

Optimal Positioning of Robotic Arms Inside the Abdominal Cavity in Pediatric Surgery

Department of Surgery

Clinical Internship II

Technical Medicine, Medical Imaging & Interventions

August 2020

Pien de Graaf
S1721569

Mentors:

Prof. E.C.J. Consten – Meander Medisch Centrum

J.L.M. Bruggink, MD – Universitair Medisch Centrum Groningen

R.H.J. de Kleine, MD – Universitair Medisch Centrum Groningen

Prof. S. Misra – University of Twente

C.M. Heunis, MSc – University of Twente

**UNIVERSITY
OF TWENTE.**



umcg

Abstract

Robot-assisted minimally invasive surgeries (RMIS) can be performed in various fields with the aid of the da Vinci Surgical System.² RMIS has several advantages to conventional laparoscopy²⁻⁹ and appears to be safe and feasible for a number of pediatric surgical procedures.^{3,10} However, the bulky robotic arms occupy significant space in the small pediatric abdominal cavity.^{3,13} Robotic port placement in pediatric surgery is considered crucial, as it has great influence on patient safety, surgery efficiency and the mobility of robotic arms without collision.^{8,9,16} The goal of this research is to design a computational model which ultimately can be employed to determine optimal positioning of robotic arms inside the abdominal cavity in pediatric surgery. Model requirements are set up and model types are evaluated, where after the model framework is elaborated using MATLAB. The framework consists of the patients' torso and target organs, complemented with a hypothetical pneumoperitoneum of the skin and fascia, trocar lines, unsafe regions for trocar placement and trocars. This forms the basis of a patient specific computational model that is part of a surgical planning strategy. However, there is still a demand for more research and further elaboration of the model, e.g. body changes due to insufflation, applying the framework to multiple children and collision detection of the robotic arms.

Key words: *Computational Model, Da Vinci, Pediatric Patients, Robotic Surgery*

I. Introduction

Robot-assisted minimally invasive surgeries (RMIS) are increasingly applied to perform procedures on deformable anatomical structures.¹ With the aid of the da Vinci Surgical System® (Intuitive Surgical, Sunnyvale, USA), RMIS can be performed in various fields such as general surgery, urology, gynecology, head and

neck surgery, and pediatric surgery.² RMIS hold great promise to enhance surgical accuracy in addition to the surgeon's dexterity.²⁻⁹ It ensures good cosmetic outcome, less postoperative pain and in general a more rapid recovery.^{4,6,7} Since the introduction of robot-assistance in surgery, laparoscopic surgeons are able to include more advanced and complex reconstructions to their repertoire.^{2,4}

RMIS appear to be safe and feasible for a number of pediatric surgical procedures.^{3,10} For complex pediatric hepatobiliary cases, RMIS seem to be ideally suited.^{4,10} In the Netherlands, pediatric RMIS are performed at the Universitair Medisch Centrum Groningen (UMCG) on children diagnosed with choledochal malformation (CM). CM is a rare congenital dilatation of intrahepatic and/or extra-hepatic bile ducts,¹¹⁻¹⁴ with an incidence of 1:59.000 newborns in the Netherlands. In about 80% of the cases, CM is presented before the tenth year of life.¹¹ CM is considered to be a premalignant condition, meaning total resection of the malformed mucosa is crucial.^{6,7,11-14} Roux-en-Y-hepaticojejunostomy is performed to recover continuity.^{4,7,11,12,14} Hepaticojejunostomy is a technical demanding procedure, which can be performed easier and more precisely using RMIS.^{4,6} RMIS are considered an emerging approach to reduce complications, compared with laparoscopic surgeries alone.^{6,14}

Within the field of RMIS, the da Vinci robot has provided several advantages to conventional laparoscopy, including three-dimensional (3D) vision, tremor filtration, motion scaling, visual immersion and intuitive movements. Moreover, seven degrees of freedom facilitate wrist-like motion, surgeons have control over their own visual field, and the surgeon's ergonomics are improved.²⁻⁹ A surgeon sitting comfortably at the console additional to fine finger and wrist movement stabilization ease suturing, knot tying and dissection.⁶ RMIS overcome some of the conventional laparoscopic pitfalls, and is therefore considered superior to laparoscopic surgery.^{2,6}

The da Vinci robot (Figure 1) contains four arms, of which one is the central camera arm. A wide variety of instruments is available with the system, rapidly changeable by an assistant surgeon. Easy arm positioning during setup is realized by the multiple positioning joints and terminal pivot joint at the attachment with the port.² First, the camera port is inserted through an umbilical curved incision.^{4-8,13,14} A pneumoperitoneum is then established by carbon dioxide insufflation with a pressure of ≤ 12 mm Hg, depending on the patient's age.^{7,8,13,14} Insertion of secondary working ports is done under direct vision of the inserted camera.^{4,5,8} In cases of pediatric patients with a small body cavity, three robotic arms are employed instead of four.^{4,6,7} However, this can lead to more difficult surgical proceedings. Therefore, working with four arms is preferred.^{7,14} There is at least one assisting laparoscopic port added for the bedside assistant, to insert sutures or additional instruments to provide suction, retract tissue or cut sutures.^{4,7,9}



Figure 1: The da Vinci Xi robotic system with a surgeon sitting at the console.

15

The da Vinci robotic system is designed for adult patients.⁷ Therefore, the robotic arms are bulky and occupy significant space, which could be a problem in a small abdominal cavity when performing pediatric surgery.^{3,13} Lack of instruments specialized for pediatric surgery requires employability of standard instruments.⁷ Robotic port placement in pediatric surgery is considered crucial to ensure a smooth case. Port placement and patient positioning have great influence on patient safety, surgery efficiency and, most challenging, maximum mobility of robotic arms without collision. Misplacing ports may lead to

serious complications and injuries, may restrict the endoscopic view and limit surgical tools' access to the target organ.^{8,9,16} Port placement decisions are based on the surgeon's experience, knowledge of the target organ position and guidelines.¹⁵ This leads to variability between surgeons and the need for a universal approach to determine optimal positioning of robotic arms inside the pediatric abdominal cavity.

Based on these notions, the goal of this research is to design a computational model, which ultimately can be employed to determine optimal positioning of robotic arms inside the abdominal cavity in pediatric surgery. This study requires the following deliverables:

1. Model requirements
2. Evaluation of model types
3. Model framework

The model enables evaluation of different port placements, usability of the number of robotic arms, in addition to the estimation of intra-operative risks. Inclusion of preoperative imaging data allows the model to be patient specific. Implementation of the model will improve surgery efficiency, counteract surgeon variability and facilitate patient specific preoperative planning. The rest of this report is structured as follows: in Section II an overview is given of the model requirements, followed by the evaluation of model types in Section III. In Section IV, the design of the model framework is elaborated, which is discussed in Section V. The conclusion can be found in Section VI. Finally, in Section VII future recommendations are described.

II. Model Requirements

Several requirements are set up, which must be able to be evaluated by means of the model. This guarantees truthfulness of the model and implementation safety. However, firstly attention must be given to patient-specific factors⁹, taking into consideration:

- Patient selection
- Surgery indications
- Previous surgery and scarring

- Medical history, co-morbidity and risk factors

The requirements are as follows:

- Ports must be distant from each other to provide adequate instrumental maneuverability and avoid collision^{8,15}
 - Distance between the robotic ports should be 6 - 8 centimeter (cm)¹⁷, however, in pediatric cases this distance between ports cannot always be realized and shorter distances may be considered
 - Increasing the distance between ports reduces the chance of external robot arm collisions, but this is only good up to a certain point; too far apart ports may result in external arm interference with the operation room (OR) table or the patient⁹
 - For a small patient, working ports should be placed more lateral^{9,10}
- Angles of the arms in relation to each other are important to maintain, so as to avoid robotic arm collisions⁹
 - The angle between the left and right working arms should be about 90 degrees, both with an angle about 45 degrees from the camera⁹
 - The smaller the patient, the wider the trocar angulation must be⁹
- The distance that separates each instrument port from the target organ should be 10 to 20 cm in order to achieve sufficient working space¹⁸
- The camera port and each lateral port should be aligned toward the target organ⁹
- Working ports should be placed at a distance > 3 cm from the superior iliac spine and costal margin⁹
- Entry ports act like a pivoting point in the abdominal surface, which creates a conic working volume around the target organ¹⁶
 - The closer the instrument is to the target organ, the bigger the range of motion with increasing angular size
 - The pivoting point is located on the fascia layer⁹
 - The point around which the robot arm will pivot should be the remote center of the trocar; the distance between the remote

center to the end of the port is 2.90 cm (therefore, the remote center is placed just at the inside edge of the abdominal cavity and 2.90 cm of trocar length is inside the patient)⁹

- The surgical tools' working volume must include the whole area around the target organ⁹
 - The working volume inside the patient might be restricted by the length of the surgical tools and the robot arms' pivoting angle at the entry ports⁹

To summarize, the model requirements described above suggest integration of several parameters into the model and demands use of imaging data. These parameters include the patient's torso size, the target organ position, the robot arms' configuration and position relative to the patient. These parameters are considered determinative factors when deciding about optimal port locations on a patients' body during the preoperative planning phase of a robotic surgery.⁹

III. Evaluation Model Types

In this section, different model types are elaborated and assessed on ability to include the model requirements from Section II. This enables determination of the best suitable model type for this application. Elaborated model types include dynamic models, programming, virtual reality and phantoms.

Dynamic Models¹⁹

Dynamic models represent behavior of an object over time. It derives equations of motions for all moveable parts and solve these using an iterative method. Dynamic models can be divided into lumped mass models, multi-body models and finite element models.

• Lumped Mass Models

This type of model consists of rigid bodies with masses, connected by springs and dampers representing connections. Lumped mass models are usually one- or two-dimensional (2D). As the

dynamic balances and other relations are yet undefined and imaging data cannot be integrated, this type of model is not suitable for this application.

- **Multi-Body Models**

This type of model formulates various joints connections, through which the number of degrees of freedom between elements can be constrained. In a multi-body formulation, also flexible bodies can be specified. It is an efficient way for complex kinematic connections, 2D or 3D. However, this type of model demands preoperative imaging of the patient in different positions to set values for kinematic parameters in addition to undefined relations, which is not realistic.

- **Finite Element Models**

In this type of model, the system is divided into a number of finite volumes, making it capable of describing local deformations and stress distribution. Finite Element Models are 3D. This type of model is less attractive for studies that involve many parameters, are associated with long computing times and availability of realistic material data is limited, making it not suitable for this application.

Programming models

Programming models act like bridges between algorithms and implementations in software. A programming model can be created in several ways, for instance using MATLAB. MATLAB allows manipulations, creation of user interfaces, implementation of algorithms and plotting of functions and data. Its functionality can be expanded by additional use of toolboxes.²⁰ An algorithm can be created that integrates imaging data, which enables visualization of the preoperative body and simulation of the intra-operative situation, performance of calculations (e.g. volumes), model and control degrees of freedom and create adjustable kinematic parameters.

Virtual Reality

Virtual Reality (VR) enables simulation of the clinical scenario, based on patient-specific

anatomy derived from preoperative imaging data. The set-up of a surgical robot can be integrated into the simulation. Generally, VR is used for motion rehearsal and the evaluation of motion feasibility.²¹ For the creation of the robot simulator, applications as V-REP and RoboDK can be used.

Phantom

Another way to evaluate placement of robotic ports is by using a phantom. A phantom can be created which mimics the human body and its properties. However, for instance, to detect whether a certain port placement causes robotic arm collision, use of the da Vinci is demanded. Additionally, a phantom is not patient specific. These factors make the use of a phantom within this study not realistic.

In terms of available facilities, feasibility and time, there is chosen to start creating the base of the framework with MATLAB. This enables use of preoperative imaging data to make the model patient specific. Additionally, it allows integration of parameters and calculations, which is the most suitable for this clinical problem.

IV. Model Framework

The model framework is created using MATLAB® Version 2018b. A Computed Tomography (CT) preoperative scan was used as an input to make the model patient specific. For this application, the CT scan would ideally be from a child. However, the provided CT scan turned out not to be sufficient. Therefore, the model design is based on the CT scan of a 20 year old patient. The MATLAB algorithm (Appendix A) is written in such way that any transversal CT or Magnetic Resonance Imaging (MRI) scan as stacked DICOM files can be employed as input for the framework. Additionally, in the context of segmentation, the scan must contain sufficient contrast to distinguish different gray values from organs and tissues.

Firstly, the patient's torso size was extracted from the CT scan, by making the image stack binary.

A Region of Interest (ROI) was selected that contains the abdomen area, to downsample the dataset. Additional functions were applied to obtain a solid white area representing the patient's abdomen. Hereafter, isosurface faces and vertices were extracted from the binary image stack, and again downsampled. The vertices were multiplied by the slice thickness and voxel size, to correct for unequal dimensions. Hereby, distances on the axes are all in millimeter (mm). The centroid was shifted towards the origin. This remains a truthful representation of the surface of the torso of the patient, see Figure 2. Target organs, in this case the liver, spleen and kidneys, were segmented from the CT scan. This was also done by binarization of the image stack and subsequent described steps (Figure 3).

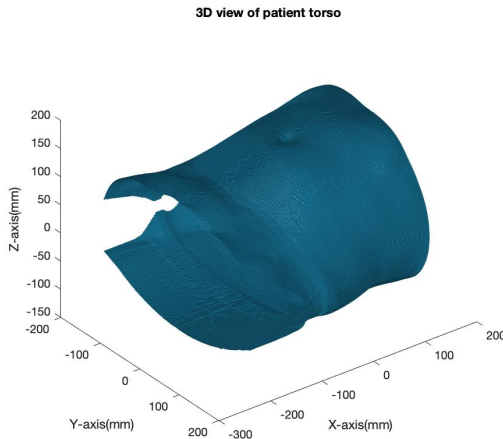


Figure 2: The segmented torso of the patient.

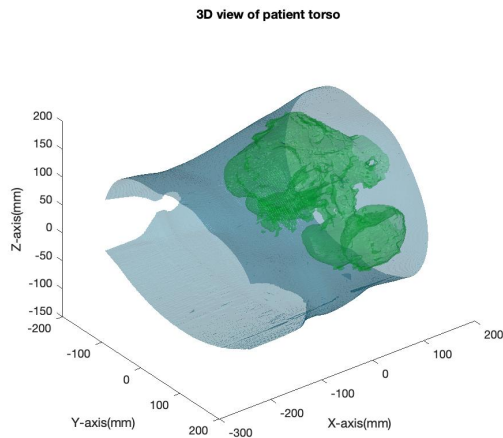


Figure 3: The torso of the patient in blue, segmented target organs in green.

Intraoperatively, the patient's abdomen is insufflated. This shape could not be derived from available data. Therefore, a hypothetical pneumoperitoneum was created by adding a part of a pointcloud ellipsoid on top of the surface of the preoperative abdomen (Figure 4). These points can all be used as trocar entry points. However, the fascia layer is where the trocar pivoting point is located. Therefore, another pointcloud ellipsoid was added to the framework. The distance between both ellipsoids is based on the thickness of the layer between the abdominal surface and fascia, measured from the CT scan. Both ellipsoids can be seen in Figure 5a and 5b.

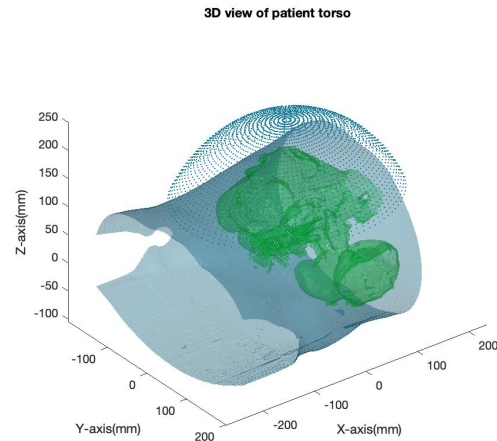


Figure 4: The torso of the patient in blue, target organs in green and a blue pointcloud representing the pneumoperitoneum (skin).

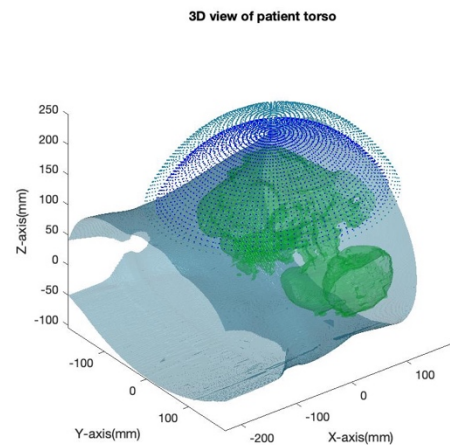


Figure 5a: The torso of the patient in blue, target organs in green and two blue pointclouds representing the pneumoperitoneum skin and fascia.

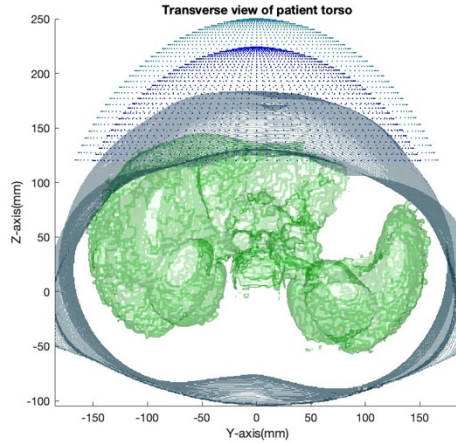


Figure 5b: The torso of the patient in blue, target organs in green and two blue pointclouds representing the pneumoperitoneum skin and fascia.

Up next, coordinates of landmark locations were stored in matrices. These landmarks include the umbilicus preoperative, the (hypothetical) umbilicus of the pneumoperitoneum and the target organ. This enables for calculations of distances between landmarks. The distance between each instrument port and target organ should be 10 to 20 cm.¹⁸ This is integrated into the model; a region on the pneumoperitoneum skin surface is colored in pink, if the distance towards the target organ coordinate is ≤ 15 cm. This region gives an indication of which locations can better be avoided during port placement, see Figure 6. The target organ coordinate represents the location of the gall bladder.

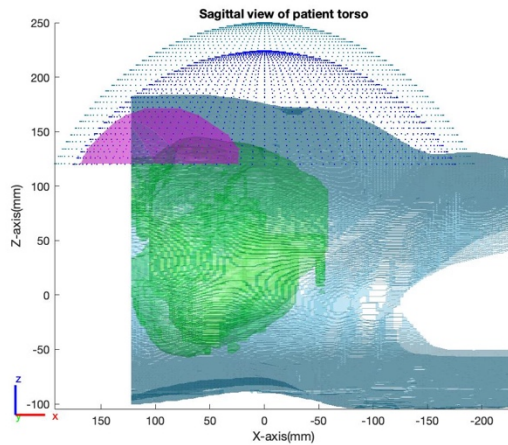


Figure 6: The torso of the patient in blue, target organs in green, blue pneumoperitoneum pointclouds and a pink region of coordinates within 15 cm from the target organ.

The camera port is inserted through an umbilical curved incision.^{4-8,13,14} Therefore, the umbilicus location of the pneumoperitoneum is considered the entry point of trocar 1. A trocar line was simulated through the entry point toward the target organ (Figure 7). The total length of the line is set at 59.2 cm, as this is the average total instrument length of the da Vinci.²² Distance between ports should be 6 - 8 cm¹⁷, meaning that within a region of 6 cm from trocar entry point 1 no additional port should be placed. In the model this was implemented as a red region around the trocar entry point (Figure 8).

In Figures 7 and 8, a bright blue point on the red trocar line is visible. This point represents the location of the pivoting point on the fascia layer. To find this location, function *skin2fascia* was written (Appendix B). This function finds the derivative of the trocar line that is closest to a point of the fascia ellipsoid. This way, the coordinate of the trocar line on the fascia layer can be calculated, for each location of the trocar entry point.

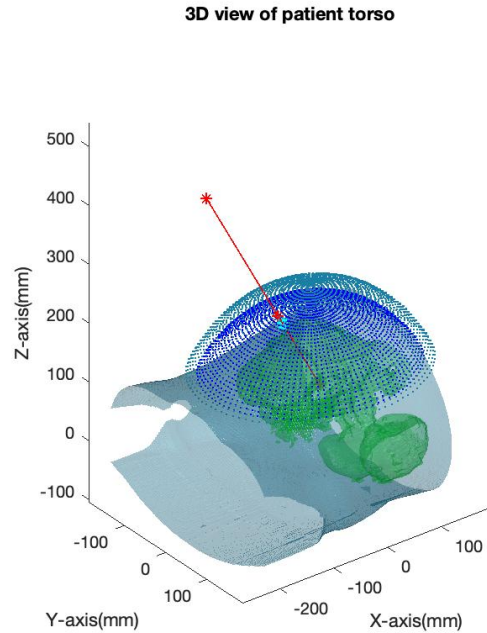


Figure 7: The torso of the patient in blue, target organs in green, blue pneumoperitoneum pointclouds and red trocar line 1 with a blue pivoting point.

3D view of patient torso

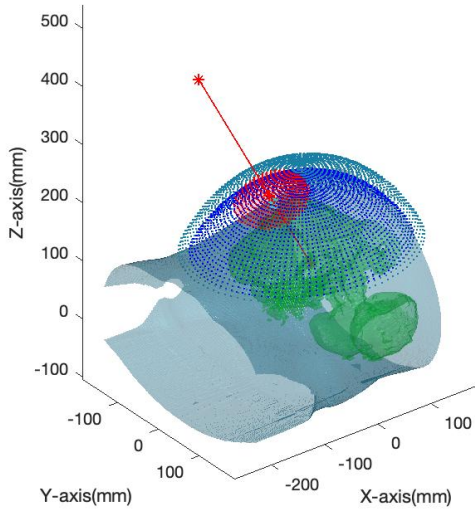


Figure 8: The torso of the patient in blue, target organs in green, blue pneumoperitoneum pointclouds and red trocar line 1 with a blue pivoting point and a red region within 6 cm from entry point 1.

The other three trocar lines were added to the model the same way described above (Figure 9). See Figure 10 for added red regions around the trocar lines on the pneumoperitoneum.

3D view of patient torso

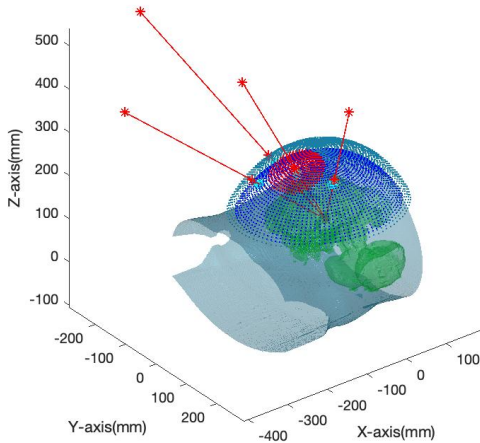


Figure 9: The torso of the patient in blue, target organs in green, blue pneumoperitoneum pointclouds and red trocar lines 1, 2, 3 and 4 with a blue pivoting point and a red region within 6 cm from entry point 1.

3D view of patient torso

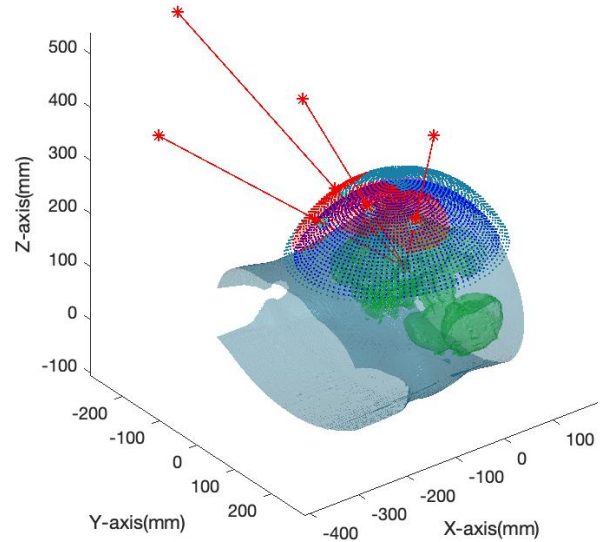


Figure 10: The torso of the patient in blue, target organs in green, blue pneumoperitoneum pointclouds and red trocar lines 1, 2, 3 and 4 with a blue pivoting point and red regions within 6 cm from the entry points.

To include trocars in the model, function `make trocar` was written (Appendix C). This function creates a cylinder over the trocar lines in the same alignment. The ratio of the trocar length inside and outside the fascia layer are integrated in this function. The inside ratio should represent a trocar length of 2.90 cm for the remote center of the trocar to be the pivot point.⁹ See Figure 11a and 11b for visualization of the added trocars. See Appendix D for figures of alternate target organs.

An attempt has been made to simulate the da Vinci robot using the Robotics System Toolbox™ in MATLAB. This toolbox provides tools and algorithms for designing, simulating and testing robots, including options for collision checking using a rigid body tree representation.²³ A rigid body tree was created; four kinematic chains that end with an end-effector represent the four robotic arms. However, time restrictions have limited extensive elaboration. Therefore, the created robot simulation is yet not ready to be merged with the model.

3D view of patient torso

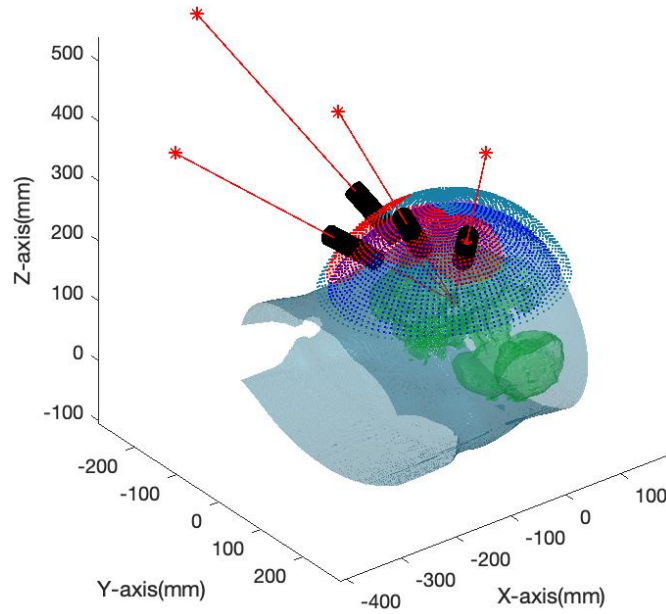


Figure 11a: The torso of the patient in blue, target organs in green, blue pneumoperitoneum pointclouds and red trocar lines 1, 2, 3 and 4 with red regions within 6 cm from the entry points and visualized trocars.

Transverse view of patient torso

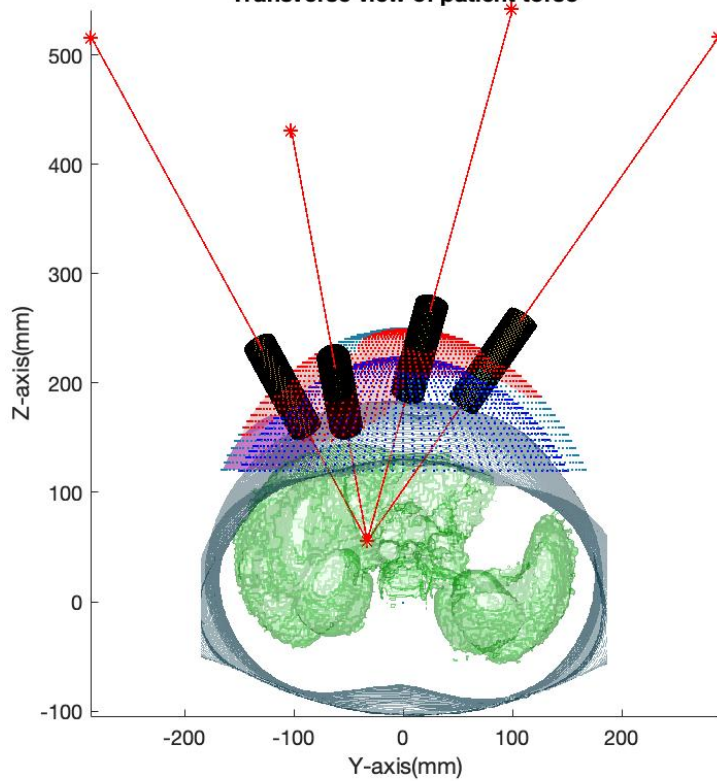


Figure 11b: The torso of the patient in blue, target organs in green, blue pneumoperitoneum pointclouds and red trocar lines 1, 2, 3 and 4 with red regions within 6 cm from the entry points and visualized trocars.

V. Discussion

Currently, the model solely demands a patient's preoperative CT scan as input. The model is implementable for every patient with available preoperative imaging data of the total abdomen. The framework does demand some parameters to be set manually for each patient (e.g. the distance between the abdominal outer surface and fascia). When implementing the model, time that currently is needed to decide where to place the ports can be reduced significantly. Evaluation of different port placements is advantageous, and the chance of port repositioning will be reduced. It can also be used to determine usability of the number of robotic arms and estimation of intraoperative risks. This will improve surgery efficiency, counteract surgeon variability and facilitate patient specific preoperative planning.

The presented framework needs to be further elaborated and there is a demand for more research. The insufflation of carbon dioxide results in a pneumoperitoneum, which deforms the patient's body shape and organs shift with respect to static preoperative imaging.^{16,24,25} Without intraoperative data available, the deformed abdominal shape is represented by a part of an ellipsoid pointcloud. Several researchers have attempted to simulate the deformations as a result of the insufflation, like Kitasaka *et al.*²⁶, Camara *et al.*²⁷, Johnsen *et al.*²⁸ and Bano *et al.*²⁵ Bano *et al.* presents a method to predict outer surface and viscera deformations by merging preoperative imaging with intraoperative skin tracking.²⁵ This method seems practicable, as it does not depend on medical imaging in the OR.

The created framework of the model is based on the preoperative CT scan of a 20-year-old. Therefore, the figures in this research give a distorted view compared to the situation in children. Guidelines for adults were retained, such as the 6 - 8 cm distance between ports.¹⁷ Lim *et al.*¹⁷ appoints this as an appropriate distance, while Chang *et al.*⁸ reckons 3 to 4 cm as sufficient in pediatric cases. However, Kajiwar *et*

*al.*²⁹ appoints a distance of at least 8 cm sufficient to prevent collision. Mattioli *et al.*⁹ mentions that the ultimate goal for proper port placement is proper angles and not so much the distance between them. Differences found in literature demand the merge of the developed framework with the simulation of da Vinci, which enables more insight of the intraoperative range of motion and collision detection of the robot.

There was no port integrated for the bedside assistant, while intraoperatively at least one assisting port is used. As arteries, veins and nerves were not visualized, the surgeon needs to keep in mind their anatomical positions. Also, the instrumentation contains an articulating joint that intraoperatively is located in the abdominal cavity, which was not integrated in the framework. The needle driver is the shortest da Vinci instrument, with a distance of 2.71 cm from the tip to the most proximal articulating joint. This distance on top of the intra-abdominal trocar length (2.90 cm) could be an issue in a child's small abdomen. For a selection of patients, it could be beneficial to adjust the depth of the port by routinely retracting the port back. This will increase the workable domain and could also improve the maneuverability of the instrument.⁹

The problem of limited working space in the pediatric abdominal cavity has led to diversity between hospitals. At the UMCG, pediatric robotic surgery is performed in patients that are at least 2 years old and weigh > 15 kilograms (kg). Kim *et al.*¹³ and Mattioli *et al.*⁹ do not recommend robotic surgery in very young, small patients. However, Dawrant *et al.*⁵ performed successful robotic surgery in five children < 10 kg. RMIS are associated with longer operating time and significant operating costs, in addition to the expensive surgical robotic system.^{3,4,6,9,14} Also, it demands a skilled robotic surgical team for a seamless workflow.^{8,14} To avoid interference between the robotic arms and the OR table, it is recommended to elevate the smaller pediatric patients using a foam padding. This way the robotic arms are allowed a greater range of motion and are less likely to collide with the OR table.⁹

Possible port locations could be selected based on performance measurements. Sun *et al.*³⁰ proposes two types: the Global Isotropy Index (GII) and the Efficiency Index (EI), which are correlated. The GII is developed to optimize the dexterity of robots, and the EI to maximize robot reachability and visibility. Computerized simulations of the da Vinci have been deployed to find optimal solutions. Trejos *et al.*³¹ selects port locations based on a performance index obtained from outputs of an inverse kinematic problem. After repeating this process, it is expected to yield optimal port locations for an individual patient.

VI. Conclusion

The framework presented in this research is intended to form the basis of a patient specific computational model that is part of a surgical planning strategy, aiming to find optimal port locations for robotic pediatric surgery. A list of model requirements was set up and different model types were evaluated, where after an algorithm was developed in MATLAB. This algorithm creates a framework that is able to visualize the surface of a patient's torso and target organs. Various elements were added, including a hypothetical pneumoperitoneum of the skin and fascia, trocar lines, unsafe regions for trocar placement and trocars.

This way, the framework enables evaluation of different port placements in the abdominal cavity of a patient. This is especially useful in case of a pediatric patient. Pediatric patients have a small abdominal cavity, limiting the workable domain of the robotic instruments along with chances of robot arm collision, posing intraoperative risks. Evaluation of port placements with a further elaborated model gives more insight of the safety and feasibility of the procedure. Also, usability of the number of robotic arms can be determined. Implementation of the model will improve surgery efficiency, counteract surgeon variability and facilitate patient specific preoperative planning. An increase of the employability of the da Vinci robot in pediatric surgery allows the advantages of robotic surgery to be exploited.

VII. Future Recommendations

Within this research, a framework is created based on a MATLAB algorithm that is able to serve as the basis of a computational model. However, in order to implement the model, there is a need for more research and elaboration. First of all, the algorithm needs to be applied on imaging data of children, preferably multiple scans of children of different ages. Also, the effects of insufflation need to be integrated into the framework. The truthful deformation of the skin enables calculating the volume of the workspace in the abdomen. When applying this for multiple pediatric patients, age differences can be evaluated. Skin deformation data can be obtained by using a 3D camera or infrared trackers, which are available at the University of Twente Surgical Robotics Lab. The trackers as well as the camera are able to give real time feedback of the position and movement of the skin, to obtain volume and size data without radiation.

The data of the exterior body can be merged with data derived from preoperative data, the interior body. Bano *et al.*²⁵ presented a method to predict outer surface and viscera deformations by using these types of data. This represents the truthful intraoperative situation and different trocar placements can be visualized with the model to evaluate its effects. For example, in pediatric patients it could be an option to zigzag trocar entry points to obtain sufficient distance between ports. However, the need of intraoperative data takes away the idea of patient-specific preoperative planning. Furthermore, the assisting port and articulating joint of the instruments can be processed into the framework.

In addition, there is a need to add a simulation of the da Vinci model to the framework to complete the model. The Robotic Toolbox in MATLAB enables working with inverse kinematics and therefore detect collision. However, applications like V-REP and RoboDK are especially designed for realizing such simulations and could therefore

be better suitable. When merging the exterior and interior body data with robotic arms' configurations, it is possible to calculate the accessible 3D points on the abdominal surface for one robot arm. This arm could be referred to as the moving arm, which could be each desired arm. The model can be elaborated in such way that if the active arm is near the threshold of a certain angle, the other arms are passively controlled to reposition,

to prevent collision to happen. The obtained outputs of an inverse kinematic problem can be used to calculate the performance index. Nevertheless, there will be a need to test the model physically on the robot to verify results. This could be realized by using a phantom, e.g. Blue Phantom³², and a (research-based) da Vinci robot, as it enables to obtain the inverse kinematics directly from the robot.

References

1. Stoyanov, D. & Yang, G.-Z. Soft tissue deformation tracking for robotic assisted minimally invasive surgery. *Conf. Proc. ... Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE Eng. Med. Biol. Soc. Annu. Conf.* **2009**, 254–7 (2009).
2. Palep, J. H. Robotic assisted minimally invasive surgery. *J. Minim. Access Surg.* **5**, 1–7 (2009).
3. Alqahtani, A. *et al.* Robot-assisted pediatric surgery: how far can we go? *World J. Surg.* **34**, 975–8 (2010).
4. Alizai, N. K., Dawrant, M. J. & Najmaldin, A. S. Robot-assisted resection of choledochal cysts and hepaticojunostomy in children. *Pediatr. Surg. Int.* **30**, 291–4 (2014).
5. Dawrant, M. J., Najmaldin, A. S. & Alizai, N. K. Robot-assisted resection of choledochal cysts and hepaticojunostomy in children less than 10 kg. *J. Pediatr. Surg.* **45**, 2364–8 (2010).
6. Koga, H. *et al.* Comparison of robotic versus laparoscopic hepaticojunostomy for choledochal cyst in children: a first report. *Pediatr. Surg. Int.* **35**, 1421–1425 (2019).
7. Chang, E. Y., Hong, Y. J., Chang, H. K., Oh, J.-T. & Han, S. J. Lessons and tips from the experience of pediatric robotic choledochal cyst resection. *J. Laparoendosc. Adv. Surg. Tech. A* **22**, 609–14 (2012).
8. Chang, C., Steinberg, Z., Shah, A. & Gundeti, M. S. Patient positioning and port placement for robot-assisted surgery. *J. Endourol.* **28**, 631–8 (2014).
9. Mattioli, G. & Petralia, P. *Pediatric Robotic Surgery; Technical and Management Aspects.* (Springer International Publishing, 2017). doi:10.1007/978-3-319-41863-6
10. Meehan, J. J. & Sandler, A. Pediatric robotic surgery: A single-institutional review of the first 100 consecutive cases. *Surg. Endosc.* **22**, 177–82 (2008).
11. de Kleine, R., Bruggink, J., de Meijer, V. & Hulscher, J. *Leerboek Chirurgie; Aandoeningen van lever, galwegen en pancreas bij kinderen.* (2020).
12. Ten Hove, A., de Meijer, V. E., Hulscher, J. B. F. & de Kleine, R. H. J. Meta-analysis of risk of developing malignancy in congenital choledochal malformation. *Br. J. Surg.* **105**, 482–490 (2018).
13. Kim, N. Y. *et al.* Retrospective assessment of the validity of robotic surgery in comparison to open surgery for pediatric choledochal cyst. *Yonsei Med. J.* **56**, 737–43 (2015).
14. Pham, H. D. *et al.* Robotic-assisted surgery for choledochal cyst in children: early experience at Vietnam National Children's Hospital. *Pediatr. Surg. Int.* **35**, 1211–1216 (2019).
15. Jessa Ziekenhuis - Robotchirurgie Da Vinci Xi. Available at: <https://www.jessazh.be/deelwebsites/urologie/prostaat--externe-genitalia--incontinentie/prostaat/robotchirurgie-da-vinci-xi>. (Accessed: 14th August 2020)
16. Reza Maddah, M. *3D visualization and interactive image manipulation for surgical planning in robot-assisted surgery.* (2019).
17. Lim, S. K., Ng, F. C., Yam, W. L. & Rha, K. H. Modified transperitoneal ports configuration and docking technique for renal surgeries with the da Vinci Surgical System Xi. *Int. J. Urol.* **23**, 801–802 (2016).
18. Kajiwar, N. *et al.* Maximizing Use of Robot-Arm No. 3 in Da Vinci-Assisted Thoracic Surgery. *Int. Surg.* **100**, 930 (2015).
19. *Design Tools: Human Body Modelling.*
20. Gilat, A. *MATLAB: an introduction with applications.* (John Wiley, 2005).
21. Turini, G., Moglia, A., Ferrari, V., Ferrari, M. & Mosca, F. Patient-specific surgical simulator for the pre-operative planning of single-incision laparoscopic surgery with bimanual robots. *Comput. Aided Surg.* **17**, 103–112 (2012).
22. Intuitive Surgical. *Da Vinci Xi Surgical System - Technical information.* (2016).
23. Robotics System Toolbox - MATLAB & Simulink. Available at: <https://nl.mathworks.com/products/robotics.html>. (Accessed: 22nd July 2020)
24. Camara, M., Dawda, S., Mayer, E., Darzi, A. & Pratt, P. Subject-specific modelling of pneumoperitoneum: model implementation, validation and human feasibility assessment. *Int. J. Comput. Assist. Radiol. Surg.* **14**, 841–850 (2019).
25. Bano, J. *et al.* Simulation of the Abdominal Wall and Its Arteries after Pneumoperitoneum for Guidance of Port Positioning in Laparoscopic Surgery. in 1–11 (2012). doi:10.1007/978-3-642-33179-4_1

26. Kitasaka, T. *et al.* Virtual Pneumoperitoneum for Generating Virtual Laparoscopic Views Based on Volumetric Deformation. in 559–567 (2004). doi:10.1007/978-3-540-30136-3_69
27. Camara, M., Mayer, E., Darzi, A. & Pratt, P. Soft tissue deformation for surgical simulation: a position-based dynamics approach. *Int. J. Comput. Assist. Radiol. Surg.* **11**, 919–928 (2016).
28. Johnsen, S. F. *et al.* Database-Based Estimation of Liver Deformation under Pneumoperitoneum for Surgical Image-Guidance and Simulation. in 450–458 (2015). doi:10.1007/978-3-319-24571-3_54
29. Kajiwar, N. *et al.* Maximizing use of robot-arm no. 3 in daVinci-assisted thoracic surgery. *Int. Surg.* **100**, 930–3 (2015).
30. Sun, L. W. & Yeung, C. K. Port placement and pose selection of the da Vinci surgical system for collision-free intervention based on performance optimization. in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems* 1951–1956 (IEEE, 2007). doi:10.1109/IROS.2007.4399354
31. Trejos, A. L., Patel, R. V., Ross, I. & Kiaii, B. Optimizing port placement for robot-assisted minimally invasive cardiac surgery. *Int. J. Med. Robot. Comput. Assist. Surg.* **3**, 355–364 (2007).
32. Blue Phantom Ultrasound Training Medical Models and Ultrasound Simulators. Available at: <https://www.bluephantom.com/>. (Accessed: 16th August 2020)

Appendix A – MATLAB script

```

%% Load CT files
dicomlist = dir('/Users/piendegraaf/Documents/TG/M2 stage/Stage 2 Meander/Scans
Meander/20/Anonymized - 30761815/Ct Abdomen/1_Abdomen 3.0 I30f 3 - 4/*');
dicomlist = dicomlist(3:end,:);
dicomlist = rmfield(dicomlist,'folder');
dicomlist = rmfield(dicomlist,'date');
dicomlist = rmfield(dicomlist,'bytes');
dicomlist = rmfield(dicomlist,'isdir');
dicomlist = rmfield(dicomlist,'datenum');

% Create image stack I
for i = 1:length(dicomlist)
    I(:, :, i) = dicomread(['/Users/piendegraaf/Documents/TG/M2 stage/20/Anonymized/Ct
Abdomen/1_Abdomen 3.0 I30f 3 - 4/',dicomlist(i).name]);
end

%% Find voxel size and slice thickness
dinfo = dicominfo('/Users/piendegraaf/Documents/TG/M2 stage/Anonymized/Ct
Abdomen/1_Abdomen 3.0 I30f 3 - 4/IM-0001-0001-0001.dcm');
voxelsize = dinfo.PixelSpacing;
slicethick = dinfo.SliceThickness;

%% Make image stack binary and select region of interest
I = mat2gray(I);
%imshow(I);

level = 0.2; % Set threshold
for i = 1:length(dicomlist)
    slices = squeeze(I(:, :, i));
    I_bin(:, :, i) = im2bw(slices, level);
end

ROI = I_bin(:, :, 20:137); % Region of interest contains abdomen area

%% Improve binary image stack
se = strel('cube', 5);
I_bin_cl = imclose(ROI, se); % Close image

% Fill image
I_bin_cl_fill = zeros(size(I_bin_cl, 1), size(I_bin_cl, 2), size(I_bin_cl, 3));

```



```

for i = 1:size(I_bin_cl,3)
    I_bin_cl_fill(:,:,i) = imfill(I_bin_cl(:,:,i),'holes');
end

% Delete table
I_bin_cl_fill2 =
zeros(size(I_bin_cl_fill,1),size(I_bin_cl_fill,2),size(I_bin_cl_fill,3));
for i = 1:size(I_bin_cl_fill2,3)
    I_bin_cl_fill2(:,:,i) = bwareaopen(I_bin_cl_fill(:,:,i),10000);
end
I_bin_cl_fill = I_bin_cl_fill2;
%imshow(I_bin_cl_fill);

%% Find surface faces and vertices, correct for voxel size and downsample
threshold = 0.5;
[f,v] = isosurface(I_bin_cl_fill,threshold); % Extract isosurface faces and vertices
v = v.*[voxelsize(1),voxelsize(2),slicethick]; % Adjust scaling for voxelsize and
slice thickness
[F,V] = reducepatch(f,v,0.5); % Reduce number of patch faces

%% Find centroid and translate
xv = V(:,1); yv = V(:,2); zv = V(:,3);

xminv = min(V(:,1)); xmaxv = max(V(:,1));
yminv = min(V(:,2)); ymaxv = max(V(:,2));
zminv = min(V(:,3)); zmaxv = max(V(:,3));

% Translate to origin
xcent = (xmaxv - xminv)/2;
ycent = (ymaxv - yminv)/2;
zcent = (zmaxv - zminv)/2 + 50;

xv = xv - xcent;
yv = yv - ycent;
zv = zv - zcent;

V = [zv,xv,yv];

%% Segment organs from CT
I_bin_org = I >= 0.41 & I <= 0.419; % Set thresholds
ROI_org = I_bin_org(:,:,5:70); % Region of interest

% Close, fill and delete noise
se = strel('cube',5);
I_bin_cl_org = imclose(ROI_org,se);

I_bin_cl_fill_org =
zeros(size(I_bin_cl_org,1),size(I_bin_cl_org,2),size(I_bin_cl_org,3));
for i = 1:size(I_bin_cl_org,3)
    I_bin_cl_fill_org(:,:,i) = imfill(I_bin_cl_org(:,:,i),'holes');
end

organs_seg =
zeros(size(I_bin_cl_fill_org,1),size(I_bin_cl_fill_org,2),size(I_bin_cl_fill_org,3));
for i = 1:size(organs_seg,3)
    organs_seg(:,:,i) = bwareaopen(I_bin_cl_fill_org(:,:,i),1000);
end

organs_seg = smooth3(organs_seg); % Smooth data

[f_o,v_o] = isosurface(organs_seg,threshold); % Extract isosurface faces and vertices
v_o = v_o.*[voxelsize(1),voxelsize(2),slicethick]; % Adjust scaling for voxelsize and
slice thickness
[F_o,V_o] = reducepatch(f_o,v_o,0.5); % Reduce number of patch faces

```

```

x_o = V_o(:,1); y_o = V_o(:,2); z_o = V_o(:,3);

% Translate to origin
x_o = x_o - xcent;
y_o = y_o - ycent;
z_o = z_o - zcent;

V_o = [z_o,x_o,y_o];

%% Create ellipsoids as pneumoperitoneum
% Skin
[ell_x,ell_y,ell_z] = ellipsoid(0,0,0,220,190,250,100); % Create ellipsoid shape
ell_co = [ell_x(:),ell_y(:),ell_z(:)]; % Store coordinates in matrix

% Remove bottom of ellipsoid
ell_row_z = ell_co(:,3);
[ell_row,ell_column] = find(ell_row_z <= 120);
ell_co(ell_row,:) = [];

% Create pointcloud from ellipsoid coordinates, skin
ell_color_mat = [0.1, 0.5, 0.65];
ell_color_mat = repmat(ell_color_mat,size(ell_co,1),1);
ell_pc = pointCloud(ell_co,'Color',ell_color_mat);

% Fascia
layerxy = 12; % Layer skin - fascia side in mm
layerz = 26; % Layer skin - fascia height in mm

[ell_x2,ell_y2,ell_z2] = ellipsoid(0,0,0,220-layerxy,190-layerxy,250-layerz,100); %
Create ellipsoid shape
ell_co2 = [ell_x2(:),ell_y2(:),ell_z2(:)]; % Store coordinates in matrix

% Remove bottom of ellipsoid
ell_row_z2 = ell_co2(:,3);
[ell_row2,ell_column2] = find(ell_row_z2 <= 120);
ell_co2(ell_row2,:) = [];

% Create pointcloud from ellipsoid coordinates, height
ell_color_mat2 = [0, 0, 1];
ell_color_mat2 = repmat(ell_color_mat2,size(ell_co2,1),1);
ell_pc2 = pointCloud(ell_co2,'Color',ell_color_mat2);

%% Landmark coordinates
umb = [-27.5 16.35 165.7]; % Coordinate umbilicus
um_pn = [-72.18 16.01 235.2]; % Coordinate umbilicus pneumoperitoneum

tar_or = [43 -33.06 55.58]; % Coordinate target organ

umpn_taror = [um_pn;tar_or];
dist_umpn_taror = pdist(umpn_taror,'euclidean'); % Distance target organ to umbilicus
pneumoperitoneum

%% Select area around landmark
dist_um_ell = pdist2(um_pn,ell_co,'euclidean'); % Distances from landmark umbilicus
dist_um_ell = dist_um_ell';

max_dist_lm = 60; % Set maximum distance from landmark umbilicus

[distum_row,distum_column] = find(dist_um_ell <= max_dist_lm); % Find points within
max distance

um_pn_maxdist = ell_co(distum_row,:); % Store points within max distance in matrix

% Pointcloud points within max distance in red
dist_color_mat = [1,0,0];

```

```

dist_color_mat = repmat(dist_color_mat,size(distum_row,1),1);
dist_pc = pointCloud(um_pn_maxdist,'Color',dist_color_mat);

%% Select area < 15 cm from pneumoperitoneum
dist_to_ell = pdist2(tar_or,ell_co,'euclidean'); % Distances target organ to ellipsoid
dist_to_ell = dist_to_ell';

max_dist_to = 150; % Set maximum distance from landmark umbilicus

[distto_row,distto_column] = find(dist_to_ell <= max_dist_to); % Find points within
max distance

to_pn_maxdist = ell_co(distto_row,:); % Store points within max distance in matrix

% Pointcloud points within max distance in pink
color_mat_to = [1,0,1];
color_mat_to = repmat(color_mat_to,size(distto_row,1),1);
pc_toell = pointCloud(to_pn_maxdist,'Color',color_mat_to);

%% Create trocar lines
line_length = 592; % Set instrument length
factor = (line_length/dist_umpn_taror) - 1; % Factor to extend trocar line length
d_line1 = diff(umpn_taror);

line1 = -factor*d_line1+um_pn; % Extended trocar line
umpn_taror2 = [line1;um_pn;tar_or];

%% Add trocar line 2
entry2 = [-122.9 -58.35 192.6];
entry2_taror = [entry2;tar_or];
dist_entry2_taror = pdist(entry2_taror,'euclidean');

dist_line2 = pdist2(entry2,ell_co,'euclidean');
dist_line2 = dist_line2';

[dist_row_line2,dist_column_line2] = find(dist_line2 <= max_dist_lm);

line2_maxdist = ell_co(dist_row_line2,:);

color_mat_line2 = [1,0,0];
color_mat_line2 = repmat(color_mat_line2,size(dist_row_line2,1),1);
pc_line2 = pointCloud(line2_maxdist,'Color',color_mat_line2);

factor_line2 = (line_length/dist_entry2_taror) - 1;
d_line2 = diff(entry2_taror);

line2 = -factor_line2*d_line2+entry2;
line2_taror2 = [line2;entry2;tar_or];

%% Add trocar line 3
entry3 = [-24.84 83.55 222.8];
entry3_taror = [entry3;tar_or];
dist_entry3_taror = pdist(entry3_taror,'euclidean');

dist_line3 = pdist2(entry3,ell_co,'euclidean');
dist_line3 = dist_line3';

[dist_row_line3,dist_column_line3] = find(dist_line3 <= max_dist_lm);

line3_maxdist = ell_co(dist_row_line3,:);

color_mat_line3 = [1,0,0];
color_mat_line3 = repmat(color_mat_line3,size(dist_row_line3,1),1);
pc_line3 = pointCloud(line3_maxdist,'Color',color_mat_line3);

```

```

factor_line3 = (line_length/dist_entry3_taror) - 1;
d_line3 = diff(entry3_taror);

line3 = -factor_line3*d_line3+entry3;
line3_taror2 = [line3;entry3;tar_or];

%% Add trocar line 4
entry4 = [-41.67 -110.8 197.5];
entry4_taror = [entry4;tar_or];
dist_entry4_taror = pdist(entry4_taror,'euclidean');

dist_line4 = pdist2(entry4,ell_co,'euclidean');
dist_line4 = dist_line4';

[dist_row_line4,dist_column_line4] = find(dist_line4 <= max_dist_lm);

line4_maxdist = ell_co(dist_row_line4,:);

color_mat_troc4 = [1,0,0];
color_mat_troc4 = repmat(color_mat_troc4,size(dist_row_line4,1),1);
pc_line4 = pointCloud(line4_maxdist,'Color',color_mat_troc4);

factor_line4 = (line_length/dist_entry4_taror) - 1;
d_line4 = diff(entry4_taror);

line4 = -factor_line4*d_line4+entry4;
line4_taror2 = [line4;entry4;tar_or];

%% Apply skin2fascia function
co_f_umpn = skin2fascia(um_pn,tar_or,ell_co2);
co_f_entry2 = skin2fascia(entry2,tar_or,ell_co2);
co_f_entry3 = skin2fascia(entry3,tar_or,ell_co2);
co_f_entry4 = skin2fascia(entry4,tar_or,ell_co2);

%% Properties for make_trocar function
troc_length = 80; % Total trocar length in mm
troc_length_in = 29; % Length trocar inside abdominal cavity = 29 mm
ratio_inside = troc_length_in/troc_length; % Ratio trocar length inside abdominal
cavity
ratio_outside = 1 - ratio_inside;

%% Visualization
figure;
hold on;

body = patch('Faces',F,'Vertices',V); % Segmented patients' body
set(body,'FaceColor',[0.1, 0.5, 0.65],'EdgeColor','none');
set(gca,'YDir','reverse')
direction = [0 1 0];
rotate(body,direction,180);
alpha(body,0.4);

organs = patch('Faces',F_o,'Vertices',V_o); % Segmented organs
set(organs,'FaceColor',[0, 1, 0],'EdgeColor','none');
rotate(organs,direction,180);
alpha(organs,0.3);

pcshow(ell_pc,'MarkerSize',15); % Pneumoperitoneum skin
pcshow(ell_pc2,'MarkerSize',20); % Pneumoperitoneum fascia

plot3(umpn_taror2(:,1),umpn_taror2(:,2),umpn_taror2(:,3),'*-r','linewidth',0.8); %
Trocar line 1
plot3(co_f_umpn(1),co_f_umpn(2),co_f_umpn(3),'*c','linewidth',2.5); % Pivot point on
fascia

```

```

pcshow(dist_pc,'MarkerSize',15); % Distance around trocar1 in red
k_entry1 = boundary(um_pn_maxdist);
trisurf(k_entry1,um_pn_maxdist(:,1),um_pn_maxdist(:,2),um_pn_maxdist(:,3),'Facecolor','red','FaceAlpha',0.1);

pcshow(pc_toell,'MarkerSize',15); % Distance < 15 cm from target organ in pink
k_taror = boundary(to_pn_maxdist);
trisurf(k_taror,to_pn_maxdist(:,1),to_pn_maxdist(:,2),to_pn_maxdist(:,3),'Facecolor','magenta','FaceAlpha',0.3);

plot3(line2_taror2(:,1), line2_taror2(:,2), line2_taror2(:,3),'*-r','linewidth',0.8);
% Trocar line 2
plot3(co_f_entry2(1),co_f_entry2(2),co_f_entry2(3),'*c','linewidth',2.5); % Pivot point on fascia

pcshow(pc_line2,'MarkerSize',15); % Distance around trocar2 in red
k_entry2 = boundary(line2_maxdist);
trisurf(k_entry2,line2_maxdist(:,1),line2_maxdist(:,2),line2_maxdist(:,3),'Facecolor','red','FaceAlpha',0.1);

plot3(line3_taror2(:,1), line3_taror2(:,2), line3_taror2(:,3),'*-r','linewidth',0.8);
% Trocar line 3
plot3(co_f_entry3(1),co_f_entry3(2),co_f_entry3(3),'*c','linewidth',2.5); % Pivot point on fascia

pcshow(pc_line3,'MarkerSize',15); % Distance around trocar3 in red
k_entry3 = boundary(line3_maxdist);
trisurf(k_entry3,line3_maxdist(:,1),line3_maxdist(:,2),line3_maxdist(:,3),'Facecolor','red','FaceAlpha',0.1);

plot3(line4_taror2(:,1), line4_taror2(:,2), line4_taror2(:,3),'*-r','linewidth',0.8) % Trocar line 4
plot3(co_f_entry4(1),co_f_entry4(2),co_f_entry4(3),'*c','linewidth',2.5); % Pivot point on fascia

pcshow(pc_line4,'MarkerSize',15); % Distance around trocar4 in red
k_entry4 = boundary(line4_maxdist);
trisurf(k_entry4,line4_maxdist(:,1),line4_maxdist(:,2),line4_maxdist(:,3),'Facecolor','red','FaceAlpha',0.1);

% Plot trocars
make_trocar(co_f_umpn,tar_or,troc_length,ratio_outside,ratio_inside);
make_trocar(co_f_entry2,tar_or,troc_length,ratio_outside,ratio_inside);
make_trocar(co_f_entry3,tar_or,troc_length,ratio_outside,ratio_inside);
make_trocar(co_f_entry4,tar_or,troc_length,ratio_outside,ratio_inside);

view(3);
camlight('headlight');
daspect([1 1 1]);
xlabel X-axis; ylabel Y-axis; zlabel Z-axis;

hold off;

```

Appendix B – Function *skin2fascia*

```
%% Find fascia coordinate
function co_fascia = skin2fascia(co_skin,co_tar_or,ell_fascia)

deriv = (co_skin - co_tar_or)/sqrt(sum((co_skin - co_tar_or).^2)); % Find derivative

for i = 1:100
    co_skin_new = co_skin - (i*deriv); % Create new point with derivative
    closest_mat = ell_fascia - co_skin_new; % Find x y z distance to ell for closest
    point
    closest_mat_l = sqrt(sum(closest_mat.^2,2)); % Calculate lengths to closest point
    [val,row] = min(closest_mat_l); % Find closest point and corresponding length

    % Save in vec
    row_vec(i) = row;
    val_vec(i) = min(closest_mat_l);
end

% figure; plot(val_vec);

% Minimal distance to the line
[~,min_row] = min(val_vec);
min_len = sqrt(sum((min_row*deriv).^2));
% Total line length
tot_len = sqrt(sum((co_skin - co_tar_or).^2));

new_len = tot_len - min_len; % New length

% Fascia coordinate
co_fascia = co_tar_or + (new_len*deriv);
```

Appendix C – Function *make_trocar*

```
%% Create trocar
function make_trocar(co_fascia_point,co_tar_or,length_trocar,ratio_out,ratio_in)

deriv = (co_fascia_point - co_tar_or)/sqrt(sum((co_fascia_point - co_tar_or).^2)); %
Find derivative

% Include trocar direction ratios
upper_bound = co_fascia_point + ((ratio_out*length_trocar)*deriv);
inner_bound = co_fascia_point - ((ratio_in*length_trocar)*deriv);

% Apply cylinder algorithm
[X, Y, Z] = cylinder2P(15,100,upper_bound,inner_bound);
hold on
surf(X, Y, Z);
```


Appendix D – Alternative target organs

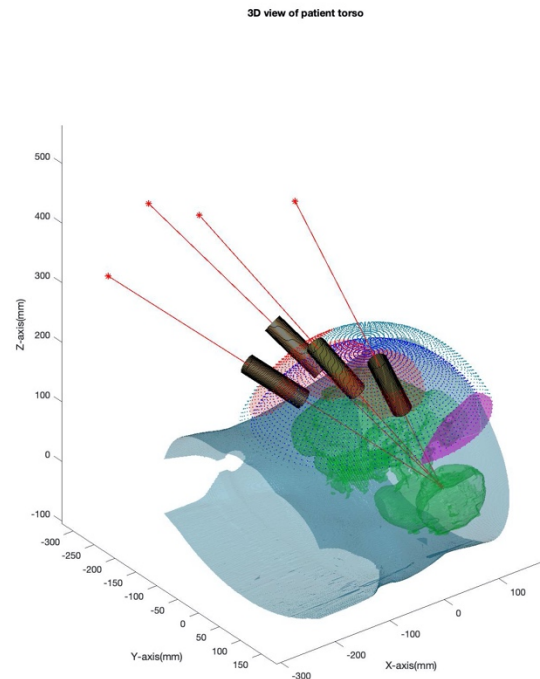


Figure 12: The target organ coordinate is set on the spleen. Changing the target causes visible differences in the variables, such as the trocar lines alignment and the position of the pink region.

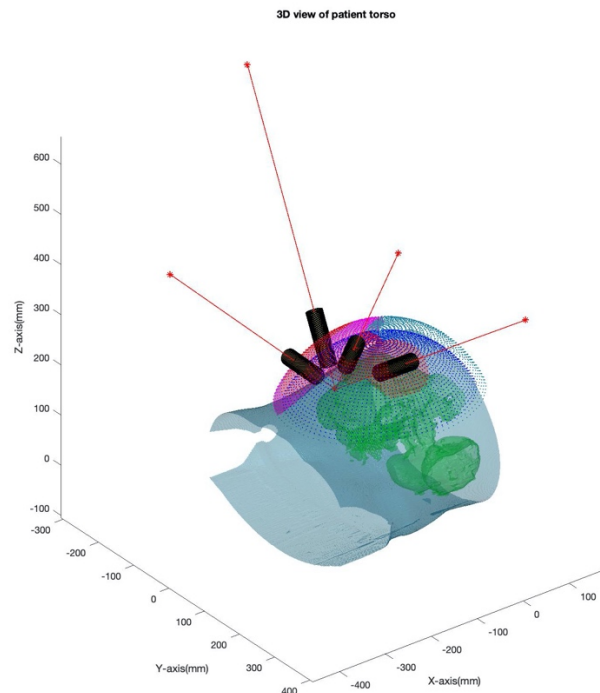


Figure 13: The target organ coordinate is set on the top of the liver. Changing the target causes visible differences in the variables, such as the trocar lines alignment and the position of the pink region. In this case, trocar entry points overlap with the pink region, which is not recommended when placing the trocars.