International Journal of Modeling, Simulation,
and Scientific Computing
(2023) 2350045 (25 pages)

DOI: 10.1142/S1793962323500459

Biomechanical modeling and simulation of a human organ using an augmented reality technique during open surgery

Aicha Ben Makhlouf*, Nessrine Elloumi†, Borhen Louhichi‡,§,‖,
Mehdi Jaidane¶, Nashmi Hassan Alrasheedi‡
and Kholoud Sandougah‡

*LATIS, ENISO, Sousse 4023, Tunisia

†SETIT, ISBS, Sfax 3038, Tunisia

‡Imam Mohammad Ibn Saud Islamic University
Kingdom of Saudi Arabia

§University of Sousse (ISSATSo, LMS), Sousse 4023, Tunisia

¶Sahloul University Hospital, Sousse 4011, Tunisia
‖borhen.louhichi@issatso.u-sousse.tn;
blouhichi@imamu.edu.sa

Received 19 July 2022
Revised 24 February 2023
Accepted 29 April 2023
Published

Basically, augmented reality (AR) technology grants innovative ways so as to manipulate and visualize a three-dimensional (3D) model of an object through superimposing computer-generated images onto another object interactively. Being able to interact in real-time with digital and spatial information offers further chances in order to manipulate medical data efficiently and easily. In fact, surgeons encounter multiple challenges handling digital patient data during surgical interventions. Multiple techniques are invested in order to visualize the operative areas, as for example fluoroscopy and ultrasound methods. The latter display certain deficiencies. Therefore, the AR technique can stand for a good candidate in order to project a 3D model of a target organ into the surgeon's perspective and view field so as to enhance the efficiency and accuracy of the medical intervention intraoperatively. The basic target of this paper is to simulate a generated biomechanical model of the liver organ and visualize its deformations through the use of an AR headset during the open surgery. The proposed approach is validated by the use of acquired CT scans of a human liver organ.

Keywords: 3D model; augmented reality; real-time; medical intervention; meta vision 2; human liver organ; finite element model; CT scans.

‖Corresponding author.

1. Introduction

For several years, health care has witnessed an overwhelming technological involvement yielding extensive research and whetting the widest interest allowing deeper insight particularly in medical imagery and surgical techniques. Within this framework, minimally invasive surgery (MIS) proves to be highly beneficial in terms of ongoing surgery and post-surgery for patients through reducing the risks of infections and bleeding incurred through the investment of instruments. Augmented reality (AR) for MIS combines multiple techniques and tools providing direct visualization during open surgery so as to facilitate and guide surgeon's navigation during surgical procedures. Unfortunately, several AR systems applied in laparoscopy consider only a rigid registration of the preoperative reconstructed model[1] and do not take into account model deformations. As a result, the efficiency of AR technology is reduced. In fact, the patient's position changes within the acquisition of the preoperative images and during the operation referring to the fact that the table is often tilted to allow the abdominal viscera to slide to the bottom of the abdominal cavity and to facilitate the access to the liver. In addition, the heartbeat can also trigger displacement in idle position. Furthermore, the instruments can move and deform the target organ, which requires new AR-based methods to visualize these deformations during the surgery operation.

Ultimately, it is noteworthy that AR integration in image guided surgery can revolutionize the workflow of visualization and interaction of the surgical preoperative data. This methodology provides a robust and promising method to visualize surgical information during the preoperative phase and to project the model during the intraoperative phase using real-time model registration in open surgery. This technique helps surgeons take better and more accurate decisions during surgical operations. Contrarily to the nonrigid registration methods,[2,3] the preoperative model projected on an intraoperative target organ surface imposes the presence of multiple fixed cameras. However, using multiple cameras, this method complicates the visualization and navigation during the surgical operation. As a result, this traditional method is not efficient enough and cannot fulfill optimal accuracy. This research work foregrounds a new AR methodology to simulate a patient liver model during an open surgery. The elaborated technique aims to not only project but also visualize a 3D reconstructed biomechanical model of a specific-patient liver taking into account intraoperative deformations with fixed parameters as well as to carry out a nonrigid real-time registration of the 3D model during surgery.

2. State of the Art

During MIS, the ongoing procedure is undertaken through the use of long sticks incorporated through the abdomen where the operating target area is visualized using a laparoscopic camera. Since the long stick manipulation is delicate within the small incision, the visual feedback is usually limited. Subsequently, challenging operations require more precision and verification. Nevertheless, the rise of

interest in AR technology application in the medical field by researchers proved to be able to solve many of the regular methods limitations, in the same manner that they introduce difficulties involving concealing a part of the organ visibility in the intraoperative phase. Hence, contrarily to laparoscopy surgery, AR in open surgery, especially when the organ is mostly visible, can offer higher accuracy and better data visualization than other mentioned methods. From this perspective, this section addresses several studies related to the current work. The pipeline starts with the liver organ segmentation from preoperative CT data and MRI images. Afterwards, 3D model reconstruction approaches are discussed. Subsequently, multiple studies are displayed to generate a biomechanical model so as to visualize the deformations of the soft-tissue hyper elastic liver organ based on the organ physiology and studied constraints. Additionally, other studies report various approaches in terms of the registration process of the biomechanical model onto the AR projection view. These approaches consider numerous algorithms to accomplish model alignment and rely upon intraoperative organ deformation in real-time.

2.1. *AR history*

AR has been regarded as a suitable technology for computer assisted surgery. Basically, the rapid increasing performance of computers and chips, advances in surgical vision and the ability to compute virtual organs preoperatively oriented the scientific community to opt for this specific path of research and therefore develop several methods to incorporate AR in surgery. AR refers to the synthesis of real and virtual imagery. Contrarely to virtual reality (VR) where the user is absorbed in an entirely artificial world, AR overlays additional information on real scenes.

Suthau *et al.*[4] asserted that AR is a successor of VR while Kipper *et al.*[5] and Azuma[6] handled AR as a further variation of VR rather than a successor. Besides, he emphasized that three characteristics are needed in order to consider an AR realistic environment, namely, a mixture of real and virtual information, a real-time interaction and a conducted simulation in a 3D environment.

As a matter of fact, Azuma *et al.*[6] offered a definition for AR technology, combining real, virtual, interactive and real time 3D registration. In the MEDical Augmented Reality for Patient (MEDARPA), Wesarg *et al.*[7] attempted to improve MIS by developing an AR system using markers superimposed to the patients' body to evaluate rigid transformations between preoperative and intraoperative images. Marescaux *et al.*[8] asserted that the first real-time enabled AR assisted laparoscopic adrenalectomy during the intraoperative phase that involved manually assisted deformable registration.

Most of these works were dedicated to implement this technology in numerous fields that can be perceived today as being available to the public, at the level of manufacturing, advertising, navigation and medical field. However, it still raises various challenges especially in the medical area, since adopting such technologies entails unsolved problems of assessment of computer assisted surgery optimization

results. Among the most prominent ones, we mention conformity of the elastic deformations constraints to the real organ deformations owing to breathing or even in view of the interaction of the organ with the surgeon's tools. Moreover, performing real-time registration of 3D models imposes the continuous development of methods and optimizations by researchers to obtain optimal and satisfactory result's as well as assist surgeons to make the right decisions.

2.2. *Projection and registration of biomechanical models*

As far as the topic of the preoperative and intraoperative registration data during Minimally Invasive Hepatic surgery is concerned, Haouchine *et al.* in Ref. 9 invested AR, coupled with the preoperative computed data models onto the endoscopic view during hepatic surgery. Their method rests on detecting the liver boundaries such as the pneumoperitoneum using a 3D point cloud segmentation. Then, the biomechanical model was generated and the rigid alignment was computed using finite element method (FEM). The alignment results were obtained using various algorithms such as the probabilistic approach and coherent point drift (CPD) for nonrigid registration. It revealed the lowest distance error providing more accurate registration with minimal computational time. However, this method displays a certain shortcoming referring to the tiny area covered by the laparoscopic camera. In order to overcome the above limitation, Plantefève *et al.*[10] introduced a geometrical reconstruction of the organ's internal structures, namely, the vascular network and the parenchyma as in the model mesh acquired from CT scans and MRI images. It involves performing the liver segmentation called skeletonization calculated through the use of an algorithm resting on Voronoi diagram and executing the Dijkstra minimum cost spanning tree algorithm so as to correct model inconsistency of the inner structures.

Next, the generation of an improved segmented map from anatomical boundary conditions was carried out using numerous iterative algorithms and position estimation to get a complete 3D tetrahedral reconstructed model. Furthermore, as soon as the obtained composite model and 3D point cloud were extracted from the stereo-endoscopic camera images, registration steps were undertaken. The obtained results revealed an improved initial registration of the liver model projection under cardiac motions during the elastic registration with an optimal estimation of position of internal structures. The transfer of the anatomical boundary conditions is extremely accurate and the quality of the registration abides by numerous variables including sensitivity to the medical images quality in the preoperative imaging liver segmentation phase.

Suwelack *et al.* in Ref. 11 developed a real-time physics-based shape matching (PBSM) method for matching a preoperative model to an intraoperative surface. The elaborated algorithm relies upon surface registration that is nonrigid with electrostatic-elastic characteristics. The biomechanical model's attributes are determined and calculated using a FEM. This method is founded on mathematical

modeling of the electrostatic-elastic registration methodology and its interconnected scheme uses surface data from the intraoperative model and volumetric data from the preoperative model. The proposed algorithm was applied investing the Simulation Open Framework Architecture software (SOFA framework). Deformed liver meshes were derived to act as input for the shape matching process.

The PBSM approach performs substantially better in terms of accuracy than the CPD algorithm, whereas even on the surface errors, it is highly dependent on the biomechanical-based regularization of PBSM and most importantly presents a multicore hardware setup for parallel problems. Santhanam *et al.* in Ref. 12 handled a method to compute physically elastic real-time deformation based on deformable anatomical organs such as a high-resolution lung model utilized during the experimentation in an AR environment. Applying a physically-based deformation method, they reconstructed a surface model obtained from a human dataset applying a linear force balance parametrized with the differential stiffness matrix, the displacement vector for each vertex and the force as a vector. Their proposed method rectified the Hauth's iterative solution reported in Ref. 13 through substituting the exponential function for a transfer function to model variations in mesh connectivity of the 3D model representation and anisotropic elasticity. Furthermore, the model definition was constrained using a nodal mass-spring model with a converging iterative method that calculates iteratively the stiffness matrix values until they reach zero or become constant. To calculate the displacement vector for each point in a 3D mesh, an explicit Euler's integration approach in Ref. 14 was selected to calculate, for each node, the force acceleration as well as the transferred force for the neighboring nodes in relation to their masses.

The simulations displayed faster convergence than the exponential one for a node that other methods mentioned in Ref. 13 and a pre-computed steady elastic deformation for their imported high-resolution lung model. Hence, the faster convergence for each node yields faster deformation simulation that is favored for AR development.

Bertrand *et al.* in Ref. 15 set forward a software that uses the AR technology to enact 3D visualizations of the preoperative data to the surgeon. Indeed, patients from the Hospital of Clermont-Ferrand were involved in the development of the Hepataug software. Hepataug loads the liver model obtained from CT and MRI data and snaps visual cues from a single laparoscopic image obtained during intraoperative phase for registration to overlay a computed deformable model of the liver onto the abdominal cavity in a separate screen in the operating room. Adagolodjo *et al.* in Ref. 2 identified a semiautomatic registration method of 3D deformable models using two-dimensional (2D) shape outlines or silhouettes extracted from a monocular camera view of a set of fixed cameras surgery setup. In this regard, the model rested upon the combination of a biomechanical model of the organ having a set of projective constraints affecting the deformation of the model. The method relied on a nonlinear biomechanical finite element (FE) model coupled with projective constraints to settle the constraints issue. In laparoscopic surgery, the

laparoscopic view is partial and uniquely a sub part of the organ's contour is visible where the presented method can handle incomplete contours having a sufficient part of the organ visible and showing convergence to a realistic configuration. In order to provide visual clues of the organ shape, the surgeon was asked to draw colored straight lines on the organ in order to specify the curvature of the surface before the registration. After the initial registration, the curvature of the texture was almost consistent with the observation pertaining to the medical images.

Subsequently, the results revealed that this approach facilitates the computation of the nonrigid 3D pose while resting uniquely on 2D data and elastic registration. In spite of relying on a single-view image obtained from a monocular endoscopic camera to carry out the alignment of the model and initial registration, the authors' method aims to illustrate AR of the intraoperative phase of liver surgery using a monocular laparoscopic camera.

Müller *et al.*[16] identified a marker-based approach that uses AR to generate a superimposed image of the kidney to facilitate percutaneous nephrolithotomy (PCNL) treatment. The setup consists of an iPad based-method with a camera positioned above the patient to register the field of intervention through a set of markers fixed on the patient's skin. The proposed method invests a phantom model to substitute the human kidney where markers are attached on the surface of a gelatine block containing the phantom model. These navigation markers are captured using the tablet that implemented image streaming and processing with the Medical Imaging and Interaction Toolkit (MITK) on both the server and the tablet. Furthermore, the system relies on streaming and exchanging data between the tracking server and the tablet via a WiFi network connection. The process works as follows: the expert scans the patient via the tablet and sends the data to the server after the compression process. Next, after the decompression step within the server side, the image is analyzed to search for the center points of the placed navigation markers through a developed algorithm.

In order to specify the orientation and the translation corresponding to the object model, a camera pose estimation was used through an approach called the perspective-n-point problem (PnP) governing the appropriate choice of PnP solver. Subsequently, they arranged multiple cameras pose estimation algorithms were arranged and tested so as to assess their accuracy with the proposed system. Thus, the conducted experiment comparing the proposed system with other methods implementing ultrasound and fluoroscopy techniques, demonstrated that the process was accelerated by 20% using the iPad technique compared to 14% using ultrasound. Despite these results, a slight change in radiation was also recorded during the process.

3. Proposed Methodology

3.1. *Approach overview*

In contrast of its use in laparoscopic surgery, AR in open surgery can provide unlimited implementation possibilities. In the case of hepatic surgery, visualizing

the organ data intraoperatively is primordial to visualize the liver inner vessels, veins, tumor locations and parenchyma. As the surgeon attempts to cut a portion of the liver for reimplantation for example, the inner structures are not visible and require a review of the preoperative data so as to estimate the cut location, which is the same case in radiotherapy. As a matter of fact, an accurate biomechanical modeling of the liver organ can overcome these procedure deficiencies. Moreover, the liver model does not realistically have a linear deformation behavior of its elastic tissues since the inner structures preserve some of the applied force, along with external body movements during the surgery. From this perspective, basic linear trigonometric modeling and elastic deformation are not accurate enough to estimate the nonrigid deformation and simulation of deformable soft-bodies such as the liver organ of any other soft tissue model. For this reason, an appropriate biomechanical modeling is elaborated and discussed in this paper.

This proposed method aims to visualize preoperative organ data in real-time by means of biomechanical 3D model rebuilding from a pre-processed reconstruction of the liver organ given CT and MRI images. The generated biomechanical model

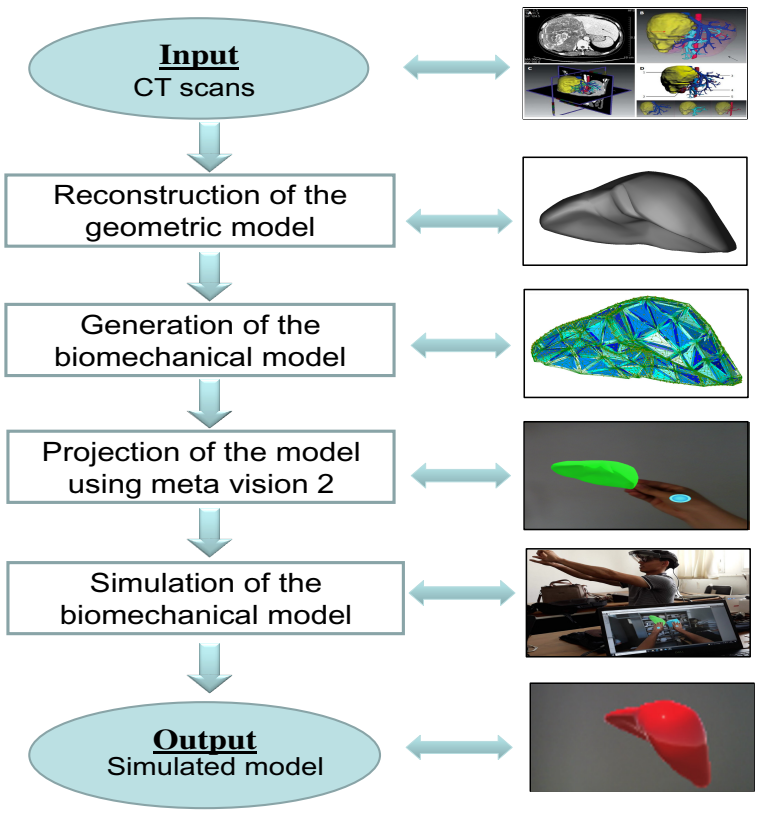

Fig. 1. Overview of the proposed approach.

is configured on the Unity environment and the projected model is simulated in three-dimensional (3D) space using the Meta 2 AR headset. The projected model takes into action the model geometry by tetrahedralization. Moreover, a new algorithm that combines the mass spring (MSM) model system and FEM properties is set forward in order to visualize model deformations generated by the user hands interactions. Hence, the physics computation inside the simulation engine relies on a solver that updates the applied forces, resolves the constraints as well as collisions and estimates updated mesh model vertex positions on multiple iterations on each physics frame.

The suggested method depicted in Fig. 1 rests upon several steps starting with initial model acquisition and reconstruction followed by the generation of the biomechanical model displaying the elastic properties of the surface model. Ultimately, the model is projected on AR space and the model interaction in addition to its corresponding elastic deformations simulation are observed.

3.2. *Reconstruction of the geometric model*

Before the biomechanical model generation, the model requires the mesh of the liver organ acquired in a model pre-processing phase followed by an extraction and segmentation phase. The final result is a mesh of the liver organ obtained from a model reconstruction step.

3.2.1. *Model pre-processing*

To reconstruct a 3D model of the liver organ, a deep learning technique needs to be applied. This technique implements a convolution neural network (CNN) fed with the input of a dataset of volumetric images having a digital imaging and communications in medicine (DICOM) format of the patient's scanned organ. The training procedure of the CNN network outputted a group of segmented images of the liver organ.

The used 3DIRCADb dataset[17] corresponds to an open dataset that contains 3D CT data from different types of liver organs varying with a large diversity and complexity of lesions.

3.2.2. *Model extraction and segmentation*

For the segmentation process, a deep learning technique using the U-net architecture is applied to obtain segmented images of the liver images from the group of volumetric images. This step relies on several sub-steps. First, data pre-processing operations are executed on the specified input as a set of images, performing image resizing and applying the windowing technique with Hounsfield Units measure to recompute pixel intensity. Subsequently, a separation of the CT images with their corresponding mask is enacted. Finally, the segmented images highlighting the liver region are then fed into the mentioned CNN model.

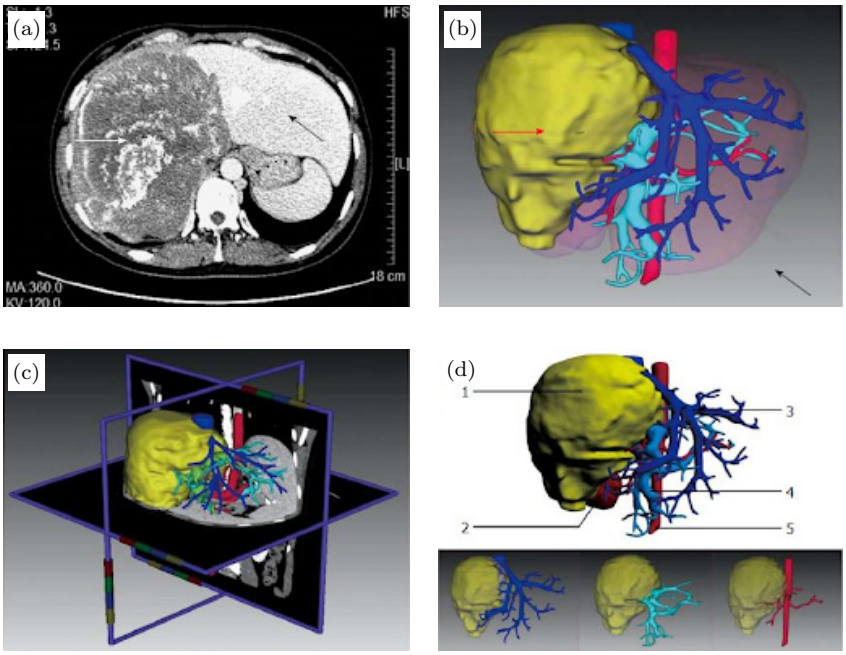

Fig. 2. Reconstruction of a model from input segmented images.[19]

The final result stands for a group of segmented images used to reconstruct the 3D model in the following step.

3.2.3. *Model reconstruction*

The 3D model reconstruction is generated using an open-source software for rebuilding computed tomography and magnetic resonance images developed by a Brazilian research team named Invesalius 3.[18] The software pipeline provides the capability to import multiple DICOM files, analyze them, export resulting ones to the STL, OBJ, volume rendering file formats and manual image segmentation.

The resulting reconstructed 3D model is exported as an OBJ format file to be post-processed by polishing any surface imperfections, which prepares for the next step (Fig. 2).

3.3. *Generation of the biomechanical model*

3.3.1. *Model representation*

After the generation of the smoothed surface corresponding to the liver model, and in order to be able to effectively display an organized mesh topology, the model geometry is checked if it represents a closed volume through the use of a

library called TetGen.[20] The suggested method invests TetGen based on a Delaunay triangulation (DT) algorithm to recompute a reconstructed clean mesh topology and triangulation.

The DT algorithm (reconstruction algorithm) performs triangulations as a subdivision of a volume into triangles or tetrahedrons and exhibits the property that the circumcircle of every triangle (or tetrahedron) does not involve any points of the triangulation.[21]

Subsequently, the reconstructed model needs to undergo several steps so as to acquire its mechanical elastic properties and simulate the model behavior inside the physics engine.

3.3.2. *Model tetrahedralization*

Model tetrahedralization is conducted using a compiled library with a C# assembly wrapper importable and usable inside unity scripts. This library is named TetGen.[20] It corresponds to a software that is also available as a library, designed to generate tetrahedral meshes for 3D weighted points and can produce the Delaunay and weighted Delaunay power diagram. In addition to the methods of acquisition of a 3D polyhedral domain, TetGen generates the constrained or incremental Delaunay tetrahedralization (DT) using the Delaunay algorithm and an isotropic adaptive tetrahedral mesh (Fig. 3). The edges and faces of the mesh (also referred to as domain boundaries) are checked and can be preserved in the resulting mesh if chosen by the input behavior. The TetGen library allows an easy multi-platform integration since it is expressed in C++ and uses uniquely the C standard library. Owing to the computational efficiency that the library offers, TetGen does not have specific requirements to compile as it is fast to build and can be invited from external sources.

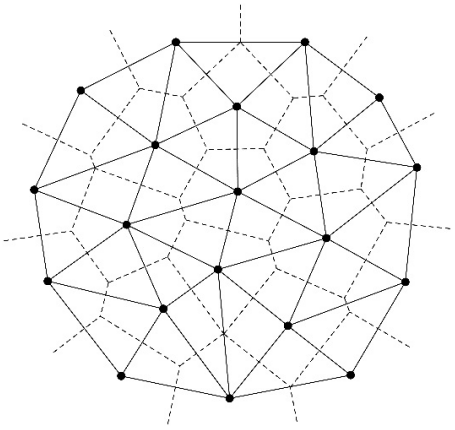

Fig. 3. Mesh triangulation with DT.

In our case, the function of unity scripts is invested through the main method of the wrapper when compiled as a dynamic library.

In this respect, the incremental DT algorithm variation is incorporated in the tetrahedralization step. Therefore, this library's role seems to be crucial to identify the mesh geometry and possess a parametrized model.

As part of the TetGen behavior execution with the incremental DT approach, the level of multiple parameters can be adjusted to generate a quality mesh

- Tetrahedralize a Piecewise Linear Complex (PLC).
- Reconstruct a prior generated mesh.
- Mesh refining to enhance its quality.
- Mesh coarsening (decrease the mesh elements).
- Output mesh data.

These parameters refine the optimized mesh triangles, count and output a proper mesh.

The DT within TetGen is implemented relying on two approaches, namely, the incremental approach of Bowyer–Watson (BW)[22] and the incremental flip algorithm of Edelsbrunner and Shah.[23]

Grounded on the input object and the specified behavior, the output of TetGen is either a DT or a quality refined tetrahedral mesh.

Example: The output of TetGen for the reconstructed liver mesh illustrated in Fig. 4 is: 1025 nodes, 5254 links, 2046 faces and 3207 tetras.

3.4. *3D model projection using Meta 2 AR*

In order to project the model in 3D space and to guarantee its interaction with Meta2 AR headset, a collision model component seems to be intrinsic for

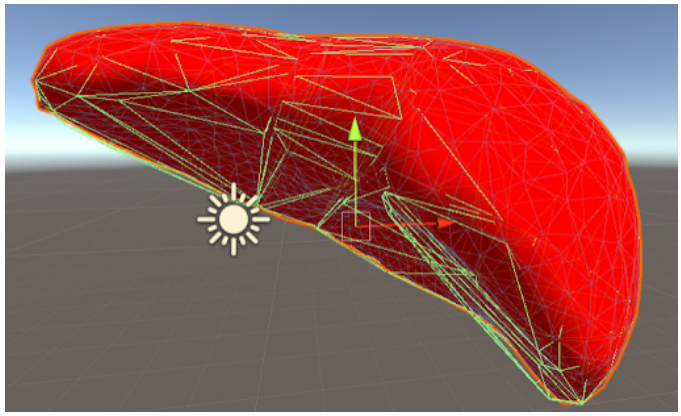

Fig. 4. Tetrahedralized model in unity.

this interaction. As simpler collision shapes (such as a box collider) can enable translational and rotational interactions, whenever the mesh simulation changes by deformation, the collision model becomes inadequate and will therefore neither integrate the model nor allow precise future interactions. Thus, determining a new collider with moderate accuracy can result in a more accurate interaction as the model deforms at runtime.

3.4.1. *Meta requirements*

Apart from the core scene components with Meta 2 AR, each object needs to refine components that require:

- Rigidbody component: With the parameters of Mass, Drag, angular drag, IsKinematic set to false, UseGravity set to true.
- Collider component: Allowing similar scene entities that have colliders to interact and entail the resulting behavior.

3.4.2. *Collision model — V-HACD library*

To be able to represent our model collision system, inserting classic convex mesh colliders to the model reduces the interaction accuracy as the collider incorporates the model largely and can transform the collider into a similar primitive mesh. Hence, convex mesh colliders are not suitable for large, complex and highly subdivided models since they require a mesh triangle count to be equal or less than 255 triangles.

As a result, experimentations were undertaken using several methods, which led to the choice of the volumetric hierarchical approximate convex decomposition

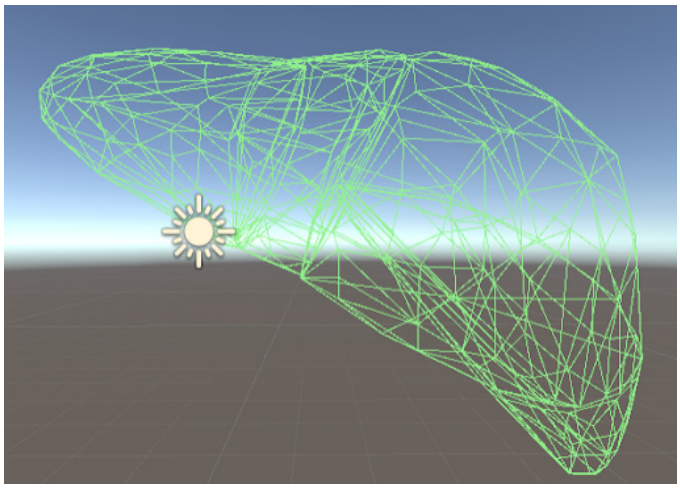

Fig. 5. Liver organ collision model decomposition of convex parts.

library (V-HACD library).[24] Since rendering techniques decrease the computation time for relatively large meshes by removing unused segments of the mesh, their physics simulation can be influenced in terms of accuracy and time taken to carry out additional computation of these complex meshes. As a matter of fact, decomposing a nonconvex mesh into a small set of approximately convex parts using the whole mesh's convex hulls which can be invested as a piecewise convex approximation of the original mesh. In Fig. 5, a visual decomposition of the liver model is foregrounded.

From this perspective, the computation of the group of convex meshes can be performed rapidly and easily. Indeed, it helps maintain the real-time aspect of the transformation of the model when deforming the model interactively as well as reduce the load on the physics simulation when the recalculation of the collision model occurs. Therefore, the above stated result constitutes a powerful impetus for decomposition of the liver organ collision model into a set of convex parts.

3.5. *Simulation of the biomechanical model*

3.5.1. *MSM model simulation*

MSM models are extensively investigated in computer graphics for simulation of various models such as clothes or deformable solids and it was only recently that its application in softbody simulation has been tackled and explored (Fig. 6). The MSM model can be applied for softbody simulation of the soft elastic tissue of the liver organ during open surgery. Additionally, the deformation computation of

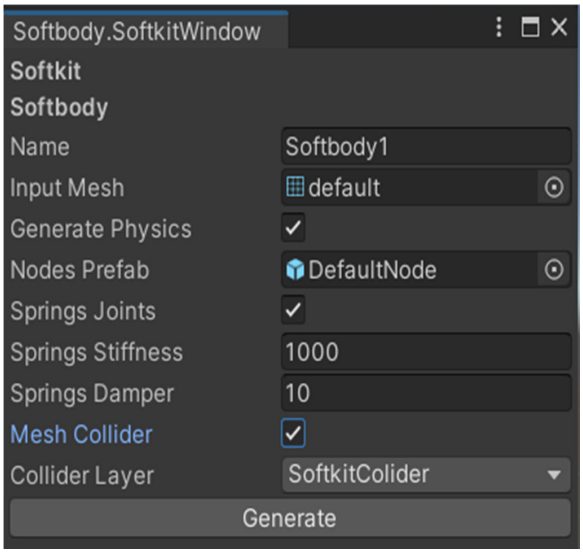

Fig. 6. Liver MSM model generation.

Algorithm 1. MSM Model Preparation Algorithm.

Input:InputMesh
StiffnessValue
DampingValue
Result: ClonedMesh
ListOfRigidbodies
ListOfSprings
Begin
vertices ← *InputMeshvertices*
vertexCount ← *verticescount*
triangles ← *InputMeshtriangles*
triCount ← *trianglescount*
/*Tetrahedralize using TetGen resulting mesh */
data ← Tetrahedralize()
InitializeTetMesh(*data*)
For i ← 1 **to***TetMeshNodesCount***do**
TetMeshNodePositions[i]← $\vec{V}$[i*3 + 0][*i**3 + 1][*i**3 + 2]
End
For i ← 1 **to** *TetMeshEdgesCount* **do**
TetMeshEdges[i] ← *dataEdges*[i]
End
For i← 1 **to** *TetMeshFacesCount* **do**
TetMeshTriangles[i]← *dataFaces*[i]
End
For i← 1 **to** *TetMeshTetrasCount* **do**
TetMeshTetras[i]← *dataTetras*[i]
End

AddTetMeshComponent()
/* Generate the rigidbodies and spring joints of the biomechanical model */
GenerateNodesAndSpringJoints()
/* Create a mesh collider that will be dynamically updated */
GenerateMeshCollider()

End

the model is challenging referring to its high dependency on the parameters of the stiffness and damping of the springs between the nodes and assuming that these node masses are equal to one unit in unity. Furthermore, the springs contractions allow the nonrigid model to possess some form of rigid behavior when contracting, retaining the model volume each time a deformation occurs.

Accordingly, the springs movement and elasticity are assessed grounded on the Newtonian dynamics.

3.5.2. *MSM model*: *Topology preparation*

To start with, the liver tetrahedralized model is generated and its geometrical topology is refined from the reconstructed model using the compiled assembly TetSharp Tetrahedralize method, which invites the native methods of TetGen inside a C++ wrapper. The input is a reconstructed mesh with the options: PLC, refine quality and coarsen checked in the TetGenBehavior parameter.

Alternatively, the input mesh might not have a clean topology such as a double vertex self-intersection or disconnected open edges, which can generate errors and might require the use of additional tools to retopologize the mesh.

Algorithm 2. MSM Generate Nodes and Spring Joints method.

```
Begin
    ListOfRigidbodies[TetMeshNodesCount]
    MSMModelRigidbodies← ListOfRigidbodies
    MSMModelStiffness← StiffnessValue
    MSMModelDamping← DampingValue
    AddMSMModelDamping←DampingValue
    AddMSMComponent(ListOfRigidbodies)
For i←0 to TetMeshNodesCount do
    gameObject← newGameObject
    gemaObject.Position←TetMeshNodePositions[i]
    rigidbodies[i] ← gameObject.Rigidbody
End
ListOfSprings[TetMeshEdgesCount]
For i ← 0 to TetMeshEdgesCountdo
    link← TetMeshEdge[i]
    idSource← link.id1
    id2 ← link.id2
    rigidbody1 ← ListOfRigidbodies[id1]
    rigidbody2 ← ListOfRigidbodies[id2]
    SpringJoint← AddSpringJoint(rigidbody1)
    SpringJointConnection← rigidbody2
    SpringJointSpring← StiffnessValue
    SpringJointDamper← DampingValue
    ListOfSprings[i]← SpringJoint
End
MSMModelSpringsCount← TetMeshEdgesCount
MSMModelSpringJoints← ListOfSprings
End
```

The output of the Tetrahedralize method stands for a mesh containing the nodes, edges, faces and tetras data saved to a TetMesh model. This mesh will be stored in a TetMesh model that is added to the mesh gameobject components list.

After tetrahedralizing the mesh with TetGen, the resulting mesh geometry is redefined to be able to generate rigidbody nodes and their corresponding spring joints. The following algorithm highlights and details how the MSM Model is constructed and prepared for the simulation.

In order to model the spring joints between the nodes, a mapping is performed and projected into the MSM model. This mapping occurs between all the nodes on the surface model where the edges extremities of each face (can be a triangle or a tetrahedra), two distinct nodes, are linked through a spring joint with the specified spring stiffness as well as damping values and the spring rest length.

The TetMesh model is used to store the mesh definition. It holds data about the node positions, triangles, tetras and edges at all times and is referenced in a MSM model entity.

Additionally, the resulting biomechanical model plotted in Figs. 7 and 8 refers to the MSM model which holds the springs constraints definition, the global stiffness and damping values of 1000 and 10 units, respectively. Moreover, the model also accounts for the list of rigidbodies created at the node positions. To clarify

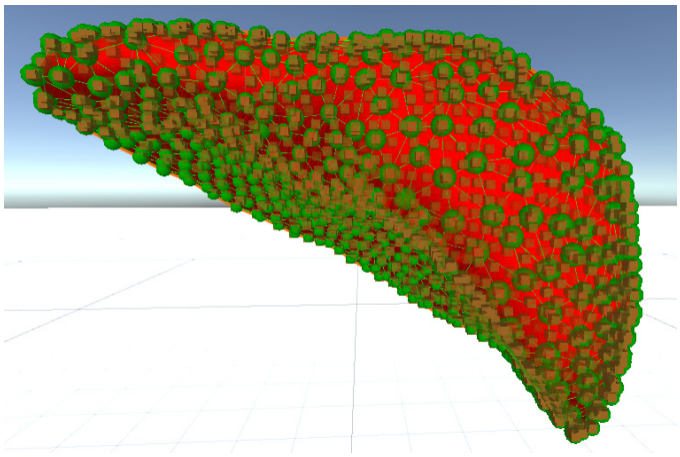

Fig. 7. Liver model generated in unity environment.

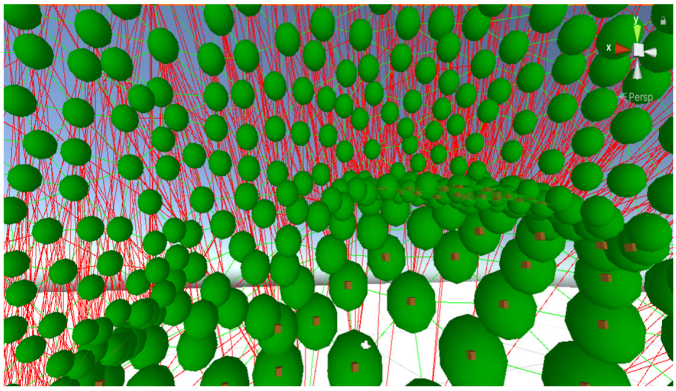

Fig. 8. The resulting model nodes and joints.

this notion, a Rigidbody corresponds to a property with a mass when added to any object, allowing it to interact with fundamental physics behavior, like forces (gravity, external forces) and acceleration. In fact, the rigidbodies are the core-computed elements of the simulation that possess a mass in the network of springs within the model.

3.5.3. *MSM solver*: *Displacement computation*

To simulate the generated biomechanical model, the presence of another core component is highly required. The MSM solver is the component that computes the simulation. Basically, the solver is based on a number of solver iteration steps implemented within the physics simulation FixedUpdate method in unity.

Algorithm 3. MSM Model Solver Algorithm.

```
Data: InputMesh
Begin
    MeshVertices[] ← InputMeshVertices
    MeshNormals[] ← InputMeshNormals
    Mappings[InputMeshVertexCount]
    maxSearchDistance = 0.00001 f
    /* OnStart method of Unity MonoBehavior
    OnStart
        For i ← 0 to InputMeshVertexCount do
            vertexVector←MeshVertices[i]
            FoundMapping←false
            minDistance = 100000.0 f
            minId=0
            For j ← 0 to MSMModelRigidbodiesCount do
                dist←CalculateDistanceBetween(vertexVector, MSMModelRigidbodies[j].Position)
                If (d < minDistance) then
                    minDistance =dist
                    minId= j
                End
            End
            If (minDistance <maxSearchDistance) then
                    Mappings[i] ← minId
                    FoundMapping ← true
                End
            End
        End
        /* FixedUpdate method on Unity MonoBehavior
        FixedUpdate
            For i ← 0 to InputMeshVertexCount do
                MSMRigidbodies[Mappings[i]].UseGravity ← true
                MeshVertices[i] ← MSMRigidbodies[Mappings[i]].Position
            End
            InputMeshVertices ← MeshVertices
            RecalculateMeshNormals()
            GenerateMeshCollider()
    EndLoop
End
```

Notably, when the simulation starts, this component, specifically the FixedUpdate loop, solves the resulting spring joints network when exposed, for instance, to gravitational force and the external spring forces are applied to the model.

Departing from the TetMesh model, each mesh vertex corresponds to a mapped node positioned near from the original vertex calculated through the minimum distance between each pair originating from the original mesh and the reconstructed mesh, respectively. As a result, these mappings are kept in track in a mappings array of indices.

First, the mesh rigidbodies' positions are updated to their predicted positions via Unity's FixedUpdate physics computation. Second, the rigidbodies' estimated positions reflecting the visualized mesh normals are recalculated to account for mesh faces movement and inversion if the applied force is too powerful. Eventually, the collision model of the mesh is recalculated independently for every physics frame.

As each deformation is not permanent, the spring joints will naturally contract due to the stiffness factor and each spring will return to its original rest length, entailing the return of the model to its approximate idle state.

For a mesh model containing $m \times n$ node masses, each node mass is associated with other node neighbors through springs. The association between neighbors is achieved in various methods that allow certain springs to be constrained based on the applied stress force. The example under study related to the mesh of the $m \times n$ masses or nodes, where each node being positioned at time t on the point $l_{ij}(t)$ with i in $[1, \ldots, m]$ and j in $[1, \ldots, n]$.

The evolution of the system is governed by the fundamental law of dynamics,[25] New-ton's second law:

$$\vec{F} = m\vec{a}. \tag{1}$$

The dimension of the force vector for a 3D model is three, typically denoted as (Fx, Fy, Fz), where Fx, Fy and Fz are the force components in the X-, Y- and Z-directions, respectively. $\vec{F}$ is the resulting force, m is the mass of the object and $\vec{a}$ is the acceleration force.

Thus, any regional deformation of springs will produce a deformation of the whole object. The Damping force parameter depends largely on the velocity of the force applied on the springs. Since they are not entirely elastic, the springs absorb a portion of the force and decrease the velocity of the particle (or node) masses related to them.

Every node of the model will absorb forces from the springs with which it is associated. According to Hooke's Law,[26] when a deformation is applied to the spring, the force exerted by the spring is linearly proportional to the difference in lengths of the springs:

$$\vec{F} = k\vec{x}, \tag{2}$$

where k indicates the spring constant and $\vec{x}$ expresses the computed difference vector in length between two nodes. The calculation is indicated in terms of the following equation[27]:

$$\overrightarrow{F_s} = k(|\overrightarrow{v_{ij}}| - l_{ij})\overrightarrow{v_{ij}}/|\overrightarrow{v_{ij}}| = \begin{cases} k(|\overrightarrow{v_{ij}}| - l_{ij}), \\ -k(|\overrightarrow{v_{ij}}| - l_{ij}), \end{cases} \tag{3}$$

where $\overrightarrow{F_s}$ is the spring force from spring (i, j) onto the node, (i, j, k) is the defined spring stiffness, $\overrightarrow{v_{ij}}$ is the vector resulting from the difference in mass position from node i to node j and l_{ij} is the idle size of the spring. Thus, the total spring force on a node at i refers to the total of all the forces applied originating from the connected springs on the node j neighboring the node i.

The solver acquires the created rigidbody nodes on surface of the mesh and the spring joints between nodes. The OnStart unity method maps the rigidbody nodes by looping through the input mesh vertices and finding the lowest distance to get a

Algorithm 4. MSM Model Simulation of Deformation Algorithm.

Input: MSM
Mappings []
MetaHand
Begin
index← *FindNearestRigidbodyIndexToMetaHand*()
rigidbody← *MSMRigidbodies*[*Mappings*[*index*]]
AddForce(*Rigidbody*, *MetaHand* * 10 *f*)
End

correct mapping. Basically, the simulation starts looping through all the rigidbodies positions, updating the visual mesh vertex positions from the rigidbodies positions, then the collision model is recomputed. If a force is applied to any rigidbody along the simulation, the solver iterates and updates its positions as well as the springs.

3.5.4. *Exposure to internal and external forces*

The computed biomechanical model is exposed to internal forces maintained by the inner structures as well as external forces such as the gravity force that pulls the model downwards, surgeon's tools moving a segment of the organ, the heartbeat and breathing or the collision of the organ with the neighboring organs due to movement. All of these registered forces underly the appropriate biomechanical modeling of the liver organ to handle these deformations introduced by any type of force during the intraoperative phase.

Naturally, the gravitational force directly affects the model right from the beginning of the simulation. As far as the proposed method is concerned, the applied forces correspond to the gravity force as well as the hand interactions introduced by the hands using the projected model with Meta 2 AR headset.

The hand interaction exerts a force through the estimation of the nearest rigidbody position investing the *FindNearestRigidbodyIndexToMetaHand* method from the hand position in world space provided by the MetaHand feature. Notably, the force vector is computed based on the rigidbody and hand position coordinates multiplied by 10 and the force value is applied to that rigidbody object via the AddForce method. Naturally, the MSM solver accounts for this change, which can be observed in the simulation.

The applied force affects the rigidbodies position predictions. Thus, the MSM solver copes with the rigidbodies and their spring joints connections, updating their positions depending upon the duration of the simulation and whether a force has been applied or not.

3.5.5. *Seamless mix of algorithms*: *MSM and FEM*

In order to elaborate an accurate biomechanical model, a combination of algorithms involving FEM and MSM could be useful in terms of tacking the benefits of both methods. In fact, the hybrid approach combines the generated model of the liver

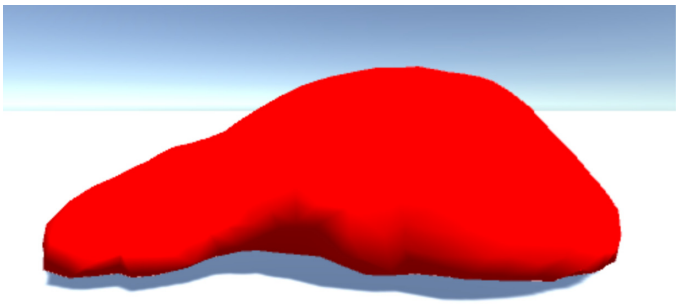

Fig. 9. Deformed mesh to gravity with MSM model based simulation.

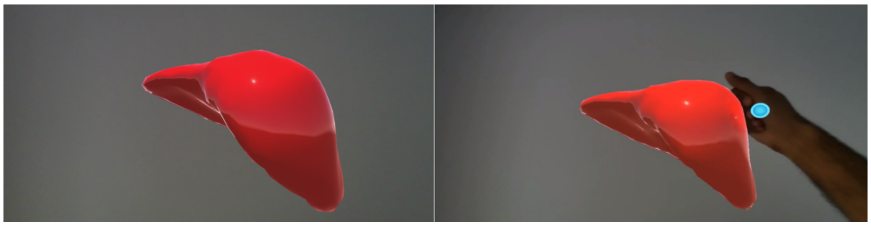

Fig. 10. Model interaction in AR space with the proposed method.

using the FEM tetrahedralization approach and the model faces mapping with its rigidbodies. Being constrained with the use of the spring joints, the springs can be affected by such constraints as the max volume to better deform the model. The main scene physics solver dependency on the number of simulation iterations along with the network of springs constitute the catalyst for the deformation and recomputation of the springs after each deformation to retain the mesh initial rest shape.

4. Experimental Results

The findings of the simulations conducted in the previous section are assessed in terms of performance and model resiliency to deformations triggered by hand interactions using the Meta 2 headset.

The simulation of the biomechanical model of the liver with the MSM model provides an elastic deformation of the model when exposed to gravity force (Fig. 9). The model is therefore perceived as deformed continuously owing to its collision with the flat surface. The tested stiffness and damping parameters with this model were experimental estimations of the spring joints elastic behavior of the surface model of the liver organ.

The interaction with the model in AR space is somewhat noticeable in Fig. 10, where the deformation is small and the model quickly retains its original position as the solver dissipates the force through the nodes and springs network.

Table 1. Comparison of model deformation results obtained by the proposed method and related method.

	Proposed method	Obi softbody[28]
Solver iteration execution time	Four iterations per 1/60 seconds timestep	Three iterations per 1/60 seconds timestep
FPS	High	High
Meta 2 AR interaction	Supported	Not supported
Deformation	Smooth	Smooth
FixedUpdate execution time (physics)	0.71 ms	0.1 ms
Liver Model Nodes count	1025	1423
Collision detection	Exists	Exists

A comparative study of the simulation methods was carried out to assess the model deformation compared to other methods relying on its ability to perform topological modifications in real-time, computation time, density of the model nodes and stability of the simulation at the cost of computation (Table 1).

In this work, four steps are implemented: the reconstruction of geometric model, the generation, the projection and the simulation of the biomechanical model using AR headset.

In order to validate the accuracy of the model as well as the speed rendering in real time, a registration step is needed to superimpose the simulated biomechanical model with the real liver. Thus, we have measured the execution time of the simulation rendering and we have concluded that although the obtained time using the proposed approach is higher than Obi Softbody method,[28] it is acceptable to be applied during open surgery. Also, the main advantage of the developed approach is that it supports the AR headset technology contrary to the existing methods.

Our model is characterized by its decomposition of nodes and edges of the mesh which refer to masses and springs, respectively, where each node is assigned to a mass value and its connected spring joint. The resulting model is prone to the gravitational constant force of 9.81 in the Y axis. To ensure the convergence of the springs, a damping force of 100 was applied and a springs stiffness value of 500 was adjusted. The model deformation to gravity force, for example, can be simulated iteratively through the model nodes or particles sum of exerted forces while updating the rigidbodies positions for each solver iteration timestep. This time step consists of 4 iterations per physics execution time step of 0.016 (or 1/60) seconds.

Moreover, model interactions with the floor due to gravity and hand interactions in 3D world space bring the surface rigidbodies to deform based on springs sum of forces to the hands world position, which initiates the springs deformation by the springs experimental damper parameter making the simulation of this deformation more difficult to trace. From this perspective, a minor drop in performance with Meta 2 AR headset in terms of the rendered frames per second was observed. Collision detection is intrinsic in terms of enabling the model to interact with other

objects. In this phase, the collision detection was computed between the liver model point vectors and the plane acting as the ground normal vectors and between the liver model point vectors and the interacting hand game object with its direction vector when testing with the Meta 2 AR headset.

The collision model recomputation with the V-HACD library[24] during each solver iteration within the frames belonging to a deformation, revealed that the decomposition of the collision model of a group of convex meshes can be recalculated fast. Basically, the collider recomputation relies largely on the render mesh with the lowest possible polygon count possible in order to generate it faster. Therefore, for this specific liver collision model and its generation parameters, retaining the accuracy of the collision model with low computation time affects the real-time aspect of the simulation, which needs experimentations and adjustments to find a mid-way point and serve the simulation. The MSM system implementation in the simulation of softbodies such as the liver model allows the modeling and prediction of the displacement of elastic dampened springs and makes it possible to perform topological modifications in real-time at no computation cost. Although it can perform fast calculations, the elastic behavior and performance of the rigidbodies depend highly upon the nodes number of the model. In addition, having linear springs simulates a linear model. However, the simulation of a complex model composed of several geometric sub-models of internal structures does not simulate the exact liver softbody deformation and hence requires a reevaluation of the spring's network as well as the computation of the internal forces of the model. A powerful showcase of complex multi-model simulation is the Sofa framework. Unfortunately, Sofa[29] which currently has a Unity plugin, held under a private license, does not support the Meta 2 AR headset at the moment. The MSM model can simulate elastic soft tissue of the hyper elastic liver model which has been a hot area of research triggering significant scientific concerns and whetting the widest interest upon volume conservation constraining and correct volume behavior. Indeed, it stands for one of the most outstanding properties of incompressible bodies used in surgery simulation like internal organs which is also applicable in FEM with other physical models and constraints restricting the FEs movements and providing an accurate simulation of soft bodies.

5. Conclusion and Perspectives

In this research work, our central focus was upon an appropriate biomechanical modeling demonstrating its outstanding properties and parameters. In addition, we experimented with a myriad of approaches to simulate the liver organ deformations. Scanning through the conducted experiments, the MSM model as well as the FEM approaches proved to be the most superior in terms of computational time. Topological modifications favored triangular and tetrahedral mesh, as they are simpler to subdivide to more triangular elements. Furthermore, topological modifications occur during surgery owing to patients' movement or surgeons' organ deformation

with the intraoperative tools. At this stage of analysis, we would assert that the basic merit of using FEM lies essentially in the likelihood of using different physical laws in the form of constitutive functions and accurate boundary conditions to the model. AR displays the potential to intensely improve medical interventions during open surgery. So far, the use of 3D image-based surgery planning and training simulations has been more and more accepted in this area. However, promising and worthwhile as it proves to be, it still needs more development and further improvement in terms of accurate biomechanical modeling especially in hepatic surgery. In this respect, this modeling constitutes a powerful impetus for future researchers to perform further investigation so as to enact a fruitful and valuable contribution to this field. As a final note, it seems that the use of biomechanical modeling will bring such a positive impact that it may revolutionize the area of hepatic surgery.

Acknowledgment

We would like to extend our appreciation to those whose contributions and support were crucial to the successful completion of this research work. We are profoundly grateful to Professor Mehdi JAIDANE and Professor Borhen LOUHICHI for their unwavering dedication, expert guidance, and invaluable mentorship throughout the entirety of this research project. their insights and encouragement were instrumental in shaping our work. We acknowledge the generous support provided by the Sahloul University hospital. Without their funding, this research would not have been possible. Their belief in the importance of our work allowed us to conduct extensive experiments and data analysis.

References

1. Su L.-M., Vagvolgyi B. P., Agarwal R., Reiley C. E., Taylor R. H., Hager G. D., Augmented reality during robot-assisted laparoscopic partial nephrectomy: Toward real-time 3D-CT to stereoscopic video registration, *Urology* **73**(4):896–900, 2009, doi:10.1016/j.urology.2008.11.040.
2. Adagolodjo Y., Trivisonne R., Haouchine N., Cotin S., Courtecuisse H., Silhouette-based pose estimation for deformable organs application to surgical augmented reality, *2017 IEEE/RSJ Int. Conf. Intelligent Robots and Systems* (*IROS*), Vancouver, Canada, 2017, doi:10.1109/iros.2017.8202205.
3. Golse N., Petit A., Lewin M., Vibert E., Cotin S., Augmented reality during open liver surgery using a markerless non-rigid registration system, *J. Gastrointest. Surg.* **25**:662–671, 2021, doi:10.1007/s11605-020-04519-4.
4. Suthau T., Vetter M., Hassenpflug P., Meinzer H.-P., Hellwich O., A concept work for augmented reality visualization based on a medical application in liver surgery, *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* **34**(5):274–280, 2002.
5. Kipper G., Rampolla J., *Augmented Reality*: *An Emerging Technologies Guide to AR*, Elsevier, pp. 1–158, 2012.
6. Azuma R. T., A survey of augmented reality, *Presence*: *Teleoperators and Virtual Environments* **6**(4):355–385, 1997, doi:10.1162/pres.1997.6.4.355.
7. Wesarg S., Firle E. A., Segmentation of vessels: The corkscrew algorithm, *Med. Imaging Image Process.*, Vol. 5370, SPIE, pp. 1609–1620, 2004, doi:10.1117/12.535125.

8. Marescaux J., Rubino F., Arenas M., Mutter D., Soler L., Augmented-reality–assisted laparoscopic adrenalectomy, *JAMA — J. Am. Med. Assoc.* **292**(18):2211–2215, 2004.
9. Haouchine N., Roy F., Untereiner L., Cotin S., Using contours as boundary conditions for elastic registration during minimally invasive hepatic surgery, *2016 IEEE/RSJ Int. Conf. Intelligent Robots and Systems* (*IROS*), Daejeon, Korea, IEEE, pp. 495–500, 2016, doi:10.1109/iros.2016.7759099.
10. Plantefève R., Peterlik I., Haouchine N., Cotin S., Patient-specific biomechanical modeling for guidance during minimally-invasive hepatic surgery, *Ann. Biomed. Eng.* **44**(1):139–153, 2015, doi:10.1007/s10439-015-1419-z.
11. Suwelack S., Röhl S., Bodenstedt S., Reichard D., Dillmann R., dos Santos T., Speidel S. *et al.,* Physics-based shape matching for intraoperative image guidance, *Med. Phys.* **41**(11):111901, 2014, doi:10.1118/1.4896021.
12. Santhanam A., Fidopiastis C., Lup F. H., Rolland J., Imielinska C., Physically-based deformation of high-resolution 3D lung models for augmented reality based medical visualization, pp. 21–31, 2004.
13. Hauth M., Gross J., Straßer W., Buess G. F., Soft tissue simulation based on measured data, *Med. Image Comput. Comput. Assist. Interv. — MICCAI* **2003**:262–270, 2003, doi:10.1007/978-3-540-39899-8_33.
14. Kaye J. M., Primiano F. P., Metaxas D. N., A three-dimensional virtual environment for modeling mechanical cardiopulmonary interactions, *Med. Image Anal.* **2**(2):169–195, 1998, doi:10.1016/s1361-8415(98)80010-8.
15. Bertrand L. R., Abdallah M., Espinel Y., Calvet L., Pereira B., Ozgur E., Bartoli A. *et al.*, A case series study of augmented reality in laparoscopic liver resection with a deformable preoperative model, *Surg. Endosc.*, **34**(12):5642–5648, 2020.
16. Müller M., Rassweiler M.-C., Klein J., Seitel A., Gondan M., Baumhauer M., Maier-Hein L. *et al.*, Mobile augmented reality for computer-assisted percutaneous nephrolithotomy, *Int. J. Comput. Assist. Radiol. Surg.* **8**(4):663–675, 2013, doi:10.1007/s11548-013-0828-4.
17. IRCAD France Institute, https://www.ircad.fr/research/3d-ircadb-01/.
18. InVesalius 3 Software, CTI research team, https://www.gov.br/mcti/pt-br/rede-mcti/cti.
19. He Y.-B., Bai L., Aji T., Jiang Y., Zhao J.-M., Zhang J.-H., Wen H. *et al.,* Application of 3D reconstruction for surgical treatment of hepatic alveolar echinococcosis, *World J. Gastroenterol.* **21**(35):10200–10207, 2015, doi:10.3748/wjg.v21.i35.10200.
20. Si H., TetGen, a delaunay-based quality tetrahedral mesh generator, *ACM Trans. Math. Softw.* **41**(2): 1–36, 2015, doi:10.1145/2629697.
21. Uwitonze A., Huang J., Ye Y., Cheng W., Li Z., Exact and heuristic algorithms for space information flow, *PLoS One* **13**: e0193350, 2018, doi:10.1371/journal.pone.0193350.
22. Rebay S., Efficient unstructured mesh generation by means of Delaunay triangulation and Bowyer–Watson algorithm, *J.Comput. Phys.* **106**(1): 125–138, 1993, doi:10.1006/jcph.1993.1097.
23. Edelsbrunner H., Shah N. R., Incremental topological flipping works for regular triangulations, *Proc. 8th Annual Symp. Computational Geometry*, Berlin, Germany, pp. 43–52, 1992.
24. Thul D., Ladický L., Jeong S., Pollefeys M., Approximate convex decomposition and transfer for animated meshes, *ACM Trans. Graph.* **37**(6):1–10, 2018, doi:10.1145/3272127.3275029.
25. Tsai T.-C., Position based dynamics, in Lee N. (ed.), *Encyclopedia of Computer Graphics and Games*, Springer International Publishing, Cham, pp. 1–5, 2017, http://link.springer.com/10.1007/978-3-319-08234-9_92-1.

26. Moyer A. E., Robert Hooke's ambiguous presentation of "Hooke's Law", *ISIS* **68**(2):266–275, 1977.
27. Haouchine N., Image-guided simulation for augmented reality during hepatic surgery, Theses, Université de Lille, 2015, https://hal.inria.fr/tel-01254439.
28. Macklin M., Müller M., Chentanez N., Kim T.-Y., Unified particle physics for real-time applications, *ACM Trans. Graphics* **33**(4): 1–12, 2014, doi:10.1145/2601097.2601152.
29. Allard J., Cotin S., Faure F., Bensoussan P. J., Poyer F., Duriez C., Delingette H., Grisoni L., February. Sofa-an open source framework for medical simulation, *MMVR 15-Medicine Meets Virtual Reality*, Vol. 125, Long Beach, CA, IOP Press, pp. 13–18, 2007.