Surgical Training Simulator

By

Muhammad Rizwan
Muhammad Usman Shahbaz
Muhammad Waleed Khan

Department of Computing
School of Electrical Engineering & Computer Science
National University of Sciences & Technology
Islamabad, Pakistan

2013

Surgical Training Simulator

By

Muhammad Rizwan	**2009-NUST-BIT-264**
Muhammad Usman Shahbaz	**2009-NUST-BIT-268**
Muhammad Waleed Khan	**2009-NUST-BIT-269**

Department of Computing

School of Electrical Engineering & Computer Science
National University of Sciences & Technology
Islamabad, Pakistan

2013

CERTIFICATE

It is certified that the contents and form of thesis entitled "**Surgical Training Simulator**" submitted by Muhammad Waleed Khan **2009-NUST-BIT-269**, Muhammad Usman Shahbaz **2009-NUST-BIT-268** and Muhammad Rizwan **2009-NUST-BIT-264** have been found satisfactory for the requirement of the degree.

Advisor: Shamyl bin Mansoor

(18/6/2013)

Co-Advisor: Muhammad Muddassir Malik

( )

DEDICATION

To Our Parents, Teachers

& the People who supported us in our lives

ACKNOWLEDGEMENTS

We would like to take this opportunity to pay our humble gratitude to Almighty Allah who blessed us with His kindness to complete this project.

We are highly thankful to our Advisor Mr. Shamyl bin Mansoor and our Co-advisor Mr. Muhammad Muddassir Malik for guiding us in the project and for encouraging us to achieve our goal. Their support has helped us to achieve many ambitious tasks and has shown us the right way to achieve our target.

We would like to offer appreciation to our parents. Their supplications, criticism and encouragements were the means by which we have completed our project in time.

Last but not the least, we would like to thank to the people who have always supported us. We are indebted to them for the bulk of the information.

Table of Contents

Table of Figures

ABSTRACT

Minimal Invasive Surgery (MIS) is a marquee of modern day surgery's evolution. In conventional procedures, surgeons used to make large incision to perform the required operation. But now using minimal invasive surgery only a few small holes are made through which instruments and camera(s) can be inserted into the body. Using these instruments (Camera, Grasper, Burner), the surgeon performs the surgery. The use of virtual reality simulators to train newbie surgeons for this evolving type of surgery is a common practice in the medical world. While mimicking the real environment, lack of haptic leaves a lot of room for improvement thus creating research and development opportunities.

CHAPTER 1 – INTRODUCTION

1.1 - Project Title:

SURGICAL TRAINING SIMULATOR

DEVELOPMENT OF HAPTIC AIDED MIS TRAINING EXERCISE USING OPEN SOURCE PHYSICS ENGINE (SOFA)

1.2 - Problem Statement:

In our project we will initially benchmark the Lapsim(a state of the art surgical simulator) to develop a Laparoscopic cholecystectomy exercise i.e. surgical removal of the gall bladder present under the liver using open source physics engine SOFA and further modify it by integrating haptics in it using openHaptics alongside Phantom Omni Hand manipulator. In addition to mapping of our 3D hand movement from real world to the virtual world, Phantom Omni Hand Manipulator will also provide force feedback as per our desire for haptic purpose. Our main focus will be on haptics. The problem is that SOFA provides us with a very limited scope for haptics. We need to modify and develop our own haptic effects using openHaptics library with more realistic effects.

The challenges involved are:

- Understanding SOFA (Open Source Surgical Simulation Library)
- Modeling Simulation Scenes in SOFA
- Programming in C++ using SOFA
- Working with Omni Haptic Device to create realistic simulations
- Providing real-time haptic force feedback

CHAPTER 2 – COMPONENTS, MODULES AND SUB-MODULES

2.1 - Hardware:

The project involves the GeoMagic product Phantom omni device, the controlling and interfacing of which is the main part of the project. A Firewire PCI card IEEE 1394 is used to communicate phantom omni with Computer.

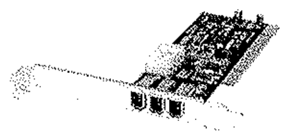


Figure 1-Hardware

2.2 - Modules:

We have divided the project in three major modules.

- Literature Review on SOFA Modeler
  - Scene Creation
- Literature Review on Instrument Modeling
  - Instrument Model
- Literature Review on Haptics in SOFA
  - Phantom Omni Integration and openHaptics in SOFA

2.3 - Block Diagram:

Block Diagam

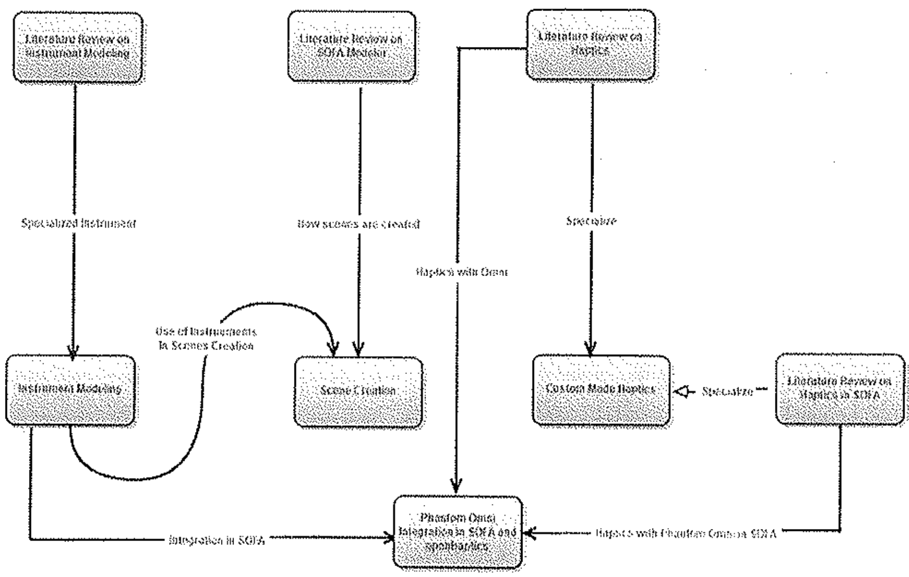


Figure 2-Block Diagram

CHAPTER 3 - LITERATURE REVIEW ON INSTRUMENT MODELING:

In the initial phase of the project, our goal was to:

- ✓ Create a good understanding of SOFA.
- ✓ Interaction with its modeler.
- ✓ Modeling of different instruments.
- ✓ Controlling the instruments with Mouse and Keyboard.
- ✓ Understanding the difference and control of Rigid and Laparoscopic instruments.
- ✓ Integration of Phantom Omni Device with SOFA.
- ✓ Controlling the instruments with Phantom Omni.

3.1 - Create a Good Understanding of SOFA

This part was relevant to the literature review of SOFA. SOFA is basically a flexible library to use different geometrical model interaction. It can also be used for their visual model and it provides different collision models of different mechanical object.

3.1.1 - Installation:

SOFA framework is downloadable from http://www.sofa-framework.org/ . Once downloaded and unzipped, place all the items in one folder and run the bat file. The version of Operating system and the compiler should be looked upon carefully. In my case the operating system is Microsoft Windows 7 64 bit and the compiler is Microsoft Visual Studio 2010 64 bit. Running the Project VC10_x64.bat file will create the Sofa.sln file which is executable in VS2010. Once all the projects are loaded, build the whole solution.

3.1.2 - Loopholes:

While installation, an error were faced that some projects require some .lib files which was not manually built by the compiler. So to solve this, we have to manually build that project to generate the .lib file and manually copy that file in lib folder.

After complete installation, run the SOFA from /bin/Modelerd.exe.

A successful installation will look like:

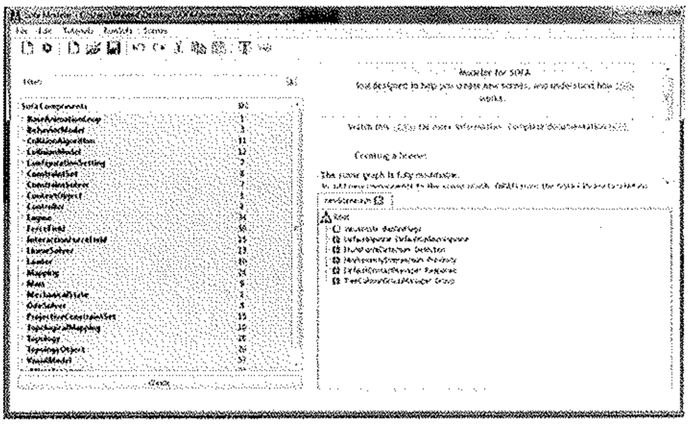


Figure 3-Successful Installation

3.1.3 - Interaction with modeler:

After installation, different tutorials were tried on the modeler to get familiar with the different attributes of SOFA and its modeler. Tutorials of falling cube and falling cubes on a floor under gravity were performed as shown in figure:

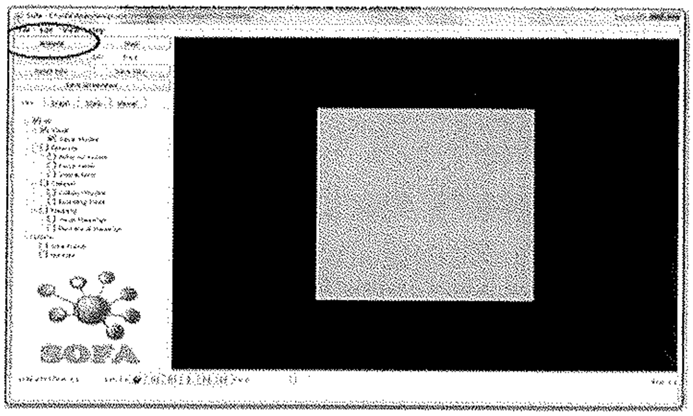


Figure 4-Interaction with modeler

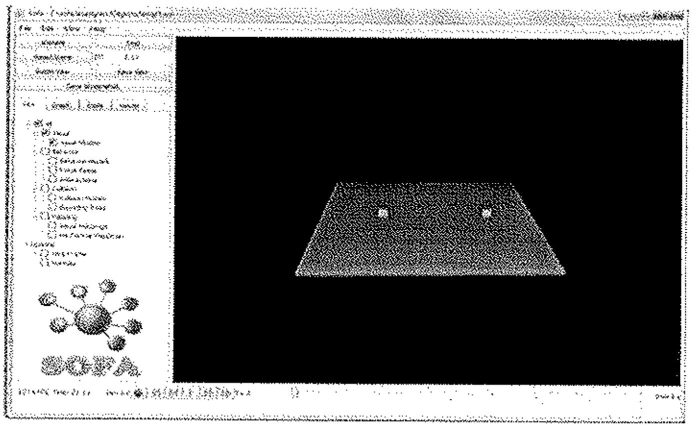


Figure 5-Tutorial Falling cube

3.2 - Modeling of different instruments:

After getting familiar with SOFA and working with its modeler, we started working on Instrument modeling. First we gather information about different types of instruments.

3.2.1 - Types of instruments:

There are two types of instruments given by SOFA that can be used in our surgical simulation.

- Rigid instrument.
- LaparoscopicRigid instrument.

As our simulation is about Laparoscopic surgery, so we started our work with laparoscopicRigid instrument. Also there is a main point in using this instrument which was the position of laparoscopicRigid instrument. During surgery the instrument is pivoted at some point as to give surgeon a feel that the instrument is inserted in the body through a hole and it is not freely movable.

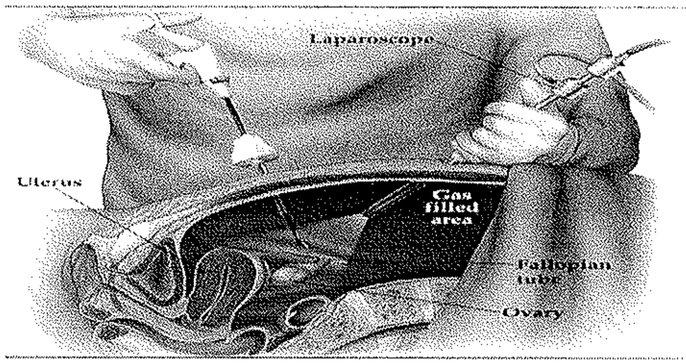


Figure 6-Laparoscopic Procedure

3.2.2 - Instrument Compatibility:

In SOFA, by default the compatibility of laparoscopicRigid instrument is less than the Rigid instrument. So in our instrument modeling, we mapped laparoscopic instrument on rigid instrument to give laparoscopic instrument the features of rigid instrument. The detail compatibility problem will be discussed in later part of the report.

3.2.3 - Instrument handling:

Then we modeled an instrument made of mechanical object of rigid type and gave a visual model as shown in figure:

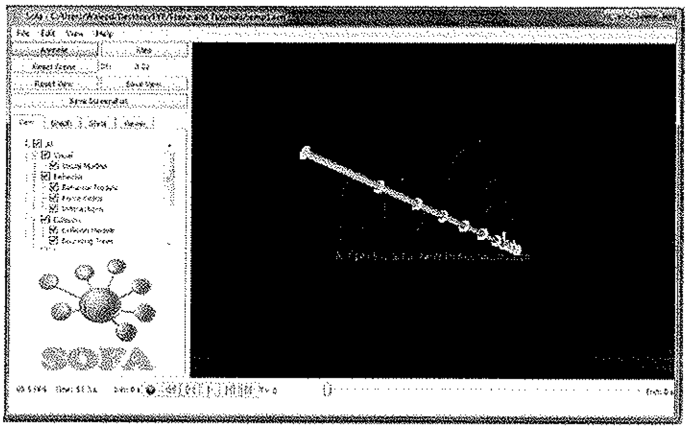


Figure 7-Rigid Instrument

At this point, we are able to control the Rigid type object using our keyboard and mouse input. Now we are going towards Laparoscopic instrument. We made a mechanical object of LaparoscopicRigid Type and mapped it on Rigid type mechanical object as shown in figure.

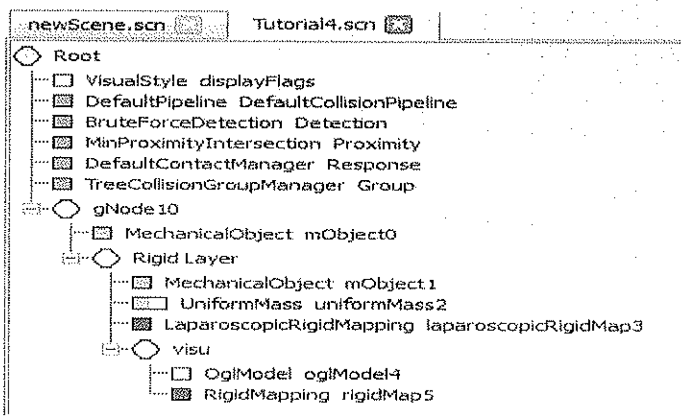


Figure 8-Tree Hierarchy

3.2.4 - Controlling the instruments with Mouse and Keyboard:

Now at this point we were unable to control the laparoscopic instrument in qt view i.e. to move the instrument around its origin. We want this control to understand the control of different instrument which will be used later to control them with phantom omni. Here we modified the code.

We opened the qtviewer.cpp and changed the control pressed event in mouse event. Here we created a pointer to the laparoscopic instrument and handled its animation and other rotation and translation.

Now when we are able to control the Laparoscopic instrument, next step is to create another laparoscopic instrument and switch the control between those two instruments. Same work is done in the control pressed event in mouse event of "qtviewer.cpp". Shown in figure:

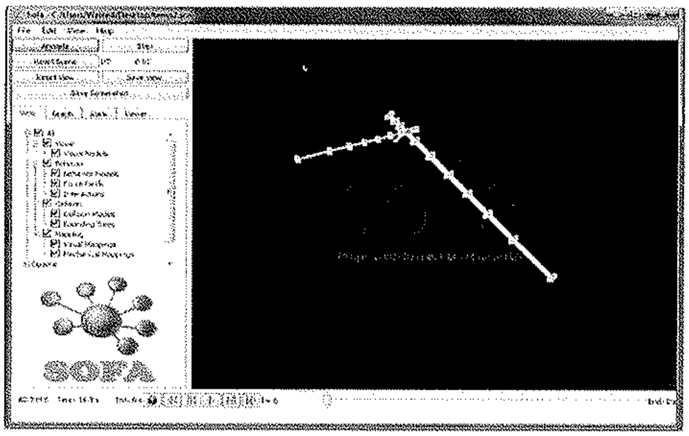


Figure 9-Switched Control of two instruments

3.2.5 - Understanding the difference and control of Rigid and Laparoscopic instruments:

Important aspect of laparoscopic instrument is that it is pivoted around one point. So it is best to use Laparoscopic instrument in this surgical simulation.

3.2.6 - Integration of Phantom Omni Device with SOFA:

SOFA provides sensible plugin to communicate with phantom omni device to provide haptic feedback. The sensible plugin requires phantom omni drivers and Open Haptics SDK. So to communicate with Phantom omni, first install these drivers and SDK. Then uncomment sensible plugin definition in sofa-default.prf file, and re-run the bat file for VS2010_64 bitt.

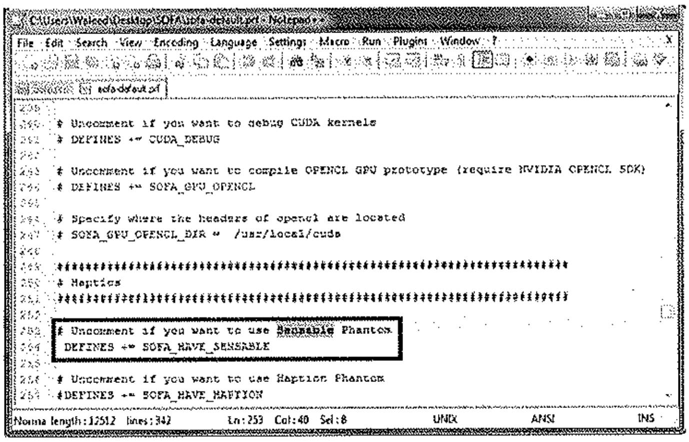


Figure 10-Sensable plugin installation

Now execute the sofa.sln file and a project for sensible plugin is created. Build the project to generate the SensablePlugin_1_0d.dll file. Now load the plugin in the sofa as shown in figure.

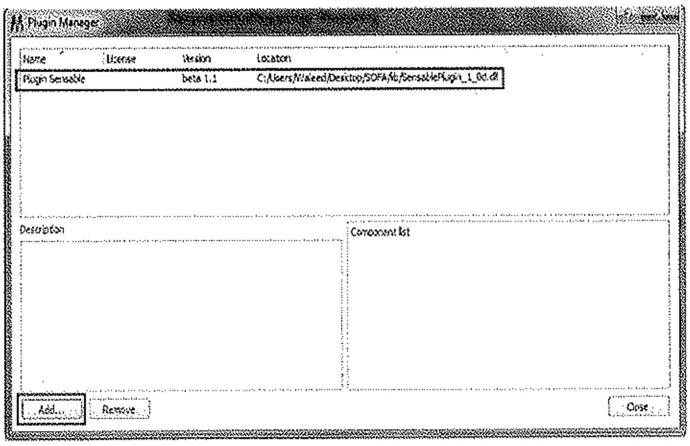


Figure 11-Plugin insertion in SOFA

3.2.7- Controlling the instruments with Phantom Omni:

Now our aim is to control the modeled instruments through phantom omni. First we create a simple exercise for rigid instrument and added omni driver and necessary constraints as shown in the figure:

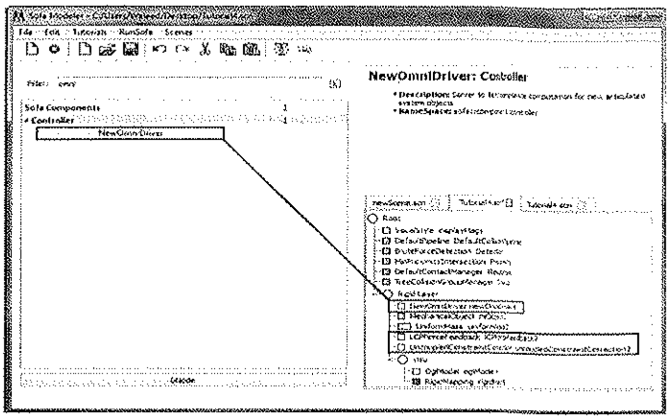


Figure 12-Addition of Omni Driver in Tree Hierarchy

Now we are able to control our rigid instrument through phantom omni. Next step is to control our laparoscopic instrument through phantom omni. For this purpose we have to create the pointer to the laparoscopic instrument and handled its animation and other rotation and translation. For this we created a pointer to the device to get the coordinates of stylus. Then we get all the objects in the tree and created the pointer of the first mechanical object of Laparoscopic rigid type. Then we passed the coordinates of device to control the movement of the stylus.

3.2.8 - Design of Instrument:

After installation of SOFA framework, as per the exercise that we are going to model we created the Instruments in 3DS Max. This was one of the most important part because once we create the instruments we will be using the same instruments while interaction with the liver.

3.2.7- Controlling the instruments with Phantom Omni:

Now our aim is to control the modeled instruments through phantom omni. First we create a simple exercise for rigid instrument and added omni driver and necessary constraints as shown in the figure:

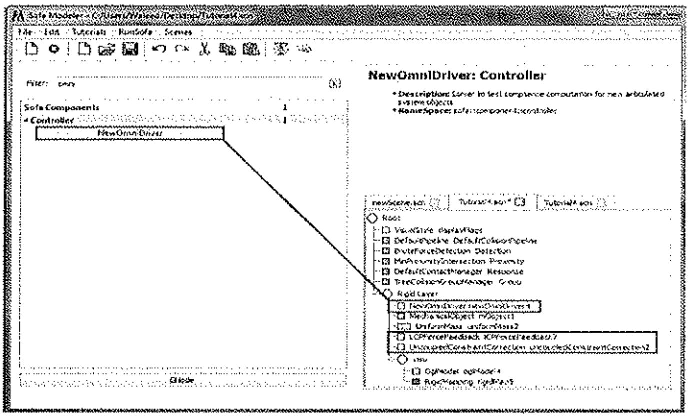


Figure 12-Addition of Omni Driver in Tree Hierarchy

Now we are able to control our rigid instrument through phantom omni. Next step is to control our laparoscopic instrument through phantom omni. For this purpose we have to create the pointer to the laparoscopic instrument and handled its animation and other rotation and translation. For this we created a pointer to the device to get the coordinates of stylus. Then we get all the objects in the tree and created the pointer of the first mechanical object of Laparoscopic rigid type. Then we passed the coordinates of device to control the movement of the stylus.

3.2.8 - Design of Instrument:

After installation of SOFA framework, as per the exercise that we are going to model we created the Instruments in 3DS Max. This was one of the most important part because once we create the instruments we will be using the same instruments while interaction with the liver.

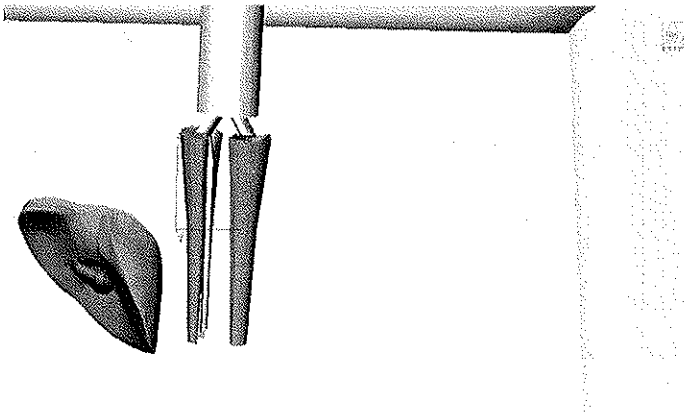


Figure 13-Design of Instruments

3.2.9 - Integrating multiple Omni:

Figure 14-A view of multiple omni's connected

Adding the Omni functionality is the most important part in creating surgical simulation. This thing would be going to be used in all future surgical applications or simulations. Unfortunately support for Multiple Omni wasn't provided in SOFA.SOFA can detect 2 Omni devices but when it comes to getting data from them it get stuck. This was a difficult task for us to be done. We started with editing the main NewOmniDriver cpp file present inside the sensible plugin.

There is a data structure in NewOmniDriver with the name "data" it has different variables like ready, stop ,x ,y ,z position holders. At any instant you just call the data variable with device name to get the position, e-g Device1->data.ready will return true if the Device1 have available data.

Whenever we add multiple Omni support after initializing that is after the calling of init() the program got stuck. The data.ready was called on both the devices and SOFA doesn't know from which device we have to get the data. We knew from start that the problem is with threading that SOFA is getting data from both devices but unable to determine from which device and at which moment. The solution is mutual exclusion which states that

"A way of making sure that if one process is using a shared modifiable data, the other processes will be excluded from doing the same thing."[1]

The solutions sounds very simple in which we integrated the pthread in C++.By using pthread we separated the onAnimatebegin() function of both the devices. As expected it should run but unfortunately again we were getting the same error on it that is "Data is not ready". All solutions were tried to check the issue but all in vain. After reviewing the built in feature of SOFA we came to know about another feature of SOFA which is known as the Boost feature. The Boost feature enables you to integrate threading inside SOFA.

We used the services of christen Duriez who is also among the pioneers of SOFA. Attached is the mail for reference:

[1] http://www.princeton.edu/~achaney/tmve/wiki100k/docs/Mutual_exclusion.html

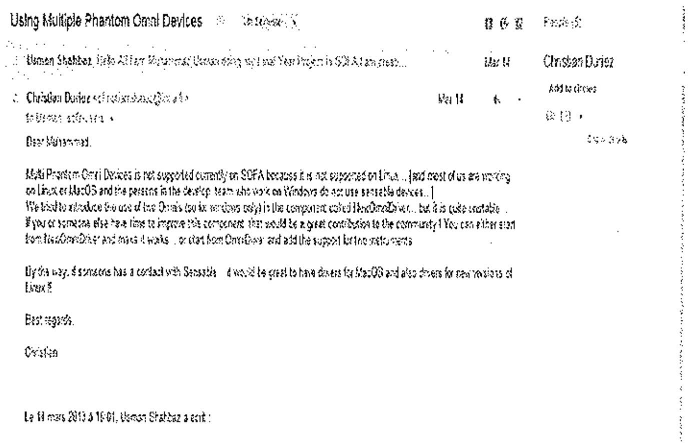


Figure 15-View of SOFA Pioneers

He adjusted the code as per the Boost feature and the multiple Omni code was working properly. The Boost feature has built in functions for yield(),start(),stop() so one wouldn't have to worry about it.

Shortly we will be updating the Multiple Omni stuff on "github" so that it would be accessible by the community and everyone could get benefit from it. In our exercise we will be dealing with two instrument which is Burner and Grasper so it was the necessary feature that should be included in our simulation.

3.2.10 - Multiple Phantom Omni mapping issue:

In multiple Omni integration, the mapping of two styluses (instruments) is performed on one mechanical object using index.

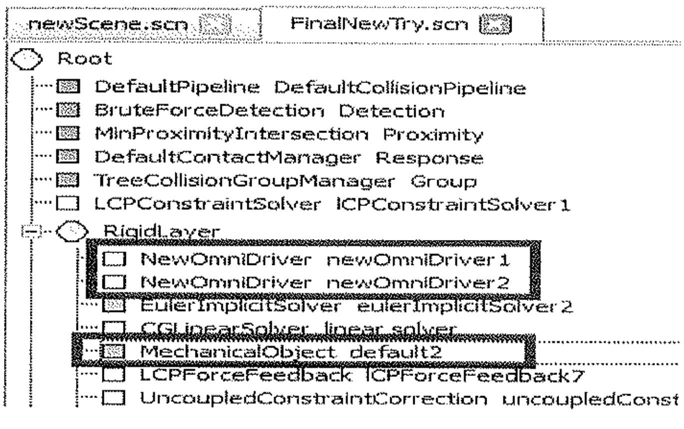


Figure 17-Multiple omni's in a tree

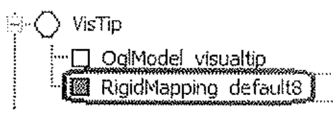


Figure 16-Mapping component

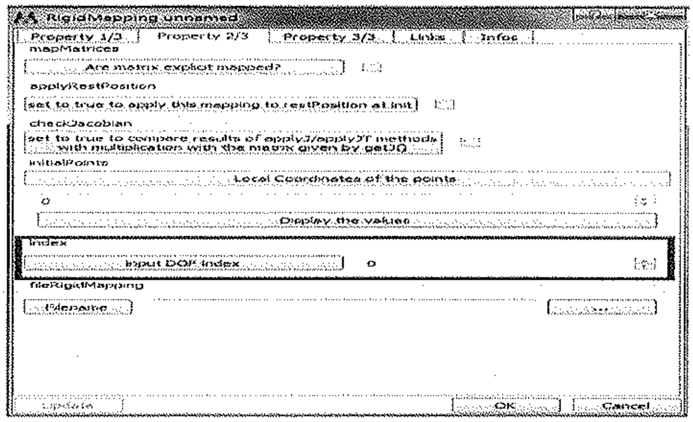


Figure 18-index property in RigidMapping

Here comes the compatibility issue. LaparoscopicRigid mapping lacks this property. i.e. support of multiple mechanical indexes to map multiple Omnis. To solve this we need to write a new component which is left over for the future development tasks. For now we continued the development with Rigid instrument.

3.2.11 - Grasper Instrument:

In our cholecystectomy exercise, an instrument is needed to grasp the gall bladder to pull it out of the fats and separate it from the ducts. So for this, we modeled a grasper instrument. Because this instrument was not provided directly by SOFA, so we had to write our own routine for this instrument. Grasper instrument basically works on Ray model. It has some functions like init(), updateRay(), deactivateRay() etc. The concept behind this is the rays coming out of the grasper instrument. We set the routine in such a way that whenever user presses the button on phantom omni, a new node is created in the tree of the Sofa Modeler. This tree is responsible for the movement of grasped object. The rays from grasper instrument are generated and they come in contact with the other mechanical object's mesh. As a result of their contact, they get attached to each other and the mechanical object gets translated where ever the grasper moves. This node is automatically deactivated as soon as the surgeon press the button on omni, i.e. release the object.

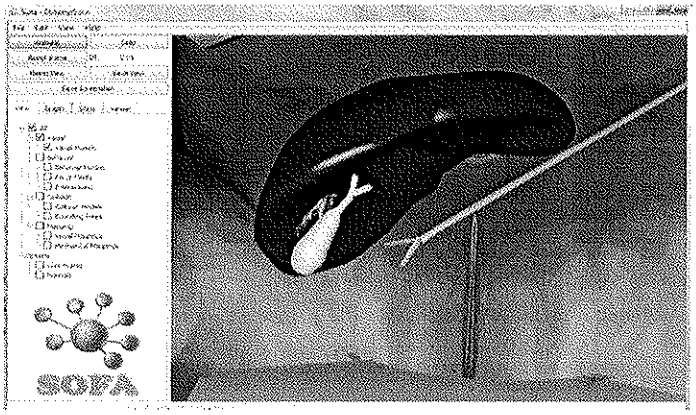


Figure 19-A view of Grasper Instrument

3.2.12 - Burning Instrument:

In our exercise, an instrument is needed to burn the fats such that the gall bladder gets separated from the liver. In our surgery we assumed that the duct which connects the liver and the gall bladder is already clipped. This is the future task to make the simulation of the duct clipping. Now to burn the fats, we modeled a burning instrument. Because this instrument was not provided directly by SOFA, so we had to write our own routine for this instrument. Burning instrument, just like grasper instrument, also works on Ray model. It has some functions like init(), updateRay(), deactivateRay() etc. The concept behind this is the rays coming out of the burning instrument. We set the routine in such a way that whenever user presses the button on phantom omni, a new node is created in the tree of the Sofa Modeler. This tree is responsible for burning rays coming out of the instrument. The rays from burning instrument are generated and they come in contact with the other mechanical object's mesh. As a result of their contact, the part of that mesh gets disappeared which in turn is the burning effect and as a result the gall bladder gets free from the liver fats. This node is automatically deactivated as soon as the surgeon press the button on omni, i.e. stopped the burning rays.

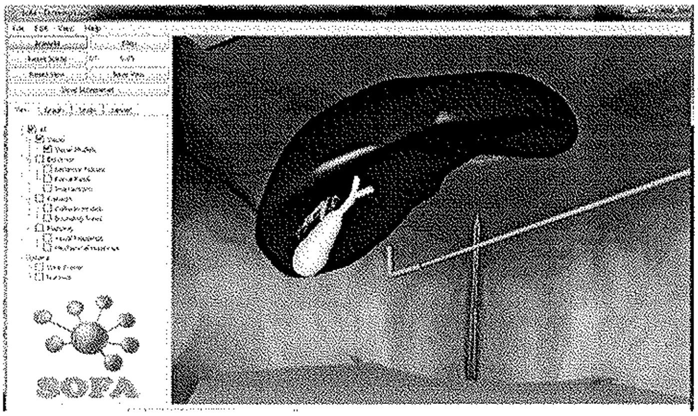


Figure 20-A view of Burning Instrument

CHAPTER 4 - LITERATURE REVIEW ON SOFA MODELER:

To get familiar with the SOFA modeler, we followed the given tutorials by SOFA community (mentioned earlier).After modeling of different instruments in SOFA, there is a major module regarding scene of our exercise.

4.1 - Scene Creation:

After the study of the mechanical properties of human liver, we were ready to create the 3D scenes on which the actual surgeries will be performed. It was a three step process:

4.1.1 - Scene Creation in 3ds Max:

The scenes (body parts) were separately created in Autodesk 3ds Max software. The screen shots are attached:

Gall Bladder:

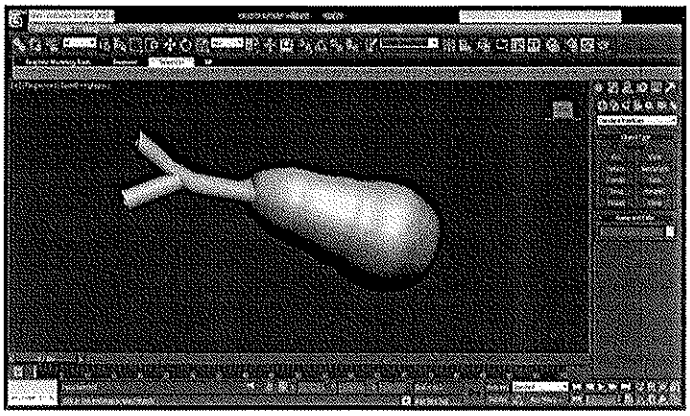


Figure 21-Gall Bladder Modeling in 3ds Max

Liver:

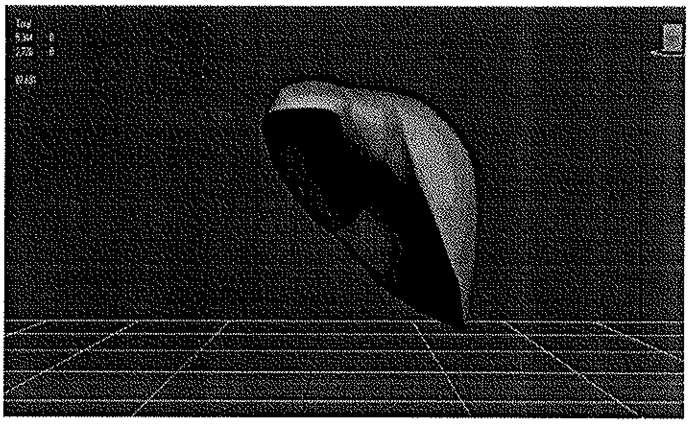


Figure 22-Liver Modeling in 3ds Max

Fats:

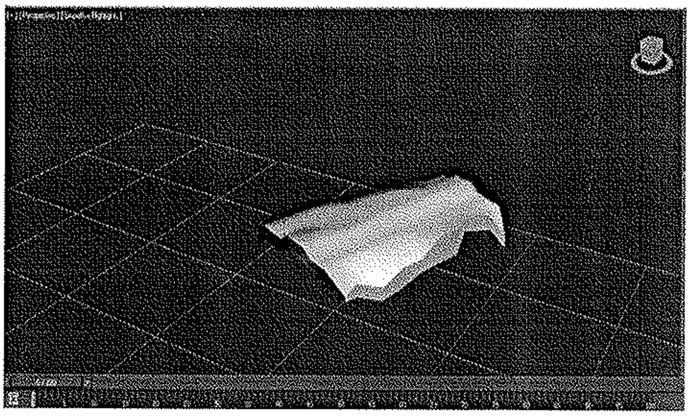


Figure 23-Fats Modeling in 3ds Max

Figure 24-Human Body Modeling in 3ds Max

Verification using STL check to generate .msh files and exporting .obj files:

The second step was to verify the images using STL check in 3ds Max. The STL check is basically used to generate the .STL file, which in turn is converted to .msh file, which is used in the SOFA for different exercises. The STL file cannot be generated correct if the STL check is not true. Screen shots attached:

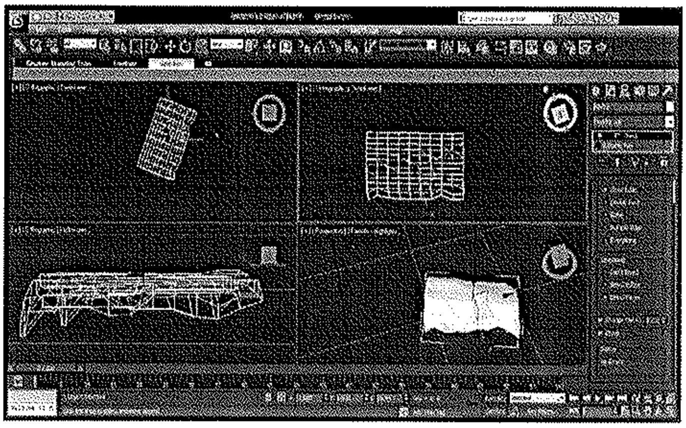


Figure 25-Verification using STL check to generate .msh files

Loading .obj files in SOFA and repositioning to correct positions:

The final step was to import the scenes (.obj and .msh) files in SOFA, moving them to the correct positions and applying different topologies so that the final scenes have same behavior as that of human body parts. Screen Shots of the scene in SOFA are attached:

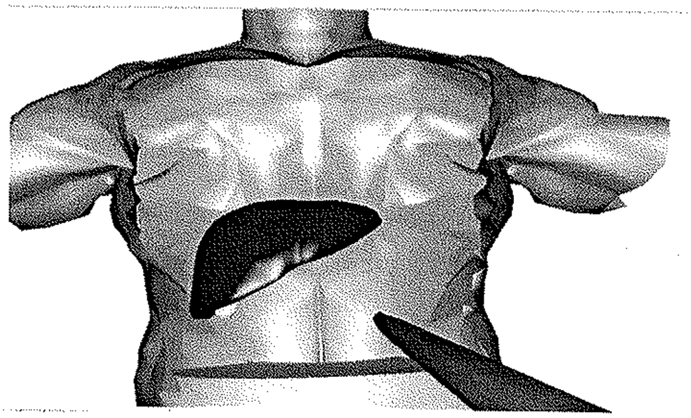


Figure 26-obj files after importing in SOFA

Checking Behaviour:

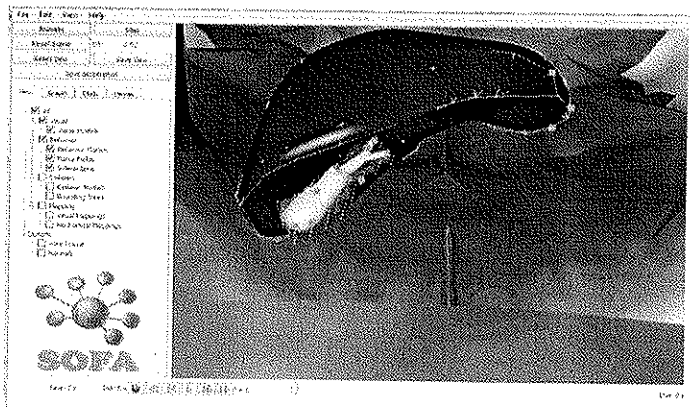


Figure 27-Checking Behaviour

Checking : Collision

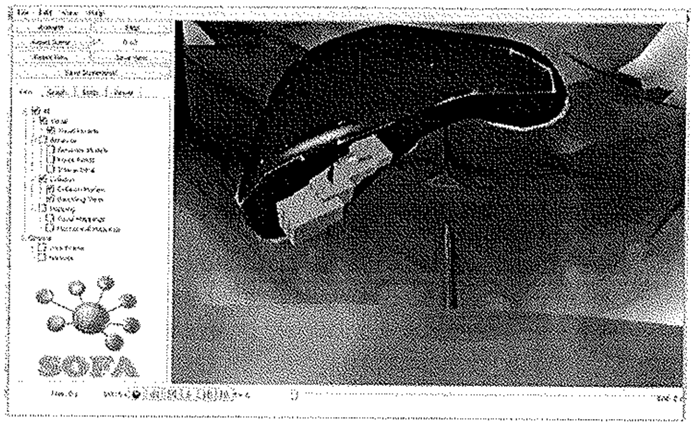


Figure 28-A view of Collision model

Checking: Mapping

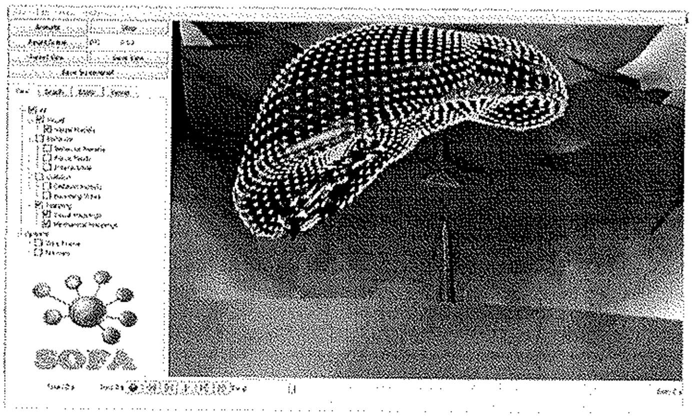


Figure 29-A view of mapping

4.1.2 - Attach Constraint:

The next step was to use the Attach Constraint in SOFA to attach the Fats layer to the Gallbladder so that when Gallbladder is pulled, the Fats layer comes with it, this behavior is also present in human body. There were four vertices of the Gallbladder that were attached with the four vertices of the Fats. Final Screen shot is attached (Gallbladder being pulled):

Figure 30-Working of attach constraint

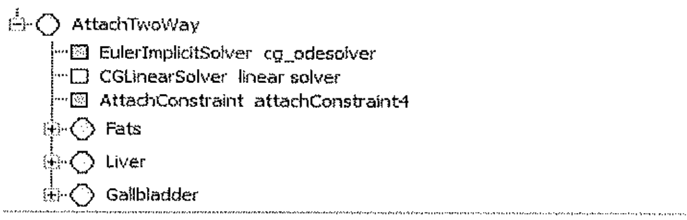


Figure 31-Attach constraint in Tree

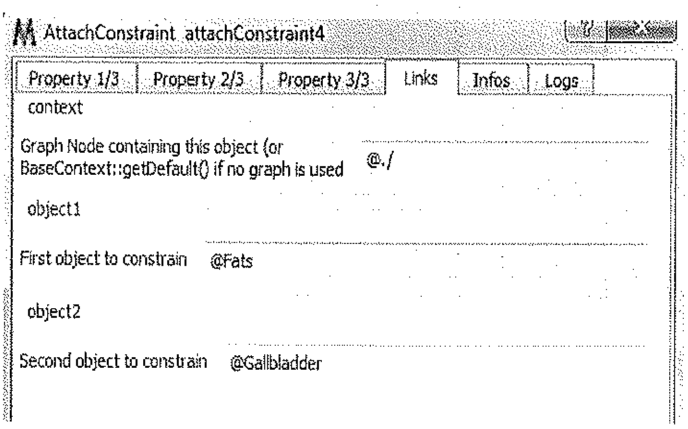


Figure 32-Applying attach constraint

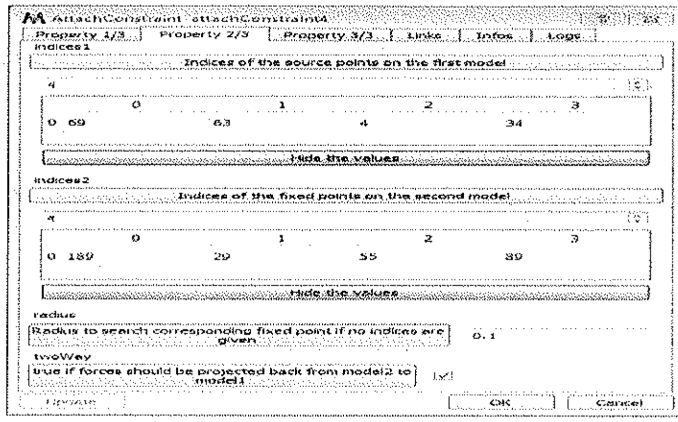


Figure 33-Attach Constraint Points

4.1.3 - Fats Burning:

The next step was to add such topology to fats so that they burn. This was done using the TetrahederalCorotationalFEMForceField exercise available in SOFA in which the mesh of a cylindrical tube is removed which looks like burning effect.

4.1.4 - Original Exercise:

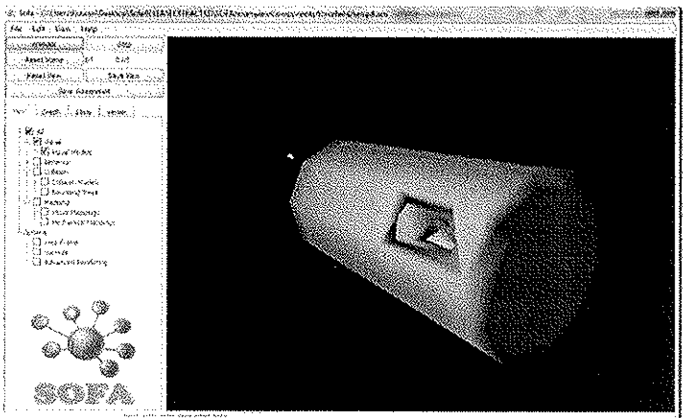


Figure 34-Fats Burning original exercise

4.1.5 - Fats Burning:

Figure 35-A view of burning property

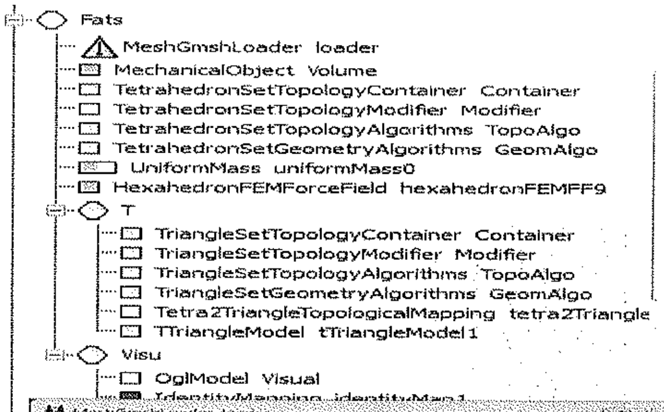


Figure 36-Fats in Tree Hierarchy

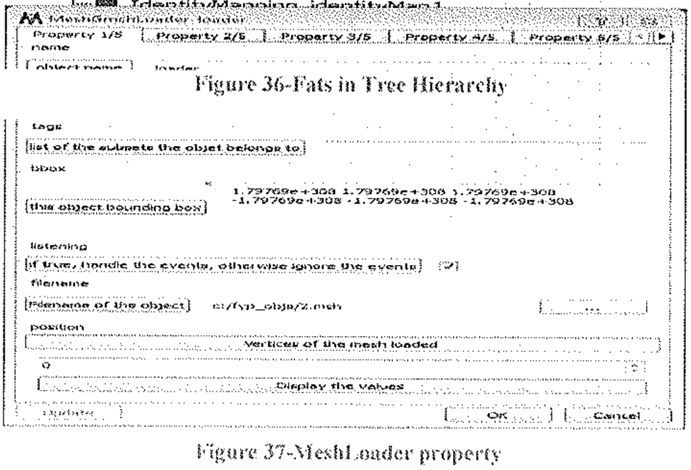


Figure 37-MeshLoader property

4.2 - SOFA modes:

There are two modes in which SOFA can be run.

- Debug mode
- Release mode

Initially our development was in debug mode. When the scene of our exercise is created, the processing cost was increased due to which our exercise got a lot slower that it was very difficult to even animate an object between two points. The reason behind is in debug mode there some processes like "log" which is running side-by-side of the program makes the modeler slow. To avoid this we shifted out development from debug mode to release mode which works perfectly fine.

CHAPTER 5 - LITERATURE REVIEW (HAPTICS):

5.1 - Study of Haptics Technology:

5.1.1 - Haptics I/O Model:

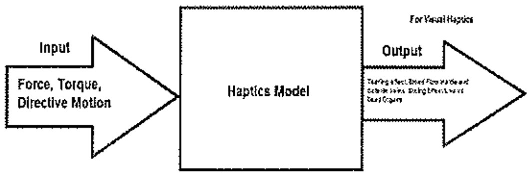


Figure 38-Haptics I/O Model

The Haptic I/O model was the first step to understanding the basics of the project so that further work may be done easily. This model was studies in detail including Visual haptics and Force Feedback Haptic.

5.1.2 - Research Papers Findings (Related to Forced feedback in surgical simulation):

- Learned from Research paper "*Haptic Handwriting Aid for Training and Rehabilitation by James Mullins, Chris Mawson and Saeid Nahavandi*" that the maximum force exerted by the phantom Omni device is 3.3 Newton on the hand. For the liver model

- From Yahoo Answers found the tissue type of human liver, which quotes Hepatocytes (Liver Cells) themselves form the bulk of "hepatic tissue" as they form columns and sinuses.

- Learned from research paper "***Biomechanical Response of Human Liver in Tensile Loading Andrew R. Kemper, PhD, Anthony C. Santago, Joel D. Stitzel, PhD, Jessica L. Sparks, PhD, and Stefan M. Duma, PhD***", that the response of human liver is not only elastic, but it is "**viscoelastic**", which means it has **elastic** as well as **viscous** properties, so our model needs to behave the same.

- Generally a Phantom Omni is designed as a positional output device and force input device, quoted in "***Haptic Handwriting Aid for Training and Rehabilitation by James Mullins, Chris Mawson and Saeid Nahavandi***", but in our project we will be giving position and force as input to control the MIS tools to penetrate or for cutting through, and the output will be the feedback force which the liver will provide as a reaction to the actions.

- The Phantom Omni device overshoots in some cases and in the project "***Haptic Handwriting Aid for Training and Rehabilitation by James Mullins, Chris Mawson and Saeid Nahavandi***" they had used PD (Proportional Differential) controller is introduced that decreases the overshoot and reduces settling time to acceptable boundaries.

- "***Specifically, with increased loading rate the failure stress significantly increased while the failure strain significantly decreased***" quoted in the research paper "Biomechanical Response of Human Liver in Tensile Loading by Andrew R. Kemper, PhD, Anthony C. Santago, Joel D. Stitzel, PhD, Jessica L. Sparks, PhD, and Stefan M. Duma, PhD" *which means that if we increase the loading rate, the point of stress at which failure occurs also increases and the failure strain decreases.*

From research paper : ***"Biomechanical Response of Human Liver in Tensile Loading by Andrew R. Kemper, PhD, Anthony C. Santago, Joel D. Stitzel, PhD, Jessica L. Sparks, PhD, and Stefan M. Duma, PhD"***

Note: Images and data below are obtained from research paper : ***"Biomechanical Response of Human Liver in Tensile Loading by Andrew R. Kemper, PhD, Anthony C. Santago, Joel D. Stitzel, PhD, Jessica L. Sparks, PhD, and Stefan M. Duma, PhD"***

5.1.3 - The Tests:

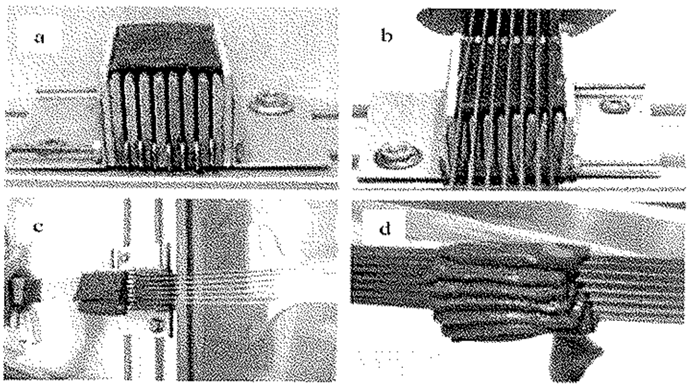


Figure 39-A view of Tests

[2] http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3242546/

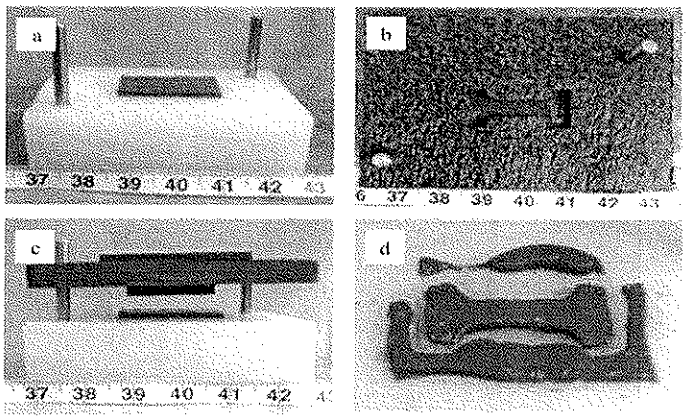


Figure 40-A view of Tests

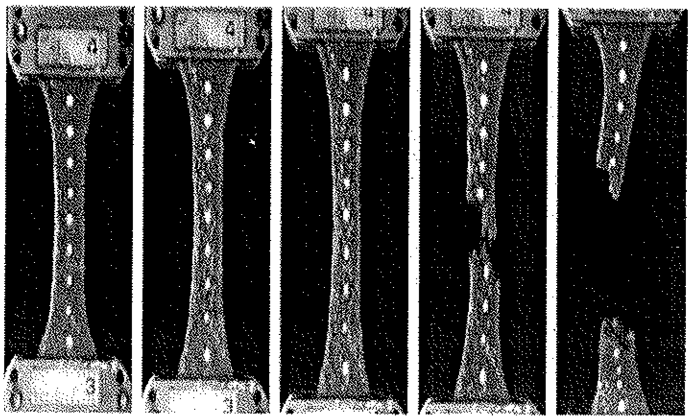


Figure 41-A view of Tests

5.1.4 - The Results:

Averages and Standard Deviations by Loading Rate.

Rate	Desired Strain Rate	Average Strain Rate	Average Failure Stress	Average Failure Strain
	(s^{-1})	(s^{-1})	(kPa)	(strain)
Rate 1	0.01	0.008 (± 0.001)	40.21 (± 21.39)	0.34 (± 0.12)
Rate 2	0.10	0.089 (± 0.014)	46.79 (± 24.81)	0.32 (± 0.05)
Rate 3	1.00	0.871 (± 0.093)	52.61 (± 25.73)	0.30 (± 0.10)
Rate 4	10.00	9.477 (± 1.964)	61.02 (± 24.89)	0.24 (± 0.07)

Figure 42-Results

- So the failure stress is between 45 to 55 kPa although it cannot be exactly quantified because it varies in most cases i-e in the tests provided by the research paper "***Biomechanical Response of Human Liver in Tensile Loading by Andrew R. Kemper, PhD, Anthony C. Santago, Joel D. Stitzel, PhD, Jessica L. Sparks, PhD, and Stefan M. Duma, PhD***" the failure stress ranged in between 7kPa to 95kPa.The failure strain however was almost around 0.3 scale.

5.1.5 - Test 1:

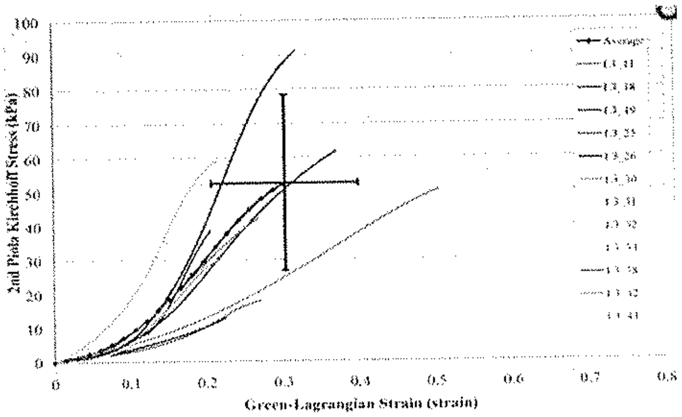


Figure 43-Test 1

5.1.6 - Test 2:

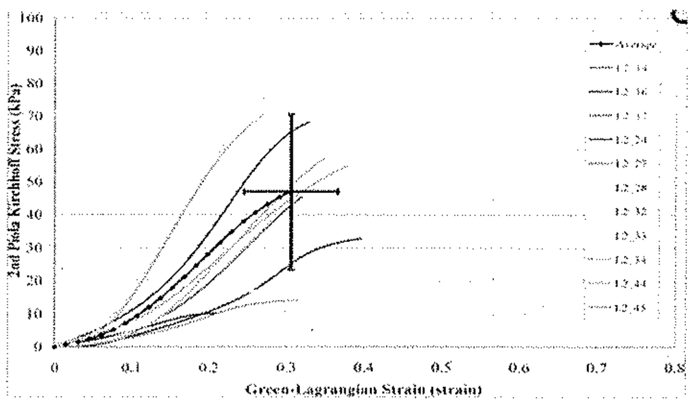


Figure 44-Test 2

➢ "Apart from a patch where it connects to the diaphragm (the so-called "bare area"), the liver is covered entirely by visceral peritoneum, a thin, double-layered membrane that reduces friction against other organs" Quoted in Wikipedia. So the friction of human liver is very less.

➢ Graph of maximum displacement with a concentrated load of 2 N applied:

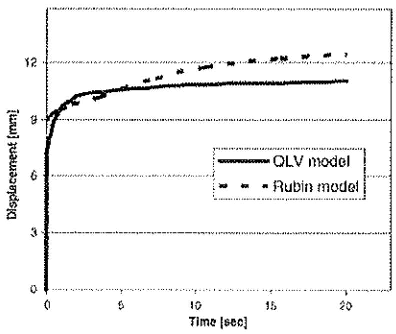


Fig. 16 Case study on the whole liver geometry: maximum displacement with a concentrated load of 2 N applied

Figure 45-Case Study on whole liver geometry

[3]http://www.sciencedirect.com/science/article/pii/S1361841507001004

- Liver Stiffness values ranging from **2.4** to **5.5** are considered as normal. Quoted in ***"Extrahepatic Cholestasis Increases Liver Stiffness(FibroScan) Irrespective of Fibrosis Gunda Millonig,1 Frank M. Reimann,1 Stephanie Friedrich, Hamidreza Fonouni, Arianeb Mehrabi, Markus W. B¨ uchler, Helmut Karl Seitz, and Sebastian Mueller"***

5.1.7 - Conclusions:

So we can use these values:

- ✓ Failure Stress: Between 45 to 55 kPa
- ✓ Failure Strain: Around 0.3
- ✓ Normal Stress : Less than 40 kPa
- ✓ Friction : Very less (Exact value to be determined)
- ✓ Stiffness value range : 2.4 to 5.5

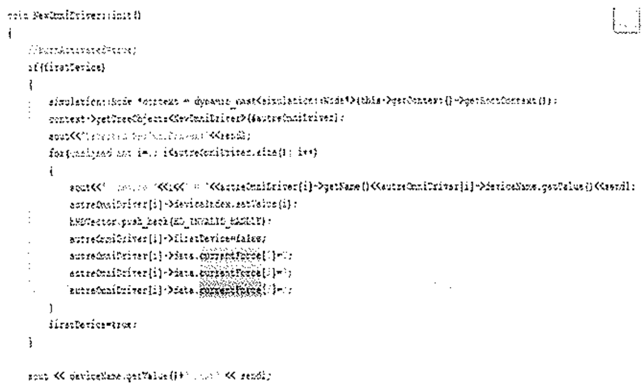
5.1.8 - Custom Made Haptic:

Custom made haptic is an important feature that determines to provide as much accurate and realistic feedback as SOFA can. We don't have pre-defined surfaces in SOFA , we are going to deal with surfaces of all types with different patches, young modulus and stiffness. Whenever it comes to haptic there are two important things

- ✓ How much force is required?
- ✓ How to send that force parameter to Phantom Omni

The first point is more related to literature review that has been done in details and we came up with some solutions as how much force is required for a liver in different scenerios.it should be kept in mind that the liver also vary from person to person. Second point is more related to development point of view and we have to set values of force as well as direction in the code.

There are many technologies like VTK scan through which you can generate 3D model of a liver and also get its runtime features like stiffness, young modulus etc. Once these things are known then the doctor can easily adjust the force as per his requirement.



```
void NewOmniDriver::init()
{
    //bornActivated=true;
    if(firstDevice)
    {
        simulation::Node *context = dynamic_cast<simulation::Node*>(this->getContext()->getRootContext());
        context->getTreeObjects<NewOmniDriver>(&autreOmniDriver);
        sout<<"[illegible]"<<sendl;
        for(unsigned int i=0; i<autreOmniDriver.size(); i++)
        {
            sout<<"[illegible]"<<i<<" = "<<autreOmniDriver[i]->getName()<<autreOmniDriver[i]->deviceName.getValue()<<sendl;
            autreOmniDriver[i]->deviceIndex.setValue(i);
            hHDVector.push_back(HD_INVALID_HANDLE);
            autreOmniDriver[i]->firstDevice=false;
            autreOmniDriver[i]->data.currentForce[0]=0;
            autreOmniDriver[i]->data.currentForce[1]=0;
            autreOmniDriver[i]->data.currentForce[2]=0;
        }
        firstDevice=true;
    }

    sout << deviceName.getValue()+"[illegible]" << sendl;
```

Figure 46-A view of control on haptics

Above is the screenshot for the current force Array that is responsible for two things

- ✓ Amount of force (magnitude)
- ✓ Direction

You can give direction an d magnitude along the three axis x , y , z.

If I set the following values for currentForce array

autreOmniDriver[i]->data.currentForce[0]=0;

autreOmniDriver[i]->data.currentForce[1]=0;

autreOmniDriver[i]->data.currentForce[2]=2;

It represents force of magnitude 2 along positive z axis.

Therefore it is in our complete control we can set the haptic as per the shape and properties of the object. We can also create a plugin through which one can change the values on runtime instead of changing in the code.

CHAPTER 6 - ALL MODULES INTEGRATION ISSUES:

- Mapping of multiple Omni's with grasper and burning instrument created a problem. The device was unable to recognize which one to handle grasper and which device is to handle burning. So whenever we run modeler, the phantom omni got stuck and need to be calibrated to make it work with normal behavior. To solved this we had to write a new routine in our code. That routine manages the autreOmniDriver[i] array in handle Event. So using this we separated the code of burning and grasping.
- In integration of the gall bladder and liver there came a point where Liver and GallBladder showed extreme pliable behaviour. That issue was solved by changing values of young's modulus in CGlinearSolver , Unchecking the two way force effect and removing the "freemotionanimationloop".
- The STL issue aroused because if a model has some open loops or the vertices are intersecting each other, the model cannot be converted to correct STL file and the converted file has 0 vertices, which in turn leads to incorrect .msh file generation and the image cannot be imported in SOFA. So to correct the STL error, we had to check these issues when we were generating 3D images in 3ds Max so that correct .STL files are obtained.

REFERENCES:

[1] http://www.ncbi.nlm.nih.gov/pubmed/18666132

[2] http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1356852/

[3]https://docs.google.com/viewer?a=v&q=cache:HeBe1Q_7MBsJ:homepages.ulb.ac.be/~labo/ISB2011/ISB2011_ScientificProgram_files/155.pdf+flat+plate+compression+tests+performed+on+whole+human+livers+by+Sparks+et+al.+(2007)&hl=en&gl=pk&pid=bl&srcid=ADGEESgjgZTroojrt24P753SvxiQMH5L0OV-rFF4Pq71ks6ucq_bChiFu9z5vfjBXBZQoQsKcxq8Qx_xVj1MP66k2NPll0K1kGX5cT_9bH0EA9W4zI5ydhBuG7py0Hz-5WsUXcszRrAO&sig=AHIEtbTeD-A7ROlrl20I6qa1hj5z-lLZ1w

[4]http://www.sciencedirect.com/science/article/pii/S1361841507001004

[5]https://docs.google.com/a/seecs.edu.pk/viewer?a=v&q=cache:kiQBSdsut1gJ:dro.deakin.edu.au/eserv/DU:30005732/nahavandi-haptichandwriting-2005.pdf+&hl=en&gl=pk&pid=bl&srcid=ADGEESiHYM79gI6gLZg3DiHQ_sgxzuHe0VyZsncym7bxa9unXye58n7q76dlxn7E_Klm8Gm1cLJFbbvBX7mAho9uP24SYpg8laWUgk46HRyQjsgtDeowgjrQ5KUdCXmcIkCQwn_bN9Tf&sig=AHIEtbQtwHmZl6EL8nSAZjfa9OXmr8iSYw

[6] http://www.sciencedirect.com/science/article/pii/S1361841507001004

[7] http://www.princeton.edu/~achaney/tmve/wiki100k/docs/Mutual_exclusion.html

[8] http://www.sofa-framework.org/

[9] http://www.sofa-framework.org/tutorials

...in - Google Chrome

...://api.turnitin.com/newreport.asp?lang=en_us&oid=335030209&ft=1&bypass_cv=1

...rences

...nitin

...ality Report

Processed on: 11-Jun-2013 21:27 PKT
ID: 335030209
Word Count: 4967
Submitted: 3

FYP Report

By Muhammad Usman Shabbaz

Similarity Index	Similarity by Source	
12%	Internet Sources:	7%
	Publications:	5%
	Student Papers:	4%

...ument Viewer

...de quoted | exclude bibliography | exclude small matches ▾ mode: show highest matches together

CERTIFICATE It is certified that the contents and form of thesis entitled [17]

"Surgical Training Simulator" submitted by Muhammad Waleed Khan (269), Muhammad Usman Shahbaz (268) and Muhammad Rizwan (264)

have been found satisfactory for the requirement of the degree. Advisor: Shamyl bin Mansoor ( ) Co-Advisor: Muhammad [13]

Muddassir Malik ( ) DEDICATION To Our Parents, Teachers & the People who supported us in our lives

ACKNOWLEDGEMENTS We would like to take this opportunity to pay our humble gratitude to Almighty Allah who blessed us with his kindness to complete this project. We are highly thankful to our Advisor Mr. Shamyl bin Mansoor and our Co- advisor Mr. Muhammad Muddassir Malik for [6]

guiding us in the project and for encouraging us to achieve our goal. Their support has helped us complete many ambitious tasks and has shown us the right way whenever we were in lost direction.

We would like to offer appreciation to our parents. Their supplications, criticism and encouragements were the means by which we have completed our project in time. [9]

Last but not the least, we would like to thank to the people who have always supported us. We are indebted to them for the bulk of the information. Table of Contents

1. 3% match (Internet from 11-Jun-2013)
http://www.ncbi.nlm.nih.gov
2. 1% match (publications)
S. Nahavandi. "Haptic Handwriting Aid for Training and Rehabilitation", 2005 IEEE International Conference on Systems Man and Cybernetics, 2005
3. 1% match (Internet from 30-Nov-2012)
http://seecs.nust.edu.pk
4. 1% match (publications)
Gunda Millonig. "Extrahepatic cholestasis increases liver stiffness (FibroScan) irrespective of fibrosis", Hepatology, 11/2008
5. 1% match (Internet from 26-Mar-2010)
http://www.procto-med.com
6. 1% match (student papers from 28-Dec-2010)
Submitted to Higher Education Commission Pakistan
7. < 1% match (Internet from 23-Jan-2013)
http://rekompozycja.pl