brain II	2.30	1.21	4.44

technical specifications, the 3D Scanner has an accuracy at 0.5 m depth resolution of 1 mm, and its spatial *x*/*y* resolution at 0.5 m is 0.9 mm.

Subsequently, we applied a subdivision filter to the surface for both the FEM and Scanner models in order to obtain two dense point clouds. Then, we defined the interest region for each point cloud by manually selecting the area where the deformation was being applied.

Once the two regions of the two point clouds were selected, we used the open source CloudCompare software to compute the error estimation. (An open source implementation of the CloudCompare application, which is used to manage and compare 3D point clouds, is available at www.danielgm.net/cc.) More concretely, we used two main functions to complete the process: register and distance procedures. The register procedure aligns two point clouds using the Iterative Closest Point (ICP) algorithm,[26] and the distance procedure calculates the distances between the two point clouds. The error was computed using the Euclidean distance from one point in the first cloud to the nearest point in the second cloud. Table 3 lists the mean errors (in mm), and Figure 8 shows the visual results of this experiment.

Our results show the error values are low enough to provide correct visual feedback. In these experiments, the visual feedback includes a gradient

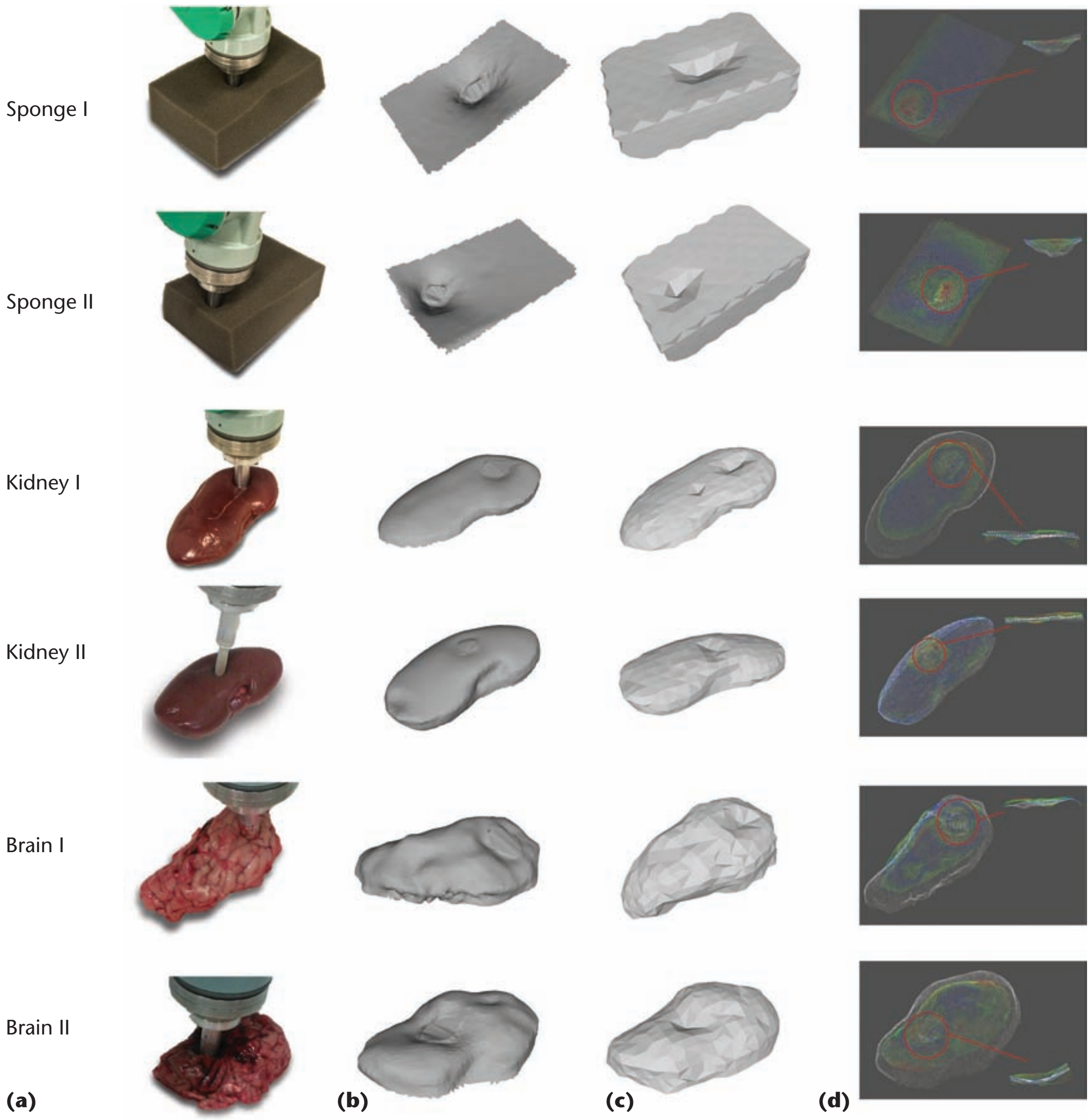

Figure 8. Comparison between the FEM formulation and scanner reconstruction for different models. (a) For each model, we show (b) the scanner mesh, (c) the FEM mesh, and (d) a color map that represents the error between both meshes (extracted using CloudCompare software).

color that illustrates the degree of deformation. Figure 9 shows the visual results for each of the deformations applied to the six samples alongside the ground truth.

Computation Time

Lastly, we computed the execution times of the object deformation module for all models. For the purposes of this evaluation, we did not perform an exhaustive study of the computation times for the marker tracking system or the matching step between the keypoints and the input point cloud. These results vary between 4 and 10 ms, depending on the model, which means the computational times are not a bottleneck for the method's performance. Thus, we have focused our attention on the physical module.

More concretely, Figure 10 presents the execution times of the mesh physical simulation step—that is, the FEM physical simulation step. To compute the mean times, we used calculations from 10 different times for the physical module and deleted the extreme values in order to discard the outliers. As Figure 10 shows, the execution times vary from 14.13 to 60.72 ms, depending on the model.

There are two reasons for this range of execution times. First, the execution time depends on the number of tetrahedrons for each model. As might be expected, the larger the number of elements, the longer the execution time will be. The sponge samples have the largest computational times because they have more tetrahedrons than the brain or kidney samples (see Figure 1). Second, the execution time depends on the time step set for each model. The smaller the time step, the longer the total execution time will be. The time step depends on the material's mechanical properties. In each case, we selected the time step to ensure the stability of the simulation.

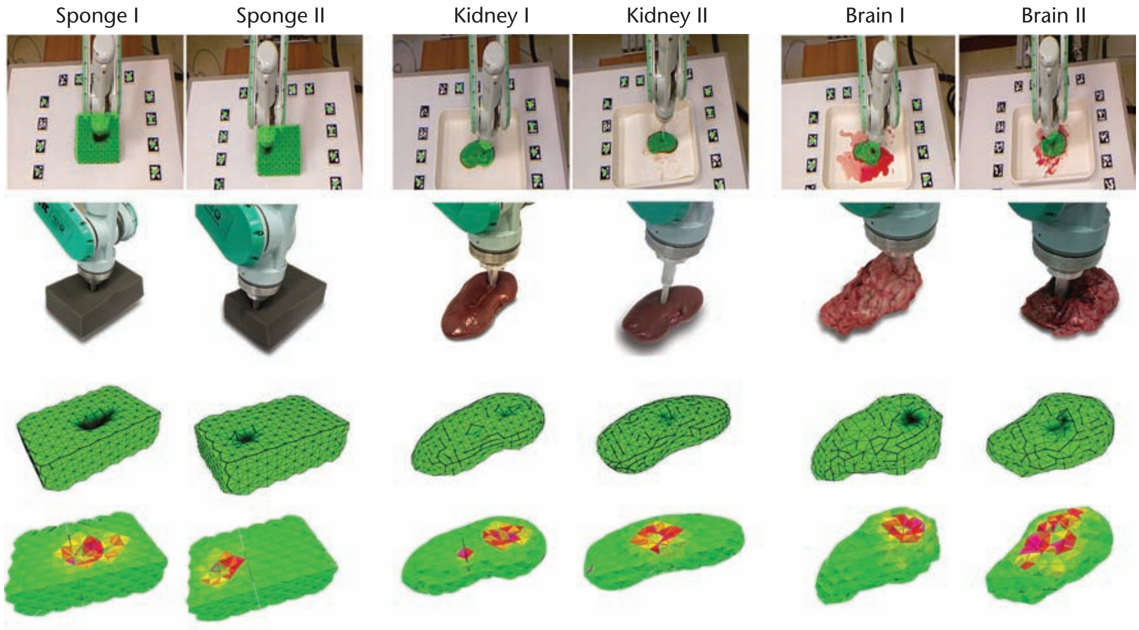

Figure 9. Visual feedback to the surgeon when the soft tissue is deformed in a robot-assisted procedure. The simulation results for the three types of examples (sponge, kidneys, and brains) were compared against the ground truth supplied by the 3D scanner reconstruction.

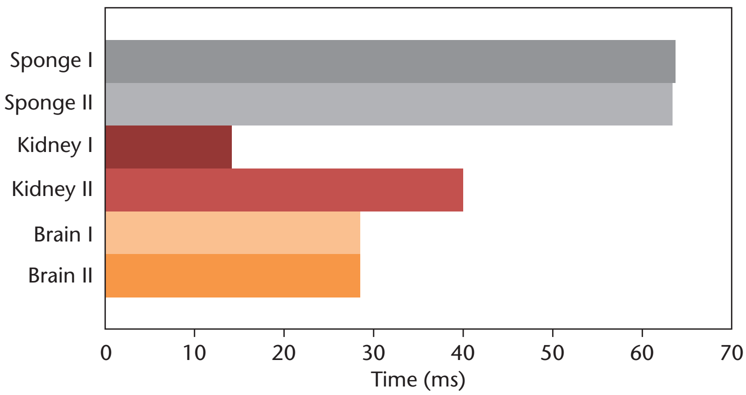

Figure 10. Execution times (in ms) of the FEM physical formulation for the different models.

Discussion

The deformation values obtained using the online FEM are smaller than those obtained by Abaqus software. This is because the input of the Abaqus simulation is the displacement recorded by the robot, and the input of the FEM is the displacement acquired by the cameras. However, these cameras do not provide the exact displacements because of occlusions caused by the tool or the material itself. These occlusions can also affect camera tracking (as with nontextured tracking systems). However, our approach can handle deformations with small occlusions (for example, the tools manipulating an object[9]). Furthermore, the modularity of the proposed pipeline will allow us to incorporate or replace certain modules. Thus, our framework's performance can be improved by using new tracking methods.[27]

In the same way, we can see that the deformations simulated with Abaqus are larger than those obtained by the scanner. This is because the scanner is not able to acquire the real deformation due to the occlusion caused by the tool. That is, the scanner acquires more accurate deformation than the cameras, but the deformation is less accurate than the one obtained with the simulation performed by Abaqus.

To improve the accuracy of the results, we could use higher density meshes, but this will increase the computation time. Depending on the concrete application where the method is applied, it will be necessary to determine the correct balance between precision and computation cost, as is the case for most real-time applications.

As we discussed earlier, the simulation approach is highly parallelizable, so the need for better accuracy could be solved using computers with a higher core count or with a GPU-based implementation. However, as Table 2 shows, the average errors obtained in the experiments with the current implementation are low enough for this approach to be considered valid and thus for it to serve as the basis for visual feedback in surgery.

In terms of accuracy and computational cost, the results of our experiments show that the proposed method returns a deformation for the tested objects that matches the theoretical and experimental results obtained. Thus, we were able to achieve a precision that enables the development of assistance surgery applications. In the future, the methodology developed can be enhanced to estimate the actual stresses on the tissues. This information could be used to prevent the robot from inflicting irreversible damage.

It may also be possible to automate the movements of the robotic arm. For example, the surgeon could select a point in the 3D model using the visualization software, and the robot then would automatically move to that point and perform an indentation.

References

1. M.A. Landeira, "Development of a General Purpose for Performance of Surgical Procedures," PhD dissertation, Univ. of Navarra, 2014.
2. S. Cotin, H. Delingette, and N. Ayache, "A Hybrid Elastic Model for Real-Time Cutting, Deformations, and Force Feedback for Surgery Training and Simulation," *Visual Computer J.*, vol. 16, 2000, pp. 437–452.
3. M.A. Landeira et al., "Desarrollo e implementación de una estrategia de gestión de singularidades para un sistema robótico redundante cooperativo destinado a la asistencia en intervenciones quirúrgicas" [Development and Implementation of a Singularity Management Strategy for a Cooperative Redundant Robotic System for Surgery Assistance], *Revista Iberoamericana de Automatica e Informatica Industrial* [Iberoamerican J. Automatic Control], vol. 12, no. 1, 2015, pp. 80–91.
4. T. Hayashi, F. de Sorbier, and H. Saito, "Texture Overlay onto Non-rigid Surface Using Commodity Depth Camera," *Proc. Int'l Conf. Computer Vision Theory and Applications* (VISAPP), vol. 2, 2012, pp. 66–71.
5. N. Shimizu et al., "Non-rigid Surface Tracking for Virtual Fitting System," *Proc. Int'l Conf. Computer Vision Theory and Applications* (VISAPP), vol. 2, 2013, pp. 12–18.
6. A.R. Fugl et al., "Simultaneous Estimation of Material Properties and Pose for Deformable Objects from Depth and Color Images," *Pattern Recognition*, 2012, pp. 165–174.
7. A. Jordt and R. Koch, "Direct Model-Based Tracking of 3D Object Deformations in Depth and Color Video," *Int'l J. Computer Vision*, 2013, pp. 1–17.
8. F. Brunet et al., "Monocular Template-Based Reconstruction of Smooth and Inextensible Surfaces," *Proc. 10th Asian Conf. Computer Vision*, vol. 3, 2011, pp. 52–66.
9. N. Haouchine et al., "Image-Guided Simulation of Heterogeneous Tissue Deformation for Augmented Reality during Hepatic Surgery," *Proc. IEEE Int'l Symp. Mixed and Augmented Reality* (ISMAR), 2013, pp. 199–208.
10. A.E. Kerdok, "Characterizing the Nonlinear Mechanical Response of Liver to Surgical Manipulation," PhD dissertation, Harvard Univ., 2006.
11. S.K. Warfield et al., "Real-Time Registration of Volumetric Brain MRI by Biomechanical Simulation of Deformation during Image Guided Neurosurgery," *Computing and Visualization in Science*, vol. 5, no. 1, 2001, pp. 3–11.
12. T. Kaster, I. Sack, and A. Samani, "Measurement of the Hyperelastic Properties of Ex Vivo Brain Tissue Slices," *J. Biomechanics*, vol. 44, no. 6, 2011, pp. 1158–1163.
13. Z. Gao, K. Lister, and J.P. Desai, "Constitutive Modeling of Liver Tissue: Experiment and Theory," *Annals of Biomedical Eng.*, vol. 38, no. 2, 2010, pp. 505–516.
14. R.J. Lapeer, P.D. Gasson, and V. Karri, "A Hyperelastic Finiteelement Model of Human Skin for Interactive Real-time Surgical Simulation," *IEEE Trans. Biomedical Eng.*, vol. 58, no. 4, 2011, pp. 1013–1022.
15. J. Hug et al., "Virtual Reality-Based Simulation of Endoscopic Surgery," *Presence: Teleoperators and Virtual Environments*, vol. 9, no. 3, 2000, pp. 310–333.
16. P.A.L.S. Martins, R.M. Natal Jorge, and A.J.M. Ferreira, "A Comparative Study of Several Material Models for Prediction of Hyperelastic Properties: Application to Silicone-Rubber and Soft Tissues," *Strain*, vol. 42, no. 3, 2006, pp. 135–147.
17. M. Sedef, E. Samur, and C. Basdogan, "Real-Time Finite Element Simulation of Linear Viscoelastic Tissue Behavior Based on Experimental Data," *IEEE Computer Graphics and Applications*, vol. 26, no. 6, 2006, pp. 28–38.
18. K. Laksari, M. Shafieian, and K. Darvish, "Constitutive Model for Brain Tissue under Finite Compression," *J. Biomechanics*, vol. 45, no. 4, 2012, pp. 642–646.
19. M. Hrapko et al., "The Mechanical Behavior of Brain Tissue: Large Strain Response and Constitutive Modeling," *Biorheology*, vol. 43, no. 5, 2006, pp. 623–636.
20. Z. Liu and L.E. Bilston, "Large Deformation Shear Properties of Liver Tissue," *Biorheology*, vol. 39, no. 6, 2002, pp. 735–742.
21. A. Wittek, T. Hawkins, and K. Miller, "On the Unimportance of Constitutive Models in Computing Brain Deformation for Image-Guided Surgery," *Biomechanics and Modeling in Mechanobiology*, vol. 8, no. 1, 2009, pp. 77–84.
22. G. Hamarneh et al., "Towards Multi-modal Image-Guided Tumour Identification in Robot-Assisted Partial Nephrectomy," *Proc. 2nd Middle East Conf. Biomedical Eng.* (MECBME), 2014, pp. 159–162.
23. N. Haouchine et al., "Single View Augmentation of 3D Elastic Objects," *Proc. IEEE Int'l Symp. Mixed and Augmented Reality* (ISMAR), 2014, pp. 229–236.
24. S. Misra, K. Ramesh, and A.M. Okamura, "Modeling of Nonlinear Elastic Tissues for Surgical Simulation," *Computer Methods in Biomechanics and Biomedical Eng.*, vol. 13, no. 6, 2010, pp. 811–818.

25. G. Irving, J. Teran, and R. Fedkiv, "Tetrahedral and Hexahedral Invertible Finite Elements," *Graphical Models*, vol. 68, no. 2, 2004, pp. 66–89.
26. P.J. Besl and N.D. McKay, "A Method for Registration of 3-D Shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, 1992, pp. 239–256.
27. Y. Park, V. Lepetit, and W. Woo, "Texture-Less Object Tracking with Online Training Using an RGB-D Camera," *Proc. 10th IEEE Int'l Symp. Mixed and Augmented Reality* (ISMAR), 2011, pp. 121–126.

Ibai Leizea *is a researcher in the Inspection and Measurement research line at IK4-Ideko. This work was performed while he was a researcher at CEIT-IK4. His research interests include computer vision, augmented reality, and photogrammetry. Leizea has a PhD in computer science from the University of Navarra. Contact him at ileizea@ideko.es.*

Ainitze Mendizabal *is a researcher in the Applied Mechanics Department at CEIT-IK4. Her research interests include physical simulation and modeling of soft tissues. Mendizabal has a PhD in mechanical engineering from the University of Navarra. Contact her at amendizabal@ceit.es.*

Hugo Alvarez *is a senior researcher in the Department of Computer Graphics and Interaction at Vicomtech-IK4. This work was performed while he was a senior researcher at CEIT-IK4. His research interests include computer vision and augmented reality. Alvarez has a PhD in computer science from the University of Navarra. Contact him at halvarez@vicomtech.org.*

Iker Aguinaga *is a senior researcher in the Applied Mechanics Department at CEIT-IK4 and an associate professor at the University of Navarra. His research interests include physical simulation and modeling of deformable objects, virtual and mixed reality, and computer vision. Aguinaga has a PhD in mechanical engineering from the University of Navarra. Contact him at iaguinaga@ceit.es.*

Diego Borro *is a senior researcher in the Applied Mechanics Department at CEIT-IK4 and an associate professor at the University of Navarra. His research interests include computer vision, augmented reality, collision detection, and haptic rendering. Borro has a PhD in computer science from the University of Navarra. Contact him at dborro@ceit.es.*

Emilio Sanchez *is a senior researcher in the Applied Mechanics Department at CEIT-IK4 and an associate professor at the University of Navarra. His research interests include robotics, haptics, and biomechatronics. Sanchez has a PhD in teleoperation and robotics from the University of Navarra. Contact him at esanchez@ceit.es.*