Development of an Autonomous Platform for Surgical Simulation and Training

By

NIHAR N. SHETH

B.S., Mechanical Engineering, Maharashtra Institute of Technology, Pune, 2012

M.S., Mechanical Engineering, University of Illinois at Chicago, 2014

THESIS

Submitted as partial fulfillment of the requirements

for the degree of Doctor of Philosophy in Bioengineering

in the Graduate College of the

University of Illinois at Chicago, 2021

Chicago, Illinois

Defense Committee:

Thomas Royston, Chair

Cristian Luciano, Advisor

Dieter Klatt

James Patton

Nishant Srinivasan, Department of Pediatrics, UIC

To my loving Parents,

for the infallible support and

for being such an inspiration to me

Acknowledgements

I would like to thank my academic advisor Dr. Cristian Luciano for the patience, guidance and support through the years. This work would be impossible without his ideas and feedback. I would like to express my gratitude towards Dr. Nishant Srinivasan and Dr. Saurabhkumar Patel for their support and immensely valuable feedback. Their conduction of the trials has been the backbone of this research. I would also like to extend my gratitude towards the Head of Department at Bioengineering Dr. Thomas Royston for the support during the period of this research.

I am sincerely grateful to my lab mate Nick Marjanovic for valuable feedback and help with the research. I would like to personally thank Dr. Rachel Yudkowsky for her engagement and valuable insight for a portion of the research. I would also like to thank Dr. Miiri Kotche for providing valuable opportunities and experiences to learn during MADLab.

Finally, I would like to thank my wife Dagny Dhumal and my parents Nitin and Rashmi Sheth for their constant love, patience and unwavering support. I am immensely grateful for all not mentioned here, but in some ways have helped with the research over the years.

Table of Contents

List of Tables

List of Figures

Abbreviations

SBT	Simulation Based Training
NPC	Neonatal pericardiocentesis
NTC	Neonatal thoracentesis
PTX	Pneumothorax
CVC	Central venous catheter
PVC	Percutaneous venous catheter
UVC	Umbilical venous catheter
EM	Electromagnetic
CEU	Central electronic unit
SW	Dassault System Solidworks
CAD	Computer-aided Design
3DsM	Autodesk 3DsMax
PLA	Polylactic acid
TPU	Thermoplastic polyurethane
PVC	Polyvinyl chloride
SSS	Surgical Simulation Software
VR	Virtual reality
App	Application
CSV	Comma-separated values
ER	Entity-relationship diagram
TF	TensorFlow

NN	Neural network
CNN	Convolution Neural Network
SCNN	Siamese Convolution Neural Network
NICU	Neonatal Intensive Care Unit
FastSTray	Fast trajectory simplification algorithm
ICS	Intercostal space

Chapter 1 - Introduction

Traditional surgical training was based on the principal of ‘learning by doing’, where the trainee surgeon over a period of time would encounter varied clinical scenarios that would gradually help them improve their skills. However, in recent years the increased accountability for the hospitals and the doctors over patients has necessitated rigorous methods of preparedness of trainee doctor (Kneebone, 2003). The need for safer ways to train and improve psychomotor skills has led to the advent of Simulation Based Training, which has created the opportunity to improve clinical and technical skills to perform simple and complex surgical procedures(Agha & Fowler, 2015). SBT training can also be used for timely refreshing skills for procedures that are rarely conducted.

1.1 Simulation-based training

SBT involves training in a simulated environment on modalities such as cadavers, animals, physical manikins or a computer-based applications. Human cadavers show realistic anatomy; however, are hard to find and extremely expensive, additionally, not all the medical trainees get an opportunity for hands-on practice. Animal models usually have different anatomy as compared to humans and the moral and ethical responsibility that comes with it makes it challenging to use this on a large scale (Hammoud et al., 2008). Computer-based applications use novel technologies, such as haptic devices, virtual reality headsets and other interfaces to provide an immersive experience for simulating surgery, but they do not provide the necessary experience of working with real instruments and some of the interfaces like virtual headsets cause headaches and other symptoms after prolonged use. Physical manikins or trainers have proved to be a viable mode of training for surgical procedures, they provide the ability to repeatedly perform a surgical procedure

(Agha & Fowler, 2015). The physical manikins can be designed with details anatomy, induced with variation of complex clinical conditions for improving trainee experience.

SBT has major advantages such as the opportunity to practice hands-on, immediate feedback of performance, experience complex surgical situations (Maran & Glavin, 2003). Research such as (McGaghie, 2012) have been able to show that skills learned during SBT have trickled down to clinical situations.

Traditionally, SBT is usually conducted in the following sequence:

1. Expert instructor explains the symptoms that would lead to the conduction of the surgical procedure.
2. Instructor explains the details of the surgical procedure.
3. Instructor demonstrates the surgical steps on the simulation modality such as a physical manikin, an animal etc.
4. Trainee practices the procedure under the guidance of the instructor.
5. Trainee memorizes the steps for the procedure.
6. Finally, instructor conducts a one-on-one assessment where the trainee performs the procedure without any guidance.

Looking at the steps above its clear that the training sessions are heavily instructor dependent. These sessions require experience instructors. The need for one-on-one assessment makes these sessions long and physically exhausting. The assessments are also subjective to the instructor conducting the session. Additionally, during the assessment the instructors regularly have to bend-over the surgical table to observe the trainee’s actions; this creates an undue stress on the instructor over the period of the training session. The need for heavy involvement of the instructor makes these sessions expensive and dependent on the availability of the instructor. Consequently, it is hard to hold these sessions on a regular basis.

The goal of this research was to create an autonomous simulator that could provide real-time feedback during training and an accurate and objective assessment of trainee performance. As a proof of concept, we are simulating neonatal pericardiocentesis (NPC) and neonatal thoracentesis (NTC).

1.2 Motivation for simulating NPC and NTC

NPC is performed when there is a pericardial effusion causing a cardiac tamponade; similarly, NTC is performed when pneumothorax is diagnosed. These conditions are highly rare and life-threatening complications in neonates. Pneumothorax in the Neonatal Intensive Care Unit (NICU) has reported an incidence rate of 1-2% and over 40% in the presence of respiratory distress syndrome(Ogata, Gregory, Kitterman, Phibbs, & Tooley, 1976)(Nowlen, Rosenthal, Johnson, Tom, & Vargo, 2002). Pericardial effusion/cardiac tamponade have an incidence rate of 0.7% to 2% (Chioukh, Ameur, Hmida, & Monastiri, 2016)(Pizzuti, Parodi, Abbondi, & Frigerio, 2010). Challenging diagnosis of these condition causes delayed intervention and treatment of these conditions require swift action and superior skills. Due to the infrequent occurrence of these conditions, alternate approaches to training in a safe and replicative environment is paramount. Especially, since these are rare conditions, accessible simulation training can help maintain skill levels.

1.3 Organization of the thesis

This dissertation is organized in 10 chapters, as follows. Chapter 2 describes the neonatal procedures in detail and evaluates the recent research on simulating both the surgical procedures.

Chapter 3 details the available surgical simulators that can automatically assess performance and their shortcomings. Chapter 4 demonstrates the setup of the simulator. Chapter 5 gives a detailed description of the fabrication process used for the physical manikin. Chapter 6 is based on the software developed for the simulator and its uses. Chapter 7 elaborates on the development of the neural networks and their training process. Chapter 8 is detailing all the testing that was conducted to validate the different parts of the simulator. Finally, Chapter 9 is the list of conclusions from our research.

Chapter 2 - Neonatal Surgical Procedures

2.1 Neonatal Pericardiocentesis

NPC is the emergency removal of air or fluid when diagnosed with cardiac tamponade caused by pericardial effusion or pneumopericardium. In other words, NPC is the removal of excess air or fluid that accumulates in the pericardial space (Fig 1**).** Cardiac tamponade is a rare and life-threatening condition, usually caused due to central venous catheters (CVC), percutaneous (PVC) and umbilical (UVC) that are administered to neonates. Pericardial effusions are usually

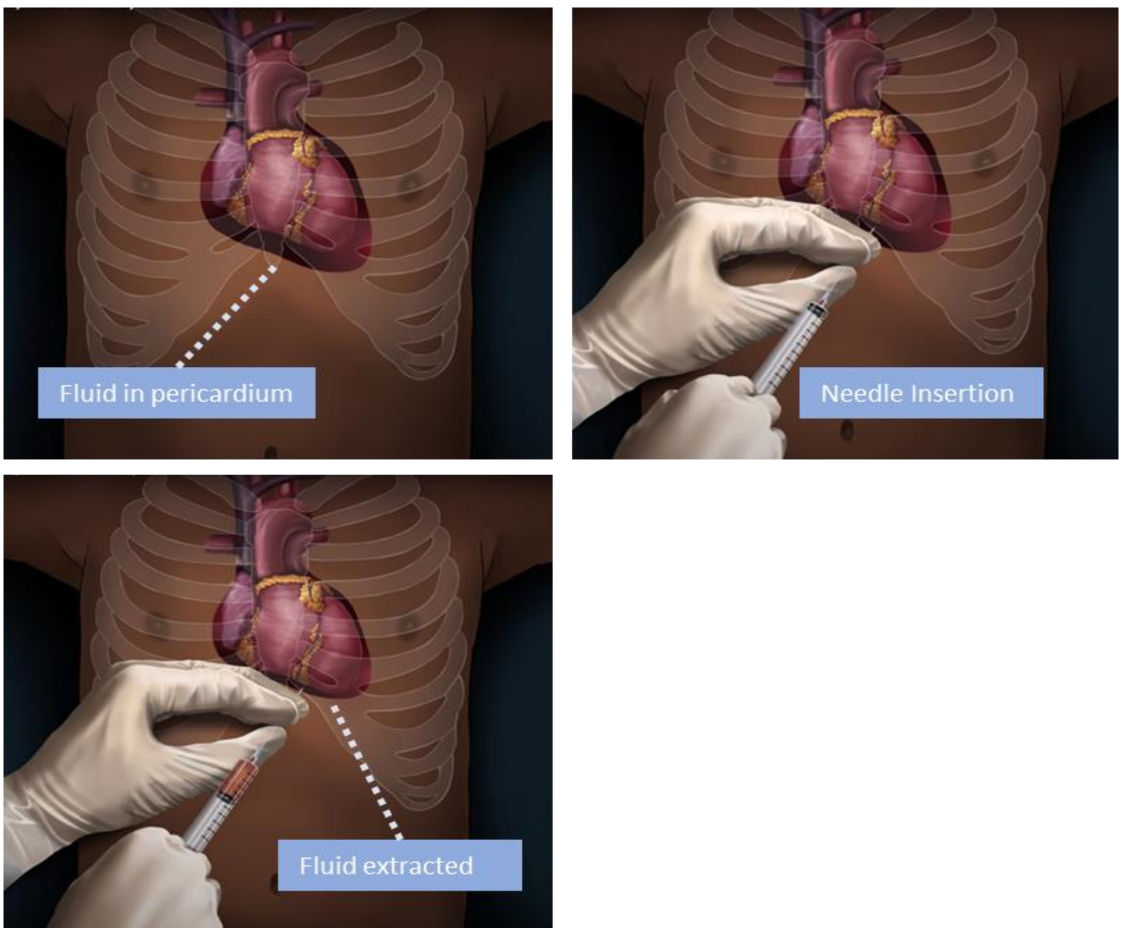


Figure 1. Emergency pericardiocentesis (sourced from OpenPediatrics™)

found in hydropic or septic neonates. Early detection and treatment are paramount, especially for neonates.

Usually, a cardiac tamponade is diagnosed through imaging, specifically an echocardiogram or an ultrasound. Furthermore, on a most regular basis pericardiocentesis is conducted under the guidance of ultrasound imaging. However, in emergency situations, such as a cardiovascular collapse, the surgical area is sanitized with betadine and a needle is inserted into the pericardium without any image guidance (Angert & Rosen, 2021). This research is focused on training for such emergency NPC mentioned above.

2.2 Neonatal Thoracentesis

The incidence of pneumothorax (PTX) in preterm infants is around 6%. Around close to 10% of infants, PTX is an incidental finding as these infants may remain asymptomatic (Liu et al., 2020). The Neonatal Intensive Care Unit (NICU) alone has reported an incidence of 1-2% and over 40% in the presence of respiratory distress syndrome. This condition is categorized under “air-leak syndromes” caused by leakage of air from a ruptured alveoli into the pleural space. The trapped and sometimes accumulating air could compress the lungs and mediastinum resulting in hemodynamic comprise (Fig 2). This is known as tension PTX. In neonates, tension PTX although rare is considered a life-threatening situation requiring immediate intervention to depress the trapped air. This procedure is called thoracentesis (TRS) and is performed using either a needle or trocar and chest-tube or a flexible catheter called “pig-tail” to remove the excess air from the pleural cavity (Wei, Lee, Cheng, Tsao, & Hsiao, 2014).

Often PTX could present with only mild respiratory symptoms or remain occult until enough air accumulates in the space to cause compressive effects. With the onset of tension pneumothorax or compressive effects, clinical condition of infants can deteriorate rapidly requiring swift action. Although, the technique of thoracentesis is well-established, suboptimal skill levels of providers due to inadequate exposure to the condition or training could result in significant complications.

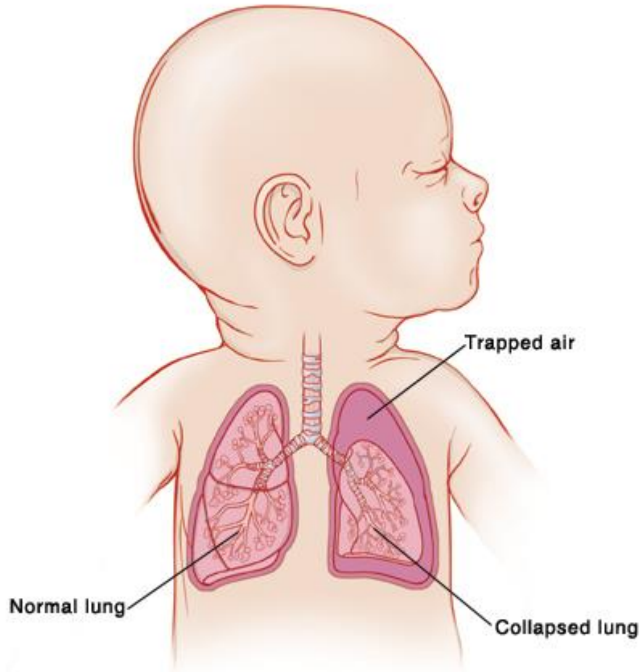


Figure 2. Pneumothorax. (Courtesy Fairview)

2.3 Existing trainers

Currently, a large segment of institutions, use the animal models such as piglets and hens to simulate NPC and NTC. However, these models have a low-shelf life and have a different anatomy as compared to real neonates. There are several alternative options that can be used. A brief description of some infant trainers commercially available or constructed for simulation are given below:

Figure 3. Trubaby X by TruCorp

2.3.1 TruBaby X

This a high-fidelity manikin made by Tru Corp, which allows for multiple surgical procedures that could be conducted on a 5-month infant (Fig 3). The manikin displays all the necessary realistic anatomical landmarks necessary to conduct needle thoracentesis. Additionally, the procedure can be performed on both sides and the silicone insert allows for upto 150 needle piercings with a 18G needle before needling replacement. However, this manikin costs upwards of $7500 with additional costs of replacing the silicone inlets. The manikin being a 5-month infant, is proportionally different from a neonate. While thoracentesis can be performed quite realistically, the manikin does not allow for pericardiocentesis.

2.3.2 Low-cost pericardiocentesis model

Proposed by Angert & Rosen (2021) is a neonatal manikin made from a repurposed resuscitation manikin from Laerdal™ and some household items (Fig 4). A liquid filled balloon in used as a heart to simulate the procedure. The internal anatomy is covered with a layer of silicone to simulate skin. Though this model is easy to fabricate, the balloon has to be replaced after a single attempt. The manikin has partial anatomical landmarks, and the internal anatomy is not realistic. Additionally, this manikin fabrication depends on the possessing a neonatal manikin that can be repurposed.

Figure 4. Construction of low-cost thoracentesis model.

2.3.3 Low-cost thoracentesis model

Gupta & Ramasethu (2014) proposes a neonatal thoracentesis model that uses electric wires to create a ribcage, which encases a supportive Styrofoam and inflated plastic bags to simulate the pleural space (Fig 5). The assembly is placed inside a toy doll and covered with sheaths of shelf liner to resemble skin and muscle. Similar to the low-cost pericardiocentesis model, this model too has partial anatomical features, and unrealistic internal anatomy. The anatomical landmarks are missing, and the use of inflated plastic bags means that it allows for a single needle insertion before needing to be replaced.

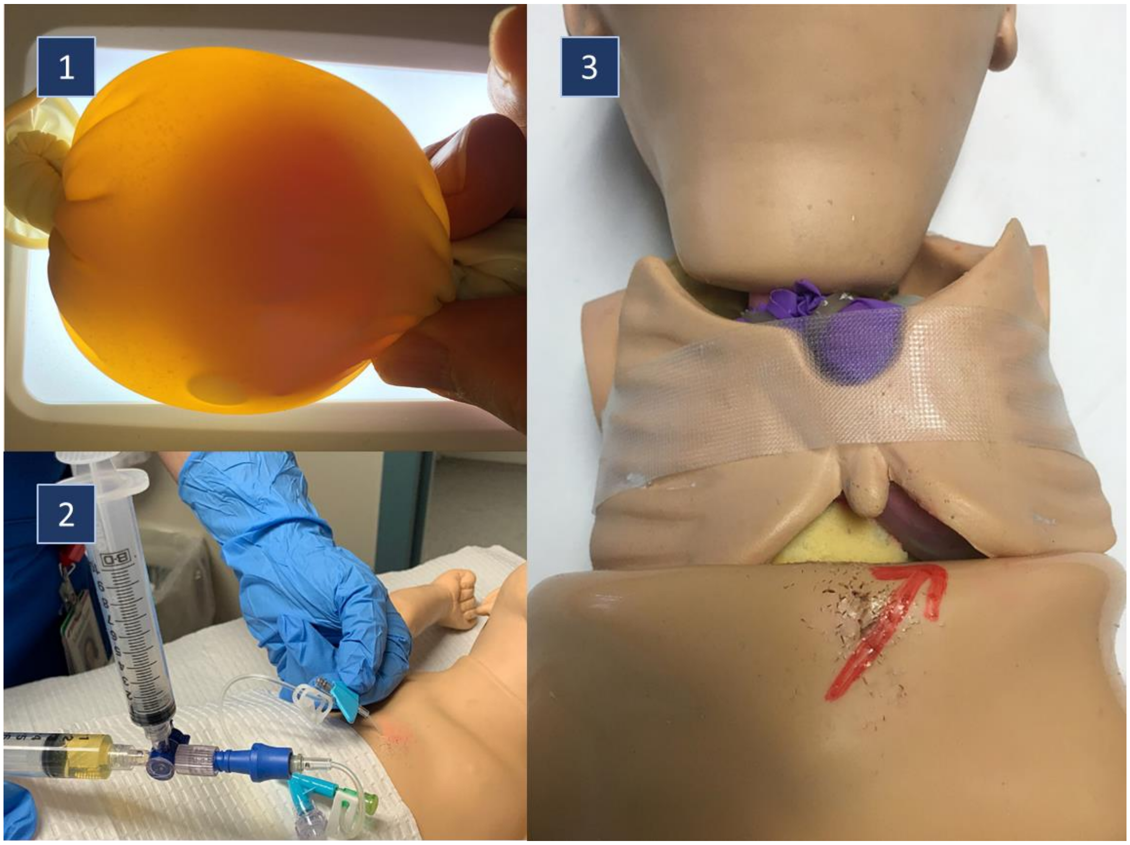


Figure 5. Low-cost pericardiocentesis mode (Angert & Rosen, 2021).

1. Balloon filled with liquid, used as the heart.

2. Pericardiocentesis being performed on the manikin.

3. Manikin with its skin pulled down to reveal the cavity.

2.3.4 Low-cost pleural pigtail trainer

Zurca et al. (2020) proposes an alternative thoracentesis model, similar to Angert & Rosen (2021), this research also repurposes a discarded resuscitation manikin, specifically a Sani Baby by Simulaids (Fig 6). 250 ml bags of saline were used to simulate the fluid accumulated in the pleura and electric wires were used to construct the ribcage. A shelf liner was used to cover the anatomy. This manikin provides an opportunity to use the Sledinger technique for the placement of the pigtail catheter, however like the previous mentioned manikins, this needs frequent replacement, has partial anatomical landmarks, and has unrealistic internal anatomy.

Figure 6. Low-cost fluid filled thoracentesis model

2.4 Chapter Summary

1. NPC and NTC are vital surgical procedures performed during life-threatening situations.
2. Training with animal or low-fidelity manikins does not provide a realistic experience.
3. High-fidelity manikins offer a realistic experience but are prohibitively expensive especially since the silicone inserts need to be replaced after several uses.
4. There are currently no trainers that can be used to train for both NPC and NTC.

Chapter 3 - Hardware Setup

3.1 Setup

The simulator is comprised of mainly two components: the simulation software and a physical manikin. The user can perform the surgical procedure on the manikin with real instruments. Each instrument is attached to an electromagnetic (EM) sensor, the information from

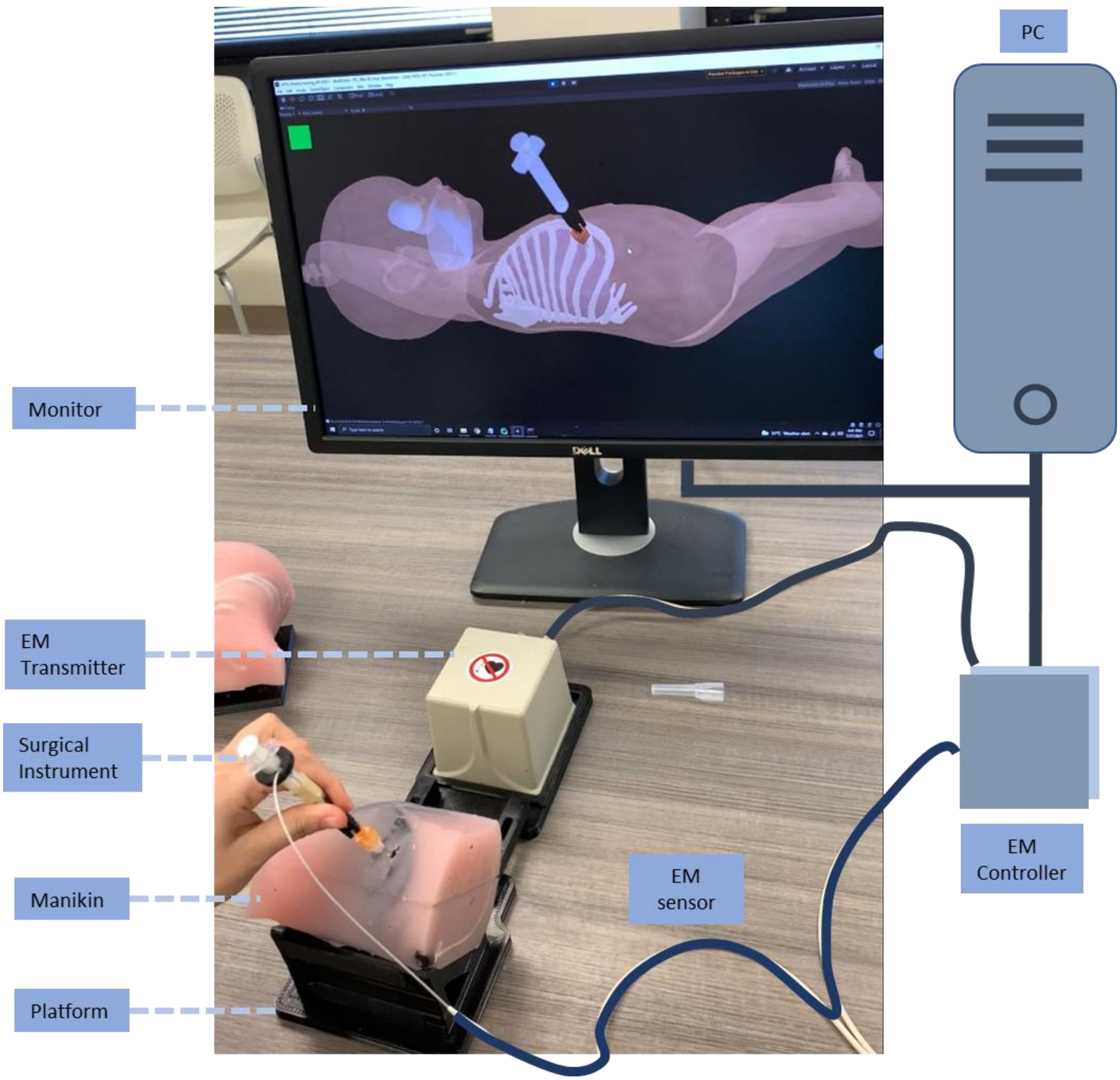


Figure 7. Simulator setup

the sensor is sent to the PC that hosts the simulation software. The information received is analyzed and displayed on a monitor attached to the PC. The comprehensive setup is illustrated in the figure above.

The following is the list of components in the setup:

1. Personal computer (PC)
2. High resolution display
3. EM tracking system
    a. sensors
    b. transmitter
    c. central electronic unit (EU)
4. Physical Manikin
5. Mounting platform
6. Surgical Instruments
7. Sensor mounts

3.2 Electromagnetic Tracking System:

There are numerous technologies to track motion such as accelerometer-gyroscope devices and multi-camera/optical sensor setups (Funke, Mees, Weitz, & Speidel, 2019) (Sarin, Bettadapura, Essa, Zia, & Sharma, 2018). However, accelerometer-gyroscopes devices only track changes in velocities and acceleration. Multi-camera setups require complicated algorithms to recreate a 3D motion in space which are computationally and memory intensive, and above all, require a line of sight to track objects. Electromagnetic tracking (EM) has low-latency and high accuracy in tracking position and orientation at high sampling rates with none of the above-mentioned disadvantages. However, the tracking accuracy is dependent on the presence of metallic objects and the distance between the EM sensor and transmitter. Since the surgical field is small and the instruments used do not affect the magnetic field, the NDI Ascension's DriveBay EM tracking system was considered suitable.

The setup comprises of a central electronic unit (CEU), transmitter and sensors. The CEU digitizes the signal from the sensor and calculates the position and orientation, per CEU at most 4 sensors can be tracked. The transmitter emits a low-intensity magnetic field; the sensors can be tracked with in this field. The EM system transmits data to the PC through a standard USB port, at a default sampling rate of 80 Hz.

3.3 Design of Sensor Mounts

The EM sensors need to be securely attached to the surgical instruments for accurate tracking. Additionally, the sensor is required to be attached in the same place and orientation on the respective instrument to avoid the need for calibration and to get consistent tracking data. For the reasons above, a set of sensor mounts were designed in Solidworks, and 3D printed in polylactic acid (PLA). The mounts easily attach to the surgical instruments in a predefined location and orientation and secure the sensor in-place, the mounts were secured to the instruments using architect's tape.

All the instruments could accommodate the designed sensor mounts, however, attempts to create a mount for the trocar-catheter and the scalpel, failed. Due to the non-rigid nature of the catheter, the mount could not be secured without hindering the functionality of the instrument, similarly due to thin profile of the scalpel, the sensor mount impeded the grip. Instead, the sensor was taped along the length of the instruments.

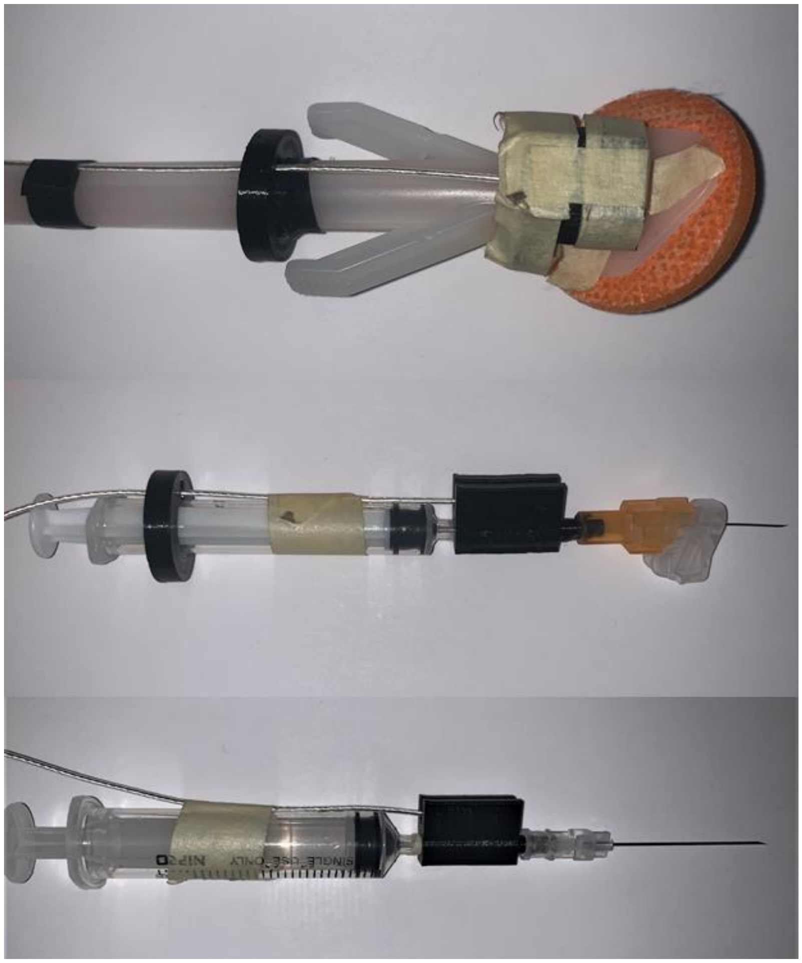


Figure 8. Surgical instruments attached to sensor mounts

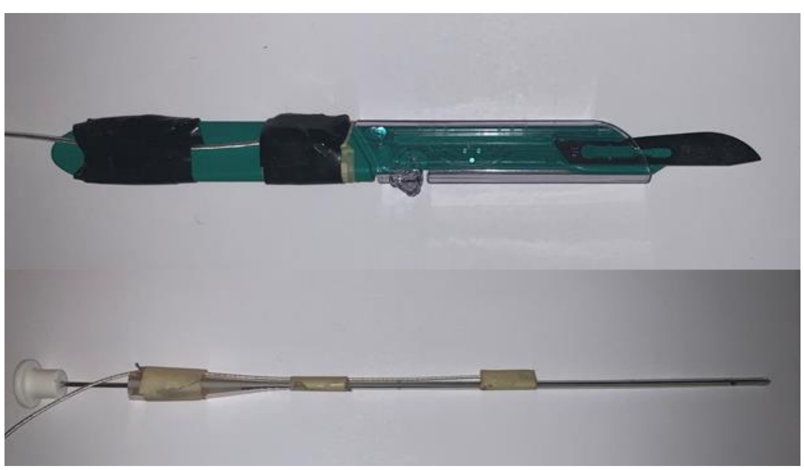


Figure 9. Surgical instruments without sensor mounts

The following series of images display the design of the sensor mounts in 2D and in iso-metric views.

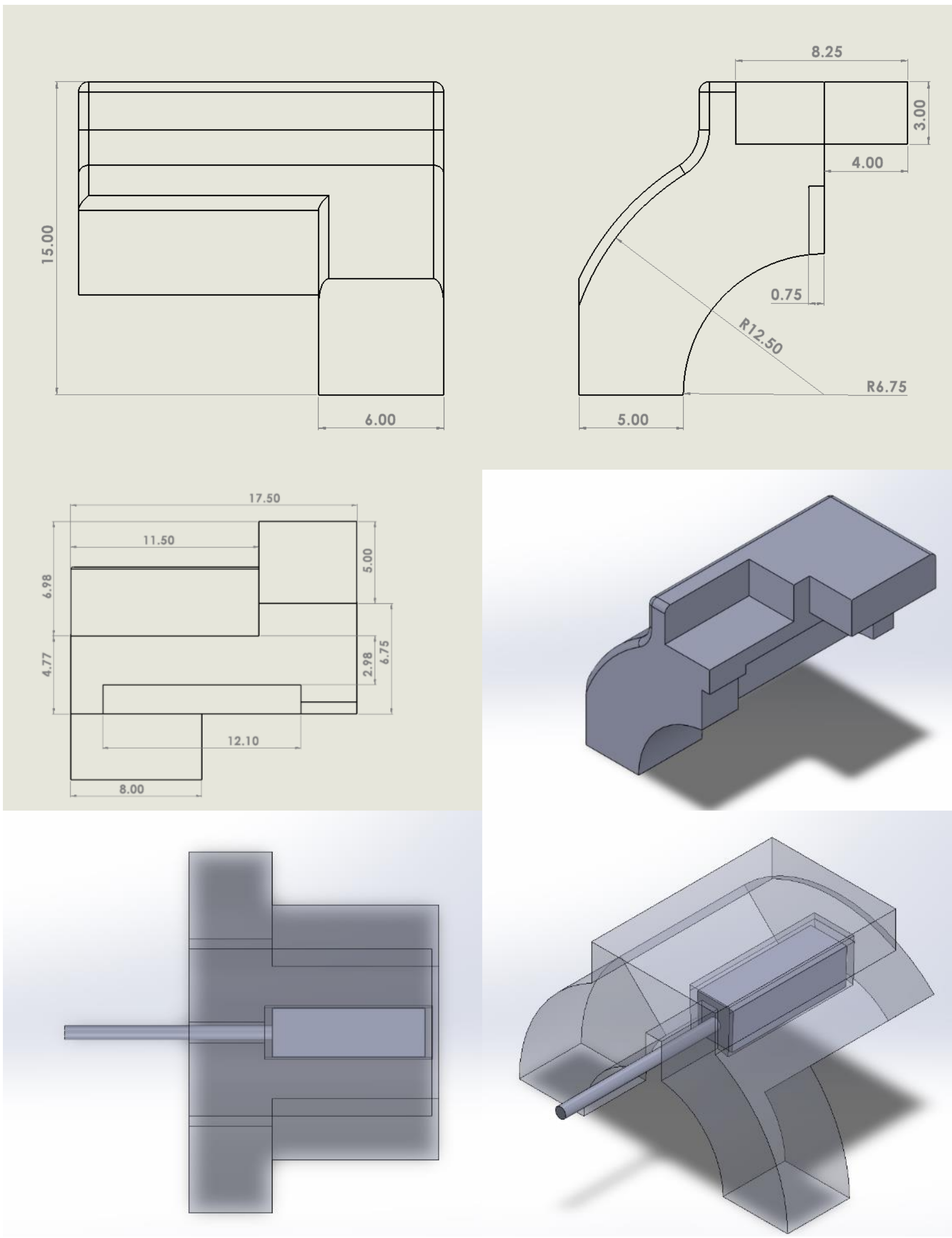


Figure 10. Sensor mount for ChloraPrep

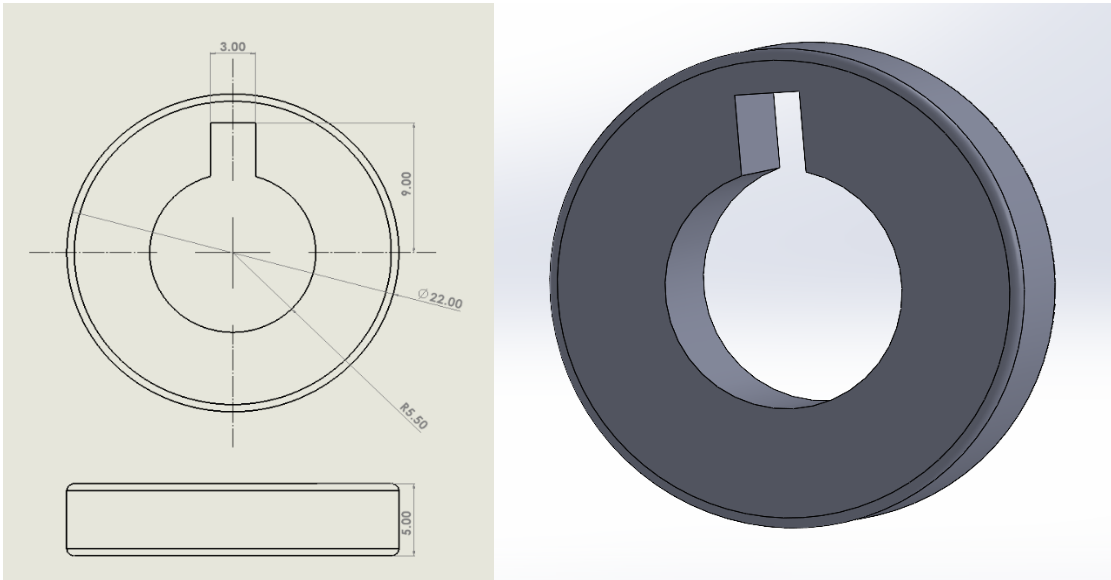


Figure 12. Cable guiding ring

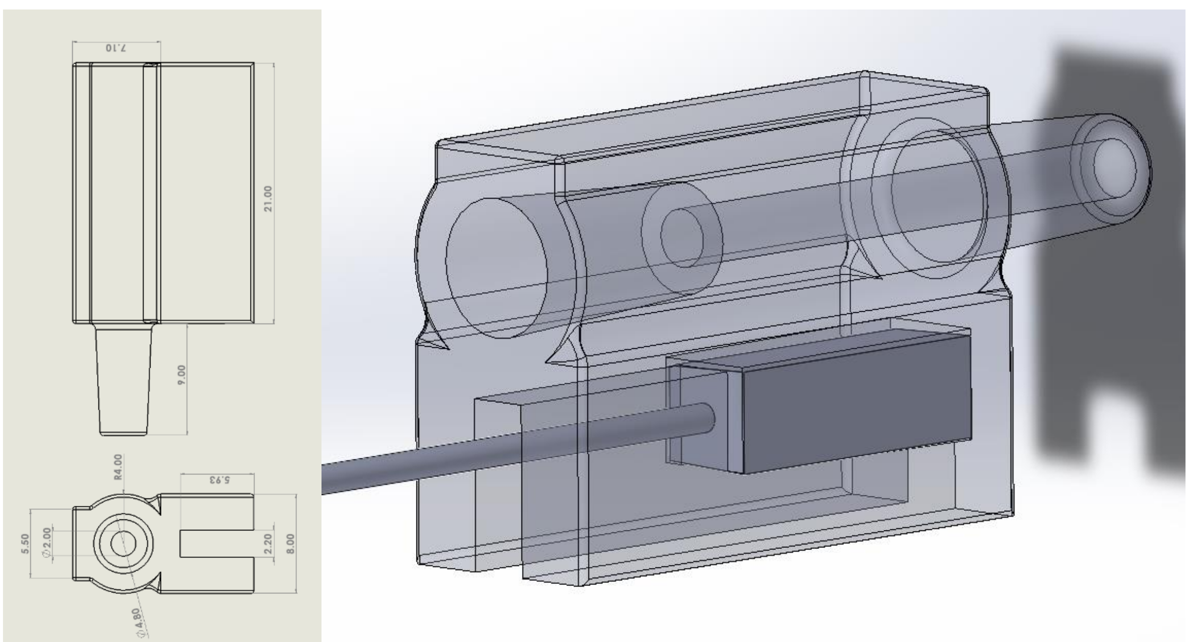


Figure 11. Sensor mount for needle and syringe

3.4 Design of Manikin Platform

The position and orientation of the manikin needed to be constant in reference to the transmitter because it was decided not to attach an EM sensor to the manikin. Additionally, the manikin needed to be placed in an orientation specific to the surgical procedure. For performing NTC, the infant in tilted to side at about 45° for ease of access. This was achieved using a custom platform that was designed in Solidworks and then 3D printed in PLA. The platform was assembly from multiple components that could be printed separately and assembled when needed. The platform has mainly two sub-platforms that hold the transmitter and manikin. However, since the

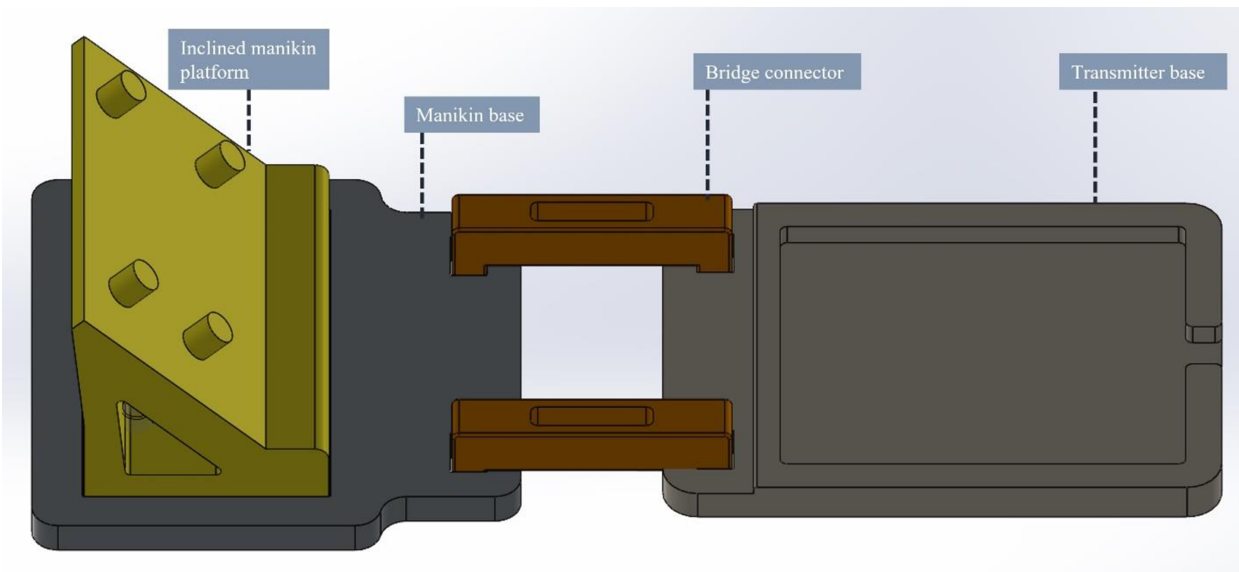


Figure 13. Platform assembly

manikin needs to be in different orientations based on the surgical procedure, the manikin base has different adaptors on which the manikin can be fixed. The two base platforms are held together using bridge connectors. The following is a list of parts that were created:

1. Transmitter base
2. Manikin base
3. Flat manikin platform (adaptor for NPC)

4. Inclined manikin platform (adaptor for NTC)
5. Bridge connector

The simple assembly of the platform has been displayed in Fig 13. The two bridge connectors join the transmitter and the manikin base. The EM transmitter can be push-fitted into the transmitter base, while one of the inclined or flat platforms can be fitted on the manikin base. The inclined platform is used for pericardiocentesis, while the flat is used for thoracentesis.

Given below (Fig 14-18) are a series of images detailing the dimensions of the parts:

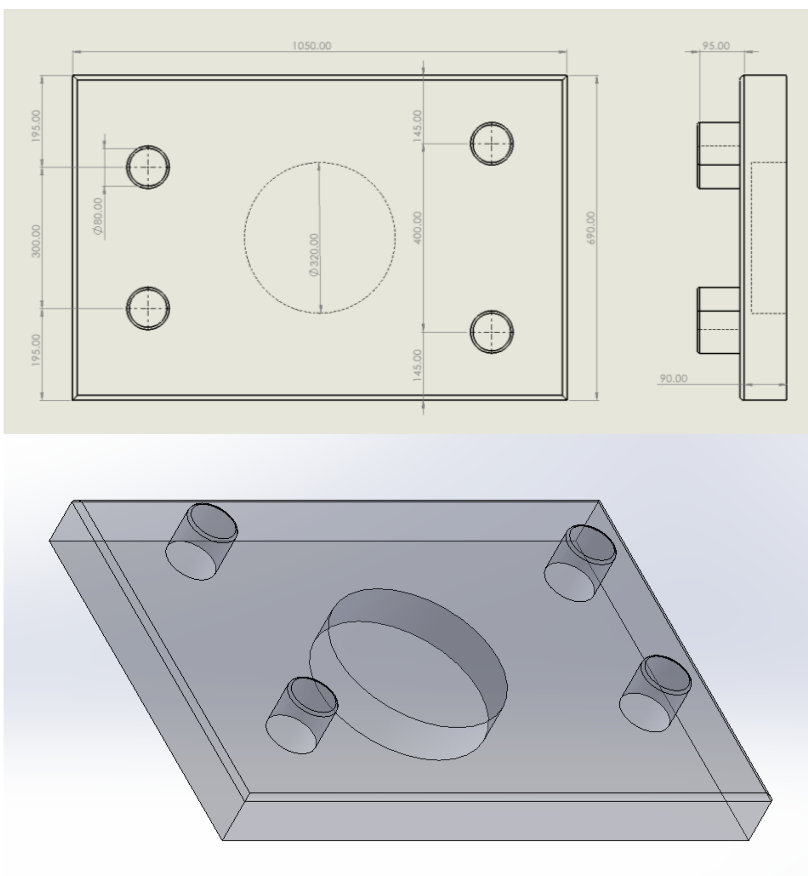


Figure 14. Flat manikin platform

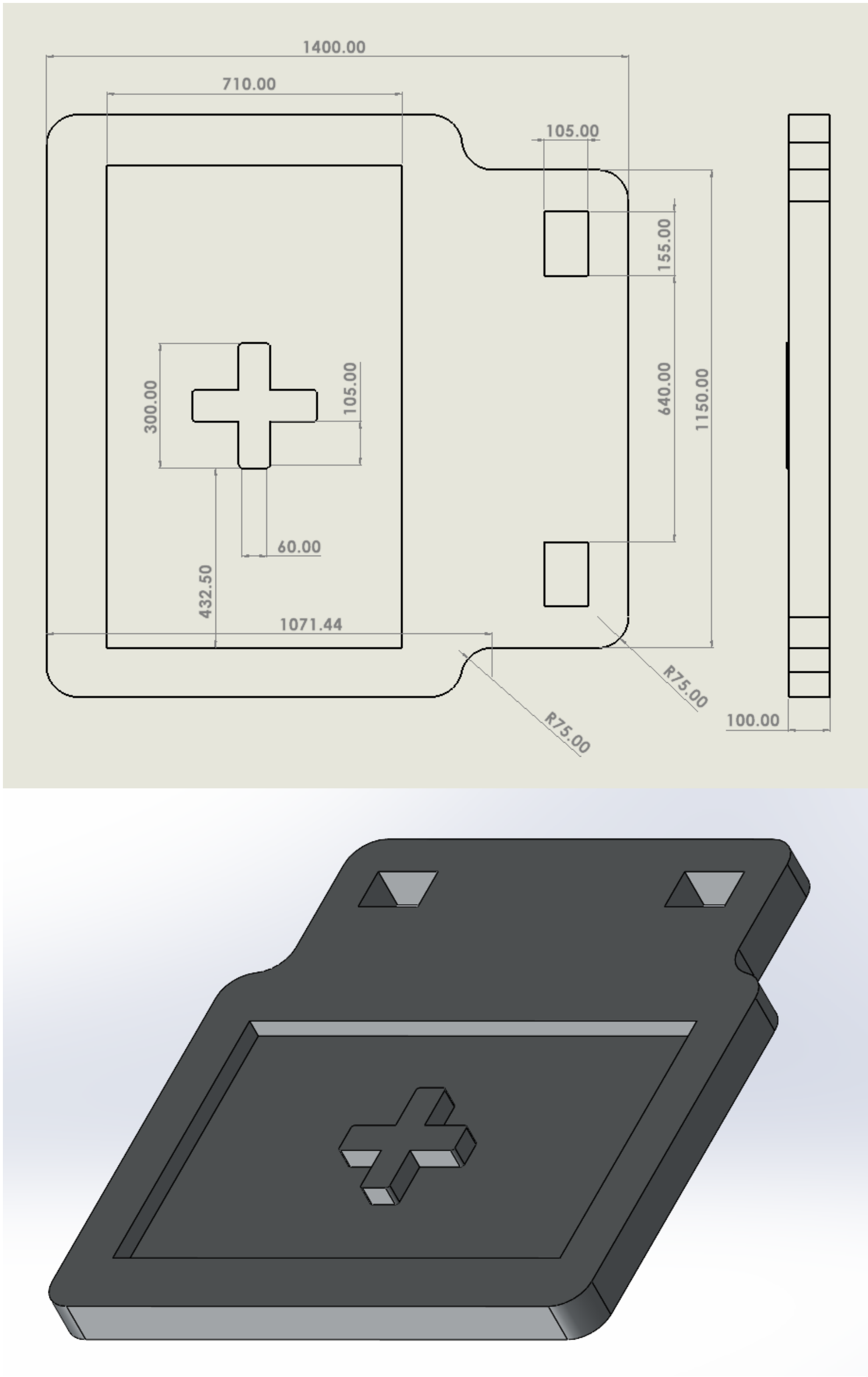


Figure 15. Manikin base dimensions.

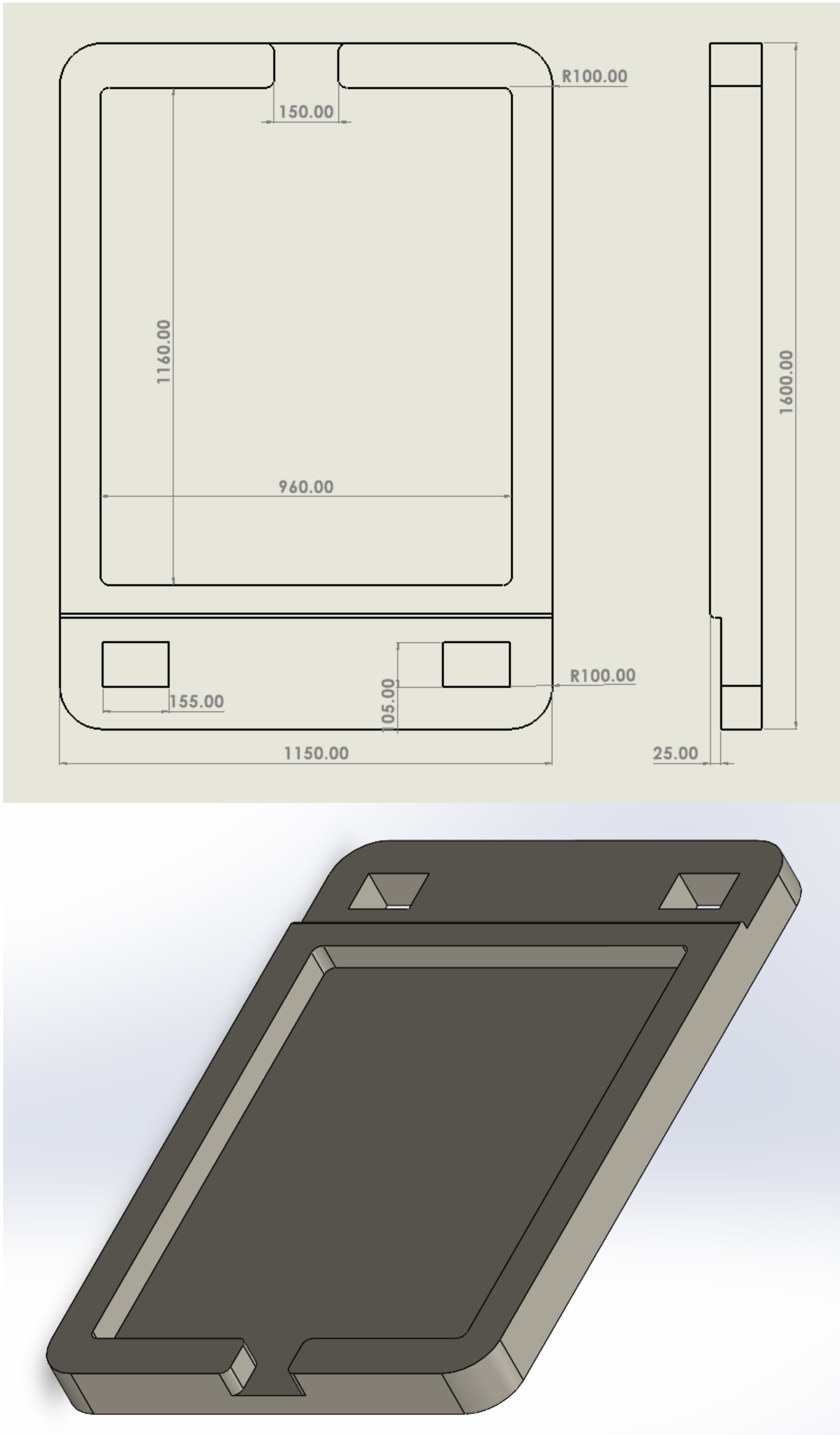


Figure 16. Transmitter base

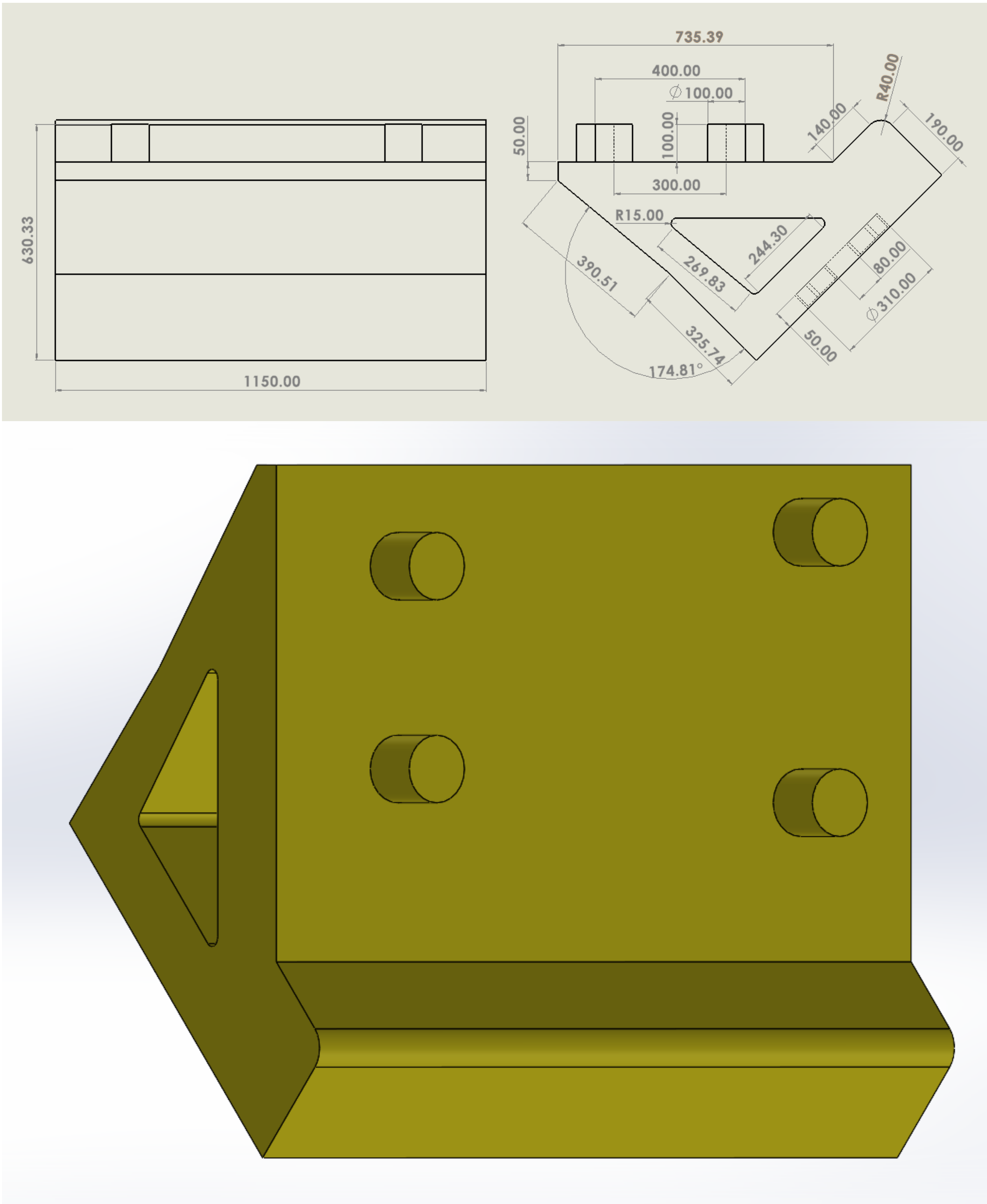


Figure 17. Inclined manikin platform

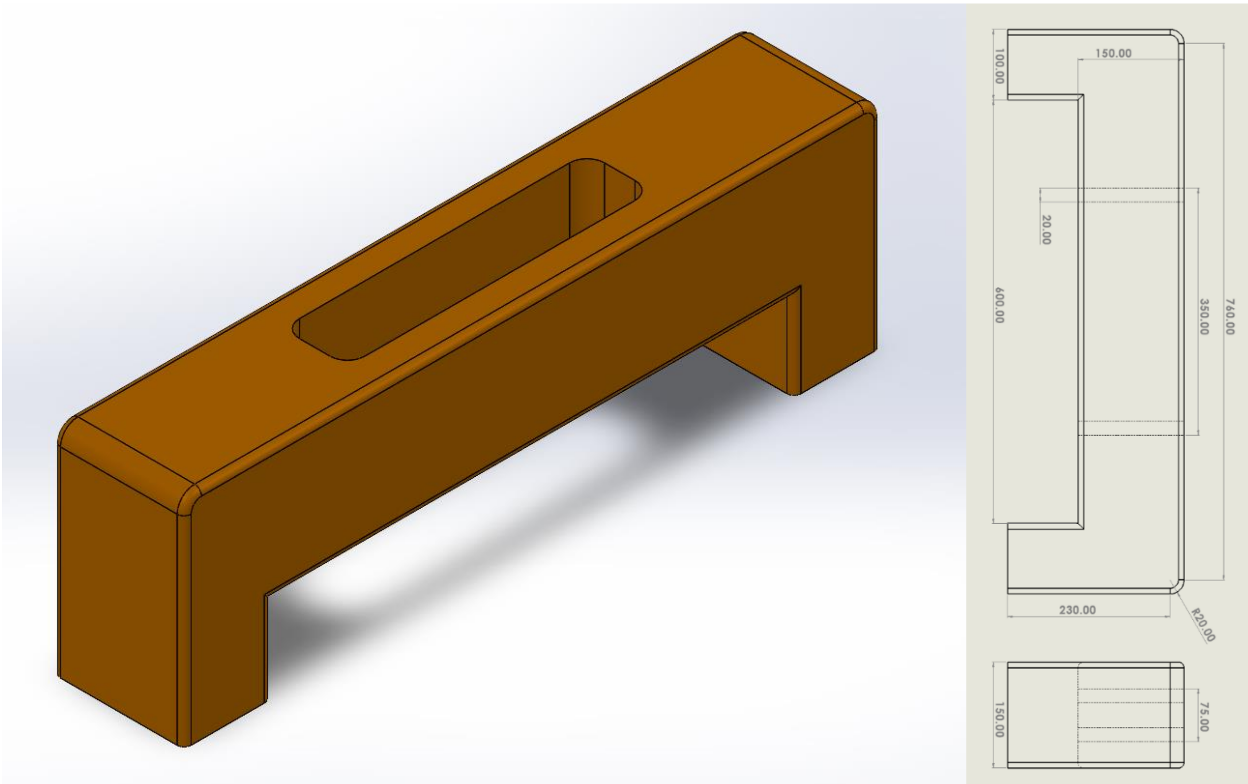


Figure 18. Bridge connector

3.5 Chapter Summary

1. The simulator comprises of two main components: a physical manikin and a virtual reality software.
2. The simulator uses an electromagnetic tracking system to interface between the physical and virtual.
3. The sensors are mounted to the surgical instruments using sensor mounts that were 3D printed.
4. The manikin and the EM transmitter are held together using a custom platform that was 3D printed.

Chapter 4 - Manikin Fabrication

Based on the hardware setup, discussed in Chapter 3, the physical manikin and the displayed virtual anatomy are geographically identical. This was necessary to accurately replicate the physical surgical actions in the virtual world. Considering the trainers discussed in Chapter 2 and by considering the opinions of expert neonatologist, the following features and were listed:

1. Presence of relevant and accurate anatomical features
2. Providing realistic tactile experience
3. Ability to perform both the surgical procedures

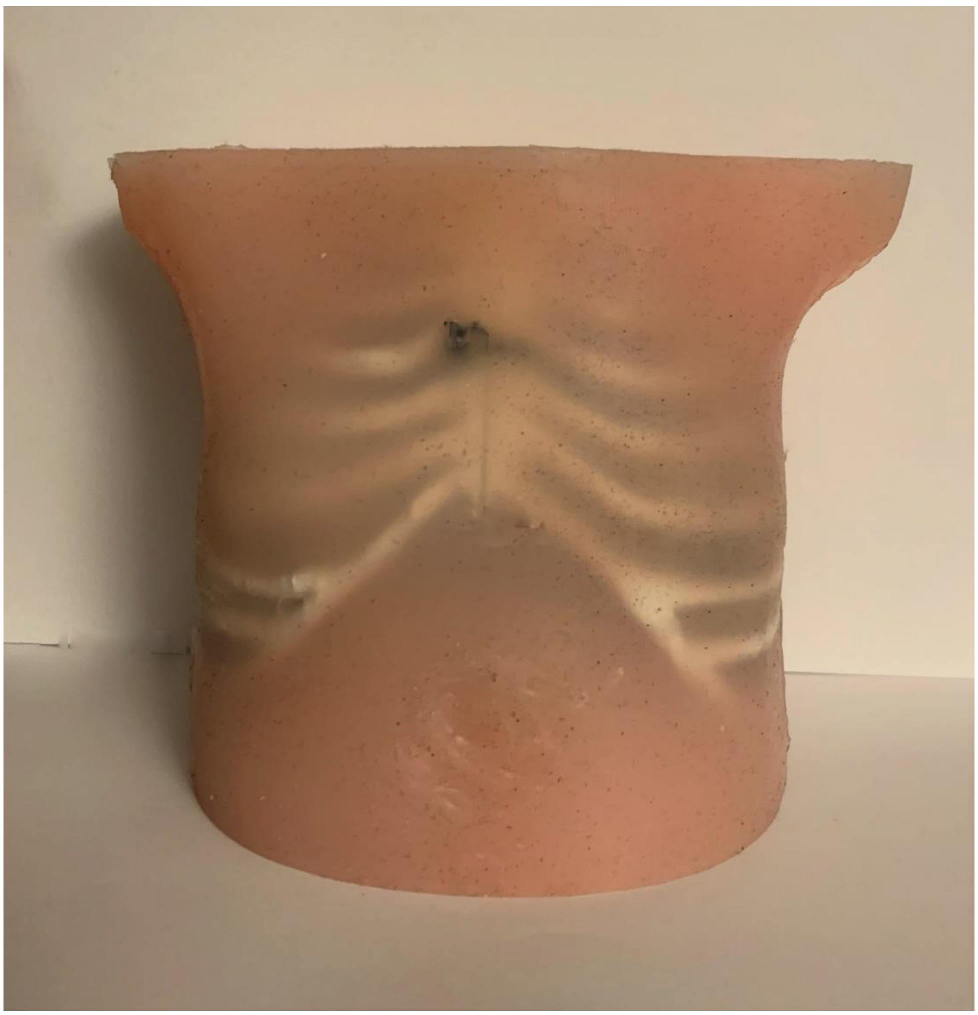


Figure 19. Fabricated neonatal manikin

4. Capability of reusing or replacing for minimal cost

The created neonatal manikin is comprised of a ribcage, heart, and a pair of pleurae, all encased in a soft silicone shaped in the form of a neonate torso (Fig 19). All the internal anatomy is 3D printed, where rigid structures are made with PLA and soft structures including operable organs are printed in thermoplastic polyurethane (TPU). All the organs or bones are printed separately with modified design elements to allow for accurate and easy assembly. Both, NPC and NTC can be performed on the manikin. A detailed fabrication process for the complete manikin is explained below:

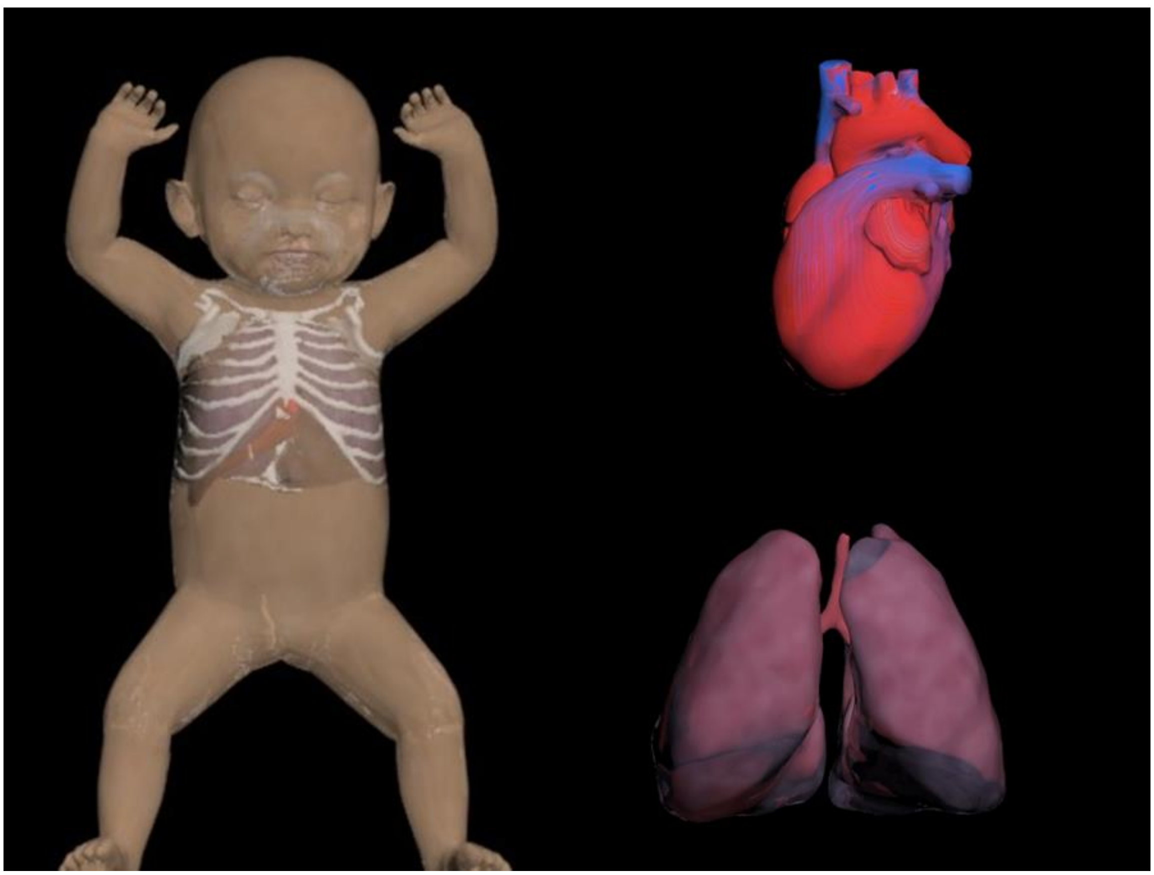


Figure 20. Virtual anatomy

4.1 Virtual Models:

For the purpose of this research, existing 3D virtual anatomy from a previous research (Susan Hayes, MS Thesis, Department of B-vis, UIC, 2017) was used. The anatomy was auto segmented from multiple neonatal CT images and then combined and processed manually in Zbrush and Autodesk 3DsMax. The anatomy included neonatal skin, ribcage, spine, pleurae, lungs, heart, and liver. However, the liver was discarded since it was not associated with the surgical procedures in interest.

4.2 Processing virtual anatomy

The 3D virtual anatomy created from the patient images is in the form of a 3D mesh. A mesh model is a set of polygons connected to each other representing the topography of the surface. The original 3D anatomy was in 3Ds format, which is a native format for Autodesk 3DsMax (3DsM). It is to be noted that the mesh objects do not have a solid volume. Hence, there is no thickness associated to these models; they are merely surface representations. Additionally, software such as 3DsM are used to create detailed 3D objects with organic structures, which is for a visual purpose. However, Computer-aided design (CAD) software such as Dassault System Solidworks (SW), modifies the mesh to a solid 3D object with an internal volume. CAD software are purposed to create engineered objects such as machine parts, tools and complicated assemblies. CAD software efficiently handles mesh files with fewer polygons and structures of mechanical nature, since it allows for complicated volume modifications with features such as shell, extrude, loft etc. CAD software also provide the advantage of modifying designs. Since the goal was to 3D print the virtual anatomy, it was necessary to print the organs with features that could be used to assemble the anatomy post printing. Additionally, due to the complexity of the organs, it was necessary to create support structures that could assist during 3D printing. To create precise engineered supports and modifications to the anatomy, SW was found to be optimum. Hence, all the anatomical meshes were exported from 3DsM in OBJ format and imported into SW. Since SW requires objects with small number of polygons, all of the anatomy was first decimated in 3DsM. The decimation process reduces the number of polygons on the mesh while keeping the geometric structure intact, the degree of decimation determines the topographic details that are lost. Most of the anatomy could not be decimated without losing vital geometric information; thus, the mesh models were cut to smaller objects, so that each model could have sufficient detail.

4.3 Processing operable organs:

Most of the available surgical trainers use internal anatomy that is cast in silicone, which provides a tactile experience similar to human tissue. However, silicone is damaged over multiple uses and is required to be replaced or recast. The other surgical trainers use items such as liquid filled balloons, plastic bags, gloves to replicated human tissue. Some of these provide a realistic tactile experience; however, most of them need to be replaced after a single use. Furthermore, for both NPC and NTC the pericardial space and the pleural space either need to be filled with air or liquid. It is challenging to create hollow silicone structures strong enough to maintain their form without collapsing. Hence, the decision was made to 3D print the operable organs. Operable here is in the context of creating organs that would allow for realistic needle insertion which was common to both NPC and NTC.

To print hollow organs, the mesh objects needed to be modified. It is possible to subtract volumes in SW to create shell like structures. This was done by cloning the object of interest in 3DsM and reducing the scale of the object to specific value that needs to be determined based on the thickness of the shell needed. The cloned object is subtracted out of the original object to create a thin surface. This fundamental process was used to create operable anatomy for this research (Fig 21).

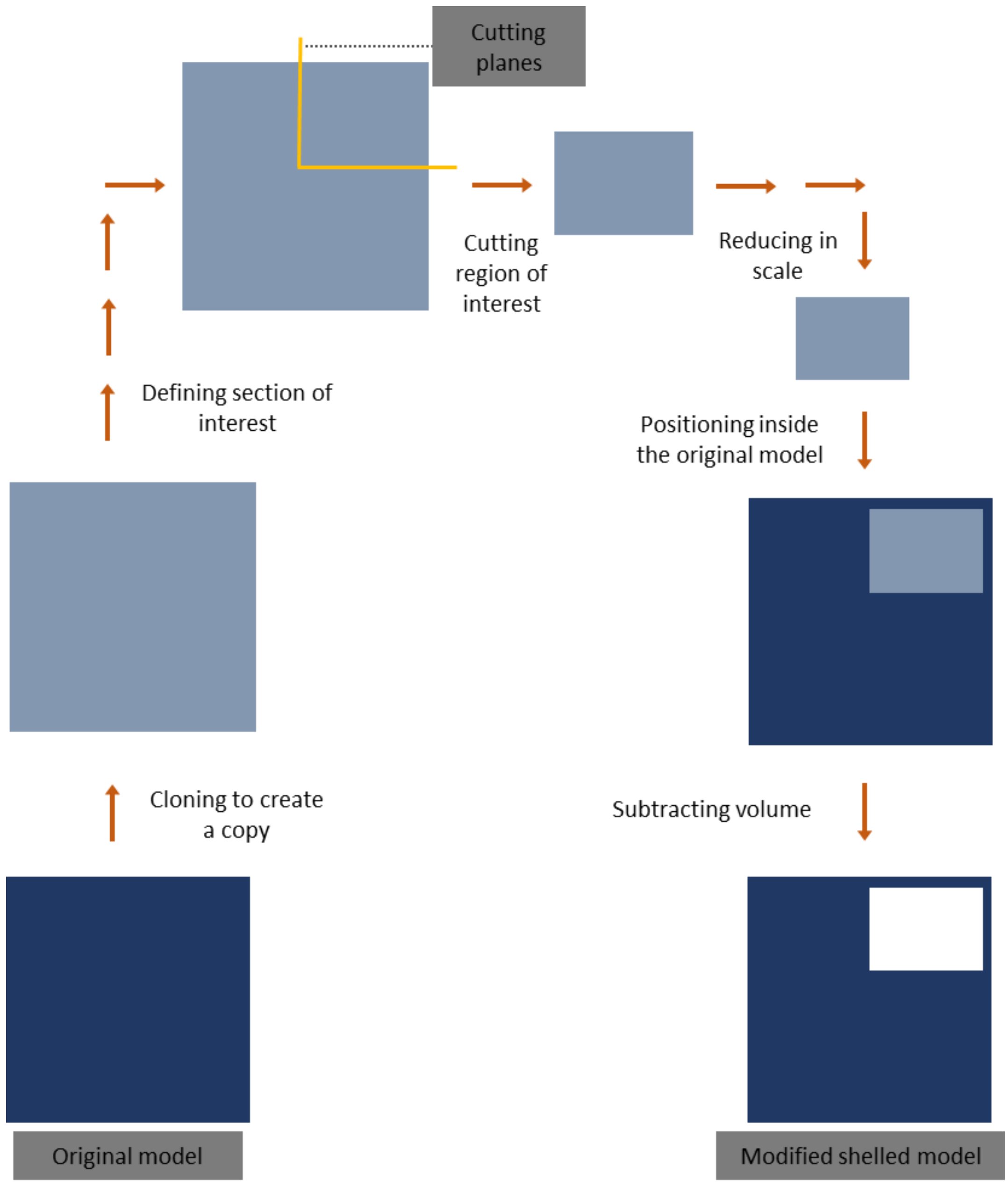


Figure 21. Process of designing operable anatomy

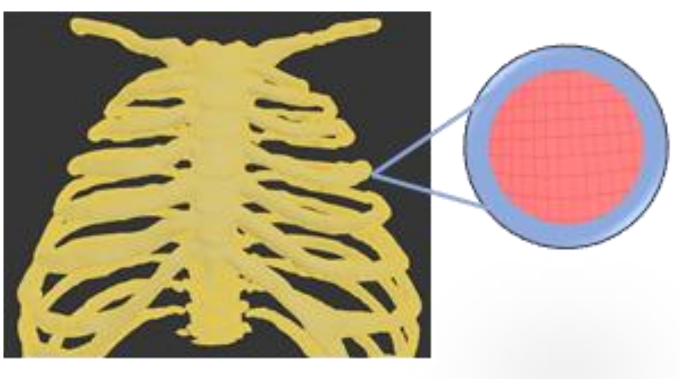
4.4 3D printing of anatomy

The basic processing explained above was used to print all of the internal anatomy. The step-by-step process for each individual part is given below:

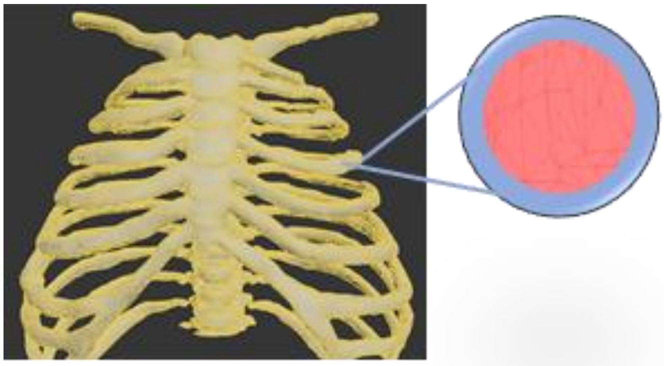
4.4.1 Ribcage and Spine:

The ribcage and spine were created as a single mesh when segmented from the scans. The combined mesh was detailed with large number of polygons, due to which the file had a large over-head. Initially the polygons were reduced using automatic decimation tools available in Fusion360. However, the polygons could not be reduced drastically without causing unwanted deformations and loss of detail. To further reduce the mesh size, the mesh was split vertically using a cutting plane and saved separately.

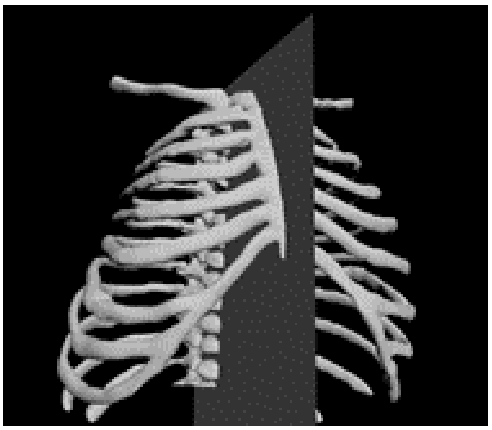
The separate meshes were imported in to Solidworks and additional features were added so that the ribcage-spine could be placed flat on a surface and the two halves could be assembled. The process is illustrated below.



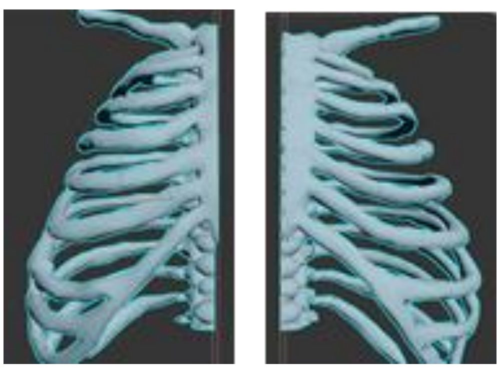
The ribcage and spine were created as a single mesh when segmented from the scans. The combined mesh was detailed with large number of polygons, due to which the file had a large over-head.



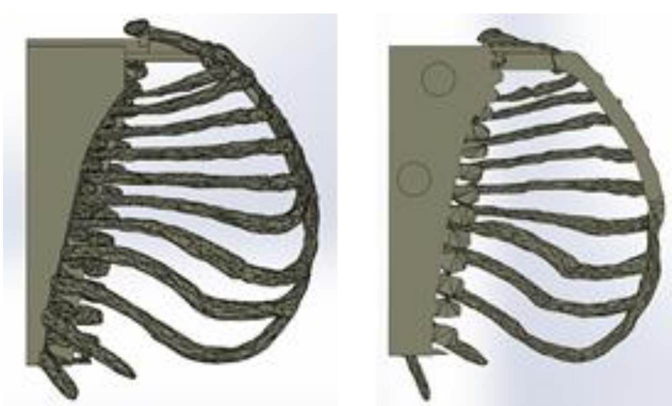
The polygons were reduced by 80% using automatic decimation tools available in Fusion360.



The polygons could not be reduced further without introducing unwanted deformations and loss of detail. To further reduce the mesh size, the mesh was split vertically using a cutting plane.



The two halves created required less memory while retaining the important features.

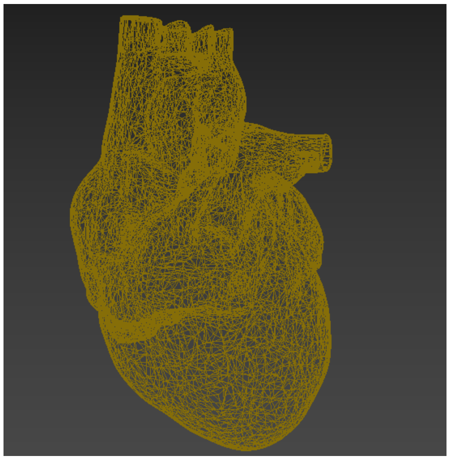


The meshes were imported in to Solidworks and a base was designed, that could support the structure while printing and could assist in lying flat when placed on a surface. Additionally, grooves were added so that two the two halves could be joined, and the pleura could be placed within.

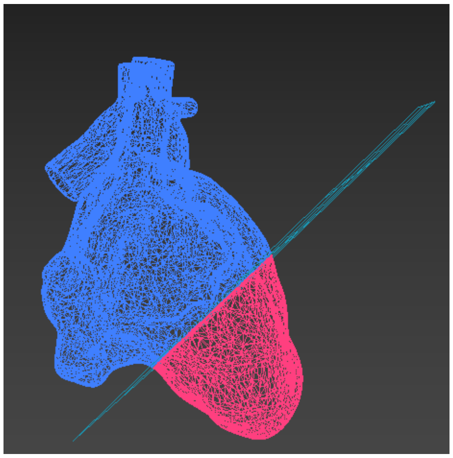
The two halves were 3D printed in PLA with a nozzle size of 0.2 mm. small structures with large over hangs such as the left clavicle, broke off during the printing process, however the majority of the anatomy was printed accurately.

Figure 22. Process for 3D printing a ribcage

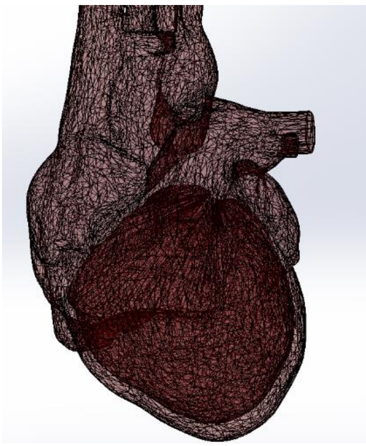
4.4.2 Heart:



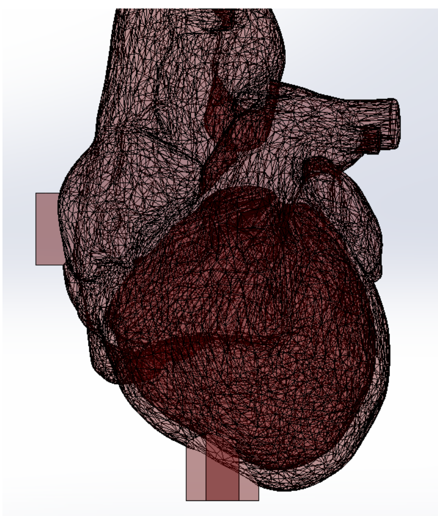
Similar to the ribcage the heart was first decimated to 3% of the original polygonal count.



A portion of the heart, resembling the pericardium was cut-off using cutting planes



After modifying the scale of the heart, the section part was volumetrically removed from the heart, creating a hollow volume within the heart in Solidworks.

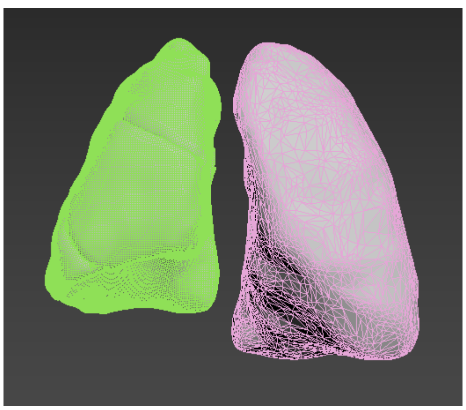


Simple peg was extruded so that the heart could be attached between the two lungs when printed. Additionally, an entry way was created at the bottom of the heart so that fluid can be filled into the pericardium chamber.

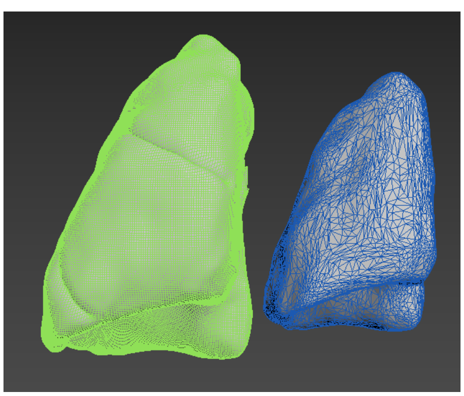
The whole heart was 3D printed in TPU at a nozzle size of 0.1mm with a wall thickness of 1mm.

Figure 23. Process for 3D printing a heart

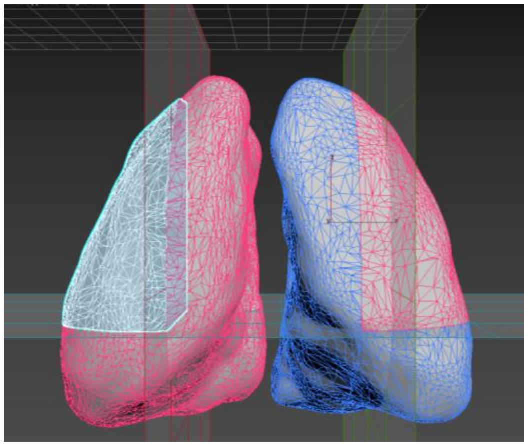
4.4.3 Pleura and Lungs:



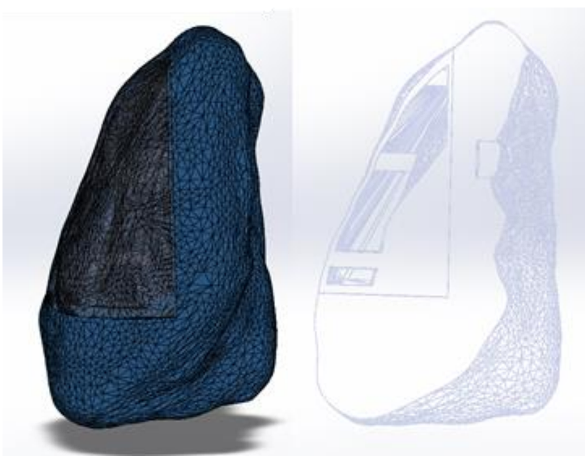
The pleura and lungs were decimated in Fusion 360.



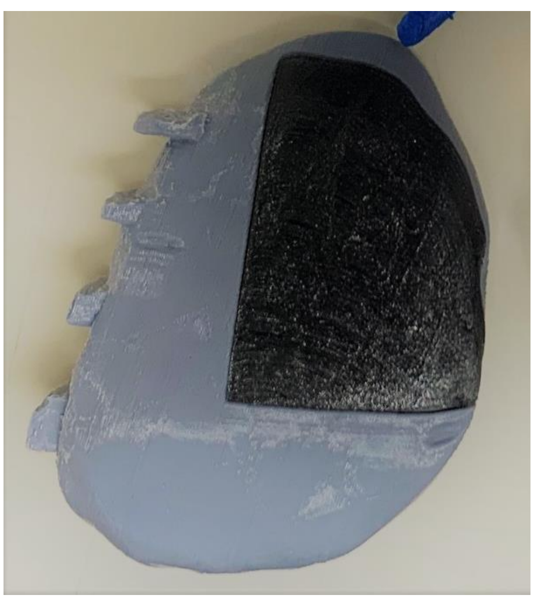
The anatomical meshes of the lungs were segmented of scans from a healthy neonate; hence the topography of the lungs was edited to induce collapsed lungs. This was achieved by cloning the lung mesh and then using Freeform tools in 3DsMax to edit the topography of the mesh.



The lungs are sectioned into two parts using a cutting place.



The modified lungs and the hollow pleura were combined to create a single model in Solidworks

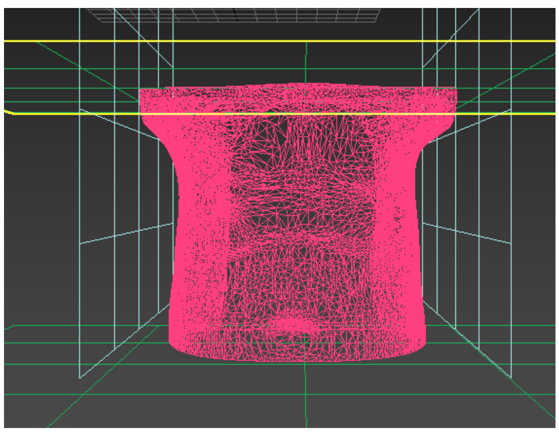


At this point the organs could be 3D printed in TPU; however, our prototypes failed because the walls were too thin. Instead, a small portion, where the needle would be inserted, was printed in TPU, while the rest was printed in PLA. The TPU portion was printed with a nozzle size of 0.1 while the rest was printed with a nozzle od 0.2mm.

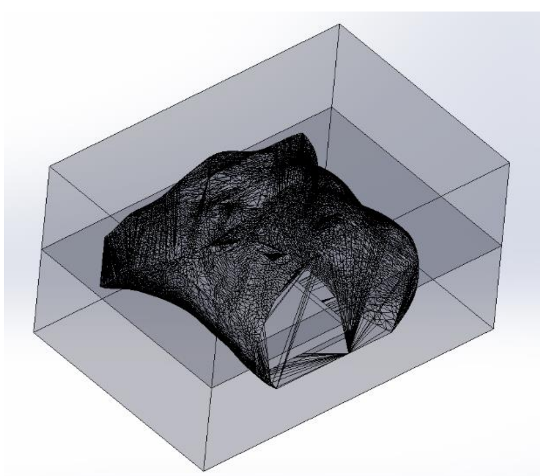
Figure 24. Process for 3D printing a combined pleura and lungs

4.4.4 Mold:

The mesh was first decimated to 25% of its original size.



The original mesh of the skin was cropped to keep the region of interest.



A volume of the skin was subtracted from a rectangular block in Solidworks

The block was then bisected to obtain the two halves of the mold. The halves were processed for efficient printing, accommodating the internal organs and a robust feature for assembly

The mold was printed in PLA at a nozzle size of 0.5 mm.

Figure 25. Process for 3D printing the mold

4.5 Casting the manikin

4.5.1 Material used for casting

Silicone was finalized as the material that would be used to cast the manikin because of it several advantages over ballistic gel, gelatin, agar and polyvinyl chloride (PVC). Silicone can be safely cast at room temperature, only required mixing a two-part solution, has a long shelf-life, is non-reactive and safe to handle. The neonatal manikin was cast in Ecoflex 30 by SmoothOn. On testing however, it was found that the cast was too dense, hence a silicone thinner was used to reduce the viscosity of the silicone.

4.5.2 Silicone density testing

A small experiment was conducted, where a range of samples with different densities of silicone were blind tested for needle insertion by an expert neonatologist (Fig 26). The neonatologist selected the sample which had 15% thinner by weight, which closely resembled the tactile experience of a neonate.

Figure 26. Silicone density test.

4.5.3 Calculating amount of silicone needed:

An approximate amount of silicone required for casting was calculated to reduce wastage. The difference between the volume of the internal anatomy including the ribcage and the torso gave the approximate amount of silicone volume require for the casting. The properties of silicone were used to calculate the amount of silicone required by weight.

The volume of each was obtained from SW.

$$Volume\ of\ Skin\ (Vs) = 877.46281\ ml$$

$$Volume\ of\ Heart\ (Vh) = 12.368.67\ ml$$

$$Volume\ of\ Left\ Ribcage\ (Vrl) = 55.16485\ ml$$

$$Volume\ of\ Right\ Ribcage\ (Vrr) = 54.95105\ ml$$

$$Volume\ of\ Right\ soft\ pleura\ (Vspr) = 8.17152\ ml$$

$$Volume\ of\ Left\ soft\ pleura\ (Vspl) = 11.65085\ ml$$

$$Volume\ of\ Right\ Hard\ Pleura\ (Vhpr) = 76.00282\ ml$$

$$Volume\ of\ Left\ Hard\ Pleura\ (Vhpl) = 80.61596\ ml$$

The volume of silicone was calculated:

$$\Delta volume = Vs - Vh - Vrl - Vrr - Vspr - Vspl - Vhpr - Vhpl$$

$$Total\ volume = 578\ ml$$

$$Silicone\ shrinkage = 1.6\ \%$$

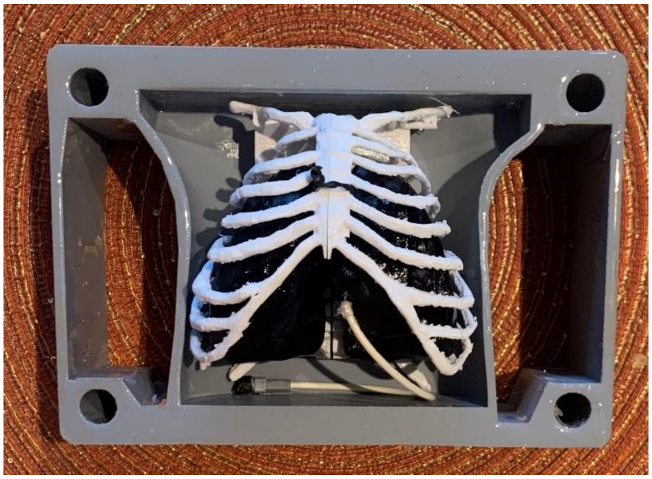
$$Minor\ leakages\ within\ mold = 3\ \%$$
$$volume\ of\ silicone\ (Vsil) = 600\ ml$$

Calculating silicone by weight:

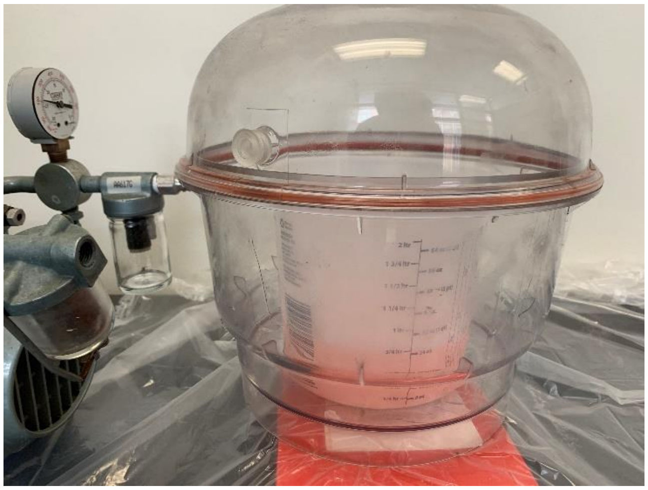
$$Silicone\ Specific\ gravity\ \rho = 1.07\ g/\ cc$$
$$Weight\ of\ silicone\ Wsil = 1.07 \times 600 \sim 645$$
$$Silicone\ thinner = 15\%\ of\ total\ mix$$

$$Weight\ of\ thinner\ Wt\ =\ 645 \times 0.15 \sim 97\ gms$$
$$Wsil\ per\ part\ = \frac{550}{2} \sim 275\ gms$$

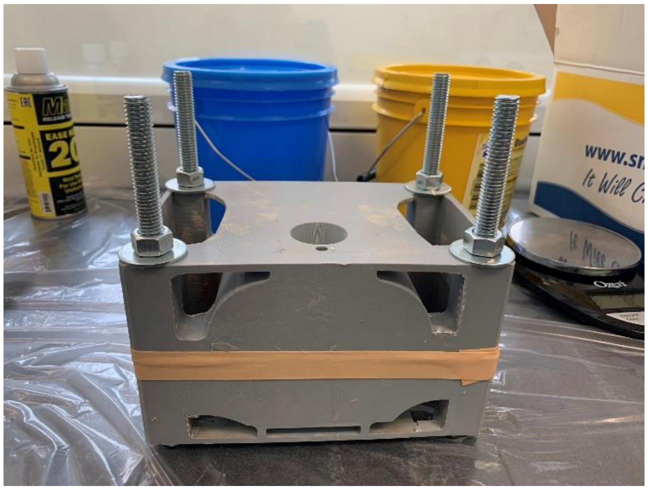
4.5.4 Casting process



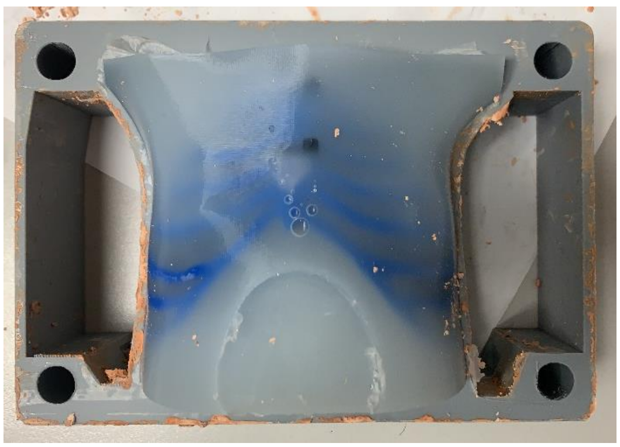
Internal anatomy assembled and placed inside mold



Silicone mixture prepared and vacuum pumped for removing air bubbles



Mold prepared and sealed for silicone pouring



Manikin pulled out of the mold

Figure 27. Process of casting manikin in silicone

4.6 Chapter Summary

1. The manikin is constructed from 3D virtual models of the skin, pleura, lungs, heart, ribcage, and spine which were segmented from real patient scans
2. The 3D meshes of each anatomical object were sliced and then decimated to be imported to SW for modifications
3. Operable organs were constructed by designing objects with hollow volumes for allowing needle insertion.
4. The internal anatomy was cast in commercial grade silicone

Chapter 5 - Software development

5.1 Software modalities

The basic software functionality involves displaying, saving and analyzing surgical steps performed on the physical manikin. The goal of the Surgical Simulation Software (SSS) is to provide an easy way to simulate surgical procedures for training and assessment with minimum involvement of the expert surgeons. To accomplish that, the software runs 3 basic modalities which are discussed below:

5.1.1 Animation and Tracing

The function of this module is to play an interactive surgical performance like an animation (Fig 28). A user can play any existing performance that was saved to file. Additionally, a trainee can trace the surgical actions using the real-time EM tracking of instruments. The user interface displays control buttons that can be used to navigate the animation and start/stop the tracing of actions. Additionally, the user can pan, rotate and zoom to explore the virtual environment.

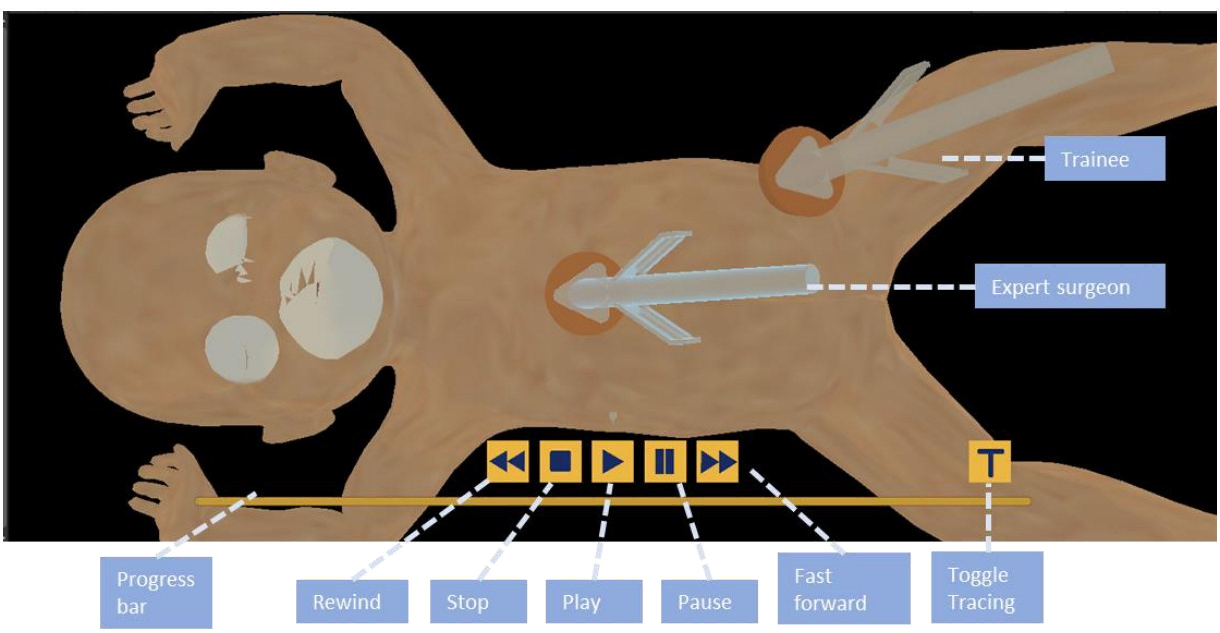


Figure 28. Animation player controls in SSS

5.1.2 Real-time guidance

This module provides a simplified guidance in real-time during the surgical procedure. A small window of the surgical action is collected, evaluated, and displayed on the indicator (Fig 29). Green, yellow and red colors indicate the degree of correctness of the surgical task. The evaluation is conducted through a neural network.

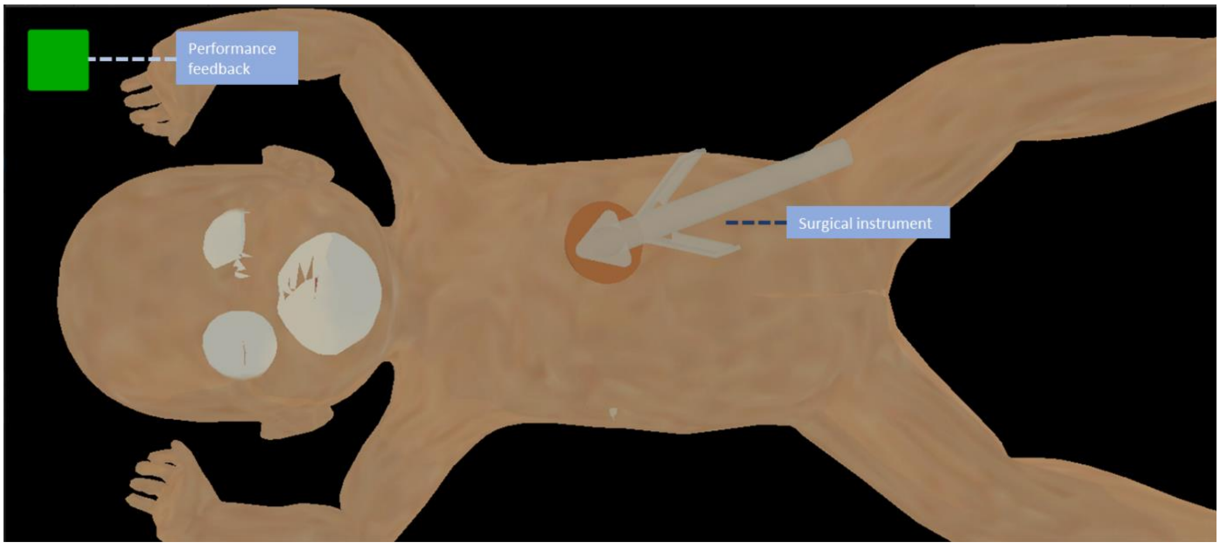


Figure 29. Real-time feedback during training

5.1.3 Performance Assessment

This module provides a detailed skill assessment at the end of the surgical procedure. The score is displayed in a table detailing the quality of performance for each surgical task within the surgical procedure (Fig 30). In this module, the performance data of the user and the assessment scores are stored to file for future reference.

Figure 30. Performance assessment

5.2 Software features

The modules described above are a combination of multiple features of SSS; which are given below:

1. Display relevant virtual anatomy
2. Display real-time motion of instruments
3. Save user information
4. Save surgical performance data (motion of instruments)
5. Load a previous surgical performance
6. Animate a loaded surgical performance
7. Evaluate surgical data for quality of performance
8. Display feedback information

These features are built with a variety of tools, which are listed below:

1. Unity software
2. SQLite database
3. Python TensorFlow (TF)
4. Python ZeroMQ API

The application was developed in Unity, which is a popular game engine that has been used to develop industrial games for multiple operating systems, such as Windows, Mac OS, iOS, Android, etc. This game engine is a collection of powerful libraries and a user-friendly interface that provides the capability to rapidly create interactive applications. The following are the features of the software:

5.2.1 Display virtual anatomy and motion tracking

The virtual anatomy segmented from neonate images are used for NTC and NPC. Similarly, during manikin fabrication, the original mesh models are decimated before importing to the application to avoid large memory over-head. Once imported, the anatomical models are calibrated for position and orientation by moving the models to coincide with physical model.

Similar to the anatomy, motion of instruments is displayed through virtual models of the instruments. These models are designed by replicating the dimensions of actual surgical instruments. Detailed drawing of the CAD models of the instruments for both the surgical procedures are given at the end of this chapter. The instruments are tracked through the EM tracker, introduced in Chapter 3. The tracking is performed at a sampling rate of 50 Hz, matching the frequency of the in-built call-back function “FixedUpdate()” in Unity. The position and the orientation of the models is updated in “FixedUpdate()”. The sampling rate is adequate to produce smooth animations and simulate detailed surgical action performed.

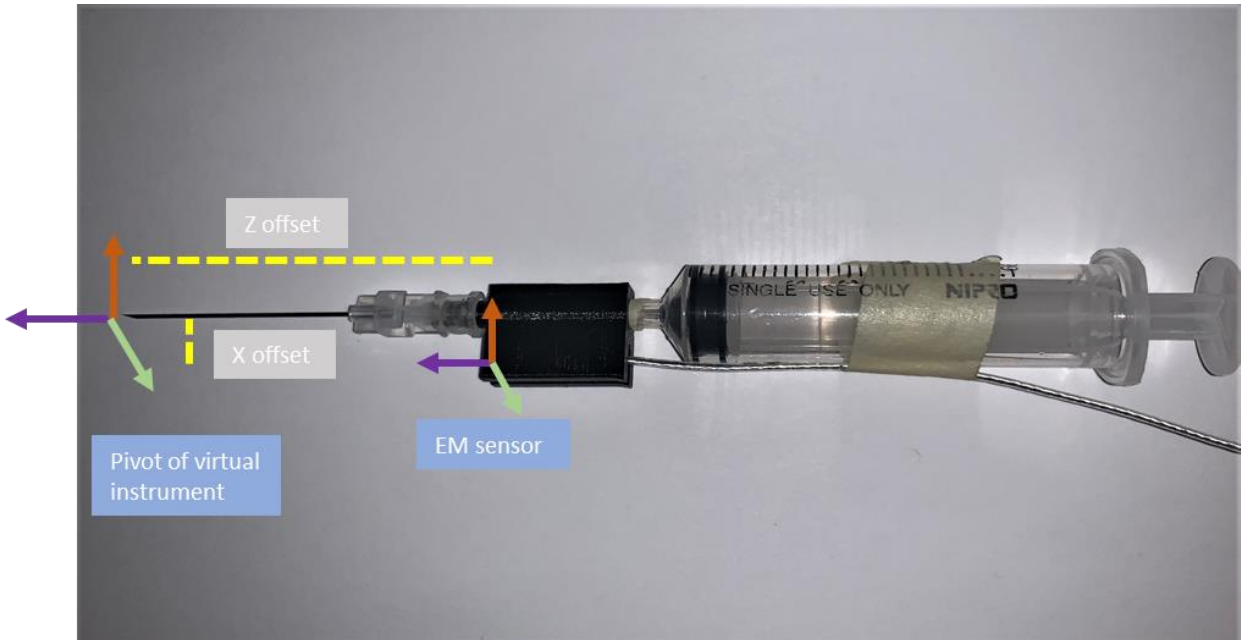


Figure 31. Calculating the offset values

The tracking data generated by the EM system is the calculated based on the 3D position and orientation of the sensor in reference to the EM transmitter. The point of attachment of the sensor to the surgical instrument can be considered as the pivot point. Similarly, for virtual models any spatial transformation is done through the pivot point on the model. It is important that the pivot point of the virtual model is in the same relative location to the physical instrument to accurately replicate the motion. If the pivots do not coincide, then a 3D offset value must be defined, so the tracker can compensate for the difference (Fig 31). Consequently, for each instrument, based on the place of attachment of the sensor the offset values are calculated and stored in the application.

5.2.2 Saving surgical performance

The EM tracker sends 6 values per sensor, 3 for the position and 3 Euler angles for orientation. A single sensor sends 300 values every second for a sampling rate of 50 Hz. This eventually utilizes large amount of storage memory. To reduce the memory requirement, two methods are utilized; a threshold filter to store only relevant information and a compression algorithm to store efficiently.

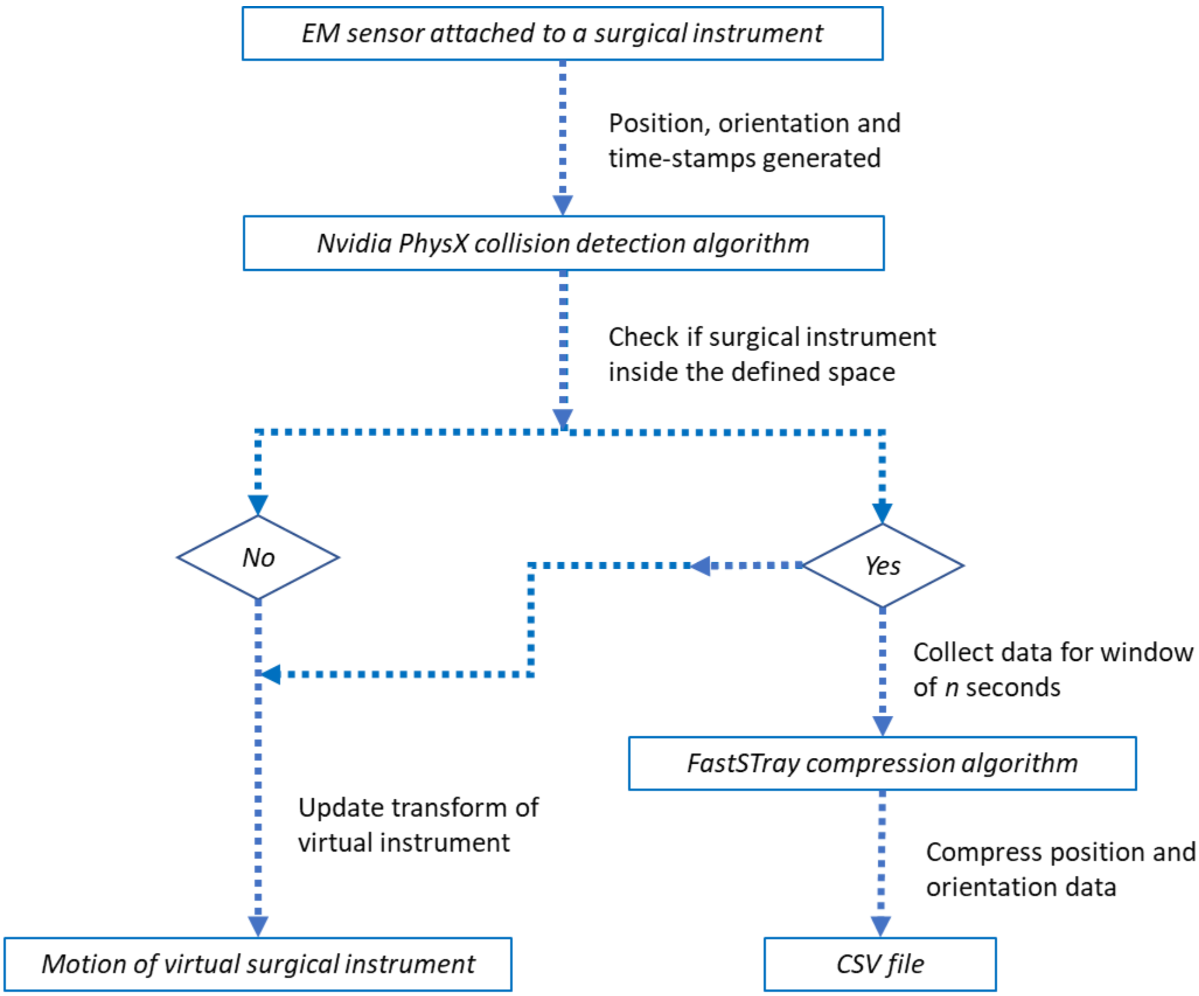


Figure 32. Process for storing tracking data

Unity has the built-in Nvidia PhysX engine, which is a physics simulation library. The collision detection in PhysX can be used to inform when two virtual objects touch each other or when one object is within the other. This feature is utilized to flag storage of spatial information during the surgical performance. A virtual box collider is used to define the surgical space within which all actions are assumed to be related to the surgical task being performed (Fig 34). Box or capsule colliders are attached to each virtual instrument. When the instruments enter this surgical space, the data storage is initialized.

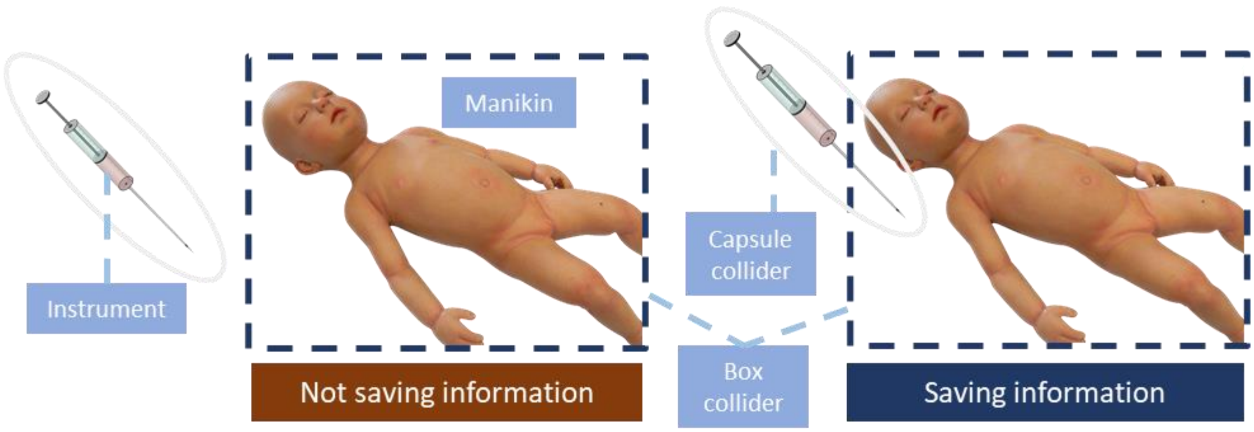


Figure 34. Collision detection

Compression algorithms are mainly divided into two parts: (1) offline, where the complete trajectory is required before compressing, and (2) online, where the compression can be done on a portion of the trajectory. Offline compression comparatively has better compression rates since all data can be processed at the same time. However, since the time required to complete the surgical task is unknow, storing all performance data in the application memory could create a large overhead in memory that could potentially crash the application. An alternative would be to first store the motion data to a file and process with a compression algorithm after the procedure is completed; however, the time taken for compressing all data at once would be long, which is

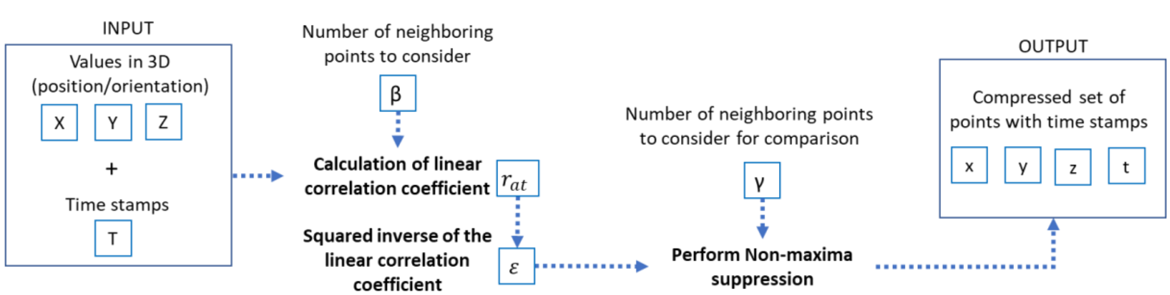


Figure 33. Process flow for FastSTray

undesirable. As a result, the software instead uses short collection windows, where the information

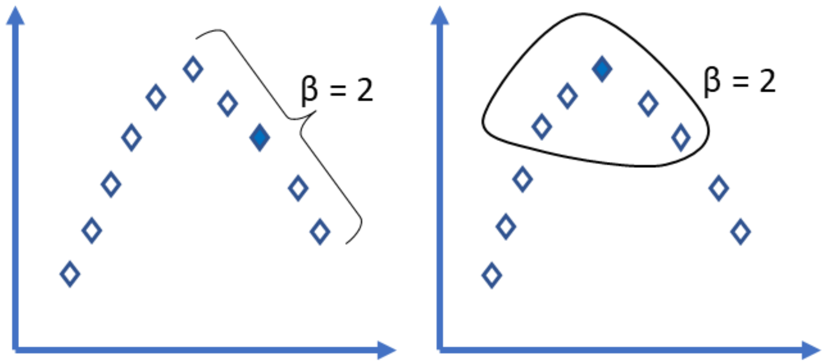


*Figure 35. Linear correlation between 2β neighbors. **Left:** High linear correlation hence this point will be eliminated **Right:** Low linear correlation, this point will be stored.*

is stored, immediately compressed, and saved to a file.

FastSTray algorithm is specifically designed for compressing 3D data of robotic surgical systems (Marino & Manic, 2016) (Fig 34). For each point on the trajectory, FastSTray calculates a coefficient that quantifies the relevance of the point, also known as an information coefficient (Fig 35). Further, applying non-maxima suppression on all the coefficients eliminates points with low coefficient values. The remaining points can be used to represent the trajectory. The two parameters β and γ in FastSTray determine the degree of compression and accuracy. 2β is the number of neighbors selected to calculate the information coefficient of a point, while 2γ is the number of neighbors to which non maxima suppression is applied. FastSTray can achieve up to 65 -70% compression while maintaining errors between 0.78-2 mm. The window for collecting the data for compression can be independently set for each sensor, along with the possibility to set the rate of compression through β and γ.

$$r_{at}(\{a\},\{t\}) = \frac{(a(i) - \bar{a})(t(i) - \bar{t})}{\sqrt{\Sigma_i (a(i) - \bar{a})^2}\sqrt{\Sigma_i (t(i) - \bar{t})^2}}$$

$$\varepsilon(P,t) = \frac{1}{(r_{at}(P.x,t))^2} + \frac{1}{(r_{at}(P.y,t))^2} + \frac{1}{(r_{at}(P.z,t))^2}$$

5.2.3 Saving user information:

Databases are used to store information in tabular form, using standard database management systems (DBMS) for a secured method of storing and querying information. Between the range of available DBMSs, SQ-Lite is small, fast, and self-contained database engine that runs on SQL. Furthermore, Unity already has an in-built SDK for SQ-lite.

A custom database was designed with tables that store user information, surgical trials by each user and assessment scores for each trial (Fig 36). The *UserInfo* table manages information pertaining to the user by storing a unique Id, user type (resident, fellow or surgeon) and years of experience. The *PerformanceInfo* table is responsible for storing all the trials conducted on the simulator, which includes the information of the surgical procedure of the trial and the name of the file containing the motion data. Finally, the *AssessmentResults* table stores the average scores received for each surgical task during the assessment phase of the training.

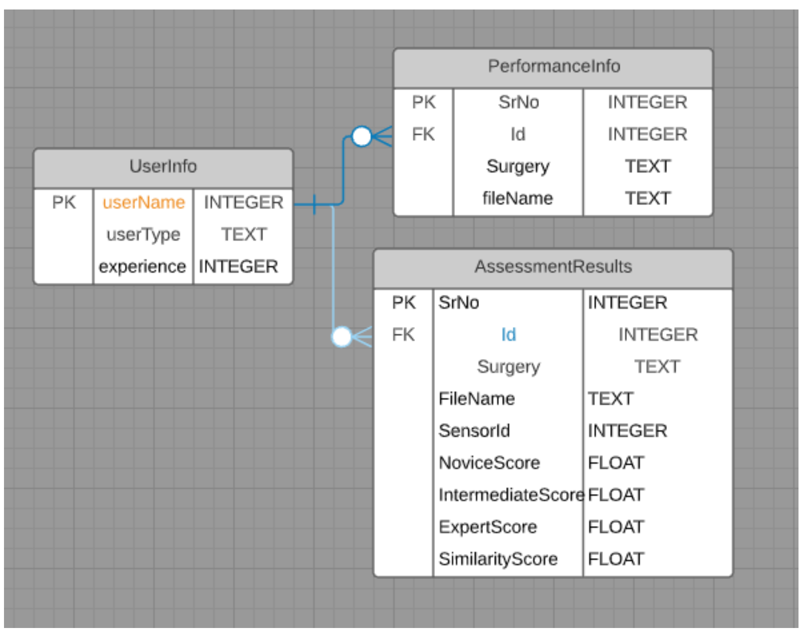


Figure 36. Database entity-relationship diagram

Tables within the database are linked to the other through one of more variables called foreign keys as shown in the entity-relationship diagram (ERD), for example the field *userName* from the *UserInfo* table is linked to the *Id* in the *PerformanceInfo* table, these relational variables make it possible to query specific information in short amount of time and safeguard accidental deletion of information. However, design of tables or selection of foreign keys can cause unnecessary cloning, storing spatial data particularly creates large amount of repetitive data such as cloning of foreign keys on every row of data. Instead, the spatial data is stored in a standard comma-separated values (CSV) format on a local PC. A unique naming convention is used to identify the CSV file, in case the performance data must be loaded into the application.

5.2.4 Animating previous surgical procedures

Spatial data stored in the files can be loaded into the software to playback surgical actions that were performed previously. First, the database is queried by the software to get the name of the file containing the data. The file is opened, and the data is loaded into the application memory. Restored data is in compressed format and is required to be decompressed before playback. Complete decompression is memory intensive, instead decompression and animation are performed simultaneously.

The missing values between two consecutive time stamps is calculated using linear interpolation for positional values and using spherical interpolation for orientation values. At every frame the position and orientation of the virtual model is updated based on the calculated α value. The animation timer (t) with a maximum time value (T) of the last data entry in the surgical performance, is updated every frame based on the time elapsed (Δt) since the last frame. For any two consecutive time stamps $(t_n \ and \ t_{n+1})$, the α is updated and used to find the missing motion data based on the latest time value t, of the animation timer. The callback function *FixedUpdate()*,

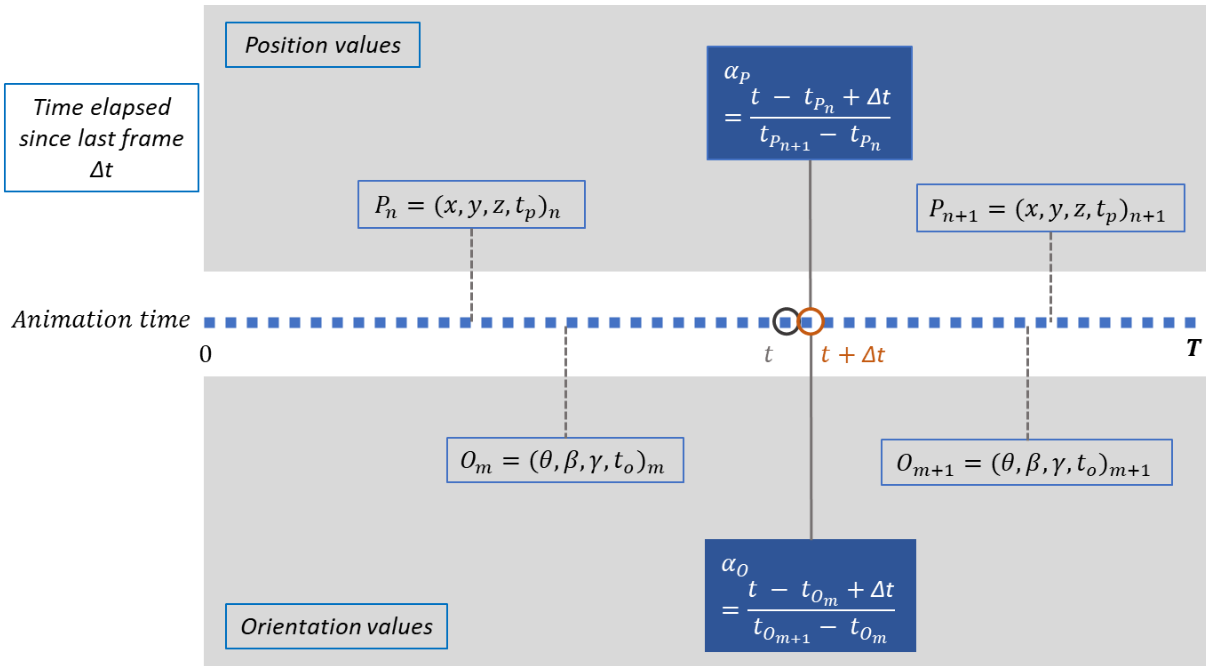


Figure 37. Calculating α values

us used to update the values since this call-back function runs at a known frame rate of 50 Hz. The linear and spherical interpolation are performed using the inbuilt Unity functions *Lerp()* and *Slerp()* respectively.

5.2.5 Evaluating a surgical performance

The evaluation of a surgical performance is performed through a neural network that has been trained using collected surgical performance data. Existing Unity API, called ML-agents, has the capability to train and load per-trained networks. However, at the time, the ML-agents API was in its development phase and the lack of clear documentation and unreliability in using some networks such as LSTMs made it an unviable option. Instead, the network is trained in Python using TensorFlow and is run in as Python application. Since both the Unity application and the Python application run independently, a communication between both was required. Consequently, ZeroMQ messaging library was used to manage the exchange of information between the Unity application and the Python application.

ZeroMQ is high level library for asynchronous messaging for distributed and concurrent applications. It supports the common messaging patterns and message ques without the use of a message broker unlike most of the middleware messaging libraries.

A request-reply message pattern is used in SSS, where the Unity application is the client, and the Python application is the server. In a continuous loop, the client sends a message, and receives a reply from the server. Every frame the updated position and orientation value from the EM tracking system is sent and collected by the python application. Sending data every frame creates a sufficient over-head, which is avoided by using multi-threading. The thread is responsible for sending the data and receiving a response. Once sufficient data is collected the data is processed, and then the neural network evaluates the data. The evaluation results are sent back to the Unity application.

5.2.6 Designing virtual instruments:

In order to accurately display the motion of surgical instruments in the virtual environment, 3D virtual models resembling the physical instruments in size and appearance were needed. There were two existing methods to create virtual models from the existing physical instruments:

1. Scanning the instrument using a 3D scanner
2. Manually measuring the dimensions and creating a 3D model using a computer aided design (CAD) software.

The accurate visual representation of the virtual instruments depended on the degree of accuracy with which the physical instruments were converted to virtual ones. Additionally, sophisticated 3D scanners were inaccessible for regular use. Hence, manual method was deemed suitable.

Sr No.	Properties	Scanning	Manual method
1.	Time required	Short, since the scans are automatically converted to a 3D object.	Long, since each dimension must be measured and replicated in a CAD software.
2.	Availability	Can be done on the phone using a specialized software, however if accuracy is important, special scanning equipment is required.	Easily done through any CAD software.
3.	Accuracy of dimensions	Low, since it's difficult to stitch 2D images to a single 3D object.	High, since the measurements are taken manually.

Table 1. Comparing scanning to manual method of converting physical surgical instruments to 3D virtual model.

The instruments were designed in Solidworks based on measurements taken using a vernier caliper and a measuring scale. The following series of images (Fig 38-43) display the instrument dimensions and their 3D renders.

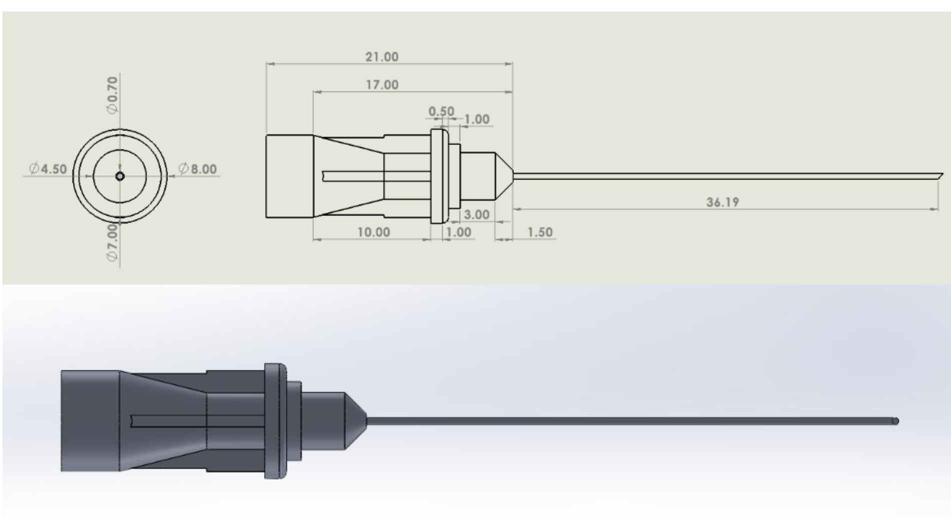


Figure 38. 22G needle for pericardiocentesis

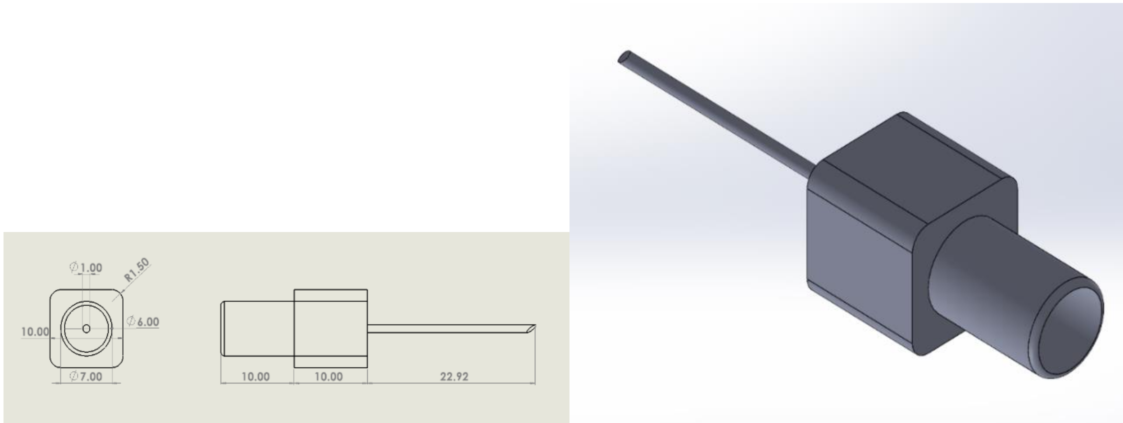


Figure 40. 25G needle for anesthesia

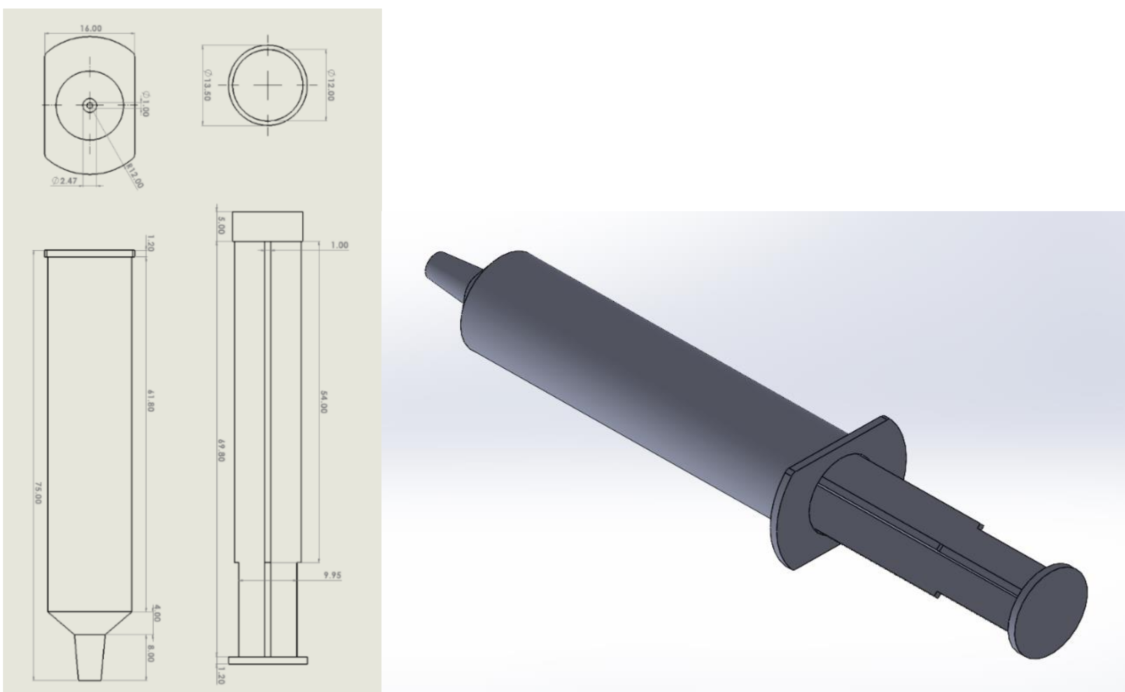


Figure 39. 10cc syringe for pericardiocentesis

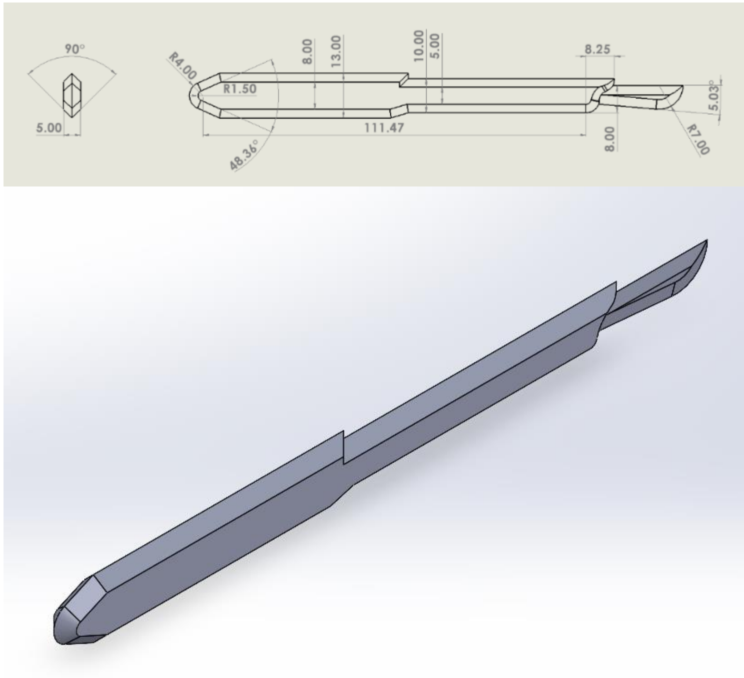


Figure 42. Scalpel used for thoracentesis

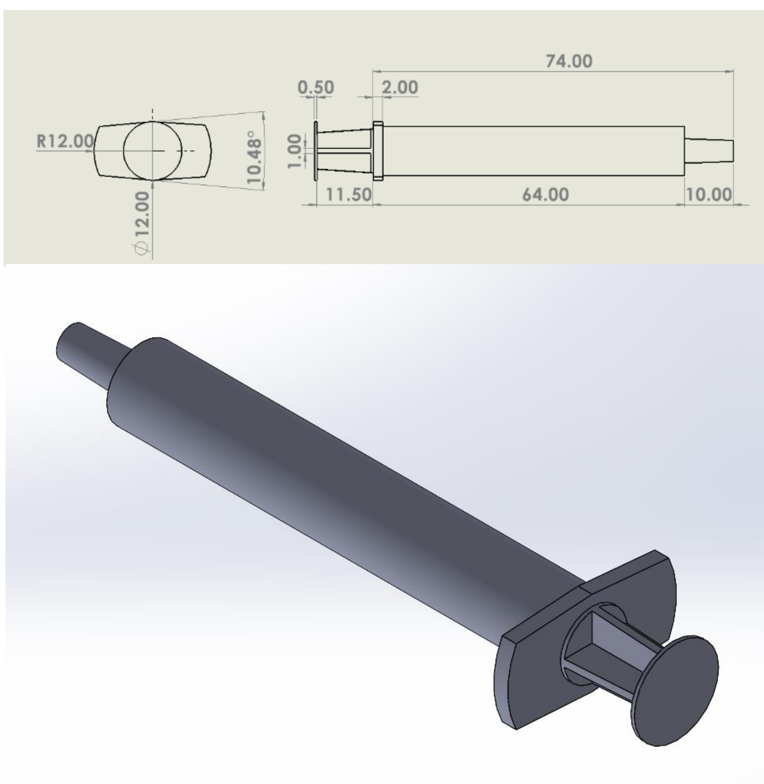


Figure 41. 10cc luer-lock syringe for anesthesia

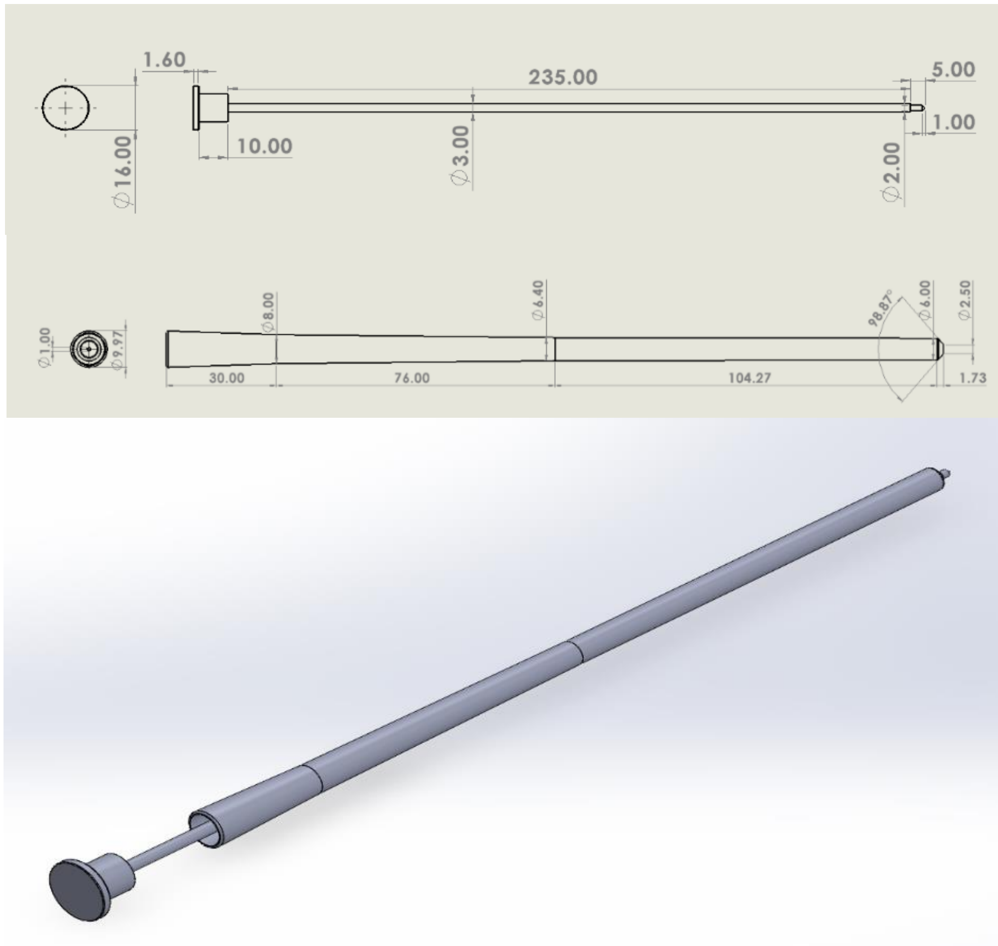


Figure 43. 8Fr/Ch trocar catheter for thoracentesis

5.3 Chapter Summary

1. The software has three modes namely, animation and mimicking, training under guidance and assessment of surgical skills.
2. The application is built in Unity C#, where it can display virtual anatomy and the real-time motion of surgical instruments
3. The virtual instruments were designed in Solidworks by measuring the dimensions of actual surgical instruments.

4. A SQL-lite database is used to store user information, history of surgical trials and the assessment results
5. The motion data is threshold filtered and then compressed by the FastSTray algorithm and saved to the PC in CSV format, simultaneously the name of the file is saved to the database.
6. The ZeroMQ library is used to communicate between the unity application and the python application.

Chapter 6 - Automatic Assessment

There has been a rise in research of automated methods of surgical skill evaluation, due to the required time involvement of expert surgeons and the objectivity of traditional methods of assessment (Castillo-Segura, Fernández-Panadero, Alario-Hoyos, Muñoz-Merino, & Delgado Kloos, 2021). Deep neural networks have had a large impact on autonomizing solutions in a myriad of fields. However, the performance of the deep neural networks is data centric, as they are dependent on large amount of data to learn autonomous behavior. Hence, its particularly difficult to apply deep neural networks for surgical skills since the data collected is comparatively very small, especially for long surgical tasks. However, there has been recent research like Wang & Majewicz Fey (2018), Fawaz, et al. (2018), Ismail Fawaz et al. (2019) that have been successful in using deep neural networks with reliable accuracy to classify surgical skills.

Most of the existing methods classify the skill levels into two to three classes based on skill levels, mainly novice, intermediate and expert. However, the data used for these methods is manually annotated through a lengthy process of assessment through standardized scoring methods such as OSATS (Objective Structured Assessment of Surgical Skills) (Niitsu et al., 2013), GRS (Global Rating Scale) (Reznick, Regehr et al. 1997), GEARS (Global Evaluation Assessment of Robotic Skills) (Goh, Goldfarb et al. 2012). Additionally, in the scenario of adding another class, the networks would need to be retrained and it would be challenging if the data would be insufficient to train the network. Instead of classification, an alternative approach of using comparison as a method of skill evaluation is used. Siamese networks are adept at learning the features that are different between the input classes, instead of learning how to identify individual classes (Chicco D., 2021). Siamese networks compared to classification networks requires less data to train. Published research (Hou, Jin, & Zhao, 2019) comparing time series data using

Siamese Convolution Neural Networks (SCNN) obtained reliable results. They were successful at showing that Siamese networks are a viable option for comparing sequential data with different lengths as compared to the traditional methods such as Dynamic Time Warping (DTW). Consequently, the goal was to employ an online Siamese convolution neural network that could identify expert surgical skill using data collected from simulation of neonatal surgical procedures.

6.1 Siamese Convolution Neural Network architecture

Siamese neural networks usually have two or more identical subnetworks. These networks share the same architecture and parameters, including weights. Unlike traditional classification networks Siamese learns the similarity between the two inputs.

The proposed SCNN has in total of 17 layers. The first 12 layers is a CNN with convolution layers followed by max-pooling and dropout layers (Fig 44). The convolution filter increases in size by a factor of 2, from 38 to 308, each layer with a constant kernel size of 2. The output from

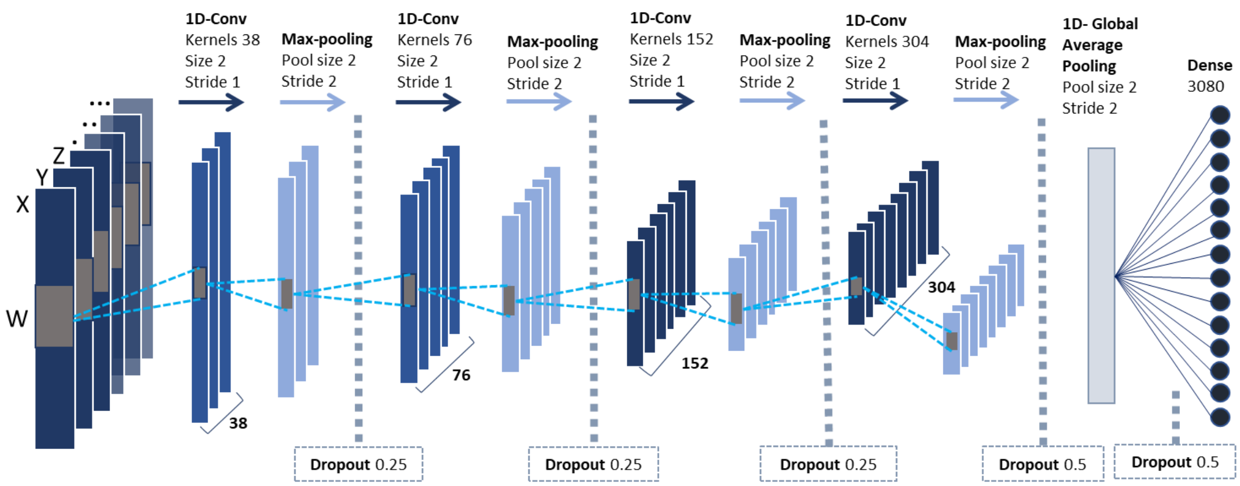


Figure 44. CNN architecture for Siamese

the convolution layers is passed through a Rectified Linear Unit (ReLu) activation before passing to a max-pooling layer with a pool size and a step size of 2. To avoid over-fitting to the surgical data each max-pooling layer is followed by a drop-out layer with a value of 0.25. Additional regularization is achieved through global average pooling (GAP). The GAP layer is followed by a dropout layer with a high dropout value of 0.5 and followed by a dense layer of 3,080 neurons.

The two inputs passed through the layers up-until the dense layer, subsequently the dense layer of the one input is subtracted from the dense layer of the other input to give a single value representing the similarity between the pair. The network uses an Adam optimizer and the binary cross entropy to calculate the loss.

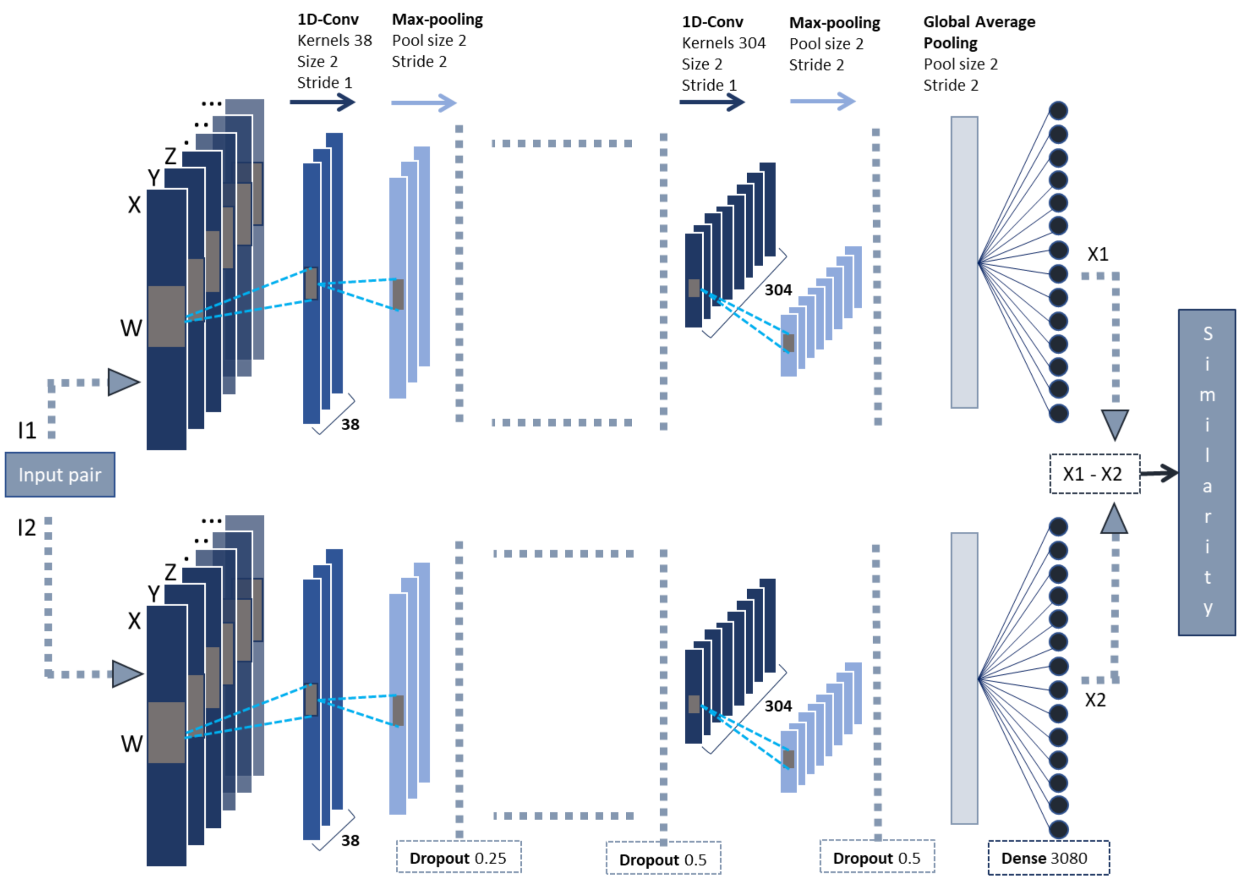


Figure45. Architecture of the SCNN

6.2 Data Collection Trials:

A simulator platform that can record and playback surgical kinematic data collected through electromagnetic (EM) tracking system was developed. It tracks the relative position and orientation of the sensors attached to the surgical instruments within a defined magnetic field created by the transmitter. It replicates the physical actions performed with surgical instruments in a virtual environment from the information received through the EM sensors. The data collected is stored in a CSV format locally on a PC. It can accurately playback the stored information as an animation; hence any discrepancies in annotating the performance could be corrected by revisiting the trials. The developed platform was used to simulate two neonatal surgical procedures, specifically, thoracentesis (THC) and pericardiocentesis (PCC).

The trails were conducted in the Neonatal Intensive Care Unit (NICU) at the University of Illinois Hospital (UI Health) with residents, fellows, and expert neonatologists. The simulation platform developed was used to collect the surgical data. The compression algorithm and the threshold filter were switched off to avoid removing any vital data. During the trials, the VR display was used to demonstrate the surgical procedure by the expert but was switched off during the data collection to avoid any distractions.

6.2.1 Protocol

The sessions were conducted like traditional SBT. The protocol used is given below:

1. Instructors gives a detailed description of the symptoms displayed by a neonate and the need for performing the procedure.
2. Instructor demonstrates the correct technique to perform the procedure.
3. The subject practices the procedure under guidance on the simulator till comfortable.
4. The subject performs the procedure under no guidance. In this phase, the data is collected.

6.2.2 Surgical tasks

Both surgical procedures were broken down to set number of surgical tasks, with a predefined sequence and technique to perform. The defined sequence and task list was as follows:

NPC:

1. ChloraPrep – Preparing the surgical area using a disinfecting applicator. Apply at the center of the surgical area and gradually move outwards in a spiral.
2. Needle Insertion – Inserting the needle 0.5 cm below the xiphoid process, slightly left of the midline at an angle of 30-40 deg to the skin while pointing toward the left shoulder. Confirm insertion by extracting liquid.

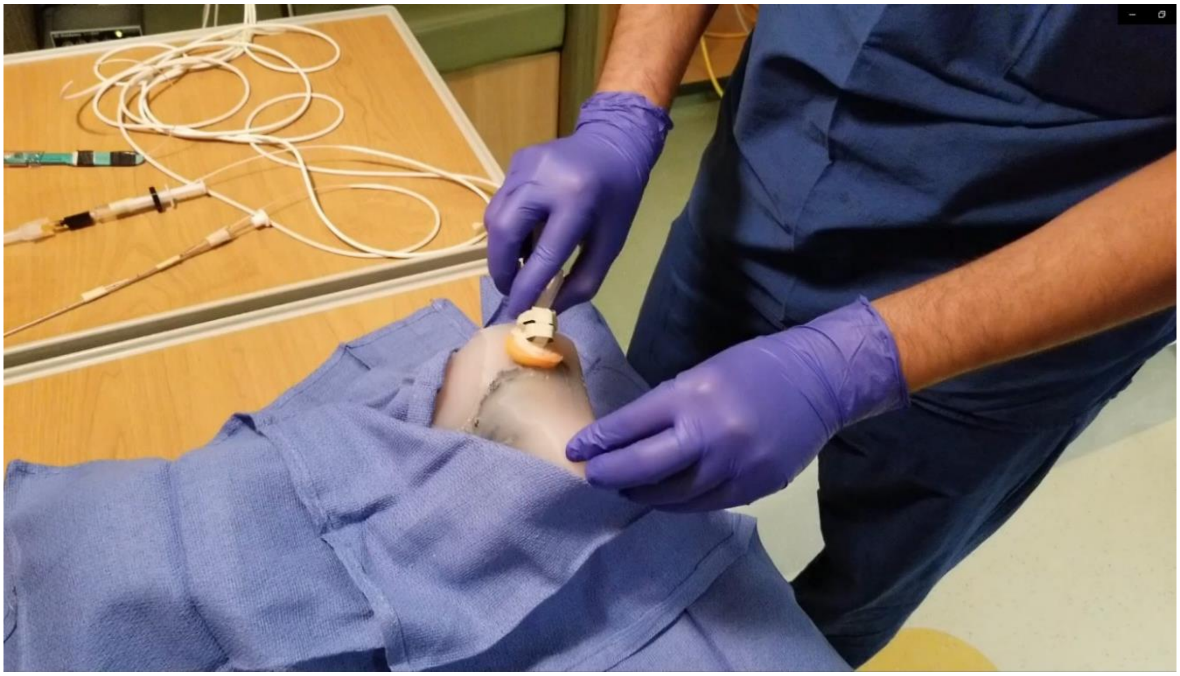


Figure 46. Surgeon preparing the surgical area during thoracentesis

NTC:

1. ChloraPrep - Same as explained above.
2. Anesthetization- Infiltrate the skin and simulate administration of anesthesia around the surgical area.
3. Incision - Create a small incision with a scalpel the layer of skin and muscle at the target insertion point.

4. Trocar Catheter Insertion - Enter 3-5 cm into the pleural space directing the tip anteriorly and superomedial.

6.2.3 Manual annotation

The data was captures at a frequency of 50 Hz. A total of 20 subjects took part in the trial. In total 46 samples were collected for PCC, out of which 3 samples were discarded due to high noise and incomplete performance. Similarly, 29 samples were collected for THC out of which 2 samples had some surgical tasks missing.

Traditionally, surgical skills are assessed using metrics such as OSATS, that provide an objective score for the performance. The score can be used to determine if the performance was satisfactory or not. To annotate the data, some of the categories form the OSATS scoring system were employed, specifically, *respect for tissue*, *time and motion*, *instrument handling* and *flow of operations*. From the 20 points that could be accumulated in total, we considered 15 and above as expert and all others as non-expert. The total number of samples for each label based on the surgical tasks are given in the Table 2.

	Non-expert	Expert	Total samples
Pericardiocentesis			
ChloraPrep	14	29	43
Needle Insertion	28	15	43
Thoracentesis			
ChloraPrep	11	18	29
Anesthetization	13	15	28
Scalpel Incision	10	17	27

Trocar Insertion	13	16	29

Table 2. Samples for each label

6.2.4 Data Processing:

The collected data using the EM system was raw motion data, evidently the data contained some unwanted information such as motion unrelated to the surgical tasks. Additionally, to train the network effectively, the input data needs to be processed with specific methods.

6.2.4.1 Extracting features:

The recorded data includes position in cartesian coordinates, orientation in Euler angles and time stamps in seconds. The orientation information was converted to quaternions to avoid the complicated scaling of Euler angles using the relation below.

$$Q = \begin{bmatrix} w \\ qx \\ qy \\ qz \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\phi}{2}\right).\cos\left(\frac{\theta}{2}\right).\cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi}{2}\right).\sin\left(\frac{\theta}{2}\right).\sin\left(\frac{\psi}{2}\right) \\ \sin\left(\frac{\phi}{2}\right).\cos\left(\frac{\theta}{2}\right).\cos\left(\frac{\psi}{2}\right) - \cos\left(\frac{\phi}{2}\right).\sin\left(\frac{\theta}{2}\right).\sin\left(\frac{\psi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right).\sin\left(\frac{\theta}{2}\right).\cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi}{2}\right).\cos\left(\frac{\theta}{2}\right).\sin\left(\frac{\psi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right).\cos\left(\frac{\theta}{2}\right).\sin\left(\frac{\psi}{2}\right) - \sin\left(\frac{\phi}{2}\right).\sin\left(\frac{\theta}{2}\right).\cos\left(\frac{\psi}{2}\right) \end{bmatrix}$$

$$where: \ \psi, \theta, \phi \ are \ angles \ at \ X, Y, Z \ axis \ respectively$$

Position			Orientation				Linear velocity			Angular velocity		
X	Y	Z	W	Qx	Qy	Qz	Vx	Vy	Vz	VQx	VQy	VQz
...	...	...	...	..	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...

Figure 47. Input features for training

To increase the number of data features to learn, tool velocities were added to the existing data. Linear velocities were calculated by differentiating position values against time, using equation given below:

$$v = \frac{\Delta x}{\Delta t}$$

$$where, \ \Delta \mathrm{x} = x_2 - x_1 \qquad \Delta t = t_2 - t_1$$

Angular or rotational velocities were calculated from quaternions and respective time values using the NumPy-Quaternion library in Python. The formula used by the library for calculating the angular velocities is given below:

$$\omega(t) = 2 * \frac{dq(t)}{dt} * conj\big(q(t)\big)$$

The final input features are summarized in Fig 47 below:

6.2.4.2 Threshold filtering:

A defined space was agreed upon by the experts, outside of which all motion data was considered invalid or unrelated to the surgical task. Threshold values were calculated based on the dimension of the cube and its distance from the origin (Eq 2). Data points with locations outside of the threshold value were removed.

	X	Y	Z
Pericardiocentesis	200 mm	200 mm	150 mm
Thoracentesis	200 mm	200 mm	200 mm

Table 3: Size of the active space

$$T_{\theta_{max}} = d + \frac{L_\theta}{2}$$

$$T_{\theta_{min}} = d - \frac{L_\theta}{2} \qquad (2)$$

$where:\ \theta = axis(X, Y\ or\ Z)$

$d = distance\ of\ the\ cube\ from\ the\ origin\ along\ the\ axis\ \theta$

$L = lenght\ of\ the\ cube\ along\ the\ axis\ \theta$

Figure 49. Active surgical space for NPC

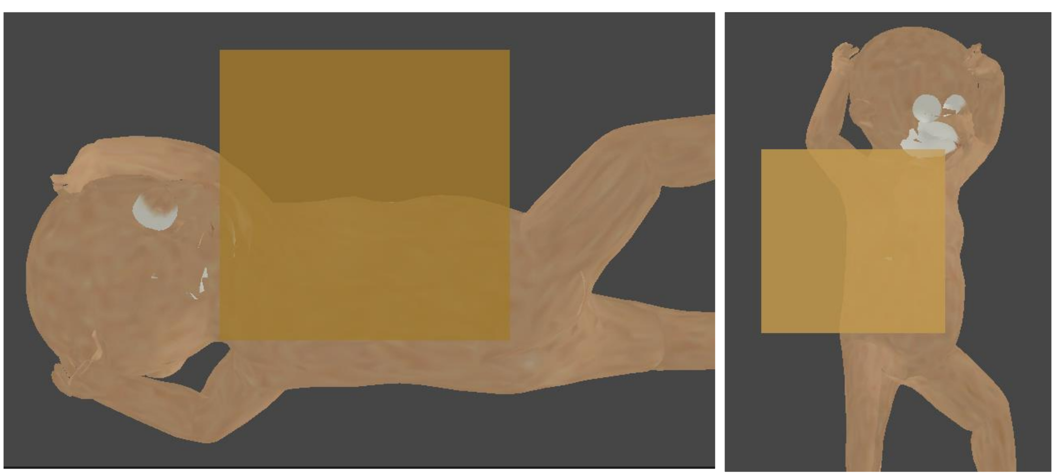


Figure 48. Active surgical space for NTC

6.2.4.3 Normalization:

To avoid biased learning and exploding gradients, feature scaling was done through min-max normalization. All the features were normalized, except the quaternions, since their values were in the range of 0-1. Usually, the max and min values for normalization represent the maximum and minimum value found from all the samples. However, outliers in the data would cause wrong scaling values. Alternatively, the max and min values can be set based on the distribution of the values per features. An example of the variation in distribution in the features for a single surgical task are given in the Fig 49. Using similar distribution plots ideal max and min values for each feature were calculated.

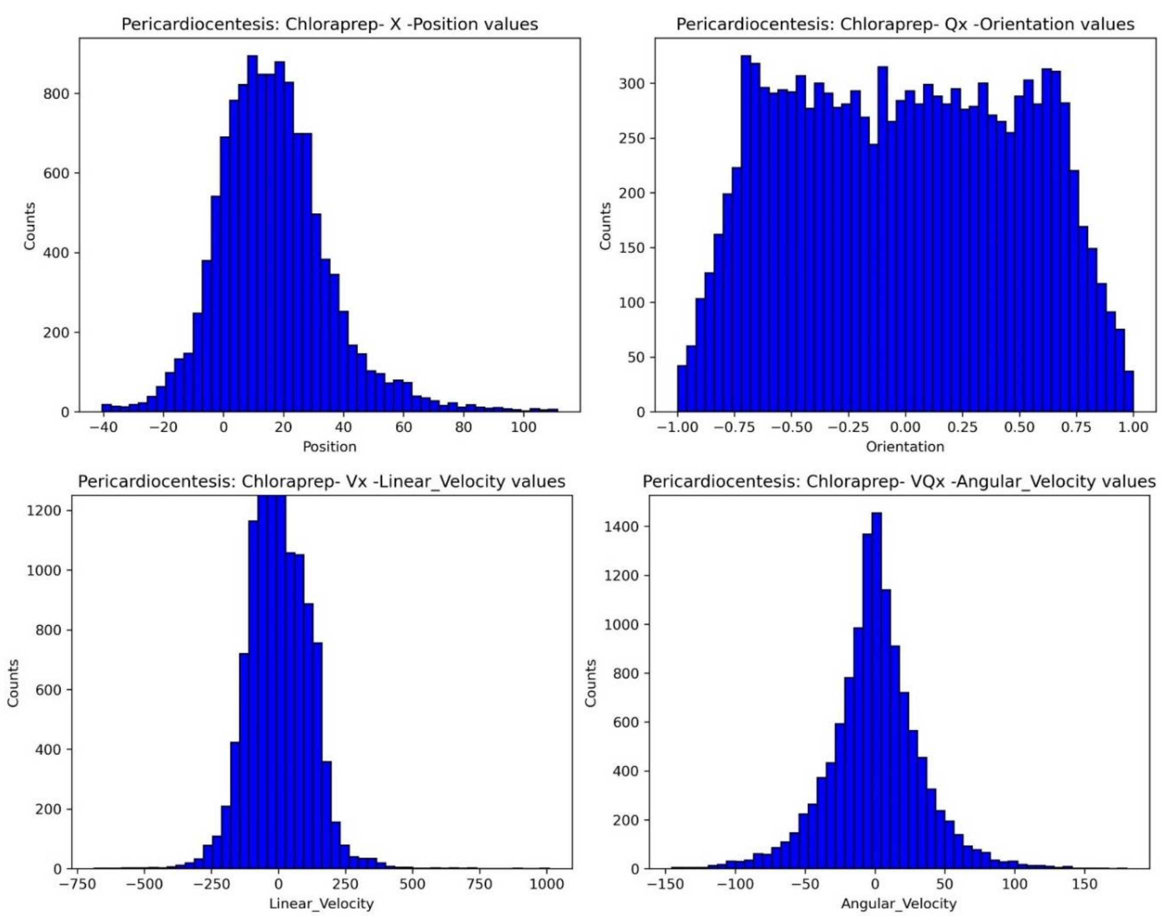


Figure 50. Feature distribution for ChloraPrep

	X	Y	Z	Vx	Vy	Vz	VQx	VQy	VQz
	Min	Min	Min	Min	Min	Min	Min	Min	Min
	Max	Max	Max	Max	Max	Max	Max	Max	Max
Pericardiocentesis									
ChloraPrep	-25	50	-250	-300	-250	-250	-75	-75	-75
	75	150	-160	300	250	250	75	75	75
Needle Insertion	0	30	-235	-100	-90	-70	-30	-25	-30
	50	100	200	80	50	90	30	25	30
Thoracentesis									
ChloraPrep	-20	80	-260	-200	-150	-200	-75	-75	-75
	70	135	-170	200	150	200	75	75	75
Anesthetization	35	132.5	-317	-150	-90	-100	-75	-75	-75
	110	200	-180	150	100	100	75	75	75
Scalpel Incision	30	70	-325	-90	-75	-90	-75	-60	-60
	115	185	-220	90	75	90	75	60	60
Needle Insertion	-35	80	-312	-100	-75	-100	-100	-100	-100
	100	170	-165	100	75	100	100	100	100

Table 4. Max and min thresholds for normalization

6.2.4.4 Augmenting Data:

Similar to previous works (Drumond, Marques, Vasconcelos, & Clua, 2018) (Wang & Majewicz Fey, 2018), we augmented our data by employing a sliding window approach. In this approach, a single sample is sectioned to multiple windows of the same size while preserving the label. The size of the window (W) and the step size (S) determine how many samples would be created from a single sample. The window size also determines the relevant time series data that would be learnt by the neural network. Effectively, the longer length of windows would represent the sample more accurately. The average time-to-complete per surgical task was widely varied as shown in Table 2, and hence the network was trained over a range of *W* to find one with the best performance.

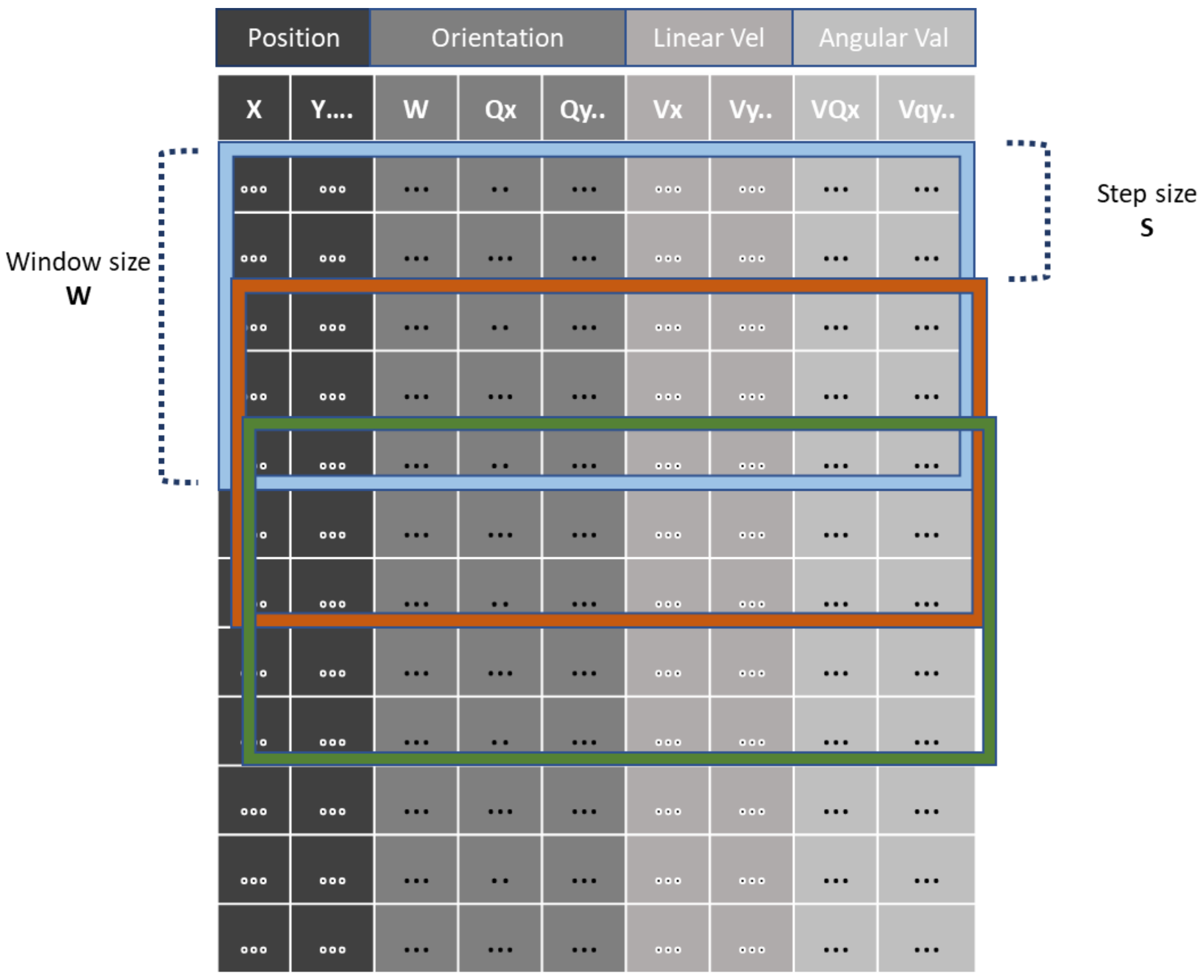


Figure 51. Data augmentation using windows

Pericardiocentesis	Average task time (seconds)	Thoracentesis	Average task time (seconds)
ChloraPrep	282.67	ChloraPrep	413.8929
Needle Insertion	2760.92	Anesthetization	1698.417
		Scalpel Incision	374.1667
		Trocar Insertion	1146.05

Table 5. Average task times for each task

6.2.5 Pair-wise Annotation:

The previously annotated data windows were split into two parts (P1 and P2) based on the labels. P1 containing expert samples while the P2 containing non-expert samples. To train the network to learn the features that differentiate between P1 and P2, N-way-One-shot method was employed. The strategy was to pair a sample only once, one with dissimilar label and one with a similar label. The interest was in learning the differences between experts and non-experts. Hence, each sample of expert in P1 was randomly paired with one sample of P2 and one sample in P1. The P1-P2 pairs were annotated as '0' while the P1-P1 pairs were annotated as '1'. The resulting total number of pairs per surgical task are given in the Table 6.

Pericardiocentesis	Total no. of input pairs	Thoracentesis	Total no. of input pairs
ChloraPrep	526	ChloraPrep	684
Needle Insertion	3338	Anesthetization	818
		Scalpel Incision	3494

Trocar Insertion	1066

Table 6: Total number of input pairs

6.3 Chapter Summary

1. Most of the deep learning models currently use classification models to evaluate surgical skills, instead here a Siamese network is used which is a comparative model.
2. The subnetworks employed for the Siamese network are CNN.
3. A total of 43 samples for NPC and 27 samples for NTC were collected to train the network.
4. The collected data was manually annotated into two groups: experts and non-experts, using an evaluation scheme similar to OSATs.
5. Linear and angular velocity from the position, orientation and time information were extracted in order to add more features for the neural network to learn from.
6. Before training, the network the dataset was processed through a threshold filter and then scaled with normalization.
7. Sliding window approach was used to augment the dataset. It was found a $W = 140$ for NPC and $W = 160$ for NTC with a $S = 10$ was ideal for training.
8. A one-shot-learning approach was used for pair creation for the training data.

Chapter 7 - Results

Overall, three separate validations focused on different aspects of the platform were conducted:

1. Training on the platform without autonomous features
2. Testing of the SCNN
3. Training on the platform

7.1 Validation of platform without assessment

The objective of the trials was to see the effectiveness of the simulator for traditional simulation-based training. The protocol set for the trials was:

1. Instructor discusses the background information regarding the surgical procedure.
2. Instructor demonstrates the procedure on the physical manikin, the physical actions are also displayed on the monitor for reference.
3. The trainee is given an opportunity to perform on the physical manikin under the guidance of the instructor.
4. Once comfortable with the simulator, the trainee performs the procedure on the manikin without any guidance.
5. Post training the trainee fills out a feedback form.

It is to be noted that the instructors were also asked to perform the procedure on the manikin and then provide their feedback in the form. Individual training sessions were held for both NTC and NPC at the Neonatal Intensive Care Unit (NICU) at the University of Illinois at Chicago (UIC) Hospital. Doctors from the NICU participating in the trials were a mix of residents, fellows, and expert surgeons, with varying years of experience.

7.1.1 Training for NTC

A formal training session for NTC was conducted for a group of 5 fellows and 3 residents by 3 expert neonatologists. The following surgical steps were conducted for NTC:

a. Palpating the ribcage to find the target insertion point.
b. Preparing the chest using ChloraPrep,
c. Infiltrating insertion area with anesthetic drug,
d. Perforating insertion point with a scalpel and
e. Inserting the chest-tube with trocar into the pleural space.

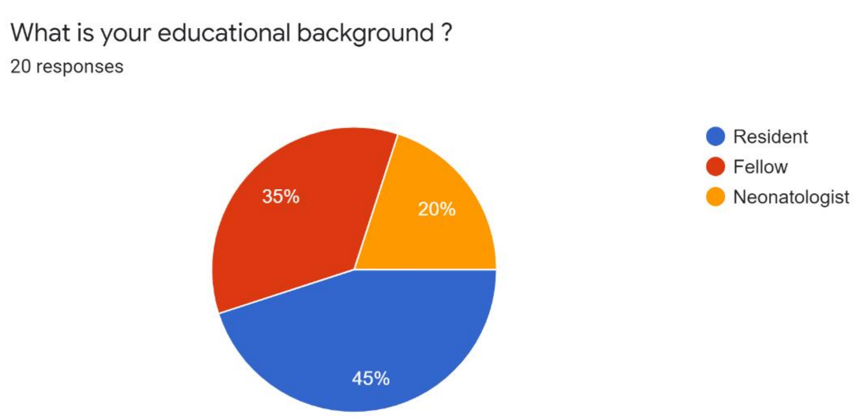
7.1.2 Training for NPC

Similarly, a training session for emergency pericardiocentesis was conducted for a group of 7 fellows and 9 residents by 4 expert neonatologists. The following surgical steps were performed on the manikin:

a. Preparing the surgical area using ChloraPrep and
b. inserting the 22G needle into the pericardium

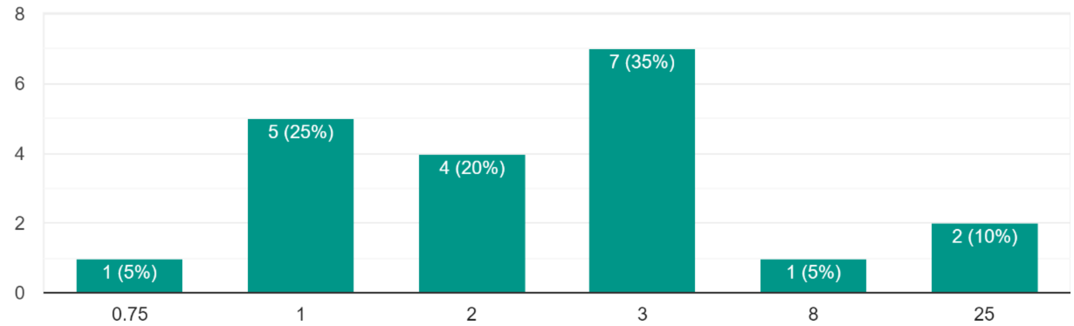
7.1.3 Results

The following results summarize the feedback received from the participants:



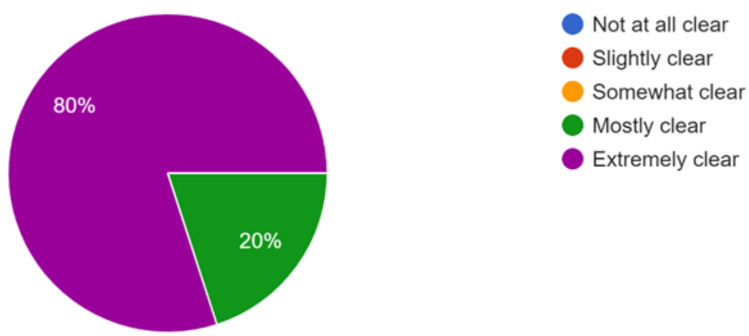
How many years have you been a resident/ fellow/ neonatologist/ surgeon ?

20 responses



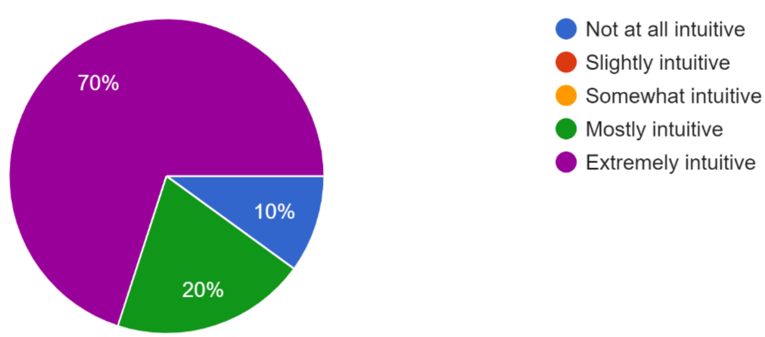
How clear was the instructional guidance to perform the surgical task ?

20 responses



How intuitive was the surgical simulator ?

20 responses



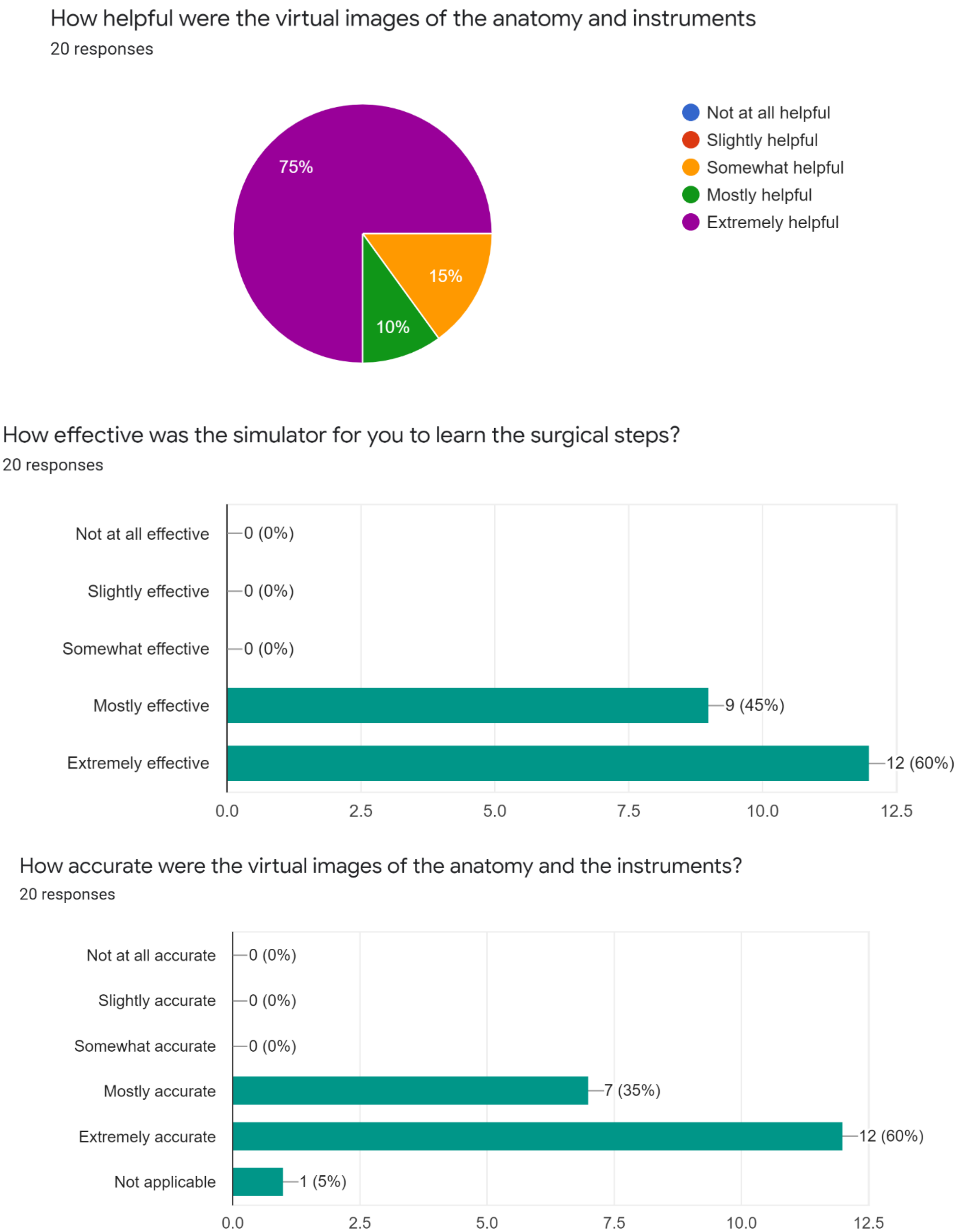


Figure 52. Feedback for training on platform without assessment

7.2 Validation of the neural network

7.2.1 Training SCNN

The neural network training was conducted for 200 epochs for a batch size of 20 at a learning rate of 0.001, where 20% of the input pairs were randomly separated for validation during the training. The seed for the random function was set to 11. We used the hold-out method to test the robustness of our network post training; a randomly selected expert and non-expert performance was omitted from the training. We trained the model in Keras using TensorFlow 2.5 on a Windows PC equipped with a dual GeForce GTX 1070 Ti graphic cards each with 8 GB memory.

Upon training the network for different window sizes, it was found that the SCNN performed better for larger window sizes. Though in Chapter 6 it was shown that the average length of the surgical task times was highly varied, it was found that the SCNN generalized best when $W = 140$ for NPC and NTC.

	Validation accuracy for window size (%)					
	60	80	100	120	140	160
Pericardiocentesis						
ChloraPrep	88.46	92.3	84.61	88.46	96.15	80.76
Needle Insertion	62.16	68.76	70.87	79.87	83.73	68.07
Thoracentesis						
ChloraPrep	60.29	72.05	73.52	79.41	94.11	91.16
Anesthetization	68.86	57.54	72.64	73.58	73.58	83.96
Scalpel Incision	65.43	85.18	86.41	92.59	95.07	95

Trocar Insertion	74.49	79.36	80.80	88.25	95.4	73.58

Table 7. Validation accuracies

7.2.2 Testing SCNN

We used 2 anchor performances from experts to compare the hold-out data to test our network. Each sample ***x*** of window size ***W*** was paired with an anchor with samples $K_0 - K_n$, the list of input pairs ***P*** was analyzed by the SCNN to give a list of similarity values ***Y***. The final

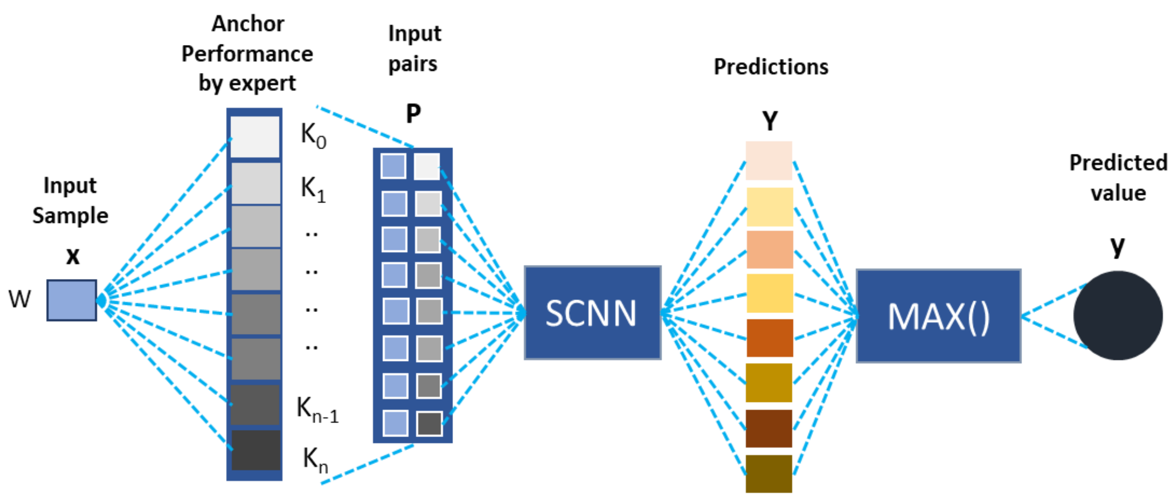


Figure 53. Obtaining similarity value for an input sample

prediction of sample ***x*** was the maximum value ***y*** from the list ***Y***. The set of predictions for all the samples within a trial are averaged to get the final similarity value. The held-out data was tested separately for each anchor. We also wanted to observe the effect of input windows on the prediction of the network, so we used a range of window sizes $w1 = 80$, $w2 = 120$, $w3 = 160$ and for simplicity we used a constant step size $s = 40$. All test results are summarized below.

Similarity values above 0.7 were considered as an expert and anything below 0.3 as a non-expert. Anything between 0.31-0.69 was considered a wrong prediction. The last column in Table 8 is the average value of all similarity values given by the SCNN. On observing the test results, we found that performed satisfactorily over all surgical tasks except the ChloraPrep in THC. Upon investigating we found that the training data was heavily skewed towards the expert label. We could potentially fix this by collecting more non-expert data.

	Surgical Task	Trial	Window size	Actual Prediction Anchor 1	Actual Prediction Anchor 2	Expected Prediction	Average Prediction
PCC	ChloraPrep	T1	w1	0.9583	**0.9961**	1	0.9729
			w2	0.9174	0.9949		
			w3	0.9867	0.9845		
		T2	w1	0.049	0.4	0	0.0325
			w2	**0**	0.101		
			w3	**0**	0.005		
	Needle Insertion	T1	w1	**0.7918**	0.6573	1	0.7057
			w2	0.7288	0.6686		
			w3	0.726	0.6617		
		T2	w1	0.3617	0.28	0	0.2685
			w2	0.29	0.2274		
			w3	0.2575	**0.1944**		
THC	ChloraPrep	T1	w1	**0.6571**	0.651	1	0.6035
			w2	0.6157	0.5786		
			w3	0.5698	0.5489		
		T2	w1	0.1202	0.109	0	0.075
			w2	0.02	0.1651		
			w3	0.021	**0.016**		
	Anesthetization	T1	w1	**0.7138**	0.722	1	0.7077
			w2	0.705	0.7095		
			w3	0.6977	0.6982		
		T2	w1	0.2725	0.2903	0	0.2416
			w2	0.2413	0.2338		
			w3	0.1993	0.2127		
	Scalpel Incision	T1	w1	0.7114	**0.8815**	1	0.7440
			w2	0.72	0.7187		
			w3	0.7075	0.725		
		T2	w1	0.039	0.1903	0	0.0595
			w2	0.022	0.0885		
			w3	0.0175	**0**		
	Trocar Insertion	T1	w1	0.7441	**0.7744**	1	0.7508
			w2	0.7327	0.768		
			w3	0.7315	0.7542		
		T2	w1	0.097	0.0864	0	0.1919

	w2	0.0653	0.044
	w3	0.0397	0.0373

Table 8: Results of the testing Siamese network

7.3 Validation of simulator with assessment

The purpose of the trial was to test the real-time feedback during the surgical training and an objective assessment provided after the training. The trial was held in the presence of 5 participants which included 4 fellows and 1 expert instructor.

The trial was conducted for training NTC. The following protocol was used to run the trial:

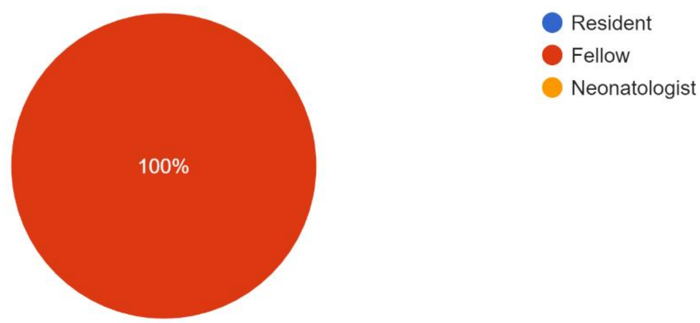
1. Watch and trace the animation of the surgical procedure performed by the instructor.
2. Perform the surgical procedure under the autonomous guidance.
3. Perform the procedure without guidance and receive the assessment results.
4. Fill out a feedback form of the experience.

   Below are given the surgical steps for NTC:

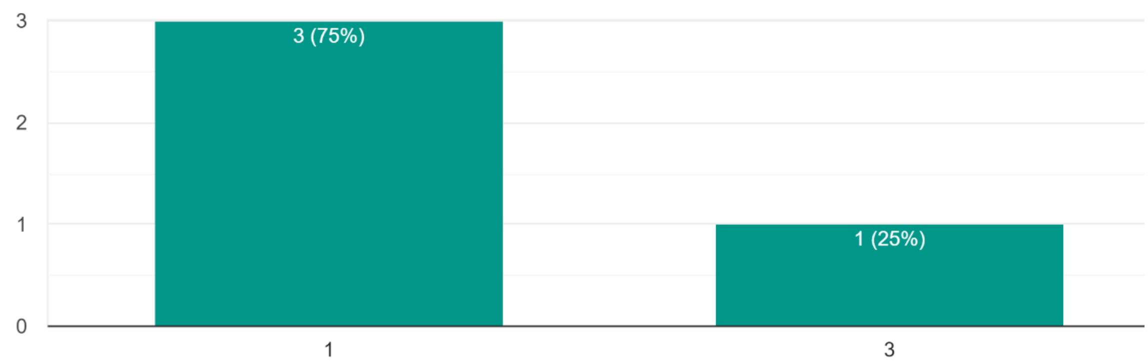
5. ChloraPrep - Prepare the surgical area by the use of a disinfecting applicator. Apply at the center of the surgical area and gradually move outwards in a spiral.
6. Anesthetization- Infiltrate the skin and simulate administration of anesthesia around the surgical area.
7. Incision - Create a small incision with a scalpel the layer of skin and muscle at the target insertion point.
8. Trocar Catheter Insertion - Enter 3-5 cm into the pleural space directing the tip anteriorly and superomedially.

The aggregated results of the feedback for the trial are given below in Fig 54:

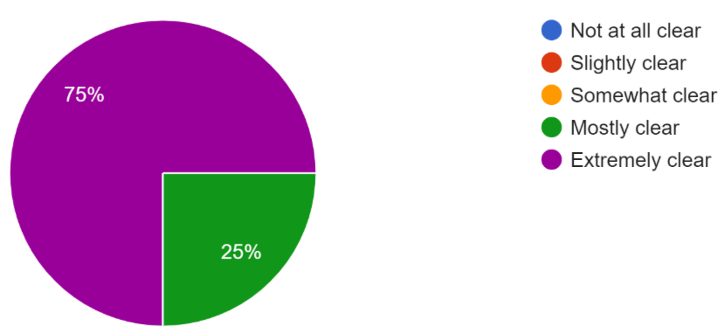
What is your educational background ?
4 responses



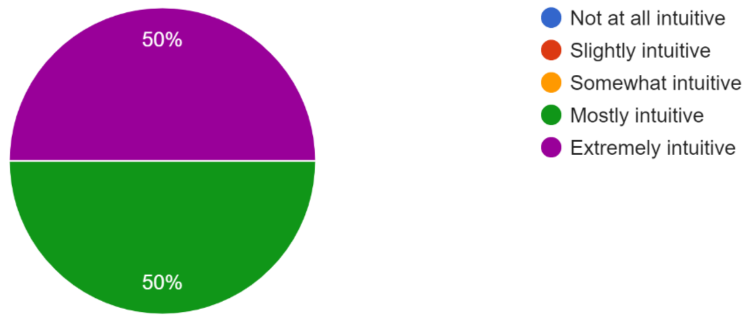
How many years have you been a resident/ fellow/ neonatologist/ surgeon ?
4 responses



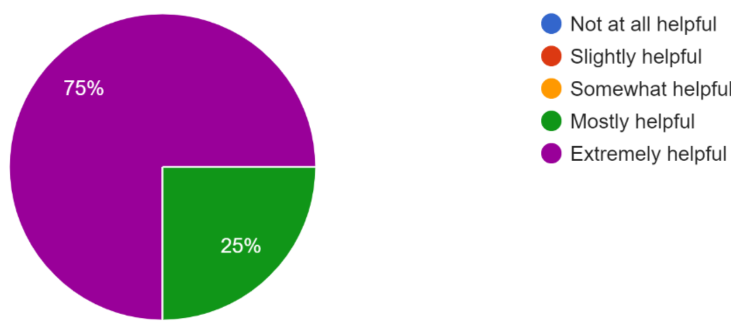
How clear was the instructional guidance to perform the surgical task ?
4 responses



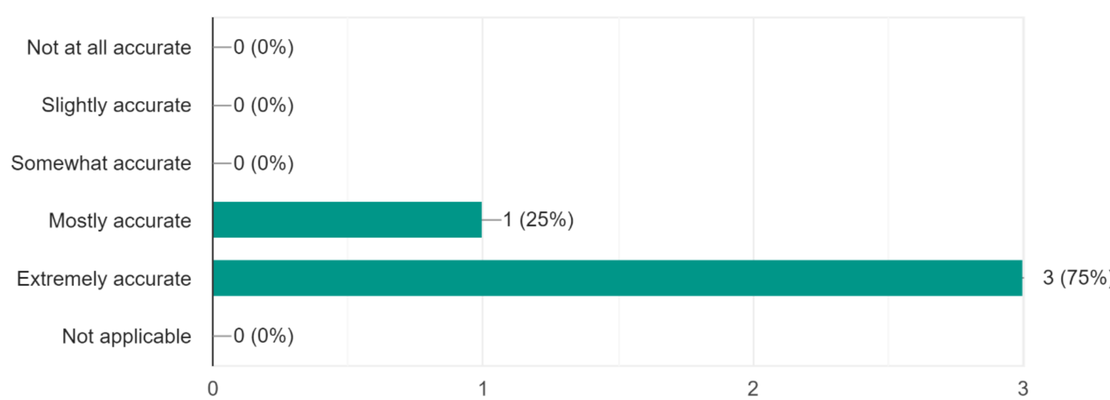
How intuitive was the surgical simulator ?
4 responses

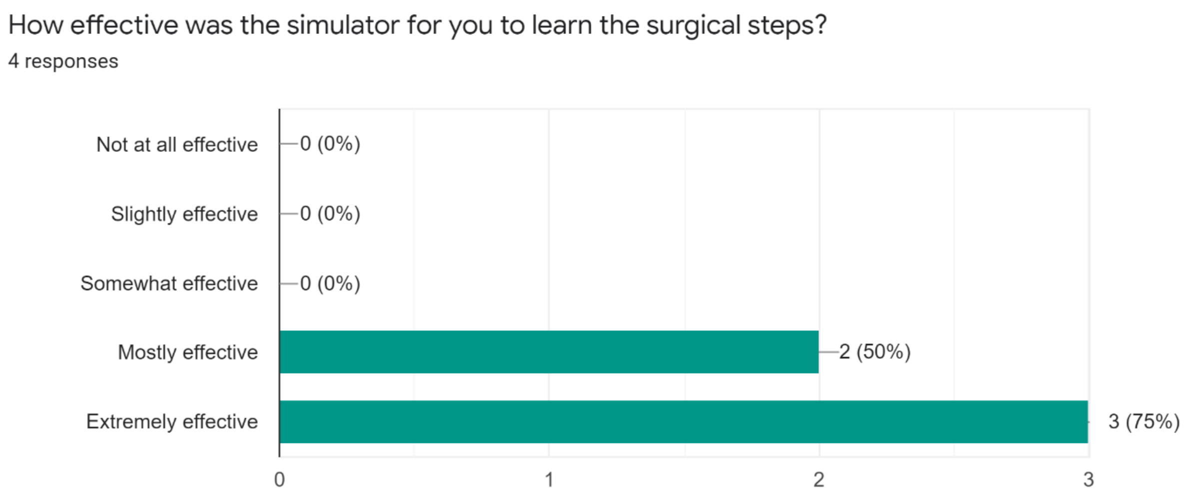


How helpful were the virtual images of the anatomy and instruments
4 responses



How accurate were the virtual images of the anatomy and the instruments?
4 responses





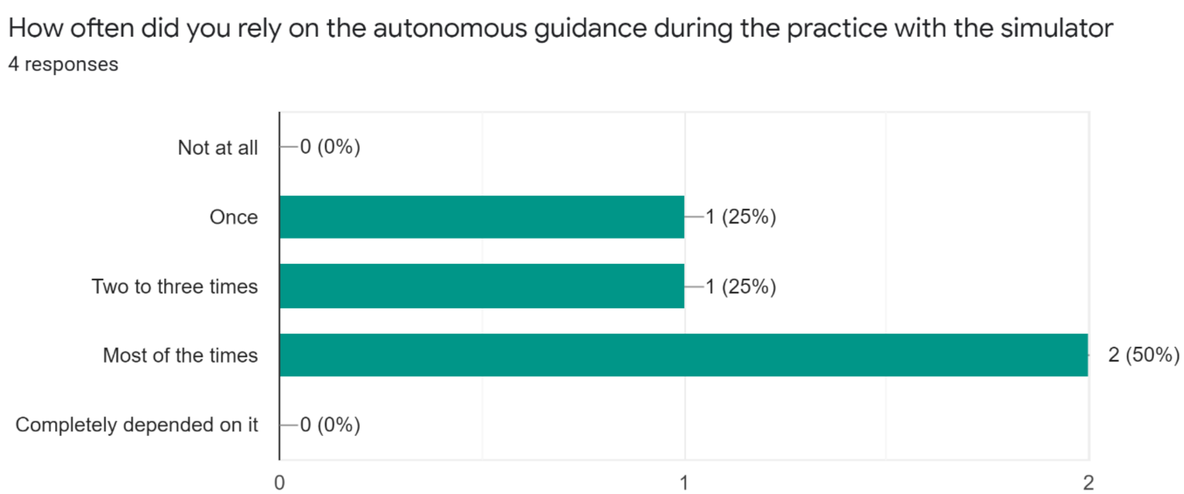


Figure 54: Feedback for autonomous training and assessment

7.4 Discussion

7.4.1 Neonatal Manikin

A novel physical manikin for NPC and NTC, where all of the internal anatomy, operable and inoperable are printed from a regular desktop printer with easily available materials, has been developed. Additionally, only the silicone and the small operable parts would need to be replaced after prolonged usage, most of the anatomy printed in PLA is reusable including the mold for casting. The fact that most of the manikin is made from easily available commercial materials, economical in cost and require no specialty equipment other than a desktop 3D printed means that a multiple manikins could be fabricated for training.

7.4.1.1 NTC

When the surgeons first performed thoracentesis on the manikin, they were able to locate anatomical features such as the nipple and the fourth and fifth intercostal space (ICS) mid axillary. The tactile feedback on inserting the needle through the silicone and the pleura into the pleural space was found to be slightly harder than in real neonate babies. Both the surgical approaches for NTC were performed on the manikin including the pigtail catheter and the trocar catheter method. However, due to the small space inside the pleural cavity, the guidewire insertion was insufficient. Additionally, the pigtail catheter was not rigid enough to pass through the pleural surface and enter the cavity. Consecutively, it was decided to omit the pigtail catheter approach for NTC, and the training was conducted with the trocar catheter.

7.4.1.2 NPC

When the experts successfully performed emergent NPC on the manikin, they found the anatomical features were accurate and the tactile feedback was very similar to the actual procedure. The expert also tested if the procedure could be conducted under image guidance using ultrasound, however the silicone that was used in the casting was found to be completely transparent under ultrasound imaging. Additionally, the needle would be invisible once it entered the pericardium under the imaging since the cavity was hollow and the surface of the pericardium was plastic. In some trials the pericardial cavity within the heart was filled with water to allow for extraction of liquid during the procedure. It was found that the liquid could be extracted during most of the trials, and it helped confirm the accuracy of the needle insertions.

7.4.1.3 Conclusions

The feedback received for the first validation is concurrent with the expert testing. The manikin fabricated shows that most of the anatomy can be 3D printed using the detailed process mentioned in Chapter 4. Most importantly, 100% of the participants found the simulator even without the autonomous capability, was effective for training of these neonatal procedures.

7.4.2 Surgical Simulation Software

The surgical simulation software can be used to train for surgical procedures through watching experts perform, mimicking expert actions, real-time guidance while learning, and finally getting an objective skill assessment of performance. Most of the participants during the trial found the virtual anatomy and instruments extremely accurate and helpful. However, the need for manually designing the surgical instruments in CAD makes it challenging to update the design when surgical instruments are changed.

The substantial sampling rate of the EM tracking system provides smooth and accurate replication of motion. However, the EM system is sensitive to metal and the accuracy rapidly declines in its presence. The software efficiently saves motion information through thresholding and the FastSTray compression algorithm. The algorithm saves data points with low linear correlation to its immediate neighbors; however, it does not consider the differentiated values because of which linearly correlated points with varying velocities are not preserved. Additionally, since this is an open loop compression algorithm, for a given segment of data, the first and the last values are always preserved regardless of their correlation, which is wasteful.

The software uses ZeroMQ to exchange information between the Unity application and the Python script handling the neural network. ZeroMQ is efficient and has low latency. Also, it provides the flexibility to develop the network in python using powerful API like TensorFlow and libraries that are efficient at data processing such as Pandas and NumPy. However, since the two applications are separate, they need to be initialized separately and any modifications made in the messages must appropriately handled in the other application. Currently, the complex messaging system cannot handle large latencies or long message queues, any of those instances causes the application to crash.

In traditional one-on-one assessment, the expert provides some amount of feedback, though its subjective, it has shown to be helpful in improving surgical skills. All the participants depended on the real-time guidance at least once during the training session. This training simulation platform currently has no means of interpreting the assessment scores to provide task specific feedback. Therefore, participants were able to correct their surgical technique, but since the guidance provides the degree of accuracy of the surgical task and not the feedback to correct the mistakes, the trainees are required to explore the technique until the correct way is found. The autonomous assessment provided is objective and gives an accurate comparison of performance to an expert through similarity scores.

7.4.3 Siamese Neural Network

A unique method for skill assessment using a SCNN has been proposed in this research. Due to the low amount of data required to train the Siamese, we were able to sufficient performance accuracy for a total of 6 surgical tasks with large variations in task times. The key reason for using a Siamese network is that it learns the features that differentiate between the surgical skill levels instead of learning to identify the skill levels. The output of the network is based on the comparison of input sample to anchor performance of the expert. During the testing, it was found that using a different anchor performance did not result in substantial variation in results; hence, it can be assumed that the trained network is generalized enough. However, because the result is the maximum value from all the similarity scores for an input sample compared to all the samples from the expert performance, it is assumed that the result signifies the similarity for the same phase of the surgical task. In other words, the network evaluates a window of input data and not the complete surgical task. Hence, any evaluation is purely based on the latest input and not based on the all the previous inputs.

The network performs fast enough to provide a real-time result for an input sample, which is the reason it could be used for the guidance during training. However, Siamese networks take much longer to train since two subnetworks need to be trained. Additionally, as compared to classification networks such as CNN, the Siamese takes much longer to process an input, since the result needs to be compared to all the expert samples.

The major drawback of neural networks is that its challenging to identify the features that are learnt during the training process. Essentially, the layers of the network are like a black box, where the only the input and the output are known. This raises an ethical dilemma especially when the application involves training surgical procedures. Since the generalization of the network depends on the completeness of the dataset, it is essential that the networks go through robust testing. The network used here is an alternative approach to the existing approaches for skill evaluation, however it is not robust enough to replace traditional assessment methods. Since the data was limited to a small number of subjects that work in the NICU, it is understandable that the dataset used here is not large and varied enough to produce an accurate and generalized trained network. However, this network can be developed further through trials and varying population. The autonomous features can be used to reduce the workload and assist instructors to assess surgical procedures.

Chapter 8 - Conclusion

This thesis provides the first end-to-end hybrid simulation platform that could be used to autonomously train and assess surgical procedures using virtual reality and neural networks. The hybrid functionality of the simulator allows for physically performing the surgical procedure with actual instruments with the added benefit of virtual reality and autonomous guidance and assessment.

The physical manikin fabricated for NPC and NTC is viable option as compared to the existing trainers. The manikin provides realistic anatomical features and tactile feedback while allowing to perform the complete surgical procedure. The fabrication process explained is general enough to apply to similar surgical procedures, where the main focus is correct needle insertion. This novelty of the physical manikin is in the fact that both the operable and inoperable anatomy for the procedures have been 3D printed.

The validation of the surgical simulation software shows that, even without the autonomous assessment, the simulator can enhance traditional methods of training for neonatal surgical procedures. The ability to save and playback a performance not only allows the opportunity for the trainees to repeatedly practice ideal surgical techniques, but also gives trainees and experts a more immersive tool to revisit surgical performances as compared to the traditional video capturing devices. The features to save surgical and user data, revisit surgical performances and communicate with python applications are generalized enough to adopt for other surgical procedures with ease. To the best to our knowledge, our software is the first to host a framework that can be used to collect surgical performance data for future applications.

With the increased need for patient safety and more accessible surgical training, there is a requirement of simulators that can be a lot less dependent on expert instructors. Deep learning is being employed in almost all technological applications, exploring all possibilities has led to valuable insight into success in autonomizing processes. We have explored a novel method of surgical training and assessment and found that this research can bring a new paradigm for simulation-based training.

We believe our contribution can be used as a base to propel new methods of skill assessment, creating repositories of surgical data and creating end-to-end surgical training.

8.1 Contributions

The following is the summary of our findings:

- Identified the drawbacks of surgical based simulation and the need for autonomous surgical simulation.
- Recognized the problems with the available physical trainers and created a novel simulation platform for training of neonatal surgical procedures.
- Developed a method to efficiently store surgical action data to file.
- Developed a method to playback a previously performed surgical procedure.
- Created the first surgical dataset for emergent NPC and NTC performed using the trocar catheter approach.
- Created a framework to send surgical data and receive skill level feedback in real-time.
- Explored skill assessment through comparison network instead of classification network using a Siamese convolution neural.
- Validated the simulator with and without the autonomous assessment.

8.2 Future Research

The physical manikin created, though reusable is difficult to disassemble, since its fully encased in silicone. In the future, the design could be modified where only the top surface of the manikin would be cast in silicone. This would require a lot less silicone and provide access to the internal anatomy housed inside the ribcage.

The current drawback with our software is that the surgical tasks involve only one instrument, further development of the platform would be necessary to handle surgical tasks involving two or more instruments. The only tracking system used is the EM system, other form of tracking such as optical tracking and pressure sensors could be used to get a comprehensive surgical data. The quality of the motion data heavily depends on the attachment of the EM sensors to the instruments, which can be a hindrance while performing the procedure.

Currently, the results of the Siamese network are satisfactory, but a larger dataset with more variation would be necessary to improve the generalization of trained model. The neural networks were trained in TensorFlow outside of the simulator platform. It would be preferable to modify the software to host the trained network within the same application without the need for a messaging library such as ZeroMQ.

REFERENCES

Pizzuti, A., Parodi, E., Abbondi, P., & Frigerio, M. (2010). Cardiac tamponade and successful pericardiocentesis in an extremely low birth weight neonate with percutaneously inserted central venous line: A case report. *Cases Journal*, *3*(1), 3–5. https://doi.org/10.1186/1757-1626-3-15

Ogata, E. S., Gregory, G. A., Kitterman, J. A., Phibbs, R. H., & Tooley, W. H. (1976). Pneumothorax in the respiratory distress syndrome: incidence and effect on vital signs, blood gases, and pH. *Pediatrics*, *58*(2), 177–183.

Nowlen, T. T., Rosenthal, G. L., Johnson, G. L., Tom, D. J., & Vargo, T. A. (2002). Pericardial effusion and tamponade in infants with central catheters. *Pediatrics*, *110*(1 I), 137–142. https://doi.org/10.1542/peds.110.1.137

McGaghie, W. C. (2012). Nihms285516. *Acadm Meded*, *86*(6), 706–711. https://doi.org/10.1097/ACM.0b013e318217e119.Does

Angert, R., & Rosen, O. (2021). Pericardiocentesis Simulation Model: Low Cost, Easy to Assemble, Effective Task Trainer. *Cureus*, *13*(4), 0–6. https://doi.org/10.7759/cureus.14475

Chioukh, F. Z., Ameur, K. Ben, Hmida, H. Ben, & Monastiri, K. (2016). Pericardial effusion with cardiac tamponade caused by a central venous catheter in a very low birth weight infant. *Pan African Medical Journal*, *25*, 1–5. https://doi.org/10.11604/pamj.2016.25.13.8731

Kneebone, R. (2003). Simulation in surgical training: Educational issues and practical implications. *Medical Education*, *37*(3), 267–277. https://doi.org/10.1046/j.1365-2923.2003.01440.x

Liu, J., Kurepa, D., Feletti, F., Alonso-Ojembarrena, A., Lovrenski, J., Copetti, R., … Li, H. L. (2020). International expert consensus recommendations for neonatal pneumothorax ultrasound diagnosis and ultrasound-guided thoracentesis procedure. *Journal of Visualized Experiments*, *2020*(157), 1–10. https://doi.org/10.3791/60836

Agha, R. A., & Fowler, A. J. (2015). The role and validity of surgical simulation. *International Surgery*, *100*(2), 350–357. https://doi.org/10.9738/INTSURG-D-14-00004.1

Hammoud, M. M., Nuthalapaty, F. S., Goepfert, A. R., Casey, P. M., Emmons, S., Espey, E. L., … Peskin, E. G. (2008). To the point: medical education review of the role of simulators in surgical training. *American Journal of Obstetrics and Gynecology*, *199*(4), 338–343. https://doi.org/10.1016/j.ajog.2008.05.002

Maran, N. J., & Glavin, R. J. (2003). Low- to high-fidelity simulation - A continuum of medical education? *Medical Education, Supplement*, *37*(1), 22–28. https://doi.org/10.1046/j.1365-2923.37.s1.9.x

Zurca, A. D., Diaz-Sanchez, A., MacKley, L., & Rodgers, D. L. (2020). Development of an Innovative Nonanimal Training Model for Infant Pleural Effusion Drainage via Pigtail Catheter Placement. *Advances in Neonatal Care*, *20*(2), 176–179. https://doi.org/10.1097/ANC.0000000000000680

Gupta, A. O., & Ramasethu, J. (2014). An innovative nonanimal simulation trainer for chest tube insertion in neonates. *Pediatrics*, *134*(3), e798–e805. https://doi.org/10.1542/peds.2014-0753

Sarin, E. L., Bettadapura, V., Essa, I., Zia, A., & Sharma, Y. (2018). Video and accelerometer-based motion analysis for automated surgical skills assessment. *International Journal of Computer Assisted Radiology and Surgery*, *13*(3), 443–455. https://doi.org/10.1007/s11548-018-1704-z

Funke, I., Mees, S. T., Weitz, J., & Speidel, S. (2019). Video-based surgical skill assessment using 3D convolutional neural networks. *International Journal of Computer Assisted Radiology and Surgery*, *14*(7), 1217–1225. https://doi.org/10.1007/s11548-019-01995-1

Marino, D. L., & Manic, M. (2016). Fast Trajectory Simplification Algorithm for Natural User Interfaces in Robot Programming by Demonstration, 905–911.

Niitsu, H., Hirabayashi, N., Yoshimitsu, M., Mimura, T., Taomoto, J., Sugiyama, Y., … Takiyama, W. (2013). Using the Objective Structured Assessment of Technical Skills (OSATS) global rating scale to evaluate the skills of surgical trainees in the operating room. *Surgery Today*, *43*(3), 271–275. https://doi.org/10.1007/s00595-012-0313-7

Reiley, C. E., Lin, H. C., Yuh, D. D., & Hager, G. D. (2011). Review of methods for objective surgical skill evaluation. *Surgical Endoscopy and Other Interventional Techniques*. https://doi.org/10.1007/s00464-010-1190-z

Oropesa, I., Sánchez-González, P., Lamata, P., Chmarra, M. K., Pagador, J. B., Sánchez-Margallo, J. A., … Gómez, E. J. (2011). Methods and tools for objective assessment of psychomotor skills in laparoscopic surgery. *Journal of Surgical Research*, *171*(1). https://doi.org/10.1016/j.jss.2011.06.034

Morais, P. A. G. (2015). Development of a patient-specific phantom model.

Cafarelli, A., Miloro, P., Verbeni, A., Carbone, M., & Menciassi, A. (2016). Speed of sound in rubber-based materials for ultrasonic phantoms. *Journal of Ultrasound*, *19*(4), 251–256. https://doi.org/10.1007/s40477-016-0204-7

Byrne, N., Velasco Forte, M., Tandon, A., Valverde, I., & Hussain, T. (2016). A systematic review of image segmentation methodology, used in the additive manufacture of patient-specific 3D printed models of the cardiovascular system. *JRSM Cardiovascular Disease*, *5*, 204800401664546. https://doi.org/10.1177/2048004016645467

Angert, R., & Rosen, O. (2021). Pericardiocentesis Simulation Model: Low Cost, Easy to Assemble, Effective Task Trainer. *Cureus*, *13*(4), 0–6. https://doi.org/10.7759/cureus.14475

Gupta, A. O., & Ramasethu, J. (2014). An innovative nonanimal simulation trainer for chest tube insertion in neonates. *Pediatrics*, *134*(3), e798–e805. https://doi.org/10.1542/peds.2014-0753

Zurca, A. D., Diaz-Sanchez, A., MacKley, L., & Rodgers, D. L. (2020). Development of an Innovative Nonanimal Training Model for Infant Pleural Effusion Drainage via Pigtail Catheter Placement. *Advances in Neonatal Care*, *20*(2), 176–179. https://doi.org/10.1097/ANC.0000000000000680

Bruno, C. J., & Glass, K. M. (2016). Cost-effective and low-technology options for simulation and training in neonatology. *Seminars in Perinatology*, *40*(7), 473–479. https://doi.org/10.1053/j.semperi.2016.08.008

Badash, I., Burtt, K., Solorzano, C. A., & Carey, J. N. (2016). Innovations in surgery simulation: a review of past, current and future techniques. *Annals of Translational Medicine*, *4*(23), 453–453. https://doi.org/10.21037/atm.2016.12.24

Lin, H. C., Shafran, I., Yuh, D., & Hager, G. D. (2006). Towards automatic skill evaluation: Detection and segmentation of robot-assisted surgical motions. *Computer Aided Surgery*, *11*(5), 220–230. https://doi.org/10.3109/10929080600989189

Sutherland, L. M., Middleton, P. F., Anthony, A., Hamdorf, J., Cregan, P., Scott, D., & Maddern, G. J. (2006). Surgical simulation: A systematic review. *Annals of Surgery*, *243*(3), 291–300. https://doi.org/10.1097/01.sla.0000200839.93965.26

Castillo-Segura, P., Fernández-Panadero, C., Alario-Hoyos, C., Muñoz-Merino, P. J., & Delgado Kloos, C. (2021). Objective and automated assessment of surgical technical skills with IoT systems: A systematic literature review. *Artificial Intelligence in Medicine*, *112*(November 2020). https://doi.org/10.1016/j.artmed.2020.102007

Liu, R., & Holden, M. S. (2020). Kinematics Data Representations for Skills Assessment in Ultrasound-Guided Needle Insertion. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12437 LNCS*, 189–198. https://doi.org/10.1007/978-3-030-60334-2_19

Khalid, S., Goldenberg, M., Grantcharov, T., Taati, B., & Rudzicz, F. (2020). Evaluation of Deep Learning Models for Identifying Surgical Actions and Measuring Performance. *JAMA Network Open*, *3*(3), e201664. https://doi.org/10.1001/jamanetworkopen.2020.1664

Xiao, X., Zhao, S., Zhang, X., Soghier, L., & Hahn, J. (2020). Automated Assessment of Neonatal Endotracheal Intubation Measured by a Virtual Reality Simulation System. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, *2020-July*, 2429–2433. https://doi.org/10.1109/EMBC44109.2020.9176629

Hou, L., Jin, X., & Zhao, Z. (2019). Time Series Similarity Measure via Siamese Convolutional Neural Network. *Proceedings - 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, CISP-BMEI 2019*, (October 2019). https://doi.org/10.1109/CISP-BMEI48845.2019.8966048

Levin, M., McKechnie, T., Khalid, S., Grantcharov, T. P., & Goldenberg, M. (2019). Automated Methods of Technical Skill Assessment in Surgery: A Systematic Review. *Journal of Surgical Education*, *76*(6), 1629–1639. https://doi.org/10.1016/j.jsurg.2019.06.011

Nguyen, X. A., Ljuhar, D., Pacilli, M., Nataraja, R. M., & Chauhan, S. (2019). Surgical skill levels: Classification and analysis using deep neural network model and motion signals. *Computer Methods and Programs in Biomedicine*, *177*, 1–8. https://doi.org/10.1016/j.cmpb.2019.05.008

Wang, Z., & Majewicz Fey, A. (2018). Deep learning with convolutional neural network for objective skill evaluation in robot-assisted surgery. *International Journal of Computer Assisted Radiology and Surgery*, *13*(12), 1959–1970. https://doi.org/10.1007/s11548-018-1860-1

Drumond, R. R., Marques, B. A. D., Vasconcelos, C. N., & Clua, E. (2018). PEEK: An LSTM recurrent network for motion classification from sparse data. *VISIGRAPP 2018 - Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, *1*(Visigrapp), 215–222. https://doi.org/10.5220/0006585202150222

Fawaz, H. I., Forestier, G., & Weber, J. (2018). Evaluating Surgical Skills from Kinematic Data Using Convolutional Neural Networks.

VITA

NAME	Nihar Nitin Sheth
EDUCATION	Ph.D., Bioengineering University of Illinois at Chicago (UIC), 2021
	Master of Science, Mechanical Engineering University of Illinois at Chicago, 2014
	Bachelor of Science, Mechanical Engineering Maharashtra Institute of Technology, Pune, India, 2012
WORK EXPERIENCE	Mar 2016 – May 2017 Medical Accelerator for Devices (MADLab) UIC, Illinois (IL)
	Jun 2016 – Aug 2016 Summer Internship PraxiCut, Chicago, IL
	Jun 2014 – Aug 2014 Summer Internship Immersive Touch, Chicago, IL
	Jan 2012 – Jun 2012 Internship Tata Motors, Maharashtra, India
	Jun 2010 – Dec 2010 Internship Tata Motors, Maharashtra, India
RESEARCH EXPERIENCE	May 2016 – May 2017 Research Assistant Mixed Reality Lab, UIC, IL
	May 2015 – Aug 2015 Research Assistant Industrial Virtual Reality Institute, UIC, IL
TEACHING EXPERIENCE	Fall, 2014 Teaching Assistant Ergonomics, Department of Mechanical and Industrial Engineering (MIE), UIC

TEACHING EXPERIENCE (Continued)	Spring, 2015 Teaching Assistant Computer Aided Design, MIE, UIC
	Fall 2015 Teaching Assistant Senior Design, MIE, UIC
	Spring 2016 Teaching Assistant Introduction to Computer Aided Design, MIE, UIC
HONORS	Beverlee Anderson Educational Scholarship (2021)
PROFESSIONAL MEMBERSHIP	Society of Simulation in Healthcare (2021)
PRESENTATIONS	Annual UIC Bioengineering Research Symposium Apr 16th, 2021
	American College of Surgeons (ACS) ACS Surgeons and Engineers: A dialogue on Surgical Simulation Mar 11th, 2021
	SSH International Meeting on Simulation in Healthcare (IMSH) Jan 19th – Mar 31st, 2021
	Society of Minimally Invasive Therapy (SMIT) 32nd Annual SMIT Congress Dec 3rd – 10th ,2020
	Society of Simulation in Healthcare (SSH) SSH SimOps Jul 16th - 17th , 2020
	ACS Surgical Simulation Summit Mar 16th – 17th , 2018 Chicago, IL
PUBLICATIONS	Sheth, Nihar N., Srinivasan, Nishant and Patel, Saurabhkumar and Luciano, Cristian J., Preliminary evaluation of a novel neural-network-based hybrid simulator for surgical training and performance assessment of neonatal thoracentesis, SUBMITTED

PUBLICATIONS (Continued)	Sheth, Nihar N., Srinivasan, Nishant and Patel, Saurabhkumar and Luciano, Cristian J., Evaluation Surgical Skills using Siamese Convolution Neural Network, SUBMITTED
	Sheth, Nihar and Marjanovic, Nicholas and Srinivasan, Nishant and Patel, Saurabhkumar and Luciano, Cristian, Standardized Process for Fabricating a Physical Trainer for Neonatal Pericardiocentesis and Thoracentesis and an Introduction to a Surgical Simulation Assistant Platform (June 5, 2020). International Meeting on Simulation in Healthcare (IMSH 2021) Technical Proceedings, Available at SSRN: https://ssrn.com/abstract=3818504