An easy-to-use 2D-3D registration process

3TU.School for Technological Design

STAN ACKERMANS INSTITUTE

An Easy-to-use 2D-3D Registration Process

Tudor Mihordea

September 2012

TU Delft
Delft University of Technology

TU/e Technische Universiteit Eindhoven University of Technology

UNIVERSITY OF TWENTE.

An Easy-to-use 2D-3D Registration Process

Tudor Mihordea
September 2012

An Easy-to-use 2D-3D Registration Process

Eindhoven University of Technology
Stan Ackermans Institute / Software Technology

Partners

Philips Healthcare | Eindhoven University of Technology

Steering Group T. Elenbaas
A. Jalba

Date September 2012

Contact Address: Eindhoven University of Technology Dept. of Mathematics and Computer Science
Postal address: P.O. Box 513, 5600 MB Eindhoven, The Netherlands
Visiting address: Den Dolech 2, 5612 AZ Eindhoven, The Netherlands
Internal address: MetaForum, room 7.090 Tel.: +31-(0)40-2474334

Published by: Eindhoven University of Technology
Stan Ackermans Institute

Printed by: Eindhoven University of Technology
UniversiteitsDrukkerij

ISBN: 978-90-444-1164-5

Abstract: VesselNavigator is a clinical application designed to offer physicians assistance during the minimal invasive procedure with the main focus on the endovascular repair of the abdominal aortic aneurysm (EVAR). The application provides a 3D overlay of the vessel structure on top of the live X-ray images to help physicians orientate and easily navigate through the blood vessels. This means that correctly aligning the 3D volume with the X-ray images (called registration) is vital in using this application.
This report presents the results of the project to improve the usability of the manual 2D-3D registration step in the VesselNavigator application, with the aim of improving registration accuracy and reducing registration time. In the registration step the pre-acquired 3D volume, often a CT-scan, is manually registered to the position of the patient on the operating table. This is done by matching the position of the volume with two X-ray images acquired from different angles. Improvements were added to help physicians translate and rotate 3D volumes in a more intuitive manner. Additionally, the user interface was enhanced to streamline the registration process and guide the physician through the steps needed for successful registration.

Keywords: Image registration, Multimodality, Endovascular Aortic Repair, Arcball, Philips Healthcare, 3D-overlay, 2D-3D Registration,

Preferred reference: T. Mihordea, An Easy-to-use 2D-3D Registration Process. Eindhoven University of Technology, SAI Technical Report, September, 2012.
A catalogue record is available from the Eindhoven University of Technology Library (ISBN: 978-90-444-1164-5)

Partnership: This project was supported by Eindhoven University of Technology and Philips Healthcare.

Foreword

A key element in creating new medical tools for physicians is the combination of multiple types of patient imaging techniques. For this we, have a whole range of tools to our disposal: X-ray, CT, MRI, echo, SPECT, PET, and many more. With VesselNavigator, we are designing an application that will use these visualizations in so-called minimally invasive procedures.

In the past, a patient with an abdominal or thoracic aortic aneurysm would have had major surgery where he or she would have been to be opened. Nowadays, the same patient will only receive a small incision, through which tools, such as guide-wires and stent-grafts, are inserted. The physician watches the live X-ray images on a monitor while he manipulates his intra-vascular tools.

There is a problem with using X-ray alone: while physicians can easily see the stent graft and guide-wire move in the live, 2-dimensional X-ray images, the vessel itself has very little contrast and is difficult to make out.

The CT scan does not have this drawback: it shows the vascular structure and does so in 3D. This is why this imaging technique modality is almost always used during diagnosis of vascular diseases. However, it has a different drawback: the 3D-image is static and therefore not usable during a procedure.

The idea behind the product we are building is the following: We will overlay the CT scan on top of the live X-ray images, such that the physician can see his tools but also know where they are located in vessels of the patient. We only have to fuse the two different datasets.

Initially this may seem like an easy, almost trivial problem: just align the two different modalities to the same position and orientation. However, the more we look at it, the clearer it becomes that this is in reality a complex problem: how do we quickly find and match features in the CT dataset to features in X-ray? What are good viewing angles for doing this alignment?

This kind of issues makes it difficult for the physician to perform a good registration, and having a tool that is very easy to use and guides the user through the necessary steps is very important. This is where Tudor came in. He made the registration in VesselNavigator as easy to use as possible, by creating interaction tools that make registration easy and implementing hints for the user.

It is the fate of anybody working on ease-of-use that if you get it right, hardly anybody will notice, the user will just start using the application. Only when you get it wrong, the user notices. This is why we are doing a thorough evaluation of the usability using tailored questionnaires.

Tudor can be proud that his work is not only being used and evaluated in multiple hospitals, but also that from the feedback we get, it is clear that Tudor has gotten it right!

Thijs Elenbaas
September 9, 2012
Philips Healthcare

Preface

This report describes the work carried out during the project representing the final part of the Professional Doctorate in Engineering (PDEng) degree program in Software Technology provided by the Eindhoven University of Technology and Stan Ackermans Institute. The project name is “An Easy-to-use 2D-3D Registration Process”.

This project consists of the design and development of an easy-to-use interface for image registration for the VesselNavigator prototype developed in Philips Healthcare.

The report is addressed to a technical audience that has a general knowledge about software design and medical imaging. Readers that are interested in the clinical background and the goals of the project should refer to Chapters 2-5. Details about the software architecture design and implementation are provided in Chapters 6-8. For the result of the project and the software process used readers must address Chapters 9 and 10.

T.I. Mihordea
October 1, 2012

Acknowledgements

Many people have contributed to the success of this nine months project. I would like to thank all of them for their support, encouragement and insights provided.

First of all I would like to thank my company supervisor, Thijs Elenbaas, for offering me the opportunity to do my final project at Philips Healthcare. I would like to express my gratitude for the guidelines, feedback and useful discussions he provided throughout the entire project.

I want to thank Andrei Jalba, my university supervisor for his helpful comments during the Project Steering Group Meetings and the reviews of this report.

Many thanks to all the people working in Philips Health care who offered me help and made my work here pleasant. I would like to thank Kirsten Zuurmond and Thijs Grunhagen for providing me with valuable insights for the clinical domain. I also want to express my gratitude to Ina Klein Teeselink for the constant feedback on the UI design. Special thanks go out to Marco Verstege whose final project (1), conducted here in the previous year, offered me a good starting point.

I want to thank my OOTI colleagues for the exciting last two years we shared. I would like to thank Ad Aerts, Maggy de Wert and out trainers for their continuous support and wise lessons throughout the whole OOTI period.

Finally, all these would not been possible without the support of my family who offered me the encouragement and motivation to pursue my studies.

Thank you all!!

T.I. Mihordea
October 1, 2012

Executive Summary

Vascular diseases are one of the leading causes of death in the developed world. Treating a vascular disease can be done in two different ways: using open surgery or using a less invasive endovascular approach. In an endovascular procedure the affected area is reached from inside of the vessel using catheters. The catheters are inserted into the blood vessel through a small incision into an artery or vein and guided to the area in need of treatment using live X-ray guidance. Blood and vessels have almost the same radio opacity as the surrounding soft tissue therefore they cannot be distinguished using X-ray images. To make them visible contrast agent must be injected into the blood stream. Both X-rays and contrast agent are harmful to the human body therefore their dose used during surgery must be reduced.

VesselNavigator is a clinical application designed to offer surgeons assistance while performing endovascular procedures such as abdominal aortic aneurysm (AAA). The main goal of the application is to help physicians navigate their instruments through the blood vessels using a small quantity of contrast agent. This can be achieved by overlaying a pre acquired 3D vessel structure of the patient on top of the live X-ray stream. The vessel structure can be extracted from a CT angiography or a contrast enhanced CT-scan done before the surgery. This 3D overlay allows the surgeon to see the contours of the blood vessels without constantly injecting contrast agent.

To correctly overlay the 3D vessel structure, its position and orientation with respect to the patient on the operating table must be known. This information can be calculated in the beginning of the surgery by aligning the volume with two live X-ray images. This 2D-3D alignment, also called registration, was the focus of the current project.

VesselNavigator application is intended to be used not only by interventional radiologists but also by vascular surgeons. These surgeons usually have less experience with imaging techniques compared to radiologists. For this reason the application must provide an easy to use registration step. This can be achieved by providing intuitive tools to interact with the 3D volume and clear visual feedback.

The main result of the current project is the increased usability of the registration step in the VesselNavigator application. Using an orthogonal interactor, a specific landmark can be aligned now using only two translations. Furthermore the Arcball interactor was introduced to offer users an intuitive way of rotating the 3D volume using the mouse. Besides handling of the volume itself, user interface improvements were added to streamline the registration process and guide the physician through the steps needed for successful registration. All these led to improving the manual registration accuracy and reducing the time needed for this process.

Table of Contents

List of Figures

1.Introduction

Abstract – This chapter provides the context for the current project and an outline describing how this report is structured.

1.1 Context

Philips Company was founded in 1891 by Gerard Philips initially as a family business oriented on light bulb production. Nowadays the company became one of the biggest electronics companies in the world and it focuses on three main areas: consumer electronics, smart lightning and healthcare. Philips Healthcare, formerly Philips Medical Systems, is one of the leading three companies in the world to offer medical devices.

Imaging systems is one of the directions of Philips Healthcare providing technical solutions. Different technologies such as X-rays, Computed Tomography (CT), Magnetic Resonance (MR) or Ultrasound (US) can be used to offer different visualization of the inside of the human body. Along with the imaging systems, integrated software solutions are provided to process the obtained images and offer assistance to physicians during the diagnostic and treatment phases.

Interventional X-ray (iXR) business unit of Philips Healthcare is focused on innovative software application that can be used in combination with an interventional X-ray machine such as the Philips Allura Xper FD20 presented in Figure 1. This machine is designed to be used in a hybrid operation room and provide real-time X-ray images of the patient during minimal invasive procedures such as an endovascular procedure.

Figure 1 – Image of the Allura Xper FD20 system. The C-arm with the X-ray source and the detector is on the left and in the upper right part the monitor displaying the acquired images is placed. The system also provides a table on which the patient lies during the procedure.

Vascular diseases affect a high number of people and are the cause of 1-3% of all deaths among men aged 65—85 years in developed countries(2). Besides open surgery, which was the only option three decades ago, nowadays minimal invasive endovascular approaches are available in most cases. In endovascular procedures the blood vessel that requires treatment is access with a catheter. The catheter is inserted into the body through a small puncture in an artery or vein and guided through the vessel, to the area in need of treatment, using live X-ray guidance.

Abdominal aortic aneurysm, a dilatation of the lower part of the aorta, is one of the common vascular diseases that can be treated using the endovascular approach. Treating this condition requires placing a stent graft into the affected part of the aorta. The blood flow will go through the stent and will not pressure the aneurysm walls eliminating the risk of rupture. The main problem in the endovascular treatment is that the blood and vessels have the same radio opacity as the surrounding soft tissue therefore they cannot be seen in X-ray images. To make them visible a contrast agent needs to be injected into the blood stream.

Both X-ray radiation and contrast agent are harmful to the human body therefore their dose used during an interventional procedure must be reduced. Software applications can be used to help surgeons orientate and navigate their instruments through the blood vessels using a small quantity of contrast agent. VesselNavigator is one of these applications. It proposes the overlay of a pre acquired 3D vessel structure on top of the live X-ray streams. This way the surgeon can see the contour of the vessels without the need of contrast injections and independent of the orientation of the X-ray arm.

In order to be able to overlay the 3D vessel structure on the live X-rays, the software needs to know its exact position and orientation with respect to the patient on the operating table. This can be done by aligning the volume with live X-ray images obtained during the procedure. This 2D-3D alignment, also called registration, is the focus of the current project.

The registration of the 3D vessel structure with live X-ray images is the first step that the surgeons must do during the procedure in order to be able to use the 3D information for overlay. For this reason, the duration of this process is directly added to the surgery time. Long surgery is, in general, associated with higher complication risks therefore a short registration process is required.

A completely automated registration process is hard to obtain due to the dynamic nature of the human body. Vessels are elastic structure and their shapes are not fixed. Furthermore during endovascular procedures the instruments used can also stretch or bend the vessels changing their shape. For this reason the first priority of the project is to offer surgeons the option to manually perform the registration. Manual registration can be later use as an initial approximation for automatic algorithms.

To manually register a 3D volume the physicians need to be able to rotate it and translate it in an intuitive way. Therefore the main goal of the project is to offer surgeons easy-to use tools to achieve the manual registration. Furthermore visual enhancements of the user interface can also help streamline the registration process.

1.2 Outline

Chapter 2 (Domain analysis) provides a short introduction to the concepts and terminology used in the rest of the report. First a short summary of the imaging modalities used in the medial domain is provided. Next an introduction to the image registration techniques is offered. In the end a more detailed description of the clinical domain of the VesselNavigator is presented.

Chapter 3 (Stakeholder analysis) provides a description of the stakeholders of the project and their goals.

Chapter 4 (Problem analysis) provides a detailed descriptions of the clinical problem the VesselNavigator application and this project in particular aims to offer a solution to.

Chapter 5 (System requirements) translates the user goals derived in the previous chapter into detailed software requirements.

Chapter 6 (System architecture) offers a high level description of the VesselNavigator application. It is intended to provide the software context for the current project.

Chapter 7 (System design) present the detailed design of some relevant parts of the project.

Chapter 8 (Implementation) provides a detailed description of the newly introduced feature.

Chapter 9 (Results) presents the results of the project.

Chapter 10 (Project Management) describes the process followed throughout the project. Furthermore the detailed planning and milestones are presented.

2.Domain analysis

Abstract – This chapter introduces some concepts needed to understand the clinical problem of the project. First a short summary of the imaging modalities relevant to the project is presented. Then we will give a short introduction into the endovascular procedures in order to help understand the motivation underpinning the design decisions as described in the following chapter. Finally a description of VesselNavigator application if given in section 2.4 to offer some more context to the project.

2.1 Medical imaging modalities

For diagnostic and treatment purposes physicians need to understand what happens inside the body of the patient. Traditionally, doctors rely on the external symptoms to draw a conclusion about certain medical conditions. Improvements in technology in the past century have offered the possibility to "see" inside of the patient in a non-invasive way (3). The Introduction of these advanced imaging techniques improved the quality of medical care allowing physicians to make more precise diagnoses and measurements with reduced impact on the patient. Some imaging techniques can be used in an operating room to provide live images of the patient making them a useful interventional tool. Other modalities have constrains that restrict their usage to diagnostic stages.

2.1.1. X-rays

The history of medical imaging starts with the discovery of X-rays. In 1895 Wilhelm Conrad Röntgen discovered this new type of radiation while experimenting with vacuum tubes (4). The new rays were able to pass through most type of materials and following studies quickly pointed out the utility for medial usage. Nowadays X-ray radiation has multiple uses both for diagnostic and during interventional procedures.

The first usage of X-rays for medical purposes was the creation of 2D radiographs. For this a patient is placed between an X-ray source and a detector. Radiation is generated by the source and sent through the body towards the detector. The rays are attenuated by the tissues it passes through. As a result the intensity of the radiation reaching the detector is dependent on the type of tissue it traverses. Because different tissues have different radiation absorption properties, visualizing the intensities of radiation on the detector provides a 2D superimposed projection of the inside of the body. The fact that bones attenuate significantly more rays than the soft tissues makes this modality very useful for diagnosing bone fractures.

2.1.2. Computed Tomography (CT)

For some applications the 2D information offered by the radiography is not enough. In 1972 G. N. Houndsfield proposed a method to overcome this limitation. He made use of computer processed X-ray images to produce a 3D representation of specific areas of the body. For his work on the new modality called computed tomography (CT) Hounds was awarded the Nobel Prize in 1979 (5)

The underlying mathematical principles of computed tomography were developed by J. Radon in 1917 (6). He showed that a function can be reconstructed from an infinite number of its projections. CT scanners make rotation scans to generate X-ray projections from all around the body. From these tomographic slices reconstruction algorithms can build a 3D representation using the Radon transform. Figure 2 offers a comparison of the CT and X-ray modalities.

CT scanners provide high details of the body and are used for diagnosing a large variety of conditions but they cannot be used during the surgery due to their construction (patient has to lie in a tube-shaped machine during the scan). Nowadays Philips provides the possibility to acquire CT-like images using an interventional X-ray machine (Figure 1) during the surgical procedure. This is achieved with a rotational scan made with the X-ray arc.

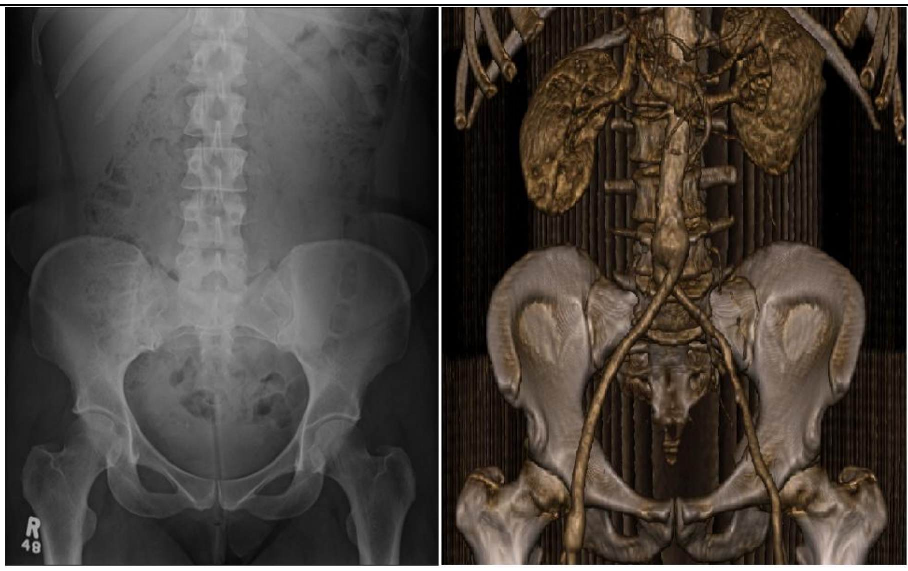


Figure 2 - Representation of abdomen anatomy using X-ray radiography (left) and a CT reconstructed volume with contrast filling (right)

2.1.3. Angiography

Angiography is a technique used to visualize the vasculature of the body. When using X-rays the vessels are not well distinguishable from the surrounding soft-tissue. One way to make the vessel stand out is to inject a radio-opaque material, also called dye, into the blood stream. This liquid will propagate through the vessel with the primary blood flow. Now the contour (and the inside) of the artery/vein can be seen using X-rays. Angiograms are widely used to diagnose vascular problems such as occlusions or aneurysms.

2.1.4. Magnetic Resonance Imaging (MRI)

Magnetic Resonance is an imaging technique that does not make use of X-ray radiation. Instead, MRI is based on the nuclear magnetic resonance principle. It uses powerful magnets to align the hydrogen nuclei in the body (7). When the magnetic field is removed, the atoms return to their initial alignment and during this process they emit energy, acting like a small radio transmitters. Different tissues react differently to this stimulus: in soft tissue the realignment time is shorter when compared to dense tissue. Specialized detectors can measure these differences after which a 3D reconstruction algorithm will generate a 3D volume.

MRI offers the best soft tissue contrast from all the non-invasive imaging techniques and it is considered safer compared to CT because it does not use X-ray radiation. On the other hand, MRI introduces some disadvantages. As with the CT scans the patient has to be almost completely enclosed by the MR machine making this modality unusable for interventional procedures. In addition the usage of large magnets makes MRI usage impossible when metals are presented. For example it cannot be used to scan patients using pacemakers or when additional instrumentation is needed. Moreover, compared to a CT scan, a MRI exam takes significantly more time and provides a lower spatial resolution.

2.1.5. Ultrasound Imaging

Ultrasound imaging is a diagnostic imaging technique making use of sound waves with frequencies higher than those of the audible spectrum. The principle is similar to the one used in constructing radar and sonar equipment. Ultrasound is sent into the patient's body and the delay of the reflected waves (echo) is measured. Based on this, images of the zone of interest can be computed. One of the advantages of this technique is that ultrasound imaging involves negligible risk. As a result it is the only modality used in visualizing the fetus during pregnancy. Because it uses low energy waves the needed equipment is more compact and significantly cheaper than many other imaging techniques. Nevertheless a disadvantage of the procedure is the fact that due to the high wavelength, the resolution capabilities are reduced and the use in advanced procedures is therefore limited.

2.2 Image registration techniques

The previous section described some of the most common imaging techniques used in the medical domain today. Each of these modalities will offer a different representation of the anatomy (2D, 3D, static, dynamic with different contrast and spatial resolution). Since each modality has its own advantages and disadvantages, it may be advantageous to combine different forms of imaging. This raises the necessity to accurately align the different images in space, a process called image registration. Registration represents the process of finding a transformation that will map the points from one of the inputs (the floating image) to corresponding points in the other input (the fixed image). A detailed classification of the registration techniques can be found in (8).

In a medical application the same anatomical feature can look different when different modalities are used. Differences in tissue seen in one modality could not be distinguished in other modality. For X-ray imaging techniques the dose the patient can receive should be limited as much as possible (details in section 4.2.1.). The lowest dose of radiation that can visualize the needed features is used reducing the image quality. For these reasons a perfect registration is usually hard to achieve in the medical domain.

The dynamic nature of the human body adds to the complexity of the registration. The same anatomic region can look different depending on the posture of the patient during imaging. For example, the CT scans are often done with the arms positioned along the head, whereas during X-ray imaging procedure the arms are placed along the body. Heart beat and breathing add additional deformation. Based on the amount of deformation and the precision needed different transformation models can be used.

Rigid transformation is a class of spatial transformation that only consists of a rotation and a translation. Approximating the registration by a rigid transformation offers the advantage of having to compute relatively few parameters (3 in the case of 2D and 6 in the case of 3D registration). This approach is preferred when the amount of deformation is small.

When dealing with significant deformation, a rigid transformation might not be enough to describe the registration. If high accuracy is required more complex models that take deformation into account are needed. This method called elastic registration is more precise but also harder to use due to the large number of parameters involved.

The registration process is dependent of the spatial dimensionality of the input images different. If the input images have the same dimensionality (2D-2D or 3D-3D) the registration process can directly spatially align the two datasets. When the input images have different dimensionality (the case of 2D-3D registration) a direct mapping is not obvious. Different spatial dimensionality means that one of the inputs has less information than the other so the registration cannot be completely done.

For example when dealing with 2D-3D registration the 2D input can be seen as a projection (in case of radiographs it's a superimposed projection, a "shadow") of the 3D input on a specific plane. In this case aligning the 3D dataset with the 2D image does not yield a unique position in space. If we move the volume orthogonal to the 2D image the alignment is preserved.

In software products, registration can be classified based on the interaction required by the users:

- Manual: In manual registration the user is responsible for finding the matching between the images. The software will provide only the tool to manipulate the two datasets in order to find this matching.

- Automatic: In automatic registration software algorithms can calculate the registration without the user intervention.

- Semi-automatic: In semi-automatic registration user interaction is required to provide a starting point for the registration process that will then be refined by an automatic algorithm.

The modality of the input also has a great impact on the registration algorithm. In different modalities images of the same anatomic feature might be presented differently. For example when trying to register X-ray on a MRI scan, using bony landmarks, it may turn out that bones are completely invisible in the specific acquisition type of the MRI. Because of this, some automatic algorithms that work for intra-modality images might require substantial modification to be used in inter-modality use-cases.

2.3 Endovascular procedures

Endovascular procedures are a type of minimally invasive surgeries aimed at treating vascular diseases. In this type of surgery the affected area to be treated is accessed via one of the body's major blood vessel (for example the femoral arteries can serve as entry point). The entire surgery is performed intra-vascular using specialized tools such, as catheters and guide wires. These devices are introduced in an artery via a small puncture and guided through the vessel using live X-ray images. The listing below presents some of the most common cases when endovascular approach is used:

- Angioplasty is used to treat an obstruction in a blood vessel generated by the accumulation of plaque on the vessel wall. Here a balloon is guided to the narrowed area where it is inflated to reopen the vessel.

- Stenting is used to place an artificial mesh tube inside of the blood vessel for treating an aneurysm.

2.3.1. Aortic Abdominal Aneurysm (AAA)

One of the more common endovascular procedures is treatment of an aortic abdominal aneurysm. This procedure, called endovascular aneurysm repair (EVAR), is one of the main clinical applications of the VesselNavigator project. To understand the reasoning behind some of the design decisions described later in this document, a short description of the clinical domain is given below.

An aneurysm is a dilatation of a blood vessel caused by a weakened vessel wall. Although the presence of an aneurysm typically does not influence the wellbeing of a patient, there is a risk that the vessel will rupture. Depending on the vessel in question, a rupture can be a life threatening situation.

The aorta is the largest artery in the human body. It originates in the left ventricle of the heart and extends down through the chest and the abdomen where it bifurcates into the common iliac arteries. The aorta carries the oxygenated blood to the entire body. Figure 3 illustrates the parts of the aorta and its main branches.

Most of the aortic aneurysms are located in the abdominal aorta (9), below the renal arteries. Almost two-thirds of them also extend down into the iliac arteries. The estimated number of 1.1 million individuals aged 50 to 84 suffer from AAA (10). Aortic aneurysms are hard to detect since in general they are asymptomatic and are usually discovered when CT-scan or ultrasound investigations are done for other reasons (11). Screening programs are initiated in some countries for early detection of the AAA (12).

The causes of aortic aneurysms are unclear but some risk factors have been related to this condition. The main factors are: (13)

- Smoking: Smokers are seven times more likely to develop an aortic aneurysm.
- Arthrosclerosis: The hardening of the artery walls due cholesterol has been linked to aortic aneurysm.
- High blood pressure: Pressure on the aorta walls weakens them and facilitates the development of aneurysms.
- Age: Most aortic aneurysms are detected in persons over 60 years old.

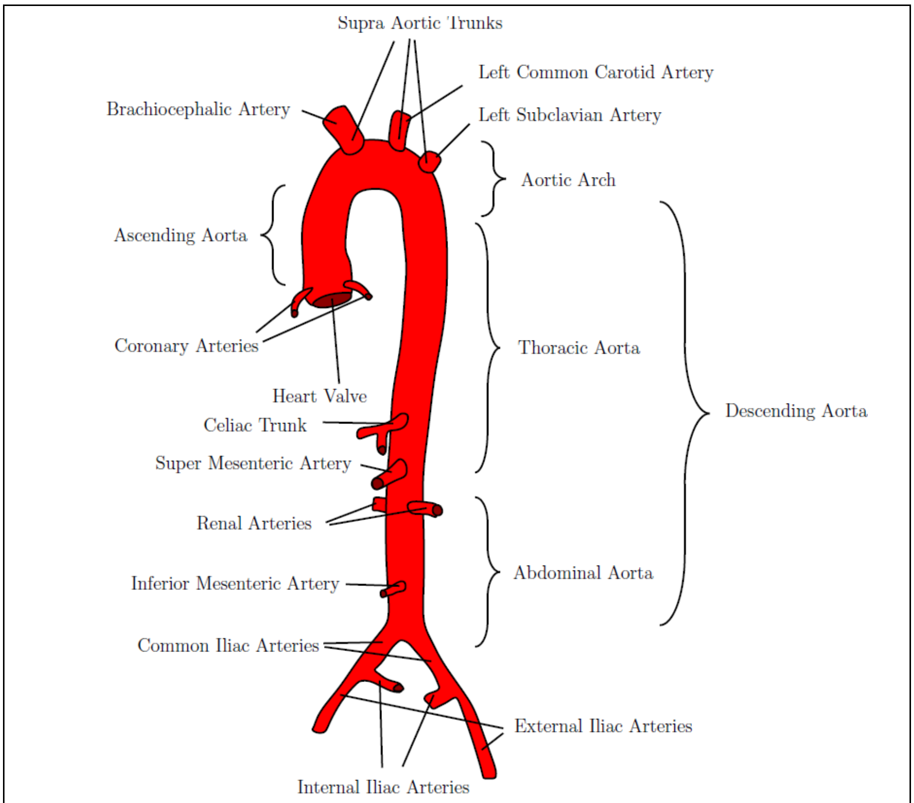


Figure 3 - Aorta and its main ramifications

In almost half the cases a ruptured aortic aneurysm results in death (14). For this reason treatment of the aortic aneurysm is aimed at preventing rupture. Surgical repair is the only efficient way of treating this condition but it carries risks due to age and age related illnesses. For small aneurysms, where the rupture risk is low, doctors might recommend active observation and try to control the advancement of the disease by reducing the risk factors. Surgery is recommended when size of the aneurysm is big (>5.5cm) and the assessed rupture risk is greater than the surgery risk.

2.3.2. AAA repair techniques

A surgical repair of an aortic aneurysm consists of replacing the weakened part of the blood vessel with a stent graft. This way the blood will flow through the graft reducing the pressure on the vessel wall and hence the rupture risk. There are currently two different approaches to this surgery: open repair and endovascular aneurysm repair (EVAR).

Open surgery is the traditional treatment of an aortic aneurysm. This procedure is done in the operating room (OR) with the patient under general anesthesia. The main steps (Figure 4) for this surgery are the following (15):

- An incision is made in the abdomen to get access to the aorta.
- The aorta is clamped above and below the aneurysm. In some cases a temporary bypass is needed to keep blood circulation in all parts of the body.
- The aneurysm is opened or the entire affected part of the aorta is removed.
- A graft is sewn in place of the aneurysm.
- The blood flow through the aorta is restored by removing the clamps. The temporary bypass, if it exists, is also removed
- The patient is closed and moved to intensive care.

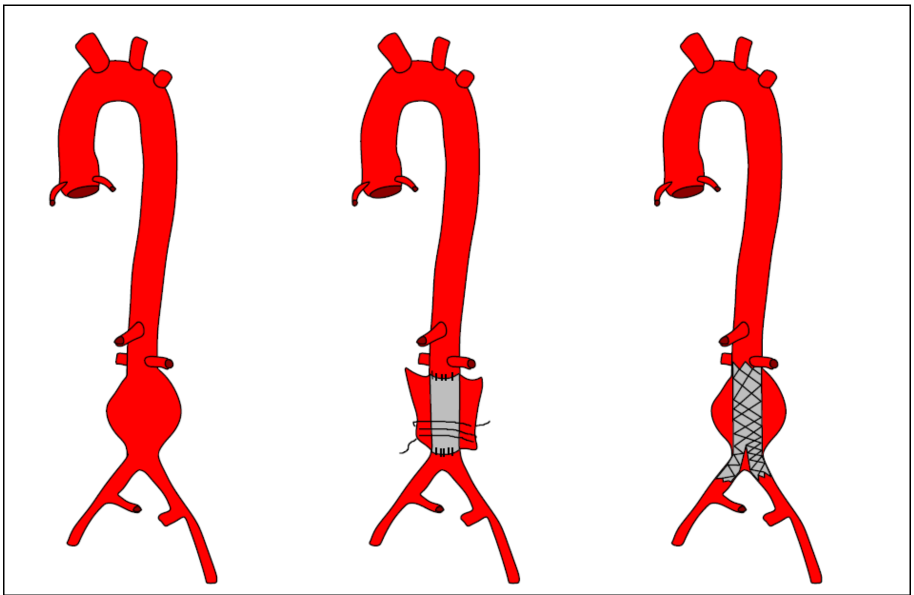


Figure 4 - Schematic representation of open surgery steps; (a) aorta with aneurysm, (b) opened aorta with the stent sewed in, (c) aorta with the graft in place

Endovascular repairs are, compared to open surgery, relatively new procedures. This approach was introduced by Juan Parodi in 1991 (16) and since then the technique has developed quickly. The procedure can be done in a radiology room or a hybrid OR and does not necessarily require the patient to undergo general anesthesia. Because of the lack of nerve endings inside of the vessel only local anesthesia might be required. The steps for this procedure are very different compared to the open surgery (15):

- An incision is made in the groin area to expose the femoral artery; then an introducer sheath is placed in to allow the insertion of the catheters and guide wires into the aorta.

- One or more guide wires are inserted into the aorta under live fluoroscopy X-ray imaging.

- Using the guide wires catheters are inserted into the aorta. Different catheters might be needed. For example catheters for injecting contrast agent to outline the vessel or the catheter that carries the stent graft are needed.

- The catheter carrying the folded stent graft is inserted into the aorta and placed into the right position. The guidance is made under fluoroscopic imaging. Contrast agent injection is needed for accurate placement. When the positioning is correct the stent graft is unfolded.

- Contrast is injected into the aorta to test the correct placement of the stent graft and to assure that no blood still flows into the aneurysm.

- All the catheters and guide wires are removed and the incision point is closed.

The low impact on the patient and reduced recovery time are the main advantages of EVAR procedures. In contrast to open surgery only small incisions are necessary. This minimizes the blood loss and reduces the time the patients have to stay in hospital. After an EVAR intervention the patient can return home within a few days. The cost of endovascular procedure is higher than open surgery because of the special equipment involved but, on the other hand, the post-operative costs are considerably reduced due to short recovery time.

The main disadvantages of EVAR are linked to the fact that it is a relatively new procedure. Open surgery is already a well proven method with a high success rate and is still the only option in some cases. Narrow vessels, the location of the aneurysm or the geometry of vessels can make endovascular procedure impossible. Moreover the usage of X-rays and contrast agent require some precautions. Nevertheless the latest studies show that endovascular procedures can have a lower aneurysm-related mortality rate.

2.4 VesselNavigator

Philips is developing the VesselNavigator application in order to help physicians clearly visualize the vessels during endovascular procedures with the focus on EVAR. The main problem in these procedures is that while the stent graft and guide wires are clearly visible on the live 2D X-ray fluoroscopy, the vessel itself is almost impossible to discern. Vessels and blood have similar radio opacity as the surrounding tissue and they cannot be distinguished using normal fluoroscopy images. To make them visible physicians need to inject contrast agent and use high X-ray levels. Both the contrast agent and X-rays are harmful to the human body so they cannot be used constantly during the procedure. Most of the time surgeons must navigate the wires without having real time information about the vessel outline.

Before the EVAR procedure a CT angiography or MR angiography scan is required to obtain detailed 3D information about the aneurysm. This information is needed for obtaining precise measurements to determine the exact types and sizes of the stent graft to be used. In addition the surgeon will use the 3D vessel structure to plan the position where he will place the stent grafts. Besides the help for diagnosis and planning it provides during pre-perative phase, the 3D vessel structure could also be used during the actual surgery for guidance.

The VesselNavigator project proposes the overlay of the pre-acquired 3D modalities onto the live images to help physicians guide through the vessels. The project proposes a workstep-based approach with four steps that is optimized for the endovascular surgery workflow. The first two steps are designed for surgery preparation and they can be done prior to the actual procedure. The last two steps are meant to help physicians during the actual surgery.

1. Segmentation: The goal of this step is to acquire the relevant information from the existing 3D volume. This step extracts the 3D structure of the relevant vessels. The bones and other details found in the original volume that would clutter the display are removed. Landmarks can also be added to the segmented volume.

2. Planning: In this step the physician can use the 3D vessel structure to select different viewing angles that will offer good visualization of specific region of interest. For example the surgeon might want to select an angle that will clearly display the point where the aorta branches into the renal arteries. This angle can be saved and during the procedure the stored information can be recalled and sent to the C-Arm to match the saved view.

3. Registration: This step is required in order to align the 3D segmented volume with the position of the patient. Manual or automatic 2D-3D registration can be used to make this alignment.

4. Live: The 3D segmented volume is overlaid on top of live X-ray images. If the volume is registered correctly with the patient then the 3D vessel structure will match the X-ray images regardless of the position of the X-ray arm.

2.4.1. Registration step

The main goal of the VesselNavigator application is to provide an overlay of a pre-acquired 3D volume on the live X-ray images during the endovascular procedure. In order to be able to present this overlay correctly, the spatial mapping between the volume and the patient must be computed. This alignment can only be determined during the procedure, when the position of the patient on the table is known. This registration can be achieved using live X-ray images. After the alignment is done, the volume will then be correctly overlaid on subsequent live X-ray images regardless of the angle these images are obtained. The registration step of the VesselNavigator is designed to offer physicians assistance for obtaining this alignment using 2D-3D registration.

Let us first consider performing registration using a 3D volume, and a 2D projection image. This is done by matching the position and orientation of the landmarks that are visible in both modalities. As section 2.2 already pointed out registering with only one image will not offer enough data about the complete 3D location of the volume since accurate depth information is missing. Moving the volume orthogonal to the X-ray image, will result in scaling of the volume projection. It is necessary to note that this scaling is contra-intuitive: due to the X-ray source – detector setup, the volume will appear smaller when it is moved closer to the detector. Figure 5 exemplifies this problem: In Figure 5 a) the volume is moved orthogonal to the projection plane (that is, the detector), resulting in a magnification of the projected image. However the effect of scaling is relatively small, and is obfuscated by other mismatches due to translation and rotation.

To solve the depth precision problem a new X-ray image from a different angle is made. As Figure 5 b) shows, the movement along a direction orthogonal to the first image is clearly visible on the second one. The volume position in space is described by three parameters, its coordinates. Fitting the volume on one of the images is equivalent to finding two of these three parameters. Fitting the volume in both images leads to a complete registration.

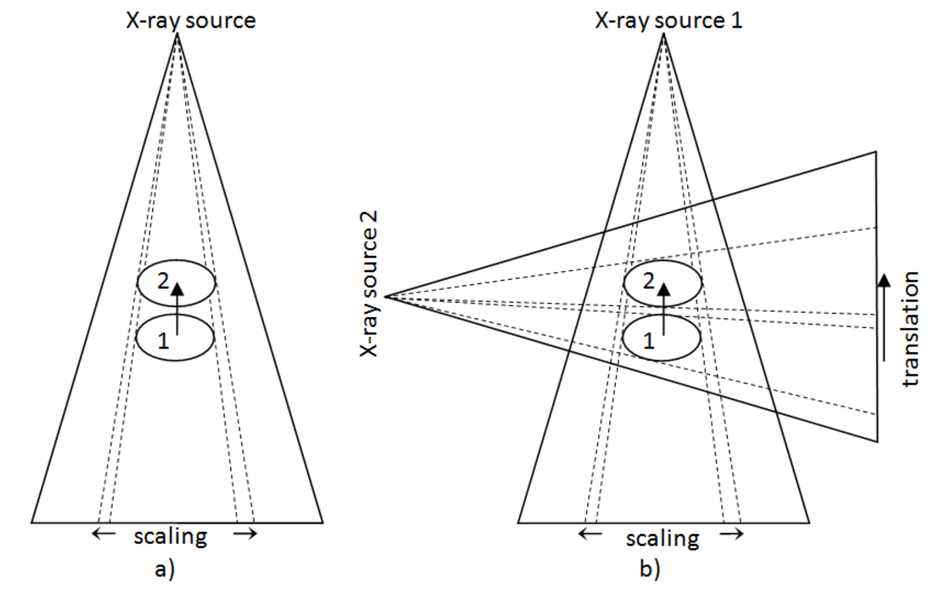


Figure 5 – The effect on translation of an object to its projection. The elliptic object is translated from position 1 to position 2 a) A translation orthogonal to the projection plane leads to scaling the projection depicted by the arrows. b) The same translation is more visible when a different projection plane is used. This time the small scaling is replaced by a larger translation on the new projection plane.

The result of the registration is also dependent on the angle between the two images. If this angle is small the images are very similar and, as a result, the depth information is still limited. For this reason maximizing the angle (up to 90 degrees) will provide a better registration. The best possible results are obtained when the images are orthogonal (Figure 5).

3.Stakeholder analysis

Abstract – In this chapter the main stakeholders of the project are presented.

3.1 Introduction

In the early stages of the project two different groups of stakeholder were identified. The first group of stakeholders consists of the persons within Philips Healthcare that are involved directly or indirectly in the development of the VesselNavigator application. The second group of stakeholders is the end users of the application developed. This groups is not limited to only the clinicians directly involved in using the VesselNavigator software but also the hospital managers who are interested in offering better care for the patients at low cost.

3.2 Philips Healthcare

Philips Healthcare provides, amongst others, imaging applications for the clinical environment. The aim is to provide hospitals with innovative solutions that can improve the physicians work and the outcome of the procedures. The domain in which Philips is providing solution is very broad. Figure 6 shows the relevant part of the organization chart for the VesselNavigator project. As the image depicts the project is carried out in the iXR - Innovation department.

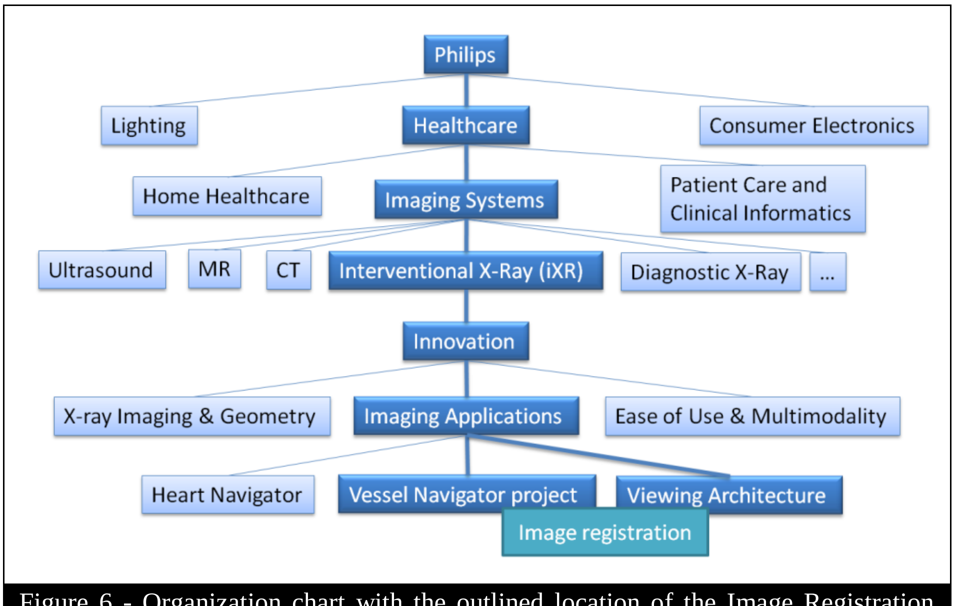


Figure 6 - Organization chart with the outlined location of the Image Registration project.

The Interventional X-ray department develops applications for interventional procedures. This software is used in combination with an interventional X-ray machine, such as Phillips Allura Xper FD20 system. This kind of machine is designed to be used in an operating room for providing live X-ray images of the patient. A C-arm containing an X-ray source and a detector is used to generate these images. The arm can be rotated around all the 3 coordinate axes (Figure 7) in order to be able to provide images of the patient from any angle.

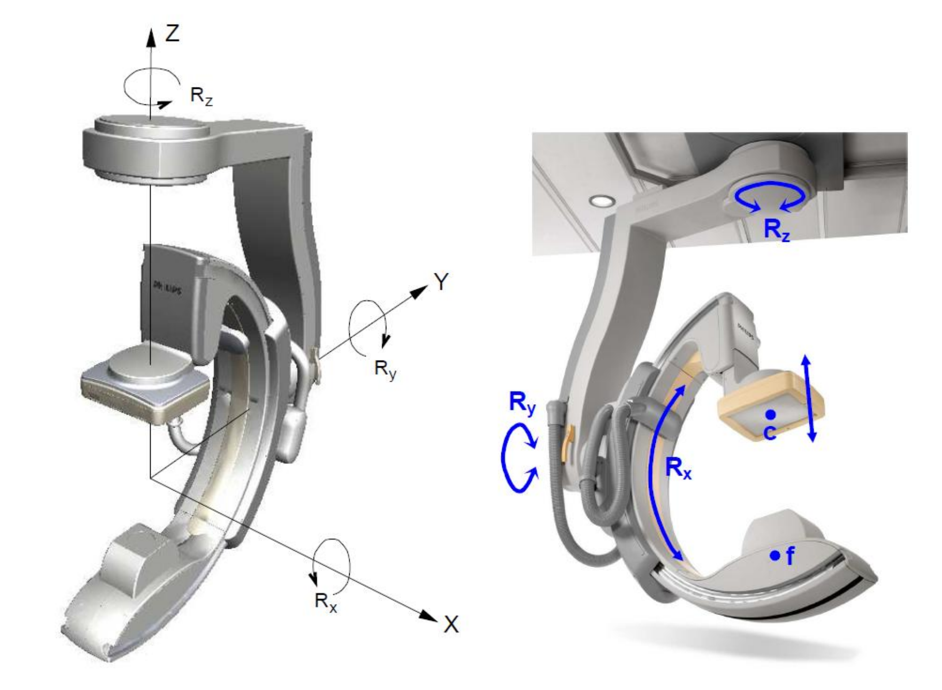


Figure 7 – The C-arm and its rotation capabilities

The Innovation department is responsible for creating prototypes for future Philips products. VesselNavigator is one of these prototypes. It is the result of the combined work of two teams. First team is composed of the people directly involved in designing and implementing the application. The second team is responsible for creating a viewing architecture (iSDK) that should offer a common ground for developing multimodality imaging application. VesselNavigator is one of the first applications that are using the new viewing architecture. This means that the collaboration between these two teams very important. The cooperation offers advantages for both teams: VesselNavigator benefits from the new viewing possibilities and in the same time the iSDK developers can use the feedback provided by the VesselNavigator team to improve their components.

As said before, VesselNavigator is in the prototype phase. In the last two years deployment in the hospitals yielded feedback that was used to iteratively improve the application. When iterations have converged to a prototype that fulfills the needs of the user and is easy to use, the application will be productized by the development department of Interventional X-Ray. A good design and reusable code will facilitate a quick transition from prototype to a productized clinical application.

In the prototype development phase new ideas are validated in the clinical environment. Clinical scientists bridge the gap between the technical and clinical domain. They possess the technical background to understand the software possibilities and are also in constant contact with the physician and the clinical trials. They provide software engineers with feedback from the hospitals and can have valuable insights about what physicians might need of find useful. Their expertise is necessary for the success of a new product.

3.3 Hospital stakeholders

The end users of the visualization applications are the physicians, and understanding their way of working is crucial for the success of the project.

Endovascular procedures are traditionally conducted by interventional radiologists, physicians who have much experience in performing minimally invasive procedures under X-ray guidance. In the past two decades improvements in the technology made minimally invasive surgeries a more popular approach. Vascular surgeons, which used to be involved exclusively in the open surgeries, also began conducting endovascular procedures. Nowadays the vascular surgery training includes endovascular techniques. From the point of view of this project both types of clinicians are potential users for the application so the main differences between them must be understood in order to meet the requirements of both.

- The first important difference between the two categories is the different backgrounds they have. Interventional radiologists have extensive training in a radiology environment and are expert users of the imaging techniques. Surgeons on the other hand undergo extensive training for open surgery and endovascular procedures and imaging techniques are fairly new fields for them. Therefore not all vascular surgeons are well versed in performing X-ray guided procedures.

- The procedure environment also differs based on the type of physician involved. Radiologists work in a radiology room which was designed and built to accommodate the X-ray machine and the additional hardware needed. The radiology rooms typically do not meet the sterility conditions of an operating room. Surgeons are used to working in an operation room (OR) which is designed for open surgery with high sterility conditions. Currently X-ray guidance in ORs is performed using mobile X-ray C-arms. However, in recent years, hybrid OR's are becoming more common. This new type of ORs usually features a high-end X-ray machine, similar to the ones found in a radiology room, but also meet the sterility requirements needed for open surgery.

Based on the differences discussed we can distinguish some advantages and disadvantages each of the two types of physicians have. The vascular surgeon has a broader surgical knowledge. This means that if complications arise during endovascular procedures he can immediately convert to open surgery. If the procedure is performed in a radiology room converting to open surgery is more complicated as it requires moving the patient to an OR. When talking about EVAR procedures, converting to open surgery is, in general, a consequence of a ruptured aneurysm. In such a case, a fast conversion to open surgery can make the difference between life and death.

As said, vascular surgeons are not necessarily specialists in using the imaging technologies. Interpreting live X-ray images displayed on a screen to orientate inside of a blood vessel can be a challenging task. For this reason clearer guidance in using the application may improve the outcome of such procedures. For example, additional information about the images orientation can be very useful but on the other hand cluttering the display with a lot of data is not necessarily helpful. The physician must be able to find the information he needs in a very short time so the data must be displayed in an intuitive way.

Figure 8 – Operating room with the Allura system installed

The way the physicians interact with the software is another very important aspect that needs to be taken into account when designing imaging applications for interventional procedures. Figure 8 presents a typical operating room setup. As the picture shows, the intra-operative setting and dynamics are very different from a diagnostic environment. During the pre-operative phase, the physician is operating the workstation from behind a desktop PC. During the surgery, the physician stands at the side of the operating table and the possibility of interacting with the workstation is considerably more limited.

In the diagnostic and preparation stage the physicians can make use of a complete workstation. This involves using a keyboard and a mouse. During the procedure the physician does not use a keyboard, but typically only a touch screen. In some cases he may choose to use a mouse, wrapped in a sterile bag. Compared to a rich interactive UI that an interventional application like VesselNavigator offers, the touch screen module (TSM) can provide only limited interaction possibilities. Figure 9 presents the TSM module available in the operating room.

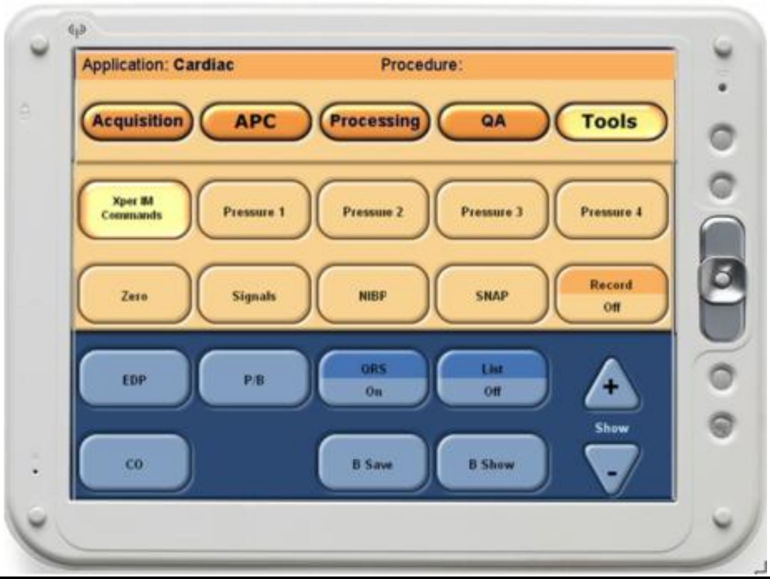


Figure 9 – The touch screen module used in the operating room

During the surgery, besides the surgeon, several other persons need to be present in the OR: anesthetist, nurses, etc. Even though they are not direct users of the software these persons are also considered stakeholders. For example the additional medical equipment used by the anesthetist can be in the way of the C-arm. The C-arm movement constraints must be taken into account when designing an application.

As said before, hospital managers are also considered stakeholders of the project. Offering better treatment possibilities would be an advantage for the hospital. Moreover improvements in minimal invasive techniques reduces the cost of the treatment by reducing the time patients are required to stay in the hospital after surgery.

4.Problem Analysis

Abstract – This chapter gives meaning to the current project by describing how the endovascular clinician goals are translated into the goals of the project

4.1 Introduction

This project was started to address needs in the clinical environment. In the previous chapter the way of working of both surgeons and interventional radiologists was described. This chapter will illustrate how the VesselNavigator application and this project in particular help physicians achieve their goals of offering better care to the patients.

4.2 Endovascular clinician challenges

EVAR procedures offer some advantages compared to the open surgery, but it also introduces some new risk factors. These factors are described in details in the following section. The main goal of the endovascular clinician is to offer the best possible care to the patient. To achieve this, he must reduce as much as possible the surgery risk factors.

4.2.1. X-ray exposure time

After the discovery of X-ray, people started to observe adverse reaction induced by them. The first signs were discovered by Rontgen when he started suffering from troubling red skin after long exposure to radiation. Further observation of people exposed to high doses of radiation showed without doubt that caution is needed when using X-ray. Nowadays the principle ALARA (As Low As Reasonably Achievable) is used in radiology field.

The effect of radiation on live creatures depends on the amount of X-ray radiation absorbed by the body. To quantify this, the sievert (Sv) unit is used. This unit measures the biological effect of the radiation to living tissue. It is different from the actual energy of the X-ray, measured in Grays (Gy). Both units are expressed in joules per kilogram but must be interpreted different. The sievert can be seen as a dose-equivalent for living tissue. Depending on the type of radiation the same amount energy (expressed in Gy units) can have different effect on the body (expressed in Sv units).

The human body is permanently exposed to radiation generated by the environment. This type of radiation, called background radiation, has natural causes and due to its low values does not present a risk. The normal value for the background radiation is around 2-3 mSv per year. For comparison, during a single CT scan of the abdomen, the body receives around 10 mSv of radiation (17).

In sufficient dose, radiation will have great impact on living tissue. It can modify the internal cell structure by ionizing atoms. When exposed to low doses of radiation the cells in the human body will not suffer extensive damage and can in general repair themselves. However, when the dose exceeds 1 Sv the cells are not able to repair the damage suffered and die. If the amount of dead cells is not high the body can just replace them. In case the body fails to replace the destroyed cells, immediate symptoms can be observed. This condition called radiation sickness is manifested usually by nausea and general weakens. The radiation can also affect the DNA that can lead to abnormal cell development. In some case the affected cells can start dividing and become cancerous. This leads to an increased cancer risk resulting from radiation exposure.

A study on 320 patients who underwent EVAR between 1998 and 2008 pointed out that in average, a patient was exposed to 7 +/- 7.1 mSv of radiation during the procedure(18). This amount is equivalent to about four years of background radiation. Besides increasing the cancer risk, X-ray exposure time can also lead to potential skin damage. Another study showed that in about 29 percent of EVAR procedure the threshold for possible radiation induced skin damage was exceeded (19).

During the procedure, the X-ray beam is focused to the region of interests. The patient is exposed to most of the radiation but other people in the operating room are also exposed, albeit to lower doses. This can be direct radiation when working directly in the beam. Furthermore, when X-rays pass through the patient's body, a small part will be scattered. Doctors and nurses are exposed to this scatter radiation. For this reason they are required to wear X-ray shielding lead aprons. Some parts of the doctor's body, like the arms or the face, require high mobility during the procedure so they cannot be protected. As a consequence, regulations limit yearly radiation dose that a physicians who carry out endovascular procedures can face. This is one of the factors determining the number of surgeries a physician can perform in a year.

4.2.2. Contrast agent

Contrast agent is used during an EVAR procedure in order to visualize blood vessels under X-ray fluoroscopy. However, the injection of the radio-opaque material into the blood stream may lead to a series of complications (20); the most important are listed below.

- Allergic reactions. The allergic reaction severity can vary. Some of them can be controlled with anti-allergic medication. In a small number of cases the allergic reaction is severe and becomes a medical emergency.

- Kidney damage. Contrast agent is removed from the body by the kidneys. Due to the nature of the dye, during this process kidneys are damage to some extent. In the case of an EVAR procedure the amount of contrast required for visualization is quite high because of the size of aorta. This combined with the proximity of the contrast injection to the kidneys leads to high concentration of dye received.

Other complications with a lower probability include blood clots that can lead to strokes or heart attacks, hypotension or even blood vessel damage. In general reducing the amount of contrast used, also reduces the risk of complications.

4.2.3. Surgery time

Surgery time has an indirect impact on EVAR procedures. A shorter surgery time implies less X-ray exposure time and less contrast agent used. Also blood loss by the patient is proportional with the procedure duration. A long procedure involves higher stress to the incision area and can increase the recovery time. Moreover the duration the patient is under general anesthesia should be as short as possible. General anesthesia is not always used in EVAR procedure. However, reducing the surgery time also minimizes patient stress when he needs to be awake during the procedure.

Surgery time can also be linked to the procedure cost for the hospital. A long operation involves keeping an OR occupied and this leads to increased costs.

4.3 Project goals

As discussed in Chapter 2, the VesselNavigator application was designed to help clinician throughout the stages of an EVAR procedure. The application has four steps that match the natural surgery steps: segmentation, planning, registration and live guidance. The third step, registration, is the focus of the current project. The primary goal of the project is to improve the third step, registration, such that the goals of the clinicians can be fulfilled better.

4.3.1. Reducing radiation exposure time and contrast dose

VesselNavigator offers physicians the possibility to overlay, on top of live fluoroscopy images, a pre-acquired 3D volume containing the vascular structure of the region of interest. Without this overlay the physician will have to use contrast injections to see the vessel outline. Since the use of contrast injection needs to be limited, most of the time the physician has to remember the outline of the vessel from the previous contrast run. Moreover the C-arm is moved during the procedure in order to get an optimal view on the region that is operated upon. After each movement the position of the vessels on the X-ray images changes and, in order to re-orientate, physicians need to use new contrast injection. VesselNavigator reduces the need to use contrast agent after repositioning the C-arm since the overlaid volume is moved and rotated synchronized with the C-arm. The physicians can immediately see the vessel structure form the new direction.

The registration is the first step that is typically performed in the operating room. The goal is to find a mapping between the acquired 3D volume and the patient position. The registration step is essential for the application. Without a good registration, clinicians cannot use the pre acquired 3D volume and hence cannot benefit from the advantages of the application.

4.3.2. Reducing surgery time

The registration step has a direct impact on the duration of the surgical procedure. The time required to register the 3D volume with the patient's position is added directly to the surgery time. Since general anesthesia is not always used, there is the possibility of patient movement during the procedure. This may result in having to redo the registration process, fully or partially.

The usage of the current 2D-3D registration pointed out several limitations that make the registration process difficult. The goal of the project is to increase usability in order to reduce the time required for the registration process. The limitations of the registration process can be pointed out by showing the way the physicians use the application. The listing below presents the steps the surgeon follows in the registration step.

1. The surgeon goes to the registration step in the VesselNavigator application. The layout of the application window is presented in Figure 10.
2. Surgeon moves the C-arm in the desired position for acquiring the first reference image.
3. X-rays are sent through the patient in order to obtain images. These images will be presented in the live window (the left window in Figure 10). Every time X-rays are sent through the patient's body, the obtained images are recorded and can be used for registration.
4. The surgeon can copy the acquired image to one of the reference windows (one of the two windows displayed in the right side in Figure 10).
5. Steps 2-4 are repeated to obtain the second reference image.
6. The surgeon translates and rotates the 3D volume with the two reference images. The manipulation of the 3D volume is done with the use of the interactors that can be selected from the toolbars.
7. Steps 2-4 can be repeated to redo the registration if needed (in case of patient movement for example).

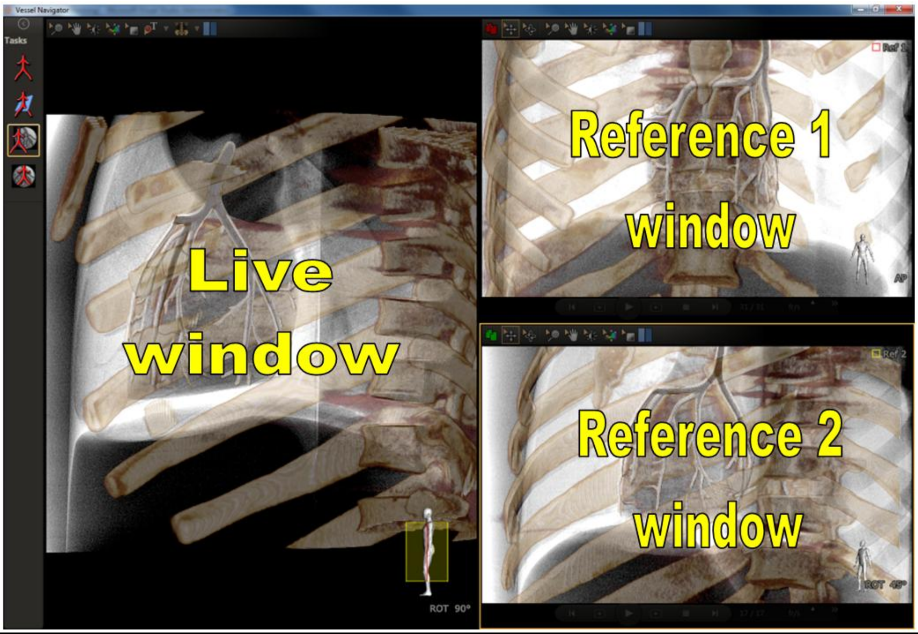


Figure 10 – Window layout for the registration step of Vessel Navigator. The live window is presented in the left side while in the right there are the two reference windows.

The first problem that the users of registration are facing is the fact that the rotation of the 3D volume, which is necessary in order to properly align it with the live X-ray images, is not an easy process. The rotation of a volume in the 3D is defined by three parameters, the rotation angles around the three coordinate axes. Defining this rotation using a mouse is a challenging task. The exiting implementation offers the user two different interactors to define a complete volume rotation. The first one, the roll interactor, transformers the horizontal and vertical mouse movement on the screen to a rotation of the volume around the Y and X axes. A second interactor is needed in order to be able perform in-plane rotation (rotations around the Z-axis). This approach does not result in a volume rotating in an intuitive way. Furthermore having to switch between interactors in order to define a complete rotation is not very user friendly. Thus, a new way of interacting with the volume is needed.

Additionally the volume also needs to be translated to achieve the registration. Section 2.4.1. described how a complete translation in 3D space can be done by aligning the volume with two images acquired from different angles. Achieving this alignment is not an easy task. To understand the problem we can use the simplified example presented in Figure 11. In order to bring a landmark located initially at ‘I’, to its correct location ‘F’ the user will follow the following steps:

1. User aligns the landmark in the first image.
2. User aligns the landmark in the second image. This second translation leads to a misalignment in the first image.
3. User needs to realign the landmark in the first image. Again this results in a misalignment of the landmark in the second image
4. etc.

A large number of steps are required to register a landmark using the method illustrated in Figure 11. Intuitively, defining a complete 3D translation should require only two 2D alignments: a first translation to align the volume with one images and a second translation in the other to correctly align the volume in the direction orthogonal to the image plane. A new translation interactor is needed to achieve this two-step alignment.

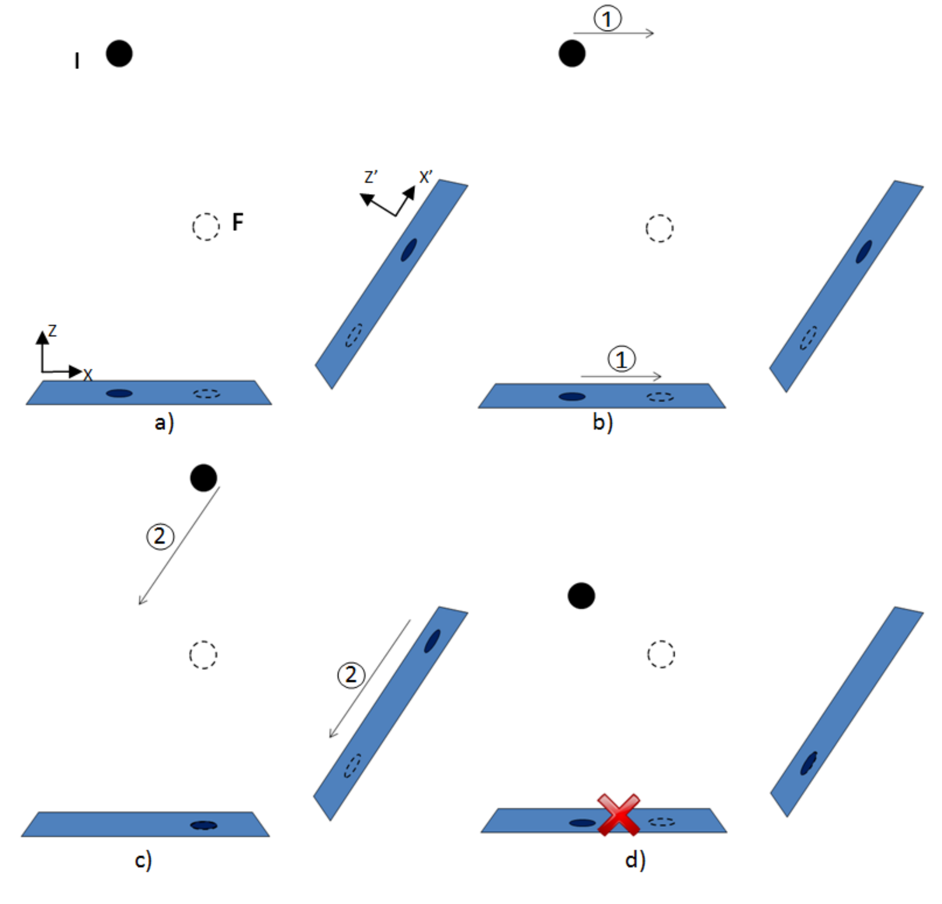


Figure 11 – 2D-3D registration steps for aligning one landmark. For simplification an orthogonal projection is shown here. Moreover the two images share the Y axis. The landmark is first aligned with the first image (1) by translating it parallel to the image plane. Then the landmark is aligned with the second image using the same approach (2). The second translation leads to a misalignment in the first image therefore additional steps are needed to achieve the global alignment.

Increased usability can also be achieved by providing suggestions to the physician during the procedure. One of the things the surgeon needs to think in the registration step is the angles from which he must acquire the reference images. The software can make suggestions about these angles based on the state of the registration process. For example if one of the reference windows is already filled the application can suggest the surgeon to acquire the second reference image such that the registration angle should be big enough for registration.

When new images are recorded the application allows the surgeon to copy them to one of the references. There are two reference windows and both of them can be used. If both references are filled coping new images will overwrite the old ones. In this case, the resulting registration angle can be the criteria for choosing one reference over the other. The application can hint the physician about which registration window to use by making these not so obvious calculations.

As Figure 10 shows, half of the application screen is occupied by the live window. During the actual registration procedure (after the two reference windows are filled) the surgeon only needs the information from the reference screens. For this reason an option to maximize these two windows will help the physician to better visualize what he is doing.

The surgery time can also be reduced by automating some of the interaction of the physician with the application. An automation that can be made is to copy the images generated by an exposure[1] run to an empty reference window (if exists). Also after the surgeons copies images to the second reference the registration process can begin. In this case the application can directly switch to a maximized view showing only the reference windows. Other graphical details can be added to the registration screen to make it easier for the surgeon to orientate. Understanding the orientation of the C-arm that lead to a specific image is hard to retrieve from the image itself. The initial prototype tried to solve this by adding to each window a small human shape (the dwarf) that has the same orientation as the images in that window. An improvement that can be made is to add a textual description of this orientation. This will offer the physician more precise information about the orientation.

1 The physician has two options to acquire live images: fluoroscopy and exposure. Fluoroscopy corresponds to a low dose of radiation. It is the method used during most of the procedure to visualize the catheters and guide wires. When anatomy details have to be visualized higher X-ray dose, exposure, is used. Exposure is usually used in combination with contrast injection to see the outline of the vessel.

5.System Requirements

Abstract –Chapter 4 described the project goals. This chapter provides a more detailed set of software requirements that derived from the project goals.

5.1 Introduction

The main goal, as defined at the beginning of the project, was the improvement of the registration step of VesselNavigator. Due to the way of working in the innovation department, requirement gathering was a dynamic process. Developing a prototype requires constant validation from the users for the newly added features. This means that a comprehensive list of requirement was not produced at project start, but rather gathered during the project. Two main directions were identified where improvements could be made:

- The existing manual registration step
- The possibility of integrating automatic registration

Based on this, short iterations were made, each iteration adding new or improved features to the prototype. After each of these iterations, based on the result and the feedback received from the users, requirements for other features could be stated or refined. The rest of this chapter presents the main requirements for the two directions of the project.

5.2 Manual registration improvements

Previous chapter discussed the need for introducing a new interactor for rotating a 3D volume. The main requirement for the new interactor is to offer users a natural an intuitive way of handling a 3D volume. For this a new way of rotating the volume needs to be introduced. The new method simulates the rotation of a sphere "glued" to the 3D volume. Besides the new rotation method (discussed in details in section 8.1) some visual features are added during the interaction to help users orientate themselves even more easily. Two features are needed:

- Graphical visualization to help user understand where the rotation center is. This will and also gives some idea about how the volume is actually rotated.
- A marker that will display the position of the point of contact with the sphere. This represents the point on the virtual sphere that was selected when the interaction was initialized.

To make use of this new rotation interactor the center of rotation must be clearly defined. This is not as trivial as it may seem. In the case of the segmentation step, it turned out that rotation was most intuitive and easy to use when the center is in the center of the viewport, whereas in the planning step is was most logical to place the rotation center in the Volume center. In the registration step, the case was even more complex: the ability to move the rotation point was required. Also, since we interact with the volume in a 2D plane, it is not always obvious what the depth position of the rotation point should be. Considering this, we decided that the selection of the rotation center should not be part of the Arcball interactor itself, but rather inject it. This resulted in a much more flexible design.

The Arcball interactor should also allow users to easily rotate around the three fixed coordinate axes. For toggling X and Y axes rotation a modifier key should be used ("shift"). Also the visualization includes the rotation axis for clarity. The rotation around the Z axis should be performed when the user starts the interaction from outside of the virtual sphere.

In previous chapter we explained the need to introduce a new way of translating the volume in order to be able to align it with two reference images without excessive iterating. This is not trivial in the case where the two translation planes are not perpendicular. A solution to this problem is to define a new translation plane for the volume. For simplicity, let us assume a 2D volume and 1D reference images. Now, when we are registering the volume with reference image #1, we define an axis perpendicular to reference image #2, and move the object only along this axis. This way when we move the volume in reference image #1, we will not see movement in reference image #2. More details about the geometry behind this interactor are provided in section 8.1 .

Figure 12 exemplifies the desired functionality of the orthogonal interactor using the same example and simplifications as Figure 11. In this case translating along the X axis in one of the reference windows will generate a translation of the volume along the Z axis of the other reference. This ensures that the alignment process in one reference will not misalign the volume with respect to the second reference.

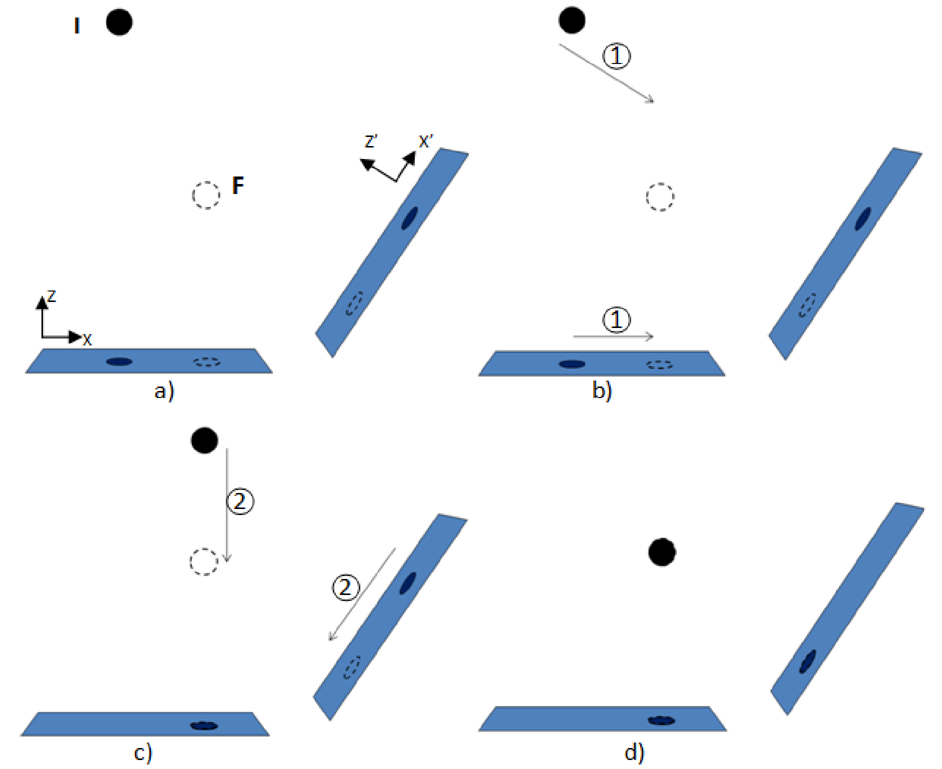


Figure 12 – Two step registration of a landmark using the orthogonal interactor. The landmark is first aligned with the first image (1) by translating it orthogonal to the second image. Using the same approach for aligning with the second image (2) will not lead to a misalignment in the first image therefore no additional steps are needed.

The other graphical improvements for the manual registration step are presented in Table 1.

5.3 *Automatic registration*

Besides improving the manual registration, the project also targeted adding automatic registration. Due to the limited amount of time a decision was made to focus on gathering available automatic registration algorithms already available in other Philips product and not on trying to develop new ones. There were two main requirements for this stage of the project

- Research on the in-house available automatic registration techniques

- Research the feasibility of integrating one of these in VesselNavigator application.

5.4 Requirements overview

The following table provides a summary of the project's requirements. A unique id is associated with each requirement for traceability purposes

ID	Description
R1	Add an intuitive interactor for rotating the 3D volumes (Arcball interactor).
R2	Add visual components to the Arcball interactor to easily trace the rotation center and the interaction starting point.
R3	Provide an interface that external components can use to modify rotation center of the Arcball interactor.
R4	Implement rotation around the fixed X and Y axes on "shift" key is pressed.
R5	Add in plane rotation (Z axis rotation) to the Arcball interactor when the interaction initiated outside of the sphere.
R6	Implement and orthogonal interactor that will offer the user the possibility to align the volume with two references in two steps.
R7	Provide a graphical representation for hinting the reference most suited for copying new X-ray runs. Functionality must be added both to the application screen and TSM.
R8	Provide angle hints that can be sent to the APC button of Allura system. Option must be available both in the application screen and TSM.
R9	Add graphical representation of the reference orientations in the live screen.
R10	Show textual representation of orientation below the dwarf.
R11	Highlight the currently selected window (live, reference1 or reference 2) in the registration step by adding a colored border.
R12	Add the option to go to a maximized view showing only the two registration windows. Option must be available both in the application screen and TSM.
R13	Automatically go to the maximized view when both references are filled.
R14	Gather information about the available registration techniques in other Philips applications.
R15	Research the possibility to integrate available automatic registration algorithms in VesselNavigator

Table 1 – Software requirements

6. System architecture

Abstract – This chapter describes the high level architecture of the Vessel Navigator application in order to provide the software context for the project.

6.1 Introduction

The VesselNavigator follow the structure of the underlying toolkit used. This toolkit uses the Model View ViewModel pattern. In the following section this pattern is presented. Next, sections 6.3 and 6.4 will show how the Viewing architecture and the VesselNavigator application are built following this paradigm.

6.2 The Model View ViewModel Pattern

The Model View ViewModel (MVVM) pattern represents a variation of the more famous Model View Controller (MVC) pattern. This new variation was introduced by Martin Fowler in 2004. The pattern is currently widely adopted in modern UI development platforms such as Windows Presentation Foundation (WPF) (21), Microsoft Silverlight, HTML5 or Android. The pattern consists of the following parts (also described in Figure 13):

- View: The View layer represents the user interface (UI). It describes how the information is presented to the user. The view is also responsible for intercepting the user interaction and translating them into application events.

- Model: The Model layer is a representation of the application domain, independent of the View. The way the Model is structured depends only on the logical relationships between the domain entities and is not connected to how these entities are presented to the user.

- ViewModel: The ViewModel represents the bridge between the Model and the View. It is responsible for retrieving the data from the Model and translating it to format required by the View. The ViewModel is notified by the Model when the data has changed and relays this notification to the View. Moreover the ViewModel responds to the UI events and updates the data in the Model if required.

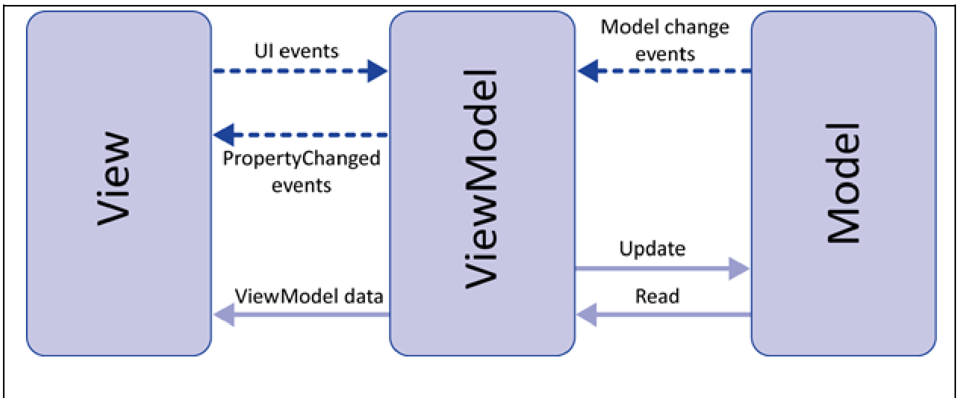


Figure 13 - The MVVM pattern components and their interaction

The main goal of the MVVM pattern is to decouple the UI design from the rest of the application logic. This is desirable since the development of the UI is often the job of a dedicated designer rather than a software engineer. In this way the designer does not need to have knowledge of the actual implementation details of the application. In the same time this separation is also beneficial for the developers who can implement the business logic without knowing how the data will be presented to the user.

When the UI and application logic are decoupled, the designers and the developers can use different tools tailored for their specific needs. The designer can use a Graphical designer that outputs its design in a description that can easily be integrated with the application. This is typically done by means of a high-level declarative language. In the case of WPF this language is Extensible Application Markup Language (XAML). XAML is a XML based language developed by Microsoft.

Completely decoupling the View layer provides the possibility to easily change the user interface without the need to modify the underlying model. This also leads to an increase testability of the application. Small mockups of the UI can be implemented in order to test parts of the logic very easy. Moreover automated tests can be written using this approach.

6.3 The Viewing Architecture

The Viewing architecture defines a common infrastructure for creating multimodality imaging applications. A software development kit, the iSDK, is available to allow software developers to easily access the components of this infrastructure. The Viewing Architecture was designed to seamlessly integrate with the WPF toolkit in order to facilitate fast prototyping development. The design of the toolkit follows the MVVM pattern (Figure 14).

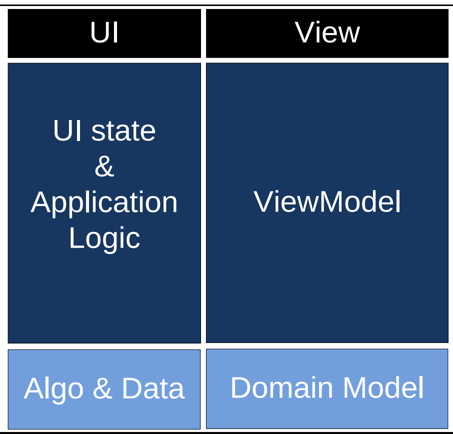


Figure 14 - MVVM of the Viewing Architecture

The View layer contains all the UI controls and widgets that can be used to build a graphical user interface. The UI can be composed of standard WPF controls. A set of these customized UI widgets and controls, called the New Experience Identity (NEI), are available in the toolkit. These components should offer a uniform UI feeling across all the multi-modality imaging application developed.

WPF does not natively offer the functionality to render complex multi-modality data (such as X-ray images or CT volumes). For this reason specialized Viewer and Viewport components are introduced. These controls, which seamlessly integrate into the WPF framework, allow users to render clinical data in the applications using customized renderers.

The ViewModel layer contains the UI state and the application logic. Within the Viewing architecture there are three mechanism that allow the communication between the View and the ViewModel

- Binding: Binding links the values of a ViewModel property to a property of a UI control (such as the caption of a button).

- Commands: The command pattern is used to connect events (such as button click or text change events) in the UI with a command property in the ViewModel.

- Interactors: Interactors are used to relay viewer (and viewport) mouse and keyboard events to handlers in the ViewModels.

The model layer contains all of the application's raw data and algorithms. Domain models are needed, for example, when the same data types are retrieved from different data sources (e.g. a CT volume can be saved in a file or a database). Also domain models are useful to separate specific algorithms implementations from their clinical usage (e.g. the same automatic registration algorithm can be used in different applications).

6.4 VesselNavigator Architecture

The VesselNavigator application is developed on top of the Viewing architecture. For this reason its architecture, also based on the MVVM pattern, is similar to the Viewing. On top of the three layer architecture, the application also separates the four tasks previously introduced. This separation does not break the three layer architectture being rather an "orthogonal" separation: each task follows the architectural model of the MVVM pattern by defining its own views and view-models (Figure 15). This workstep based separation is not such that there are no shared viewmodels or views between the worksteps

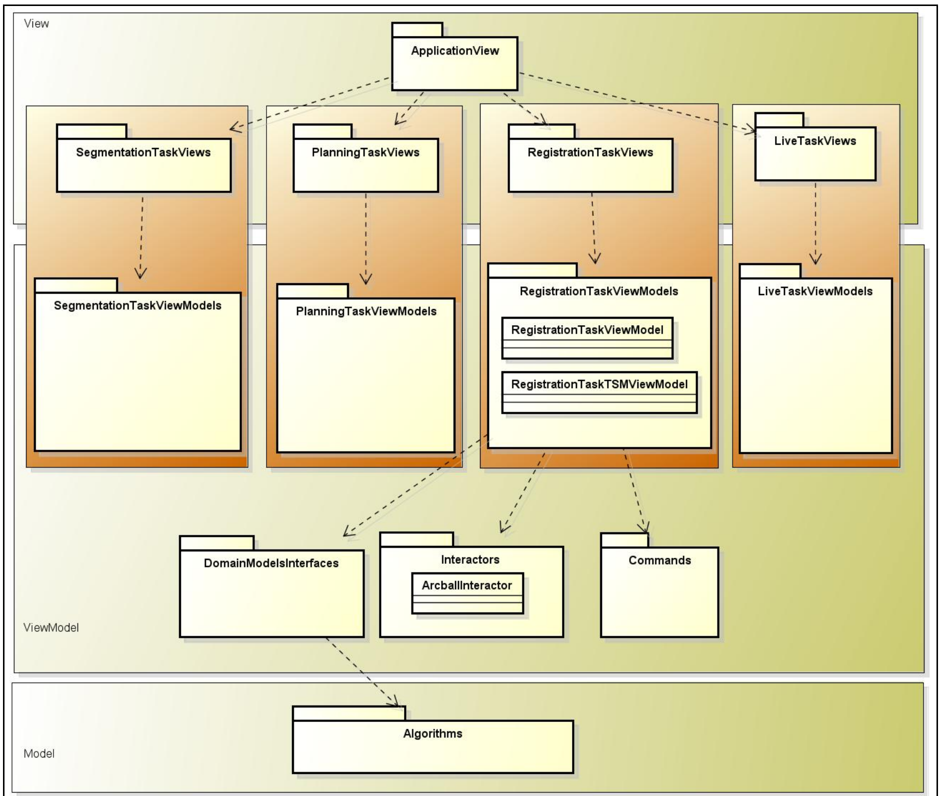


Figure 15 - Architecture of VesselNavigator application

The view layer contains the UI definition of the application. The *ApplicationView* holds the XAML description of the applications structure. For example, it divides the screen into a side panel on the left, which allows navigation between the four application steps, and the region for the views of the selected task in the right.

Each task contains its own UI definition. For example, the *RegistrationTaskViews* contains the UI definition of the registration step. XAML code for the views, viewports, toolbars and the controls needed for this step are defined here.

The ViewModel layer contains ViewModels for each task. The RegistrationTaskViewModels holds all the logic needed to keep the UI of the registration step in a consistent state. It is responsible for relaying the user events to their corresponding command or interactor handlers. Some of these handlers are not specific for the registration step; as an example the Arcball interactor is used in the entire application. The common components (interactors, commands, etc.) are not part of any task and can be placed in the ViewModel layer of the application, or even in the ViewModel layer of the Viewing architecture if there is the need to reuse them in other clinical applications.

Registration is the first step done in the OR. Therefore, while registering, the surgeon may want to use the TSM rather than a mouse. As a result, the ViewModel layer for this task contains view-models to reflect the state of the TSM user interface (the *RegistrationTaskTSMViewModel* block in Figure 15).

The model layer contains domain models, components that are independent on the UI. The algorithms for automatic registration should be stored in the model layer since they are not depending in any way on the UI. These types of algorithms can also be used in different applications. The domain models cannot be directly displayed on the screen. View-models are used to interface them (the *DomainModelInterfaces* block in Figure 15).

7.System Design

Abstract – This chapter presents the detailed design of some relevant parts of the project

7.1 Arcball interactor

The Arcball interactor was introduced to offer users an intuitive way to rotate with 3D clinical datasets. Figure 16 shows the classes involved in the design of the Arcball interactor and the connection with the Viewing architecture.

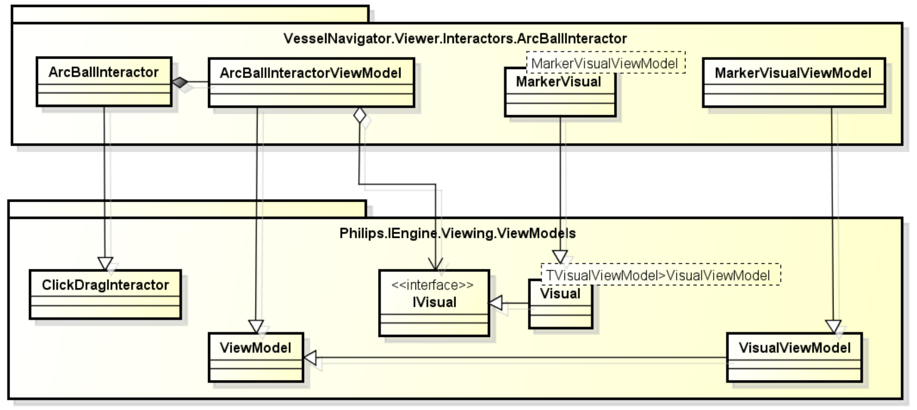


Figure 16 – class diagram presenting the design of the Arcball interactor and the connection with the Viewing architecture

The ArcBallInteractor class contains the code to perform the rotation. When the interactor is created, the "spaceToRoll" parameter is sent representing the 3D scene that will be rotated. The user events are routed to the interactor from the Viewing architecture by extending the ClickDragInteractor.

Besides the actual rotation, some graphical features are also displayed during the interaction. As previous chapter discussed the clinical data cannot be displayed using standard WPF controls. Instead a custom build viewport control is needed to make use of the specialized rendering engines. The graphical features the Arcball interactor introduces have to be displayed on top of clinical data therefore the same rendering engine must be used.

To display any elements in a viewport control the pattern described in Figure 17 is used. The following elements can be distinguished:

- *IX* is a domain model interface. It is used to decouple the data models from the data sources.
- *XVisualViewModel* represents the actual data to be rendered. It contains all the information needed to render the specific data. Most *VisualViewModels* contain a
    - *Space* that encapsulates the position and orientation of the object to be displayed
    - *DescriptionViewModel* that contains all the "raw" data needed.

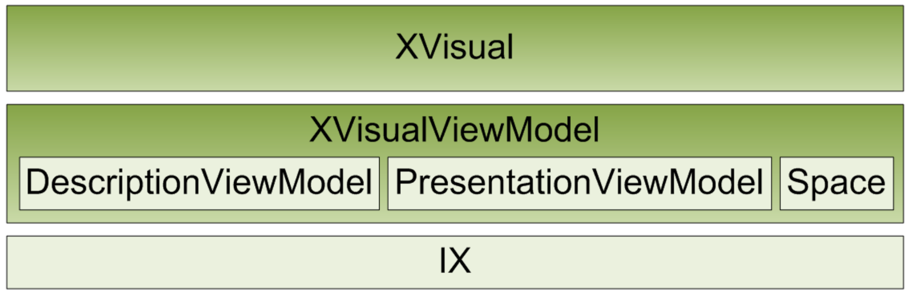


Figure 17 - The design of the ViewModel layer for features displayed in the Viewports

- PresentationViewModel that contains “presentation” data. This is information that is only used for rendering and is not specific to the object. For example color information is usualy stored in the PresentationViewModel.

- *XVisual* contains code that map the *VisualViewModel* to objects used by the rendering engine.

The Arcball interactor uses the same pattern to display its graphical elements, the marker:

- *MarkerVisualViewModel* contains the information needed to display the marker: the position of the rotation center and the current position on the sphere. The *MarkerVisualViewModel* holds only description information since “presentation” data, such as elements colors, is not configurable. Therefore, there was no need to split the *VisualViewModel* into a *DescriptionViewModel* and a *PresentationViewModel*.
- *MarkerVisual* contains code that translate the description information contained in the *MarkerVisualViewModel* to visual components that the rendering engine can display in the *viewport*.

A notification mechanism is implemented in the Viewing architecture to inform the *Visual* when the *VisualViewModel* is changed. By using this mechanism, which is based on the Observer pattern, the *VisualViewModel* is decoupled from the *Visual*. This has the advantage that different *Visual* versions can be used without changing the *VisualViewModel*. Therefore the details describing how the marker will be presented on the screen are localized in the *MarkerVisual* class.

One of the requirements of the Arcball interactor was to provide a way for external components to specify the location of the rotation center. To achieve this ArcBallViewModel was introduced. This ViewModel has a *Space* parameter in its constructor. This *Space* will represent the 3D scene in which all the graphical elements of the Arcball are placed: the *VisualsSpace*. Moreover the origin of the *VisualsSpace* is the initial rotation center. To modify the center of rotation, transformations can be applied to *VisualsSpace*.

The *ArcBallInteractor* is responsible for intercepting the user events and based on them do all the geometrical calculations to determine the rotation required. It does not know about the graphical elements that are displayed on the screen. The *ArcBallViewModel* is used to obtain this decoupling. This class is responsible with creating all of the *Visuals* needed (like the marker) and update their ViewModels. The user has to manually add these Visuals to the *Viewport* (a usage example is presented in section 8.1.4.).

7.2 Angle Hints

Angle hints were introduced to provide physicians suggestions about the position of the C-arc that will provide a good image for registration. Section 9.5 presents the values of the suggested angles. In this section the focus will be on the design of this feature and its integration into the VesselNavigator application.

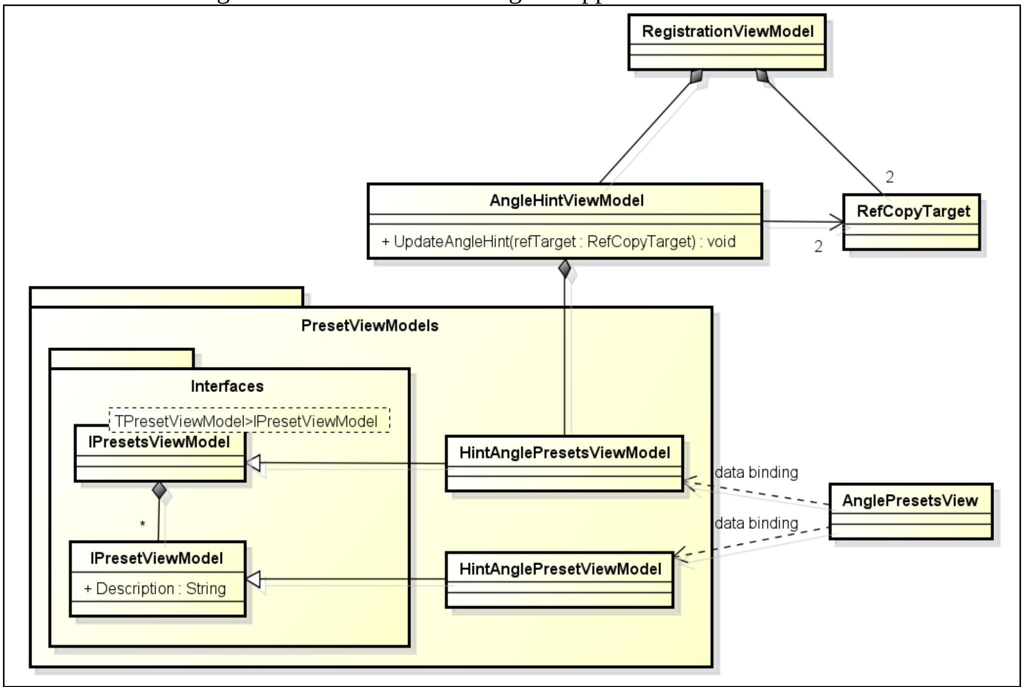


Figure 18 - class diagram presenting the design of the Angle Hints feature

The classes involved in the design of the angle hints are presented in Figure 18.

- *RegistrationViewModel* contains all the ViewModels that describe the state of the registration screen.
- *RefCopyTarget* class holds information about the X-ray runs contained in a reference screen. One instance of this class is created for each reference screen.
- *AngleHintViewModel* contains the logic for calculating the angles that will be hinted to the physician.

Presets are used in many places in the VesselNavigator application when the user needs to select from a list of similar settings the one that he wants to use. For example different visualization settings for the CT volumes are stored as presets. For this reason a *Presets* infrastructure was defined to have a uniform usage pattern. The *Presets* infrastructure defines two interfaces must be implemented. The first interface *IPresetViewModel* acts as a domain model interface. It only contains a Description property but other details can be added as needed in the implementing classes. The second interface, *IPresetsViewModel*, holds a list of available presets (described by an *IPresetViewModel*). Since presets are used to select between multiple options in the UI, the interface also contains information about the selected preset and the action to be taken when the *Preset* is selected.

To display the hint angles in the UI, the existing “Presets” infrastructure is used. For this the two interfaces described above are implemented by HintAnglePresetsViewModel and HintAnglePresetViewModel classes. The user interface is described in XAML code (the AnglePresetsView in Figure 18). WPF data binding is used to keep the synchronization between the UI and the selected preset in the *HintAnglePresetsViewModel*. Moreover a WPF command is used to link the user action of selecting a specific angle with the handler responsible to send that angle to the Allura machine.

Angle hints are used during the registration step which is the first step of the application designed for usage in the OR. Therefore the physician might only have the possibility to interact with the application using the TSM. For this reason the angle hints must be available in the TSM interface. Using WPF standard commands and data bindings provides the advantage of a loose coupling between the UI and the ViewModel layer. Following the MVVM pattern, the TSM UI can be easily connected to the same *HintAnglePresetsViewModel* as the AnglePresetsView. In this way the two interfaces are kept synchronized

8.Implementation

Abstract –This chapter describes implementation details of the added functionality. First a detailed description of how the Arcball interactor is presented, next we will describe C-arm angle calculation and the orthogonal interactor in “cone beam space”.

8.1 Arcball Interactor

The Arcball interactor offers users a fairly natural way of interacting with a 3D volume using the mouse. The main problem here is that, while mouse can only move on a plane, a 3D volume can be rotated in all three dimensions. In order to transform the 2D mouse movement to an intuitive 3D rotation the Arcball interactor proposes to simulate the rotation of a virtual sphere. This sphere is fixed to the 3D volume therefore when it rotates the volume also rotates. The following steps define how the sphere rotates:

1. **Select the sphere center**. The sphere will always rotate around its center (while rotating the sphere, the center point remains fixed)
2. **Select an initial point**. This is the first point on the sphere the user selects with the initial click. In order to make the Arcball interactor intuitive, when the user moves (drags) the mouse, this point should follow its movement. In other words the user should “feel” like he rotates the sphere by holding his finger on this point.
3. **Define the rotation.** When the mouse is dragged the sphere will rotate such that the restriction previously stated is satisfied: while dragging the initial point should always be the projection of the mouse position on the sphere surface.

8.1.1. Selecting initial point

On the screen the sphere will appear as a circle (deformed circle in case of perspective projections). The initial point that the user clicks can be either inside or outside of this circle (see Figure 19). In the first case, the user can select for example point A, a point inside of the circle. The projection of point A on the sphere will be the selected initial point. In the other case the user selects a point B, a point outside of the circle. In this case point B1, which is the closest point to B on the circle, is actually selected. The initial point will be the projection of B1 on the sphere surface.

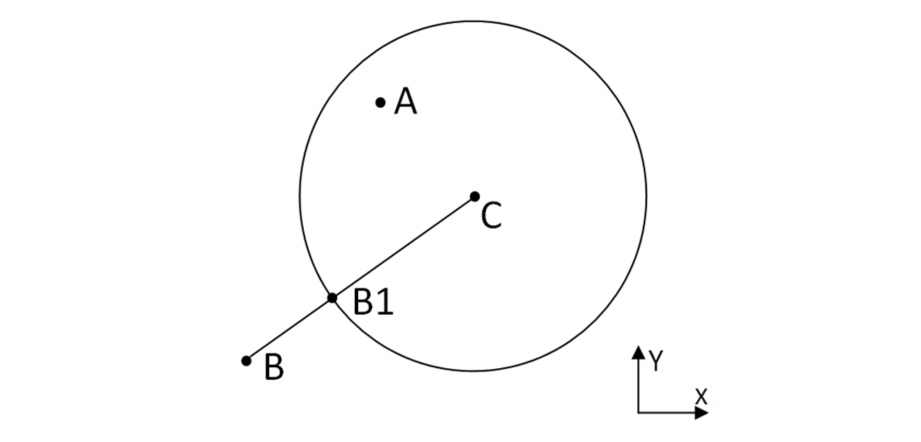


Figure 19 – Example of how initial points are selected on the sphere from the user’s point of view. The circle represents the projection of the sphere on the screen.

8.1.2. Rotation

When the user drags the mouse on the screen the sphere should be rotated such that the initial point will follow the mouse movement. The rotation of the sphere can be split into a series of "elementary" rotations that will move the initial point successively through the discrete set of points representing the mouse path.

An object in 3D space has six degrees of freedom. This means that six parameters can completely describe its position and orientation. Two restrictions were imposed to the sphere rotation. First one stated that the center of the sphere should not move during the rotation which is equivalent to reducing the spheres' degrees of freedom by three. The second restriction specified that the selected initial point on the sphere should follow the mouse movement (on drag). To achieve this two rotations are required (see Appendix D: Calculation of rotation and angulation from C-arm orientation matrix) reducing with two the degrees of freedom of the sphere. Since we only restrict five of the six freedom degrees, there is not a unique rotation that will satisfy the two conditions. In the example presented in Figure 20, moving from A to B can be done by rotating the sphere in two ways:

1. On circle[2] 1 from A to B
2. On circle 2 from A to an intermediate point C1 and then on circle 3 from C1 to B

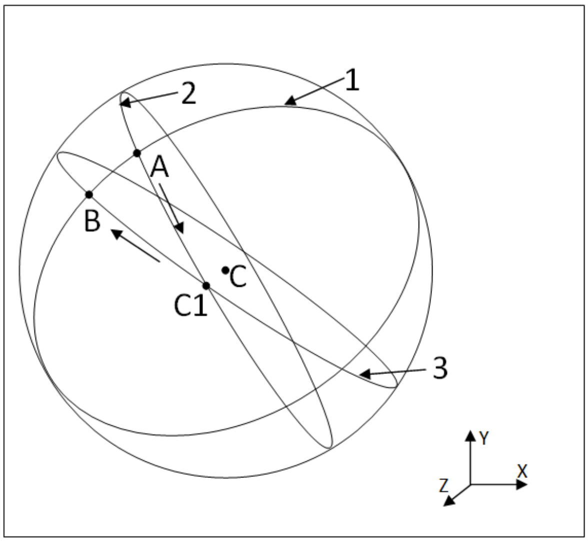


Figure 20 – example presenting different possibilities to rotate the sphere so that point A goes to position B

The first option is the most natural way to perform an "elementary" rotation. However, if we use this approach for performing "elementary" rotations, the final orientation of the sphere will not depend only on the final position of the mouse, but also on the path the mouse followed. Figure 21 exemplifies this problem. In this example, the initial point is represented by the blue sphere. In order to bring it from position A on the screen to position B, the user has two possibilities:

1. Move the mouse following path 1, equivalent to a 90 degrees rotation around X axis. Figure 21 b illustrates this rotation.
2. Move the mouse following path 2 (rotate 90 degrees around Y axis) then path 3 (rotate 90 degrees around Z axis). Figure 21 b and c present these two rotations.

[2] Rotation on a circle is the rotation around the axis that goes through the center and is perpendicular to the circle plane

The orientations of the body in Figure 21 b and c are different although the initial point moved from A to B in both cases.

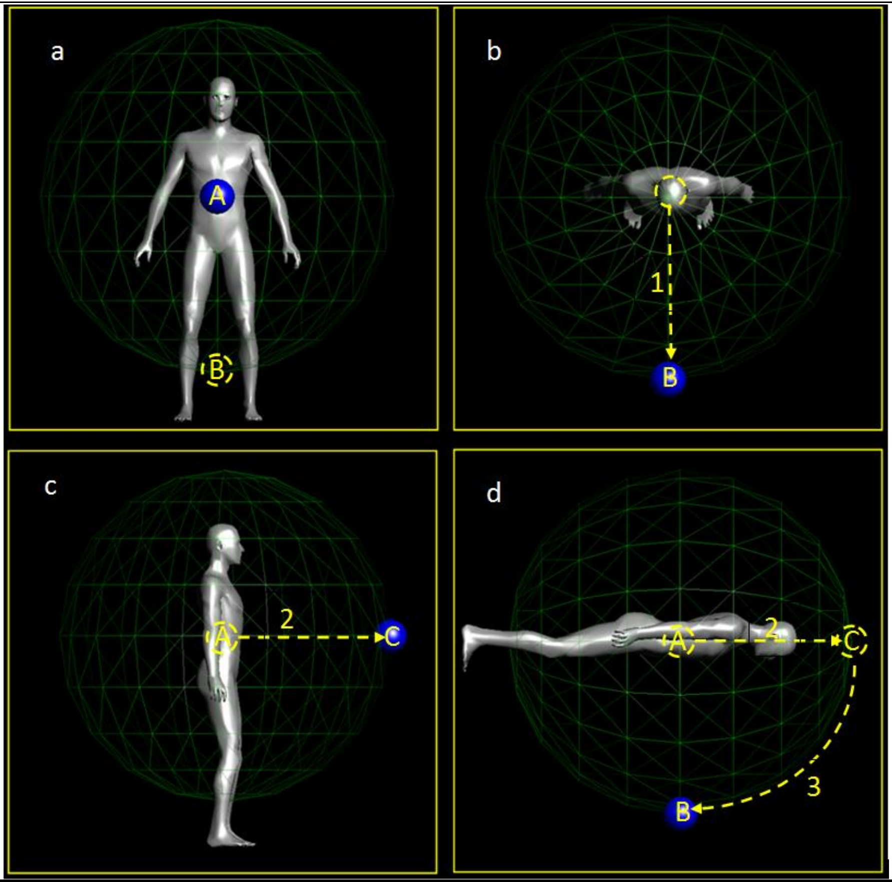


Figure 21 – Exemplification of the orientation problem for the first rotation method presented. The initial point A is brought to position B in two different ways: following a 90 degrees rotation around X axis (b), and following a 90 degrees rotation around Y axis (c) and then a 90 degree rotation around Z axis (d)

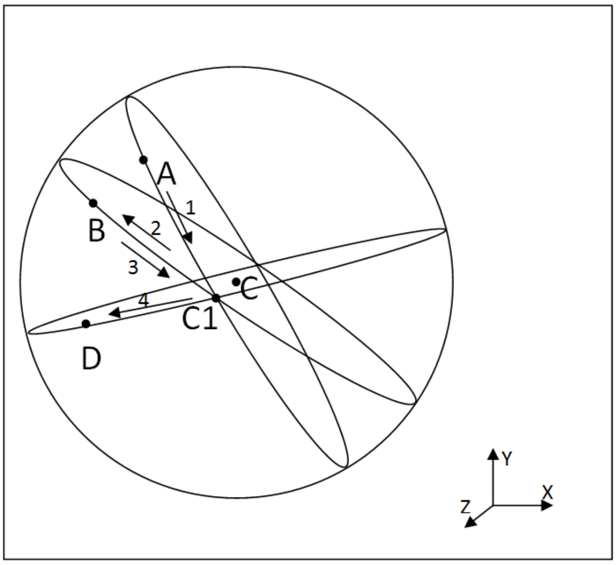


Figure 22 – Example showing how the second rotation method solves the orientation problem. Rotations 2 and 3 are inverse transformation therefore they cancel each other.

The second option presented requires two rotations, forcing the path of the initial point to go to an intermediate location (C1) on the sphere. This way of rotating preserves the orientation even when different paths are followed by the mouse. Figure 22 illustrates this behavior. Here the sphere is moved so that point A goes to B following the path A->B then B->D. In this case the four rotations presented in the figure are executed. Rotating from C1 to B and then from B to C1 does not have any effect (rotations 1 and 2 are inverse transformations). This means that applying only rotations 1 and 3 has the same effect as applying all the four initial ones and hence, point B does not influence the final orientation.

No assumptions were made about the position of the intermediate point C1. This means that the behavior presented before will be preserved regardless of the where this point is. To make the rotation more intuitive we choose C1 to be the point on the sphere closest to the screen (this point will have the same projection on the screen as the center). In this case moving the mouse horizontally through the sphere center (the projection of the sphere center) will generate a rotation around the Y axis while a vertical mouse movement through the same point will generate a rotation around the X axis.

8.1.3. Fixed axis rotations

Besides the free rotation, the interactor offers the possibility to fix the rotation axis to any of the coordinate axes (X, Y and Z).

For visualizing this fixed-axis rotation it is good to realize that, in this case the initial point cannot follow the mouse to every point on the sphere. In Figure 23 the user drags the mouse from point A to point B and the rotation is locked to the axis $\vec{v}$. In this case the sphere will rotate around axis $\vec{v}$ with angle α where α is the angle between AA’ and BB’.

In case of locked axis rotation the position of the initial point will remain on a circle. This circle is the intersection of the sphere with a cylinder. The axis of the cylinder is the rotation axis and its radius is the distance from the initial point to the rotation axis.

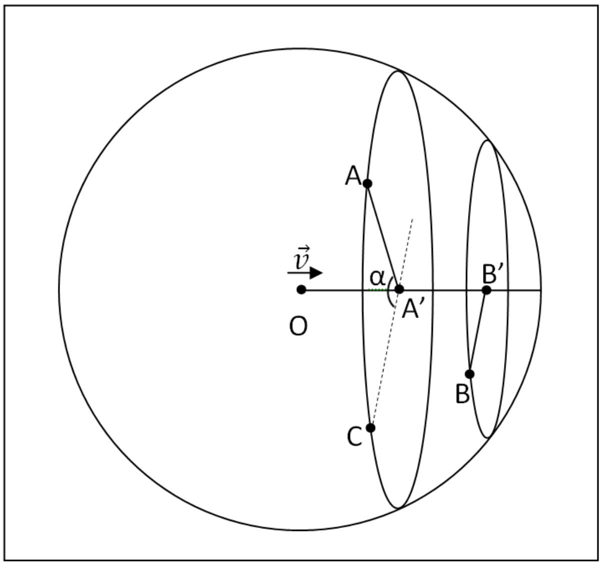


Figure 23 – example of fixed rotation axis $\vec{v}$. Dragging the mouse from A to B will generate a rotation with angle α. The final position of point A will be C.

Z-axis rotation

The Z axis rotation or “in-plane rotation” is performed when the initial click point is outside of the sphere.

If we are using an orthogonal projection view, the rotation axis is the line that goes through the sphere center and is perpendicular to the screen plane.

However, if we look at our scene in inverse perspective as described in section 2.4.1. , the rotation axis should be defined as line that goes through the center of the sphere and the projection origin (Figure 24).

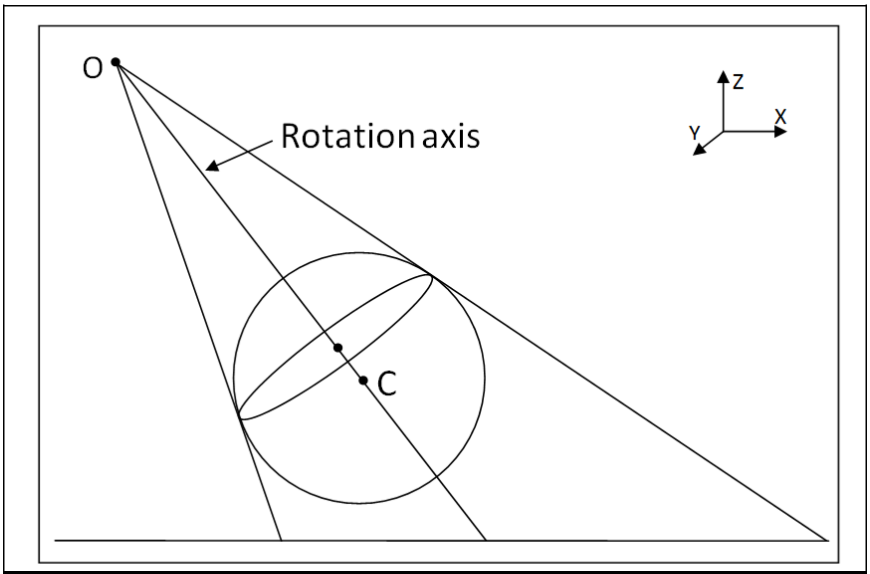


Figure 24 – Rotation axis for Z-axis locked rotation.

X and Y-axis rotation

The X or Y axis rotation is performed when the “Left Shift” key is hold while dragging. Axis X is selected if the initial mouse move was in vertical direction while Y axis is selected when the initial mouse move was in horizontal direction.

In an orthogonal view the X and Y axis of the 3D space are the same as the X and Y axes of the screen. For this reason when locking the rotation axis to X, for example, the initial point will only move on a vertical line on the screen (the point moves on a circle which looks like a line when projected on the screen). To keep this behavior in case of perspective projection, it is not possible to use the coordinate axes of the 3D space. Figure 25 shows the rotation axis that will be used when the user selects to lock the rotation on the X-axis when the initial point is A. If we rotate the selected axis point A will only move in the vertical direction on the screen (point B represents the final position of the point). The orientation of the rotation axis is dependent on the relative horizontal position of the initial point to the sphere center. The dotted axis represented in Figure 25 shows the rotation axis that will be used when the initial point will be to the left of the rotation center.

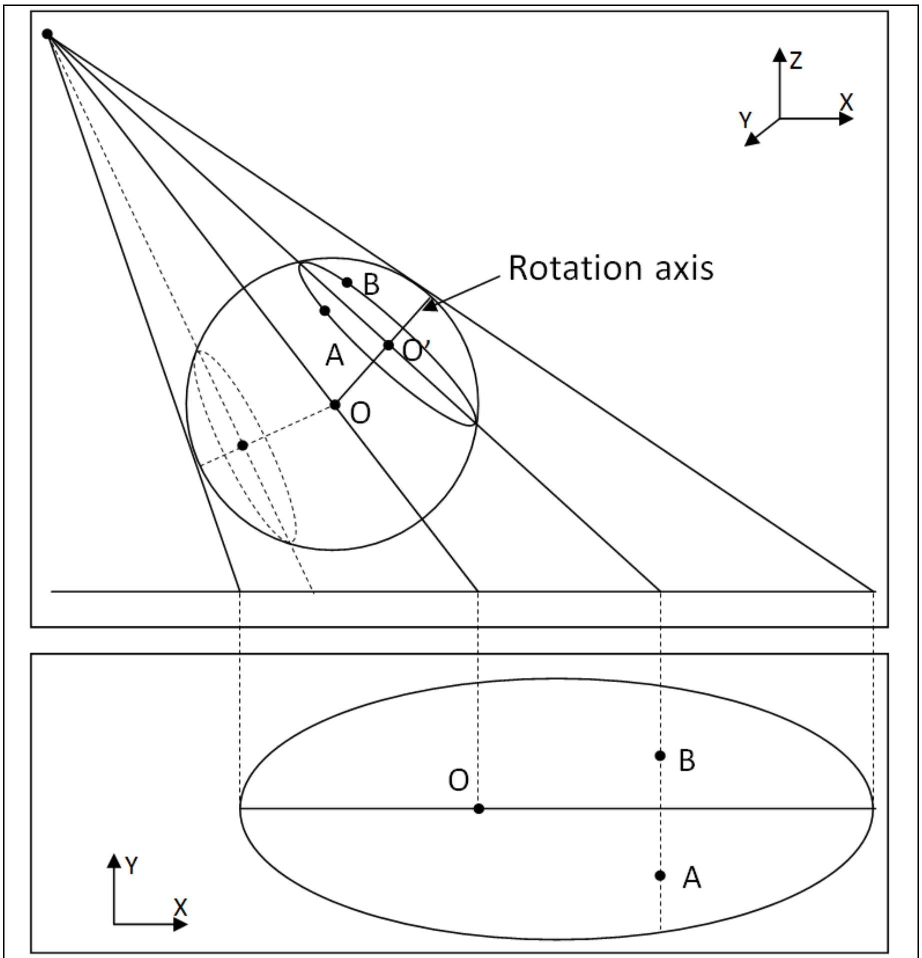


Figure 25 – The rotation axis for locked X-axis rotation for perspective projection. The bottom part represents the screen projection of the elements presented in the upper part.

8.1.4. Usage example

Let us now give a usage example for the Arcball interactor:

```
1 var arcballViewModel =
       new ArcBallInteractorViewModel(DisposeStack, visualsSpace);
2 var arcballInteractor = new ArcBallInteractor(
      InteractorTrigger.LeftMouse,
      spaceToRoll,
      arcballViewModel);
3 ViewportViewModel.Add(ArcBallInteractor);
4 ViewportViewModel.Add(arcballViewModel.Visuals);
5 ViewportViewModel.Add(
      new LayerViewModel(DisposeStack, arcballViewModel.OverlayVisuals));
```

In line 1 a new view model is defined for the Arcball interactor. The space in which this view-model is defined is important since its origin is the initial rotation center.

Line 2 defines the interactor with the left mouse as the trigger. “spaceToRoll” is the space the interactor will rotate. This can also be the camera space for orthogonal projection.

To make the graphical elements of the Arcball visible (the center axes and the position marker) they must be manually added to the viewport (line 4). In some cases it is preferred to make these elements (semi-transparently) visible even if they are obstructed by other objects in the “spaceToRoll”. To achieve this we can add duplicate visuals, with a certain transparency, in a higher viewport layer (line 5).

For the correct usage of the Arcball the user must correctly define the three spaces involved: the camera space, the space to roll, and the visuals space. For example if it is desired to rotate a specific volume, it makes sense to place the visuals in a sub-space of the volume space. For a complete description of the possible configurations see: Appendix C: Space configurations for Arcball interactor

8.2 C-arm angle calculation

The physician has the possibility to visualize parts the patient’s body using 2D or 3D modalities. When looking at these images it is important that he knows their position with respect to the patient. For example when looking at an X-ray image it is important to know the configuration of the C-arm that generated that image. During an endovascular procedure knowing the orientation of the live X-ray images is crucial for accurate navigation. Moreover in the case of the VesselNavigator application in the planning step different viewing angles for a volume can be saved to be recalled during the procedure. This requires calculating the position of the C-arm that will generate that specific viewing angle.

To completely describe the orientation of a point in 3D space three parameters are needed (position is known). In the same way to completely describe the position of the C-arm that generated a specific X-ray image, three rotation angles are required. These are the angles of rotation around the three coordinate axes (Figure 7 displays the C-arm and how it can rotate).

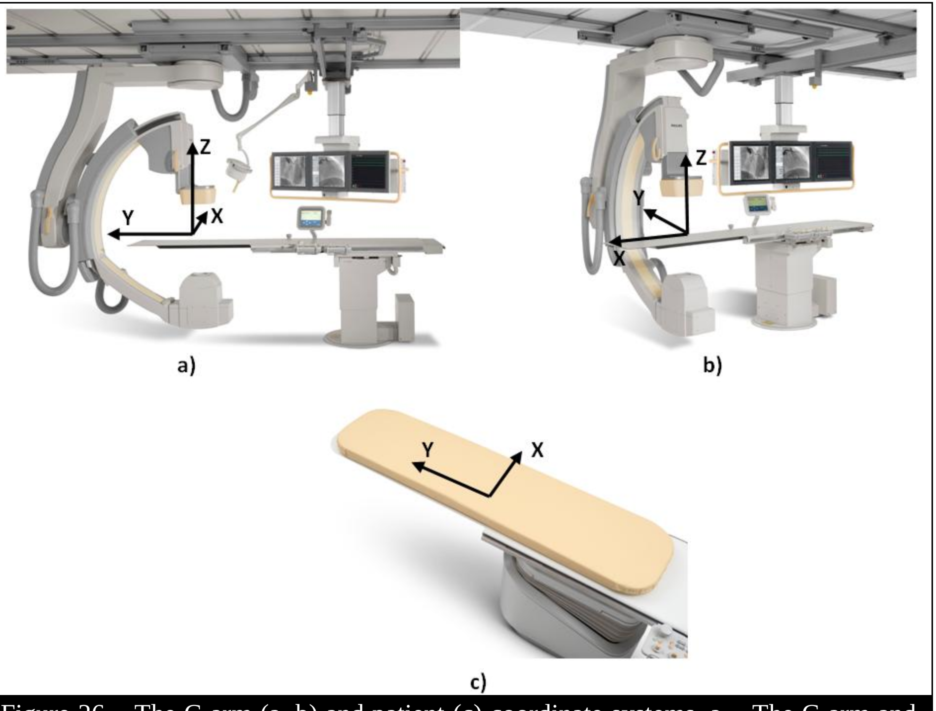


Figure 26 – The C-arm (a, b) and patient (c) coordinate systems. a – The C-arm and patient coordinates match. b- The C-arm and patient space are rotated 90 degrees to each other.

The C-arm configuration that generated a specific X-ray image is described by the direction the X-rays were sent to the patient (the normal to the image’s plane). Knowing this direction does not describe the complete C-arm orientation. Rotating the C-arm around it will result in an in-plane rotation of the X-ray image. This rotation does not change the information presented in the image so it can be ignored. For this reason, only two angles are needed to describe the direction of X-rays.

The orientation information presented to the physician is expressed in patient's coordinates (Figure 26 c): the rotations around the X (angulation) and Y (rotation) axes. When the C-arm is in the "head" position (Figure 26 a) these correspond to the angles the C-arm is rotated around its X and Y axes. When the C-arm is in a different position (rotated around its Z-axis as in Figure 26 b) the two coordinates systems (C-arm and patient) are not the same anymore. Therefore the angulation and rotation are not the same as the C-arm X and Y rotations.

The orientation displayed on the screen is always the angulation and rotation (the rotations around X and Y axis of the patient coordinate system). This two angles are, in general, different from the one the C-arm is actually rotated therefore they must be calculated. The algorithm for this is presented in Appendix D: Calculation of rotation and angulation from C-arm orientation matrix

8.3 Orthogonal interactor

Section 8.3 5.2 already discussed the benefits of using an orthogonal interactor to speed up the process of align a landmark with two one-dimensional reference images. In the real life scenario the volume needed to be registered is a 3D object and the reference images are two-dimensional. In this case aligning the volume with one reference image requires a 2D translation of the volume. Using a similar approach, as in the simplified example, when aligning the volume with image #1 we can translate it in a plane orthogonal to the image #2. This way the volume's projection on the plane of image #2 will only be translated in one direction (the intersection of the volume translation plane and the plane of the image #2). Although in this case aligning a landmark with one image will still cause the landmark's projection to move in the second image, the complete registration can still be done in only two steps. After the first alignment the landmark will only have to be aligned on a direction orthogonal to the first image plane.

As we explained before, when aligning the volume with one image we want to translate it in a plane orthogonal to the second image. This means that the translation plane should contain the normal to the second image plane. There are an infinite number of possible translation planes from which we can choose. Figure 27 presents two such possibilities.

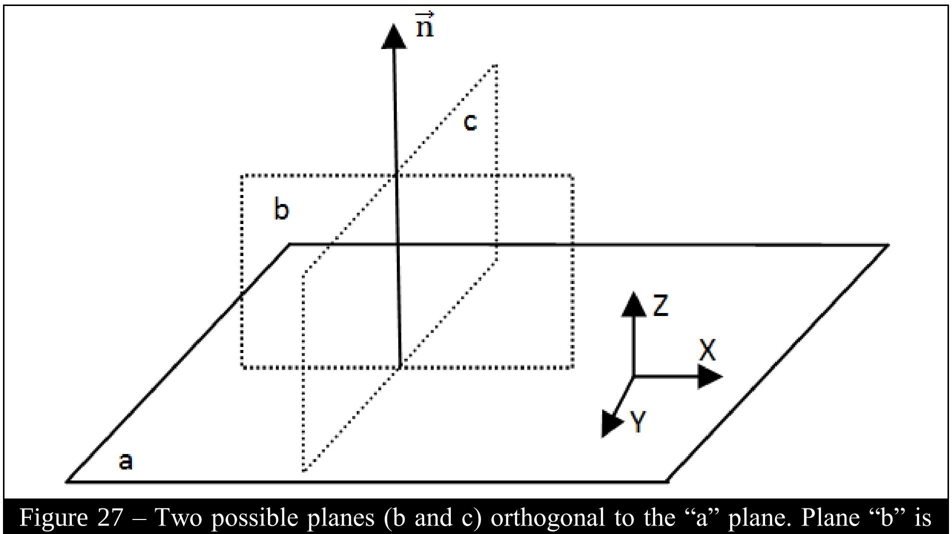


Figure 27 – Two possible planes (b and c) orthogonal to the "a" plane. Plane "b" is defined by the $\vec{n}$ direction (normal to the "a" plane) and the X-axis of the "a" plane. Plane "c" is defined by the $\vec{n}$ direction and the X-axis of the "a" plane.

To make the orthogonal interactor more intuitive we can choose the translation plane from one of the two possibilities presented in Figure 27. These 2 options are "convenient" since the volume's projection will only be translated horizontally or vertically during the alignment with the other reference image.

When an inverse perspective is used the normals to the plane in different points are not parallel. Figure 28 exemplifies this problem for the simplified case of a 2D volume. Here $\vec{a}$ and $\vec{b}$ are the directions normal to the image plane in two different points (the projections of points A and B). In this case if the point A is already aligned with the first image we would like to move the volume parallel with the $\vec{a}$ direction when aligning it with the second reference. Translating along the $\vec{a}$ direction will cause a translation of the projection of B point. Similarly if point B is already registered we would like to translate along the $\vec{b}$ direction which will cause a translation of the projection of A point.

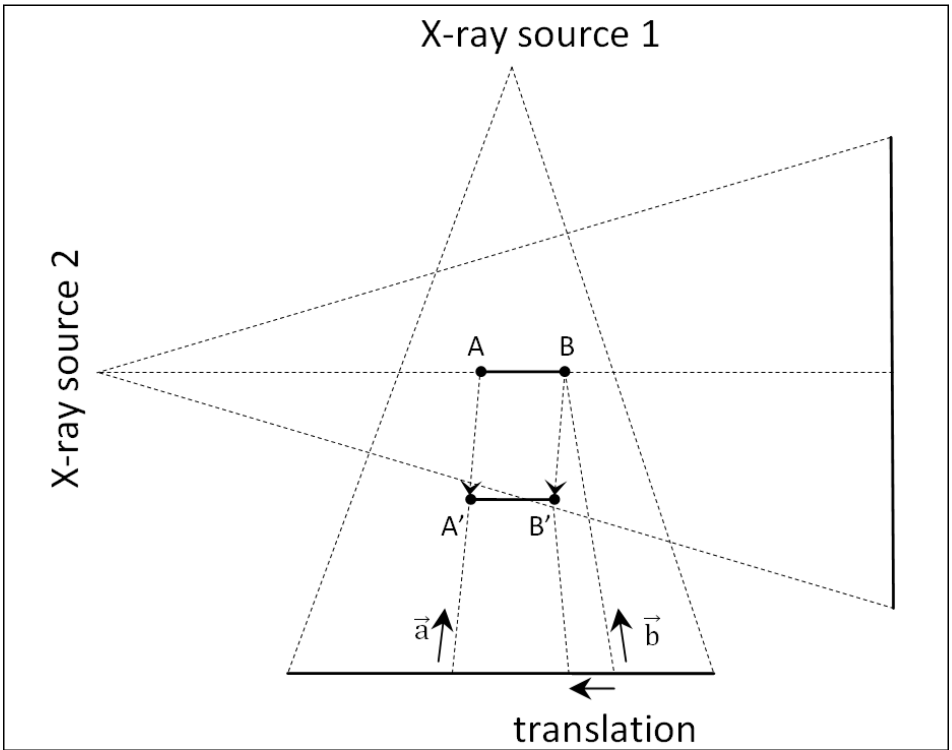


Figure 28 – Orthogonal translation of a volume in inverse perspective projection. Translating the volume along the $\vec{a}$ direction (the normal in the point where A is projected) will lead to an orthogonal translation of the A point (its projection will not move) but the projection of the B point will be translated.

The previous example showed that in order to define the good translation direction we need to know what point we are interested in aligning. This is not possible in our case because when aligning the volume in one image we are missing the depth information. For example the point A and B in Figure 28 are projected in the same point on the second reference image hence we cannot distinguish between them based on the information presented in this image. This means that we cannot know if we need to translate the volume along the $\vec{a}$ or the $\vec{b}$ direction. We can use the depth of the volume center or even a hit test to obtain a good approximation of the depth of the point we are interested in translating

9. Results

Abstract –This chapter describes how the requirements of the project were fulfilled.

9.1 Introduction

The main goal of the project was the improvement of the registration step of VesselNavigator application. The initial focus was on improving the usability of manual registration. For this a series of requirements derived from the needs of the stakeholders. Section 5.4 provides this list of requirement. In this chapter, using application screenshots, we will explain how these requirements were fulfilled.

9.2 Arcball interactor (R1-R5)

Section 8.1 describes in details the way the Arcball interactor works. Besides the actual rotation, graphical elements were required to help users orientate (Figure 29 displays these components).

- To visualize the rotation center, during the rotation, a set of three axes are displayed. These axes rotated in the same time with the volume helping the user understand how the volume rotates.
- To show the position of the initial point on the sphere a marker was added. This marker is composed of a small cone “glued” to the sphere in the initial point and a line that unites it to the rotation center. The cone orientation changes with the sphere rotation offering users feedback on how the rotation is actually preformed.

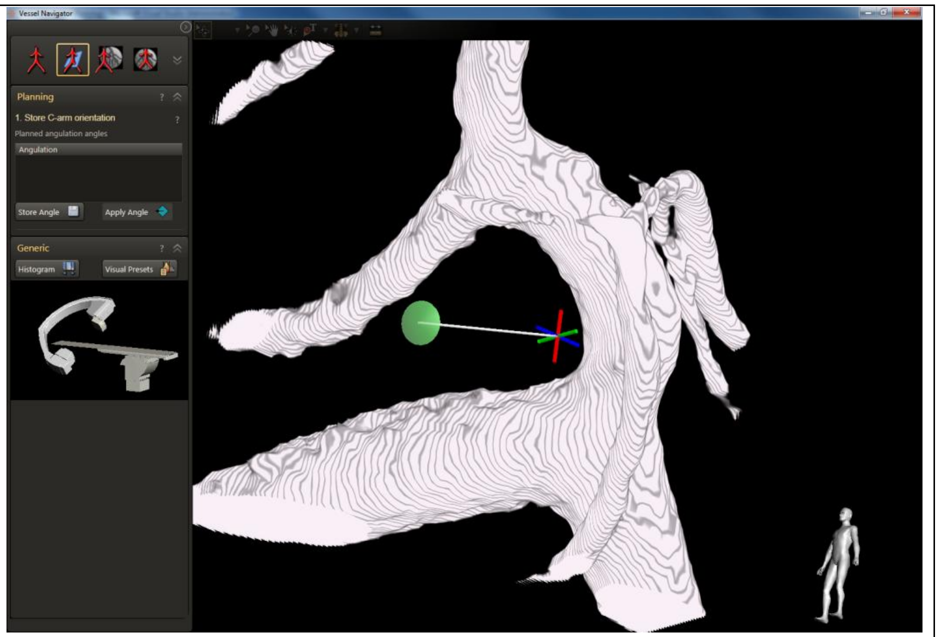


Figure 29 – Screen capture from the VesselNavigator application depicting the graphical elements added by the Arcball interactor. The three axes can be seen in the center of the screen. The marker shows the initial point, the sphere contact point, and is connected to the rotation center with a line.

Rotation around the three coordinate axes was also implemented. For the rotation around the X and Y axes the marker had to be modified to make the rotation more clear. Showing the rotation axis instead of just the rotation center provides a better visual feedback. Figure 30-30 present the fixed axes rotation.

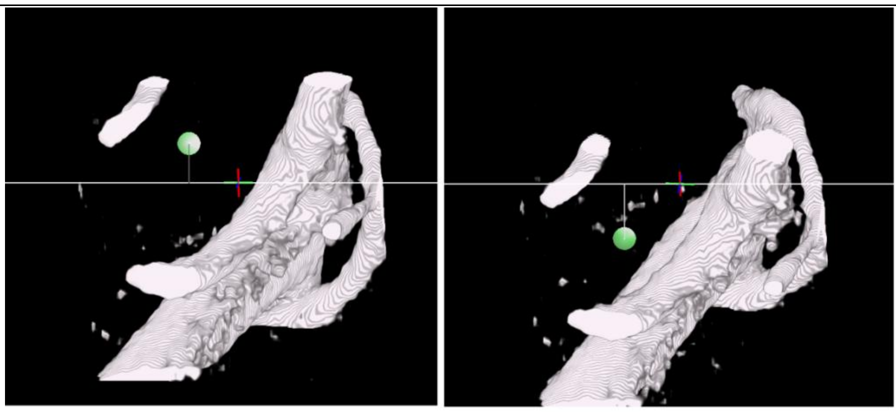


Figure 30 - example of X-axis rotation

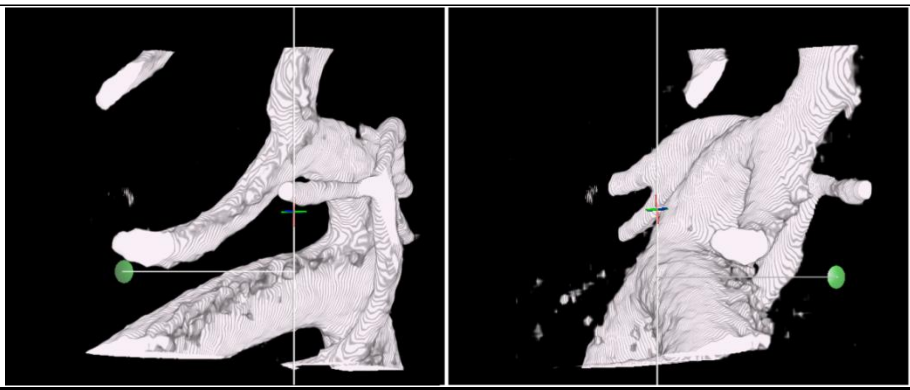


Figure 31 - example of Y-axis rotation

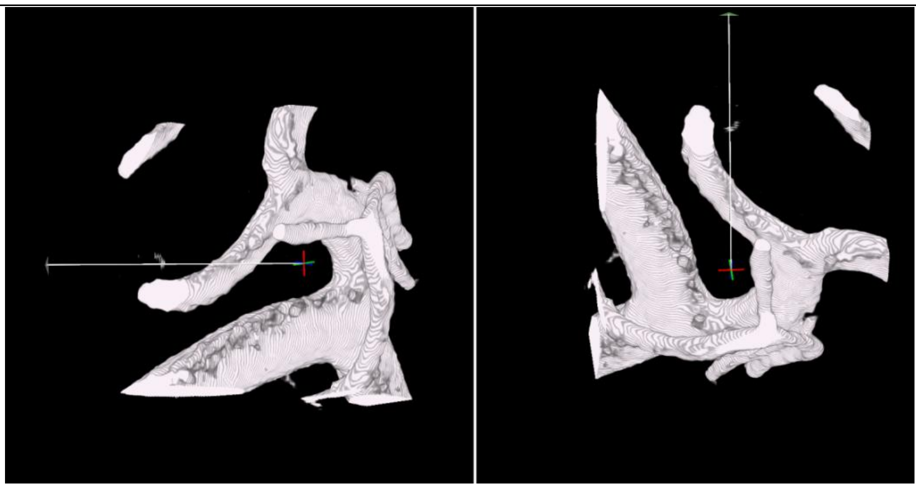


Figure 32 - example of Z-axis rotation

9.3 Orthogonal interactor (R6)

The following screenshots provide an example of how the orthogonal interactor can be used to register a volume with two reference X-ray images in just two steps. Figure 33 shows a CT volume of a thorax phantom overlaid on two X-ray images acquired from two different angles. In this initial configuration we can see that the volume is unregistered since it doesn't match in any reference. In Figure 34 the volume is aligned in the first reference but is still unaligned in the second one. In Figure 35 the volume is aligned in the second reference. This second alignment did not break the alignment in the first reference therefore the volume is completely aligned in after the two steps.

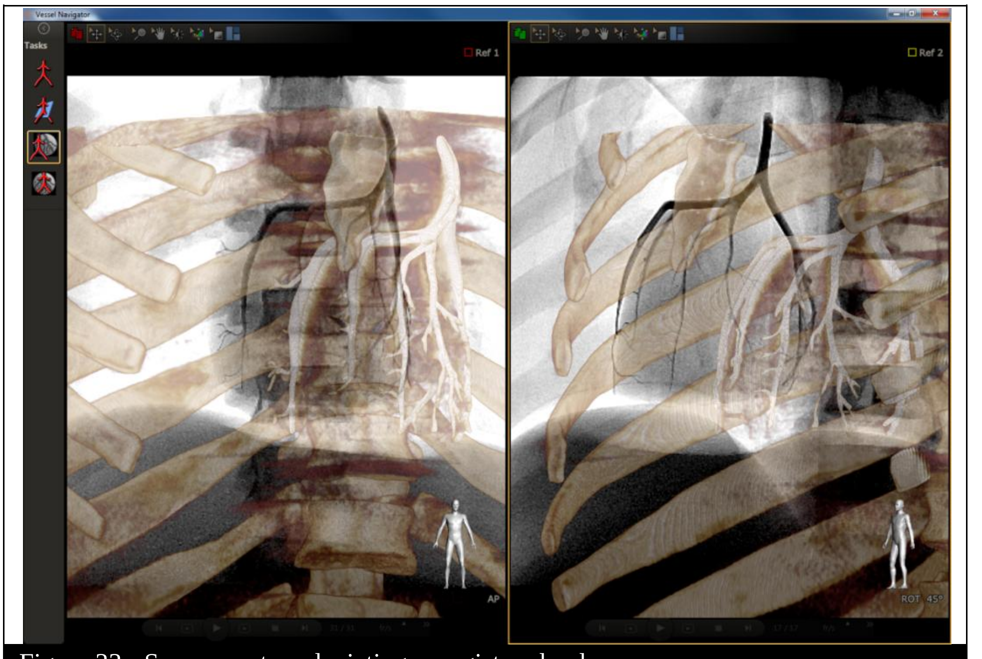


Figure 33 - Screen capture depicting unregistered volume.

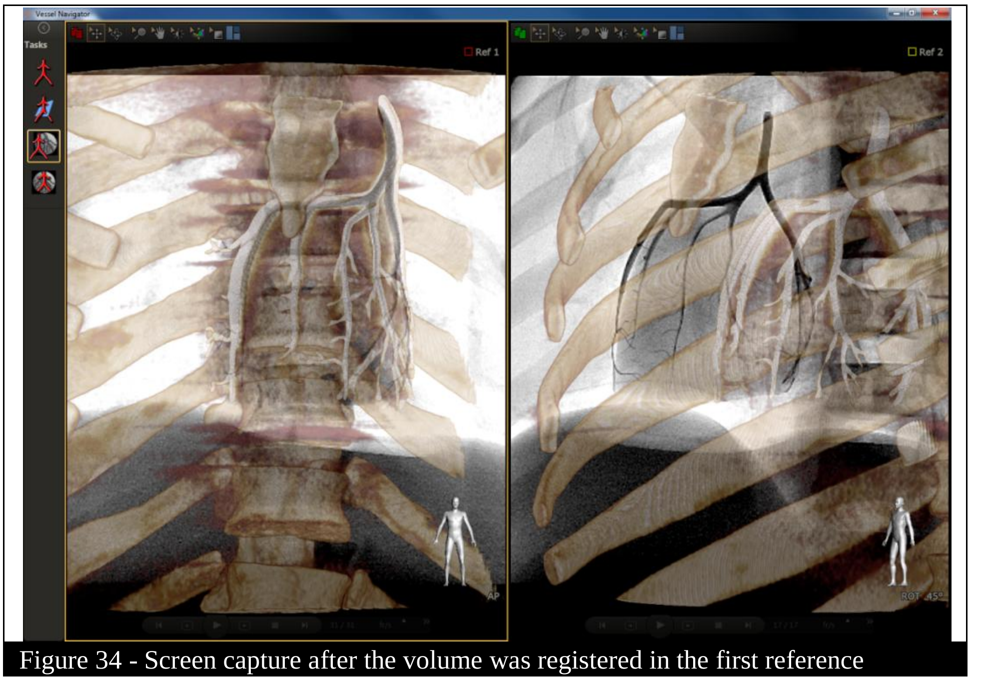


Figure 34 - Screen capture after the volume was registered in the first reference

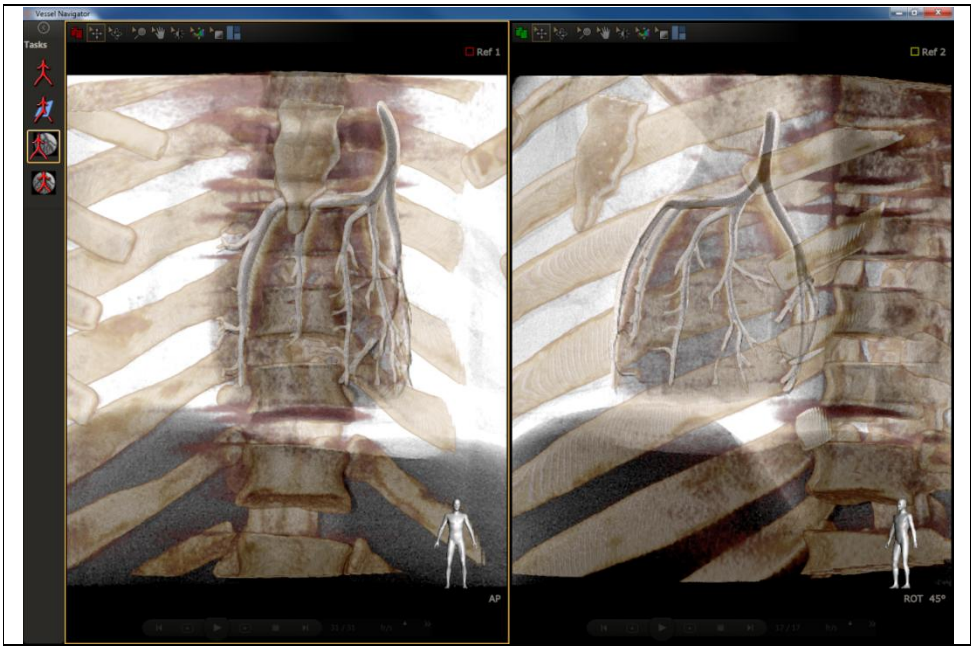


Figure 35 - Screen capture showing the fully registered volume. The registration in the second reference screen did not affect the previous alignment in the first reference

9.4 Reference copy hint (R7)

In the beginning of the procedure both reference screens in the registration step are empty. Whenever the physician starts sending X-rays to get real time images from the patient these images are recorded. After one of these runs he has the option to copy it to one of the reference screens to use it for registration. When both references are filled the registration process can start. In order to be able to make a good registration the images must be acquired from different angles. The angle between the two images must be as large as possible (ideally 90°) for greater precision. The physician has the possibility to copy the latest X-ray run to one of the references even though this is not empty (he can override the previous reference). To make this process faster the application will suggest the reference screen that should be used to copy the latest run. Since this is just a visualization hint the physician will still be able to copy a run in either of the two references.

When hinting the surgeon about one reference screen, its copy button will turn green and blink. To make the hint more obvious in the other reference the copy button will turn red. The hint will also be reflected on the TSM where the copy button associated with the hinted view will also blink (Figure 37). Deciding which reference screen to hint is influenced by their content:

- Both reference screen are empty – The application will suggest first reference
- One reference screen is empty - The application will suggest the empty reference if the angle between the current image and the one in the filled reference is big enough (>20°). The already filled reference is suggested in the other case.
- None of the reference screens are empty – In this case the application will try to suggest the reference screen that will generate the highest registration angle. The angle threshold (20°) is also necessary here to suggest one reference. In the example presented in Figure 36 the second reference is suggested since the angle between the latest run and the first reference is 90° (compared to only 45° angle with reference 2)

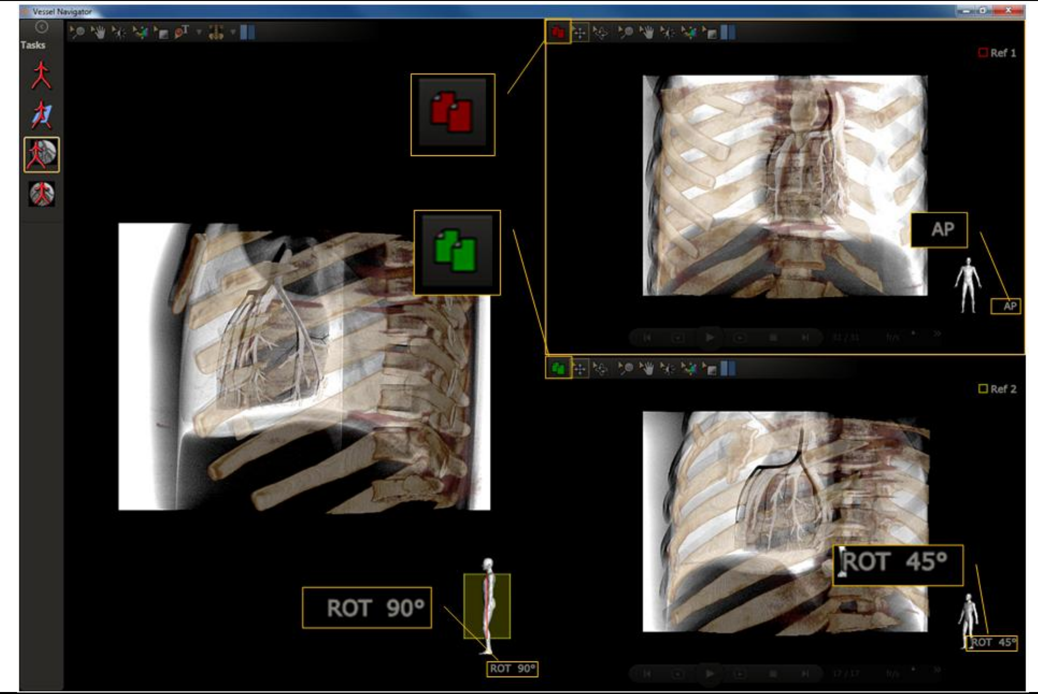


Figure 36 – example of a reference copy hint.

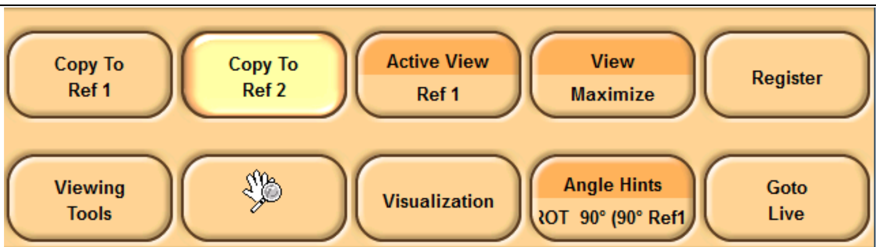
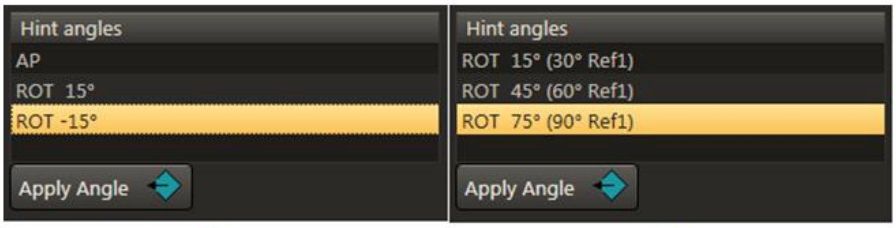


Figure 37 – the visualization of the copy hint on the TSM

9.5 Angle hints (R8)

For a good registration process the angle between the two reference images must be as big as possible. To do this the physicians need to bring the C-arm in the right position for the first reference, take an X-ray shot, then move the C-arm to the second position and repeat the procedure. In order to try to help physicians, suggestions for these angles were added to the Vessel Navigator application. Initially if both reference screens are empty some default angles are hinted (the values are presented in Figure 38 a). In later stages the hinted angles are calculated to offer a good registration angle with the latest reference copied. Figure 38 b presents the angles suggested when the after in the first reference was filled with an image obtained with the orientation selected in Figure 38 a.



a) b)

Figure 38 - screen capture showing the angle hints. On the left the initial hint angles are presented. The right image presents the angles calculated after first reference is filled.

When an angle hint is selected the information is sent to Allura machine and the APC (Automatic Position Control) button will be activated. The physician can then bring the C-Arm in the desired position by pressing this button. Sending hint angles can also be done using the TSM (Figure 37)

9.6 Graphical representation of reference orientation (R9)

This new feature offers clinicians an intuitive way of understanding the orientation of the images in the reference screens. In the initial version of the application the orientation of the C-arm during image acquisition was presented by showing a dwarf in the bottom right corner of the screen. The orientation of the dwarf corresponded to the orientation of the patient relative to the C-arm (screen capture in Figure 36 shows an example).

In the registration step three different dwarfs are presented each with a different orientation. It would be much easier for clinicians to get an overview if all this information was displayed combined. For this in the live screen cut-planes through the dwarf are displayed to show the orientation of the references. Different colors are used to distinguish between the references. The figure below (Figure 39) presents the dwarf from the Figure 36 example. The view is rotated to clearly see the cut-planes.

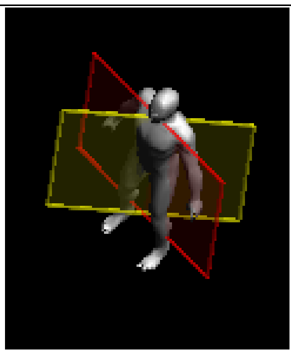


Figure 39 - Cut-plane representation of the reference orientations

9.7 Other graphical improvements (R10-R13)

Besides the features described above, some other small graphical improvements were added to improve usability. A very simple thing as a yellow border around the selected view (in Figure 36 first reference is selected) can help physicians get a more intuitive understanding of the UI. The need for this clarification is even more

The information presented in the live screen is not used during the actual registration process. Therefore an option to have a new viewing mode that contains only the two references was added. The switching between the maximized view (Figure 33) and the normal view containing also the live screen (Figure 36) can be done manually or in some cases automatically. For the manual transition the physician can use the toolbars (last button on the toolbars in any screen) or the TSM (fourth button in Figure 37). After filling both reference screens the physician will start the registration process therefore automatically going to the maximized view reduces its interaction with the application. Whenever new X-ray images are taken, the application will automatically go back to the normal view so that the new images are visible on the screen.

Whenever live or saved X-ray images are presented in the application their orientations should also be displayed. One option is to display the “dwarf” rotated so that its orientation corresponds to the orientation of the images in the screen. This offers a very intuitive way of understanding what the direction of X-rays was. However the "dwarf" alone did not offer enough orientation information. Small angle differences were impossible to notice. To fix this issue a textual representation of the C-arm position is now also displayed under the dwarf (Figure 36).

10. Project Management

Abstract –This chapter describes the relevant issues regarding project management. The process used during the project is described followed by a more detailed presentation of the project planning.

10.1 Process

From the beginning of the project its time span was fixed: nine months. This meant that a good project management was needed in order to achieve the objectives of the project in the limited amount of time. As the trainee, I was the main responsible person for the project's management process. First step I made was to arrange and agree with the stakeholders a process to be followed. In the Innovation department, a lightweight form of Agile (22) approach is used in order to provide a very low project management overhead for the prototype development. I realized that adhering to this work style would be the best approach for the project.

Developing a prototype application turned out to be challenging from the project management's the point of view. As described in section 5.1 , a complete set of requirements couldn't be gathered in the beginning of the project. Instead, an iterative approach was needed. New features were added to the application and, based on the feedback received, new requirements emerged. The planning of the project followed this iterative approach.

Project Steering Group (PSG) meetings were organized on a regular basis (about one per month) throughout the project duration. The purpose of these meetings was to update both TU/e and Philips supervisors about the status of the project. Also the next steps to be taken in the project were agreed in the meetings. This offered a periodical validation of the planning and assured that the project in on the right path.

10.2 Breakdown structure

The previous section stated the iterative nature of the project. The work done in the project can be broken down into several work packages. These packages can be visualized in the figure below.

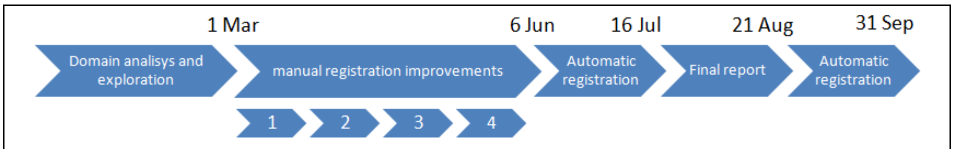


Figure 40 –work breakdown structure

First two months of the project were used to get some background for the project (the "domain analysis and exploration" block in Figure 40). Initially some technical training was needed. The first step I made was to get familiar with C# and the WPF framework. Next, learning the fundaments of the viewing architecture as used in the VesselNavigator was required. This was a very dynamic process. Attending an internal Philips workshop provided me with a starting point for using the viewing architecture. Based on this I started developing a prototype implementation for the Arcball interactor. This "learn while doing" approach turned out to be a very effective way to gather the required technical background.

The first two months also offered me the possibility to get some context information about the clinical domain. Meetings with my Philips supervisor offered me a basic understanding about the clinical problem the VesselNavigator project is focused on. Discussion with the clinical scientists provided me with some details about the way the surgeons work and how they use the application. Moreover in this period I participated in an X-ray course and got familiar with the Allura X-ray system. In this initial part of the project I got sufficient information and training to be able to get actively involved in development of the VesselNavigator application.

The second part of the project was focused on improving the manual registration step of the VesselNavigator application. To achieve this, four iteration were made each adding new features:

- The first iteration consisted of adding small improvements for the registration screen. Highlighting the active window and offering the possibility to maximize it were the main features. This iteration was also used to get familiar with the code of the application.

- In the second iteration the reference hints and the maximized view (with only the two references) were introduced. Moreover, the orthogonal interactor and the option to go to the maximized view automatically after an exposure were added.

- In the third iteration the angle hints were implemented. Besides this, using the feedback received, updates were made to the features added in the previous iteration.

- The fourth iteration introduced the graphical representation for the reference screen orientations. Also the values of the hint angles were modified based on the feedback received from the users.

In this part of the project the Arcball interactor, which was implemented in the beginning, was integrated into the VesselNavigator application. This process involved constant updates to the interactor based on the feedback received. For this reason, each of the iterations described above, also represented an iteration for the Arcball interactor.

In the third part the focus moved to the automatic registration. Due to the limited amount of time available for this topic, a decision was made to investigate the possibility of using registration algorithms already available. Therefore, a research was made to see what registration algorithms existed in other Philips Healthcare products. Based on the outcome of the research and of a feasibility study, one algorithm was selected to be integrated in the VesselNavigator application.

The last part of the project consisted in the realization of this report that should provide a complete overview of the work done.

10.3 Milestone Trend Analysis

Figure 41 presents the milestone trend analysis (MTA) for this project. It illustrates the dynamic nature of the project. Because of this just a small number of milestones could be defined in the beginning of the project. Adding new requirements and thus new milestones was a continuous process. This can be seen in the graph as the amount of lines that initiate after the project start. Also the short horizontal lines show the short iterations made.

Due to the use of short iterations, there were just a small number of changes in the planning that required a milestone delay. This was a result of the hidden complexity that some features (presented below) had.

- The integration of the Arcball interactor into the VesselNavigator project required adding support for inverse perspective projection. Compared to the initial implementation the perspective projection presented some new challenges that needed to be overcome

- The creation of a prototype for the automatic registration involved some unexpected issues (e.g. library incompatibilities) that needed to be solved. Solving these issues took more time than the actual interfacing of the automatic registration algorithm.

A significant change in the planning occurred in mid June. A request from Philips to combine the presentation with one of my colleagues moved the date of the final presentation to the end of August. This meant that the part of the automatic registration implementation had to be moved after the presentation in order to have sufficient time to write the final report.

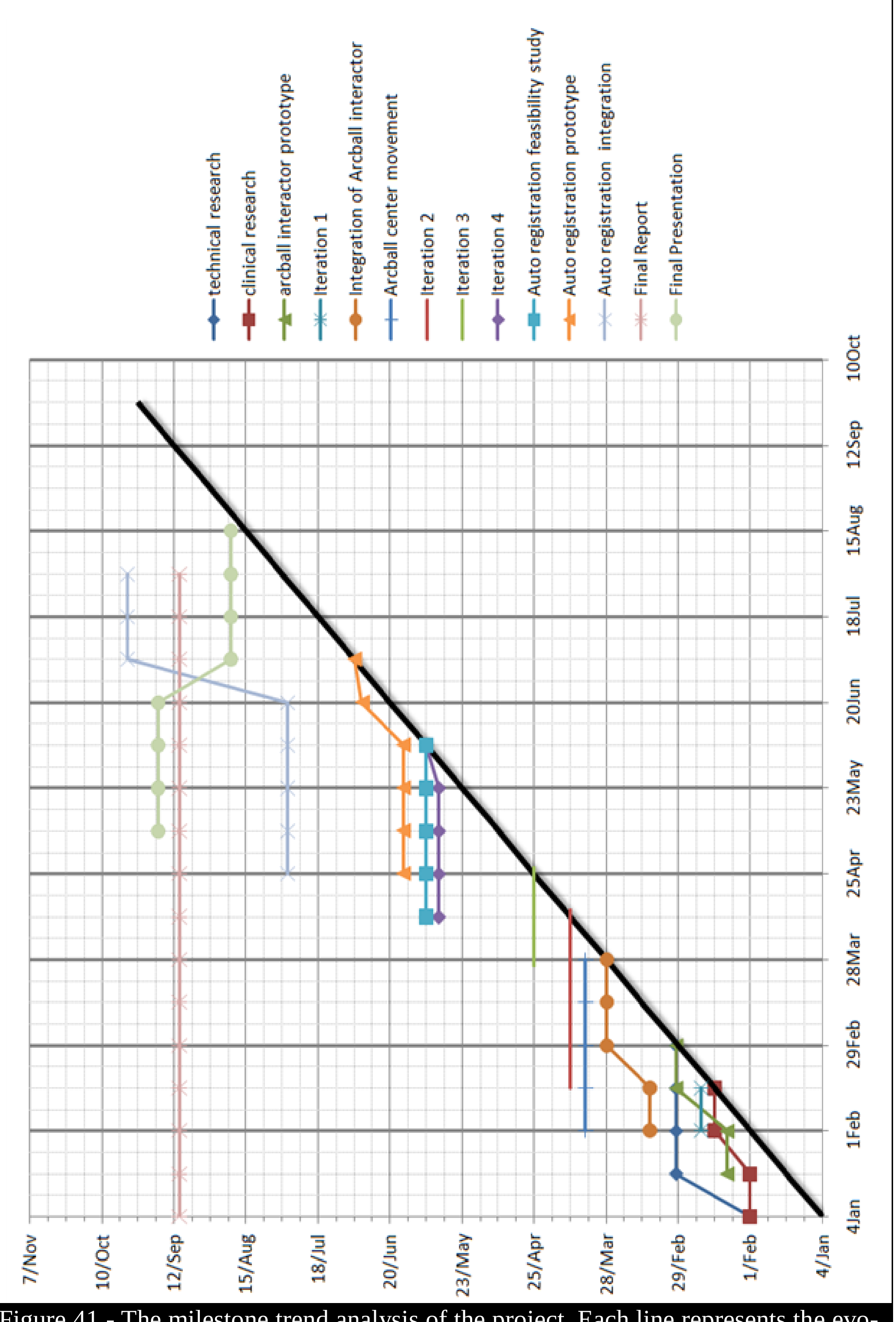


Figure 41 - The milestone trend analysis of the project. Each line represents the evolution of the planned date for a milestone.

Glossary

AAA	Abdominal Aortic Aneurysm
ALARA	As Low As Reasonably Achievable
CT	Computed Tomography
CT-A	Computed Tomography Angiography
EVAR	Endovascular Aortic Repair
MRI	Magnetic Resonance Imaging
MTA	Milestone Trend Analisys
MVVM	Model View ViewModel
TSM	Touch Screen Module
US	Ultrasound
WPF	Windows Presentation Foundation
XAML	Extensible Application Markup Language

Bibliography

1. **Verstege, Marco.** *A Framework for Flexible Vessel Segmentation and Interaction.* s.l. : Eindhoven University of Technology, technical report Stan Ackermans Institute, 2011. 978-90-444-1078-5.

2. *Abdominal aortic aneurysm.* **N Sakalihasan MD, Prof R Limet MD, OD Defawe PhD.** 9470, s.l. : The Lancet, 2005, Vol. 365.

3. *Biomedical imaging modalities: a tutorial.* **Raj Acharya, Richard Wasserman, Jeffrey Stevens, and Carlos Hinojosa.** s.l. : Computerized Medical Imaging and Graphic, 1995, Vol. 19.

4. **Glasser, O.** *Wilhelm Conrad Rontgen and the early history of the Roentgen rays.* San Francisco, CA : Norman Publishing, 1993.

5. *Sir Godfrey N Hounsfield.* **Oransky, Ivan.** 9439, s.l. : The Lancet, 2004, Vol. 364.

6. *X-ray computed tomography.* **Kalender, Willi A.** s.l. : Phys. Med. Biol., 2006, Vol. 51.

7. *Magnetic Resonance Imaging.* **Robert R. Edelman, Steven Warach.** s.l. : The New England Journal of Medicine, March 11, 1993.

8. *A survey of medical image registration.* **Viergever, J. B. Antoine Maintz and Max A.** s.l. : Medical Image Analysis, 1998, Vol. 2.

9. *Abdominal Aortic Aneurysm.* **Kranendonk, A. F. Baas and S. E.** s.l. : Clinical Cardiogenetics, 2011.

10. *The EVAR Landscape in 2011.* **FRANK J. CRIADO, MD, FACS, FSVM.** s.l. : Endovascular Today, 2011.

11. *Abdominal Aortic Aneurysm .* **GILBERT R. UPCHURCH, JR., M.D., and TIMOTHY A. SCHAUB, M.D.** s.l. : American Family Physician, Vol. 73.

12. *Management of Abdominal Aortic Aneurysms Clinical Practice Guidelines of the European Society for Vascular Surgery.* **F.L. Moll, J.T. Powel, G. Fraedrich, F. Verzini, S. Haulon,M. Waltham, J.A. van Herwaarden, P.J.E. Holt, J.W. van Keulen, B. Rantne, F.J.V. Schlosser, F. Setacci, J.-B. Ricco.** s.l. : Eur J Vasc Endovasc Surg, 2011.

13. *Abdominal Aortic Aneurysm : Pathogenesis and Implications for Management.* **Jonathan Golledge, Juanita Muller, Alan Daugherty and Paul Norman.** s.l. : Arteriosclerosis, Thrombosis, and Vascular Biology, 2006. 1079-5642.

14. *Variation in Death Rate After Abdominal Aortic Aneurysmectomy in the United States.* **Justin B. Dimick, MD, James C. Stanley, MD, David A. Axelrod, MD, MBA, Andris Kazmers, MD, MSPH, Peter K. Henke, MD, Lloyd A. Jacobs, MD, Thomas W. Wakefield, MD, Lazar J. Greenfield, MD, and Gilbert R. Upchurch, Jr., MD.** s.l. : Annals of Surgery, 2002.

15. *Open Surgery versus Endovascular Procedures.* **Branchereau, M. Jacobs and A.** 2007.

16. *Transfemoral Intraluminal Graft Implantation for Abdominal Aortic Aneurysms.* **J.C. Parodi, MD, J.C. Palmaz, MD, H.D. Barone, PhD.** 6, s.l. : Annals of Vascular Surgery, 1991, Vol. 5.

17. *Health effects of ionising radiation from diagnostic CT.* **Diego R. Martin, MD, PhD and Richard C. Semelka, MD.** 9524, s.l. : The Lancet, 2006, Vol. 367.

18. *The impact of radiation dose exposure during endovascular aneurysm repair on patient safety.* **Jones C, Badger SA, Boyd CS, Soong CV.** 2010, Journal of vascular surgery.

19. *Radiation exposure during endovascular aneurysm repair.* **Weerakkody RA, Walsh SR, Cousins C, et al.** s.l. : British Journal of Surgery, 2008, Vol. 95.

20. *Iodinated Contrast Media and Their Adverse Reactions.* **Daftary, Jagdish Singh and Aditya.** s.l. : Journal of Nuclear Medicine Technology, 2008, Vol. 36.

21. **MacDonald, Mattew.** *Pro WPF in C# 2010: Windows Presentation Foundation in .NET 4.* s.l. : Apress, 2010.

22. **Cockburn, Alistair.** *Agile Software Development: The Cooperative Game (2nd Edition).* s.l. : Addison-Wesley Professional, 2006. 0321482751.

Appendix A: Calculate point on sphere algorithm.

If the user clicks on a point on the screen (P_s), the corresponding point on the sphere (P) is selected. This is the intersection of the sphere and the ray that passes through Ps and is parallel to $\vec{v}$ (Figure 42). $\vec{v}$ is a vector orthogonal to the screen plane (its direction is dependent on the projection type).

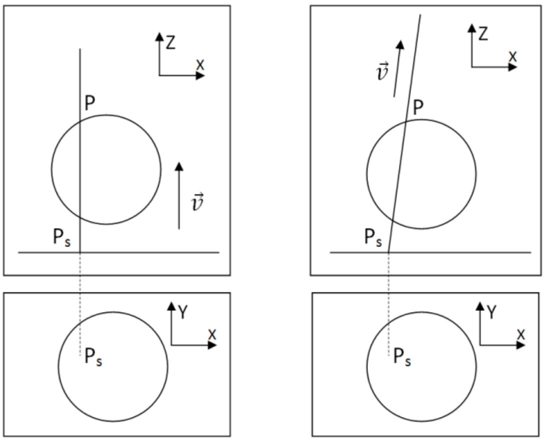


Figure 42

Sphere equation: $||\vec{P} - \vec{C}||^2 = r^2$

Line equation: $\vec{P} = \vec{P_S} + t \cdot \vec{v}$

$||\vec{P_S} + t \cdot \vec{v} - \vec{C}||^2 = r^2$

$||\vec{P_S} - \vec{C} + t \cdot \vec{v}||^2 = r^2$

$(\vec{P_S} - \vec{C})^2 + 2t \cdot (\vec{P_S} - \vec{C}) \cdot \vec{v} + \vec{v}^2 t^2 = r^2$

$\Delta = \left((\vec{P_S} - \vec{C}) \cdot \vec{v}\right)^2 - \vec{v}^2 \left(\left(\vec{P_S} - \vec{C}\right)^2 - r^2\right)$

If $\Delta \geq 0$ then the line intersects the sphere. In this case the two intersection points are:

$$t_{1,2} = \frac{-(\vec{P_S} - \vec{C}) \cdot \vec{v} \pm \sqrt{\Delta}}{\vec{v}^2}$$

$\vec{P}_{1,2} = \vec{P_S} + t_{1,2} \cdot \vec{v}$

We choose the point P that will be on the visible part of the sphere.

If $\Delta < 0$ then the line does not intersect the sphere. In this case we need to calculate point P as described in Figure 43

The calculations are different depending on the projection type:

For orthogonal projection (Figure 43 a), P has X and Y coordinates of point P_S and Z coordinate of point C.

Point P_S can be calculated using:

$\vec{P_S} = \vec{C_S} + r \cdot (\vec{A} - \vec{C_S})$

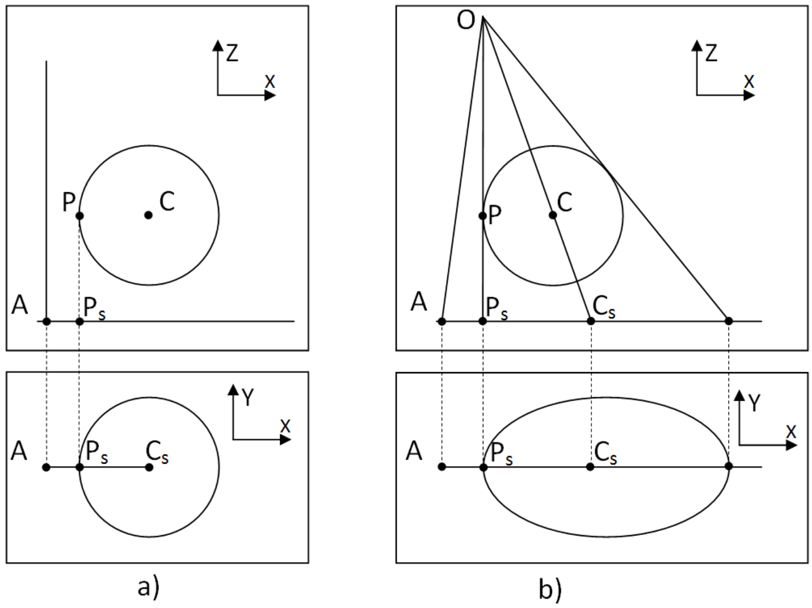


Figure 43

For perspective projection (Figure 43 b) we use the following algorithm:

Sphere equation: $||\vec{P}-\vec{C}||^2=r^2$ (1)

OP line: $\vec{P}=\vec{O}+t(\vec{P_S}-\vec{O})$ (2)

ACS line: $\vec{P_S}=\vec{A}+t'\cdot(\vec{C_S}-\vec{A})$ (3)

P is the intersection of sphere with OP line

$||\vec{O}+t(\vec{P_S}-\vec{O})-\vec{C}||^2=r^2$

$t^2(\vec{P_S}-\vec{O})^2+2t\cdot(\vec{P_S}-\vec{O})\cdot(\vec{O}-\vec{C})+(\vec{O}-\vec{C})^2-r^2=0$

Since OP is tangent to the circle, the equation above has a unique solution.

$\Delta=\left((\vec{P_S}-\vec{O})\cdot(\vec{O}-\vec{C})\right)^2-(\vec{P_S}-\vec{O})^2\left((\vec{O}-\vec{C})^2-r^2\right)=0$

Point PS is on ACs line:

$\left((\vec{A}+t'\cdot(\vec{C_S}-\vec{A})-\vec{O})\cdot(\vec{O}-\vec{C})\right)^2-$

$(\vec{A}+t'\cdot(\vec{C_S}-\vec{A})-\vec{O})^2\left((\vec{O}-\vec{C})^2-r^2\right)=0$

$at'^2+bt'+c=0$

$a=[(\vec{C_S}-\vec{A})\cdot(\vec{O}-\vec{C})]^2-(\vec{C_S}-\vec{A})^2\left((\vec{O}-\vec{C})^2-r^2\right)$

$b=2[(\vec{A}-\vec{O})\cdot(\vec{O}-\vec{C})][(\vec{C_S}-\vec{A})\cdot(\vec{O}-\vec{C})]-$

$2[(\vec{A}-\vec{O})\cdot(\vec{C_S}-\vec{A})]\left((\vec{O}-\vec{C})^2-r^2\right)$

$c=[(\vec{A}-\vec{O})\cdot(\vec{O}-\vec{C})]^2-(\vec{A}-\vec{O})^2\left((\vec{O}-\vec{C})^2-r^2\right)$

Solving the equation we obtain two values for t'.

Using (3) two values for PS can be calculated. The point we are looking for is the one closest to A from the two.

Using (2) point P can be calculated.

Appendix B: Calculate rotation angle for axis rotation

Rotation around an axis is described in Figure 44.

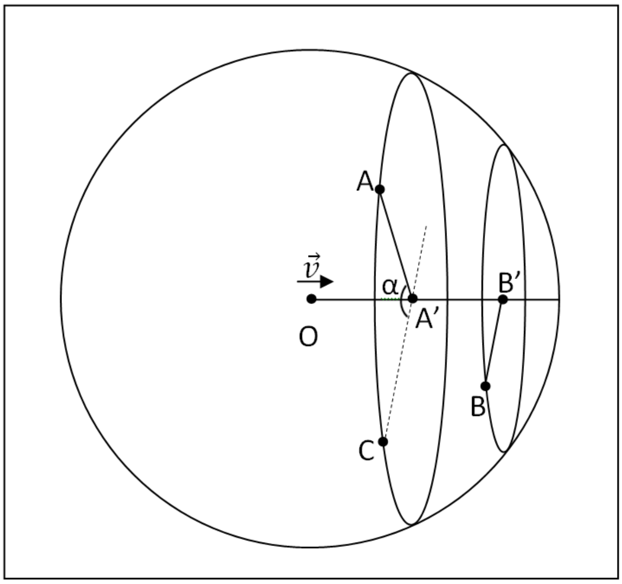


Figure 44

$$\alpha = \operatorname{acos}\left(\frac{\overrightarrow{AA'} \cdot \overrightarrow{BB'}}{\sqrt{\overrightarrow{AA'}^2 \overrightarrow{BB'}^2}}\right)$$

$$\overrightarrow{A'} = \vec{O} + \frac{(\vec{A} - \vec{O}) \cdot \vec{v}}{\vec{v}^2}\vec{v}$$

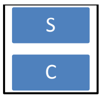
$$\overrightarrow{B'} = \vec{O} + \frac{(\vec{B} - \vec{O}) \cdot \vec{v}}{\vec{v}^2}\vec{v}$$

Appendix C: Space configurations for Arcball interactor

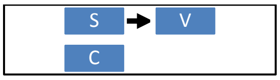
When using the Arcball interactor there are three spaces involved:

1. camera space (C) – this is the space of the viewport camera
2. space to roll (S) – this is the space the interactor will rotate (it is the space specified in the constructor of the Arcball)
3. visuals space (V) – this is the space in which the Arcball visuals are placed.

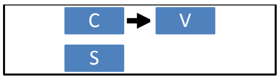
In order for the Arcball interactor to work properly the relations between the three spaces involved is important. The interactor is meant to rotate the space to roll. To get a consistent behavior in some cases the visuals must also be rotated. The listing below presents all the possible configurations taken into account. In each case the graphical representation shows the scene graph; the arrows point from the parent to the child space.

1. 

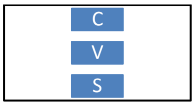
S and C are not connected

    1.1. 

    Visuals are placed in the space to roll. No extra rotation is needed

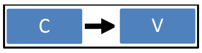
    1.2. 

    Visuals are not in the space to roll. Extra rotation needed.

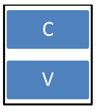
    1.3. 

    Visuals are not in the space to roll. Extra rotation needed.

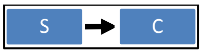
2. C==S space to roll is camera space

    2.1. 

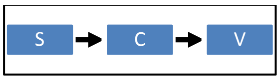
    Visuals are in camera space. Extra rotation required

    2.2. 

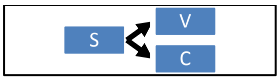
    Visuals are not rotated but the camera rotates. No extra rotation needed

3. 

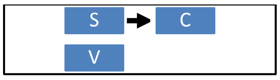
Space to roll is an ancestor of camera space. Equivalent case with 2

    3.1. 

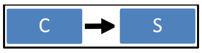
    Equivalent to case 2.1.

    3.2. 

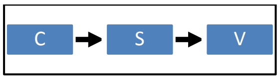
    Both camera and visuals spaces are rotated. Extra rotation needed.

3.3. 

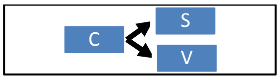
Equivalent to case 2.2.

4. 

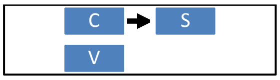
Space to roll is a subspace of camera.

4.1. 

Equivalent to case 1.1.

4.2. 

Equivalent to case 1.2.

4.3. 

Equivalent to case 1.3.

The cases where the visuals space is an ancestor of either camera space or space to roll are not taken into account since they don't quite make sense. These configurations are invalid when using the interactor.

Appendix D: Calculation of rotation and angulation from C-arm orientation matrix

On the application screen two angles must be displayed: the rotation and the angulation (the rotation angles around X and Y axes in patient's coordinates).

The rotation of an object in space can be described by the rotation around the three axes. The following matrixes describe these transformations.

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos a & -\sin a \\ 0 & \sin a & \cos a \end{bmatrix}$$

$$R_y = \begin{bmatrix} \cos b & 0 & \sin b \\ -\sin b & 1 & \cos b \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_z = \begin{bmatrix} \cos c & -\sin c & 0 \\ \sin c & \cos c & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The C-arm orientation is described by a rotation Matrix M

$$M = R_z * R_y * R_x$$

The new coordinates of a point P after a C-arm rotation are

$$P' = M * P$$

The rotation (a) and angulation (b) also define a rotation matrix:

$$M' = \begin{bmatrix} \cos b & 0 & \sin b \\ -\sin b & 1 & \cos b \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos a & -\sin a \\ 0 & \sin a & \cos a \end{bmatrix}$$

$$= \begin{bmatrix} \cos b & \sin a \sin b & \cos a \sin b \\ 0 & \cos a & -\sin a \\ -\sin b & \sin a \cos b & \cos a \cos b \end{bmatrix}$$

To obtain the corresponding angles *a* and *b* from a rotation matrix M the following condition must be satisfied:

$$M * \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = M' * \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

This means that the Z axis will have the same orientation if either M or M' is applied. The left side of the equation is the known part:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos a \sin b \\ -\sin a \\ \cos a \cos b \end{bmatrix}$$

Solving for *a* and *b* we obtain:

$$b = \begin{cases} atan2(x, z) & , \cos a > 0 \\ atan2(-x, -z) & , \cos a < 0 \end{cases}$$

$$a = \begin{cases} \mathrm{atan2}\left(-y, \sqrt{x^2 + z^2}\right) & , \cos a > 0 \\ \mathrm{atan2}\left(-y, -\sqrt{x^2 + z^2}\right) & , \cos a < 0 \end{cases}$$

We have 2 solutions:

$$a_1 = \mathrm{atan2}\left(-y, \sqrt{x^2 + z^2}\right),\ b_1 = \mathrm{atan2}(x, z)$$

and

$$a_2 = \pi - a_1,\ b_2 = b_2 - \pi$$

The solutions can also be seen in the figure below:

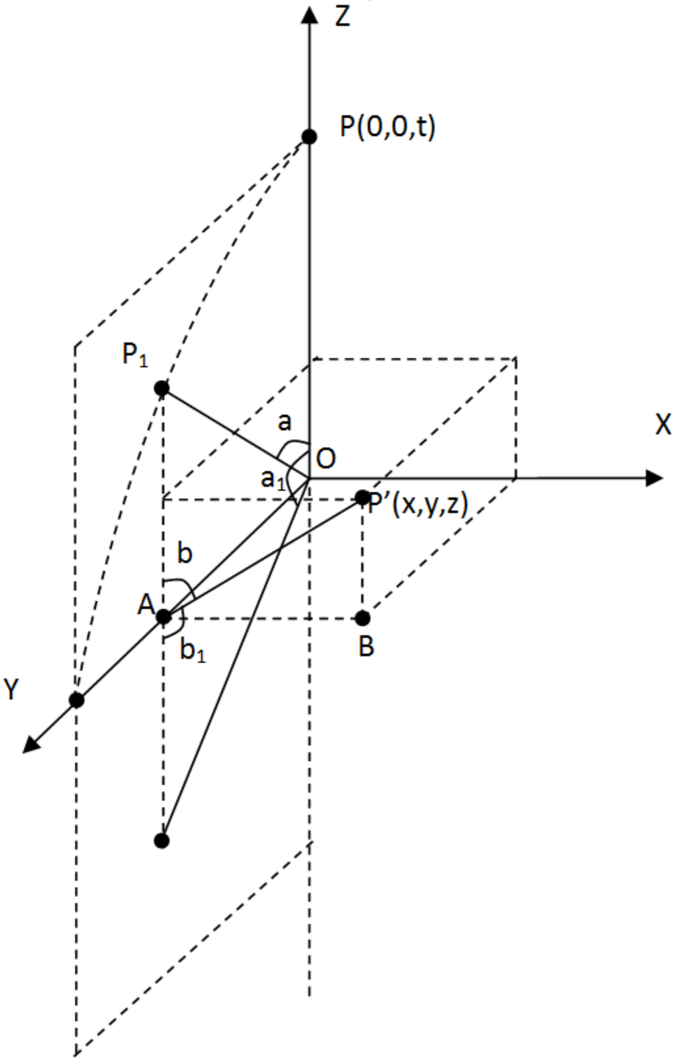


Figure 45 - the two solution to rotation and angulation (a, b) and (a1, b1)

About the Author

Tudor Mihordea graduated from the "Politehnica" University of Bucharest, Faculty of Automatic Control and Computer Science in 2008. During his studies he worked as a JAVA programmer developing applications for mobile phones. His final project was developed at Testing Technologies IST GmbH in cooperation with Fraunhofer FOKUS Institute from Berlin. The project's goal was to create automatic test cases for BPEL processes using TTCN-3. After graduation he worked as JAVA EE programmer.

From 2010 until 2012, Tudor Mihordea worked at the Eindhoven University of Technology, where he followed the Software Technology (OOTI) program from the 3TU.School for Technological Design, Stan Ackermans Institute. Tudor worked for his graduation project at Philips Healthcare on improving the usability of the registration step of the VesselNavigator application. Upon completion of the project, Tudor received his Professional Doctorate in Engineering (PDEng) degree in October 2012.

3TU.School for Technological Design, Stan Ackermans Institute offers two-year postgraduate technological designer programmes. This institute is a joint initiative of the three technological universities of the Netherlands: Delft University of Technology, Eindhoven University of Technology and University of Twente. For more information please visit: www.3tu.nl/sai.