Ing. Simon Gunacker, BSc

# Examination of dissected aortas in Virtual Reality

**MASTER'S THESIS**

to achieve the university degree of
Diplom-Ingenieur

Master's degree programme
Computer Science

submitted to

**Graz University of Technology**

Supervisor

Dr. Dr. Jan Egger
Institute for Computer Graphics and Vision

Univ.-Prof. Dipl-Ing. Dr.techn. Dieter Schmalstieg
Institute for Computer Graphics and Vision

Graz, Austria, July 2018

In cooperation with the

CAFÉ lab
Computer Algorithms for Médicine

and as part of the TU Graz LEAD Project

AORTIC DISSECTION
MECHANICS - MODELING - SIMULATION

# Abstract

The human organism is a highly complex system that is prone to various diseases. Some diseases are more dangerous than others, especially those that affect the circulatory system or the aorta in particular. The aorta is the largest artery in the human body. Its wall comprises several layers. When the intima, i.e. the innermost layer of the aortic wall, tears, blood enters and propagates between the layers causing them to separate. This is known as aortic dissection (AD). Without immediate treatment, an AD may kill 33% of patients within the first 24 hours, 50% of patients within 48 hours, and 75% of patients within 2 weeks. However, proper treatment is still subject to research and active discussion. By providing a deeper understanding of aortic dissections, this work aims to contribute to the continuous improvement of AD diagnosis and treatment. This goal is reached by presenting AD in a new, immersive visual experience: Virtual Reality (VR). The visualization is based on a CT scan of a real human patient suffering from AD. Given the scan, relevant visual information is segmented, refined and put into a 3D scene. Further enhanced by blood flow simulation and VR user interaction, the visualization helps in better understanding AD. The current implementation serves as a prototype and is considered to be extended by minimizing user interaction when new CT scans are loaded into VR (i) and by providing an interface to feed the visualization with simulation data provided by mathematical models (ii).

## Affidavit

*I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used.*
*The text document uploaded to TUGRAZonline is identical to the present master's thesis dissertation.*

| | |
|---|---|
| Date | Signature |

# Acknowledgment

First of all, I would like to thank my advisor Dr. Dr. Jan Egger. Without his expertise in the wide field of medical imaging I would not have been able to finish this work. I would also like to thank my supervisor and head of the institute Univ.-Prof. Dipl.-Ing. Dr.techn. Dieter Schmalstieg for giving me the possibility to participate in this project.

My personal thanks goes to God. I truly believe that it was him who lead me to the point where I stand today. Further thanks goes to my wife Anna. She was constantly encouraging me throughout this work and I would not have finished my studies without her support. Her english skills were a great help. Even though he does not understand it yet, I would also like to mention my two year old son Lukas who unwillingly sacrificed hours of playtime with his daddy often coming home late at night.

Last but not least, I would like to thank my parents: they spent lot of time and money in my previous education and they taught me to never give up when learning. In addition, I would like to thank my relatives: if they would not have spent precious hours looking after Lukas, it would not have been possible for me to finish my work that fast.

Finally, I would like to thank my country. Without the financial support of our social system, it would not have been possible for me to go to university at all.

# Contents

# List of Figures

# List of Tables

# 1 Introduction

The **aorta** is the main artery in the human organism. As such, it is **of crucial importance**.

When the innermost layer of the aortic wall tears, blood enters the tear and propagates between the layers causing them to seperate. This disease is known as **aortic dissection** (AD). Like any disease affecting the aorta, AD has a **high risk of mortality**. Yet our understanding of AD is still inadequate and therapeutic measures are far from mature. This is due to the fact that medical research is often based on clinical cases and, when it comes to AD, these cases are rare, often misdiagnosed, or even diagnosed post mortem only. Even if AD is properly diagnosed in an early stage, the emergency situation requires prompt action instead of copious research.

As a consequence, computed simulations enjoy great popularity in medical research, especially when it comes to lethal diseases. However, even though there is a lot of research done on AD, **medicine is missing a comprehensive AD simulation considering more than just parts of the overall problem**. But even if such a simulation would exist, an extensive, yet **easy-to-understand visualization is at least as important as the simulation itself**.

Given that both, comprehensive simulation and extensive visualization exist, **medicine might be able to gain a deeper understanding** of AD. This understanding will allow medicine to diagnose AD at an earlier stage and with higher certainty. Moreover, medicine will be able to improve therapeutic measures given that the understanding of AD could be deepened in some way.

As a matter of fact, this works contribution is a fundamental one: given underlying computer simulations, it provides a way to visualize AD in a realistic manner. Based on Virtual Reality (VR), the solution provides **a visual user experience that is immersive**: while conservative methods might require a researcher to sight a huge amount of medical images, VR allows the researcher to examine the problem *from the inside*. As such, the given solution is likely to do what it's made for: help medical researchers in better understanding AD. The solution is based on real-patients CT scans. Given these, the aorta is segmented. In a next step, the segmented aorta is converted into a 3D model that can be examined in VR. Along with the model comes texture generation: to provide a realistic representation of a real aorta, the walls have to be rendered using realistic colors and patterns. Given the model, the solution is enhanced by blood flow visualization and the implementation of user interaction: while VR allows free movement out-of-the-box, a user should be able to fly through the aorta in order to be able to examine the whole vessel. Besides the users capability to fly, he apparently requires the ability to control speed and direction of the flight. Using the controls shipped with a common VR system, the visualization allows the user to interact with the virtual environment he is placed in.

**Evaluated results** present some quantitative measures suggesting an immersive user experience. As such, the implemented solution serves as a mighty tool when it comes to the task of understanding a disease that may kill any of us: AD.

# 2 Medical Background

This chapter provides basic knowledge on (i) aorta and (ii) aortic dissection (AD). It concludes by pointing out the crucial role of medical imaging, computed simulation of aortic dissection, and visualization of simulation data.

The **aorta plays an important role** in the human organism. This fact is argued in section 2.1. As a consequence of the aorta's importance, diseases affecting the aorta are serious. One of the most serious aortic diseases is AD. To understand AD, a basic understanding of the aorta's **histology** is required. The aorta's histology is discussed in section 2.2. In addition, understanding the aortas **anatomy** is necessary for classifying different types of AD. Anatomy is covered in section 2.3

Given some basic knowledge on the aorta itself, the chapter continues by dealing with AD. **AD** is defined as a **tearing of the innermost layer** of the aortic wall **causing blood to flow between the layers** forcing them apart. This definition as well as a brief classification of the disease is given in section 2.4. Section 2.5 continues with pathology. **Pathology** describes AD in detail: (i) it's direct formation and temporal course (**pathogenesis**), (ii) it's origin (**aetiology**), and (iii) the impact it has on the organism (**pathophysiology**). Altogether, the study of pathology points out that AD can cause a life-threatening situation. However, as discussed in section 2.6, there are different **classes of AD**. Some of them have a higher risk of mortality than others. The **epidemiological studies** briefly sketched in section 2.7 support that observation. Studying the **symptoms of AD** in section 2.8 shows that AD is hard to diagnose: there exists a broad variety of symptoms. All of them could have various causes. This fact emphasises the importance of a fast and a rigorous **diagnosis**. Modern approaches of diagnosis are subject to section 2.9. Finally, proper diagnosis is crucial for deciding the right **therapy** which is outlined in section 2.10. Criado stated that **our understanding of AD is not yet complete** and **therapy is just at the beginning** [1]. As concluded in section 2.11, the life-threatening situation an AD may cause and the currently cursory understanding of the disease is the reason for this work: complex computer simulations and visualizations may help when it comes to gaining more knowledge on how to manage AD.

## 2.1 The aorta's role in the organism

Understanding the role of the aorta in the human organism allows to comprehend the risk coming from diseases that affect the aorta, AD included.

The human organism is a highly complex system. It is made up of trillions of cells [2, 3]. A cell is a separate, autonomous and self-sustaining unit, that can feed, grow and multiply [4, 5]. To do so, it needs nutrients. When processing these nutrients, the cell discards waste products. One of these waste products is $CO_2$ (carbon dioxide). It is generated when the cell gains energy by burning nutrients using $O_2$ (oxygen). Therefore, in order to keep a single cell and subsequently the whole organism itself alive [4], every cell needs to be supplied with nutrients and $O_2$. At the same time, it must be ensured that $CO_2$ and other waste products are transported away from the cell and excreted from the body. All processes comprising the supply of cells and disposal of waste products from cells are together known as **metabolism** [4].

In metabolism, **blood**, often referred to as the liquid organ, plays a crucial role: it is responsible for the delivery and removal of substances [6]. From the heart, it is transported through a dense network of blood vessels to the individual cells. Together, the **heart and vessels** are important components of what we call the **circulatory system** [7].

The **heart** is an approximately fist-sized, muscular, and hollow organ [8]. It is divided into four chambers: left and right ventricle and left and right atrium. With rhythmic contractions, the heart pumps blood into blood vessels. The blood vessels are named differently depending on their role and size: vessels that carry the blood away from the heart are called **arteries**, whereas vessels that lead the blood to the heart are called **veins**.

The largest artery in the human body is the **aorta**. It starts from the left heart extending down to the abdomen, where it splits in the two common iliac arteries. All arteries branch off from the aorta. Figure 2.1 shows how the blood circulates through the body and Figure 2.2 shows a schematic sectional image of the heart. Systematically, the circulatory system is divided into a **systemic circulation** and a **pulmonary circulation**. Starting from the left ventricle, the blood is pumped through the **aortic valve** (figure 2.2, 1) into the aorta and thus into the systemic circulation. Via the arteries, which branch out more and more finely, it finally arrives in the smallest blood vessels, the capillaries. There, oxygen and nutrients are released to the cells. In return, the blood absorbs carbon dioxide and other waste from the cells. In the intestine, it releases various waste materials and absorbs new nutrients. Then, the blood flows through the capillaries into the veins. From there it enters the **right atrium** (figure 2.2, 2), which is separated from the right ventricle by the **tricuspid valve** (figure 2.2, 3). After the blood reaches the right ventricle, it passes the **pulmonary valve** (figure 2.2, 4) and travels to the lung. When arrived in the lung, it releases the carbon dioxide and absorbs oxygen. Then it flows through the veins back into the **left atrium** (figure 2.2, 5) and from there via the **mitral valve** (figure 2.2, 6) back into the left ventricle where the cycle begins again. In addition to the fact that the aorta is an important traffic route in the circulatory system, it has another important

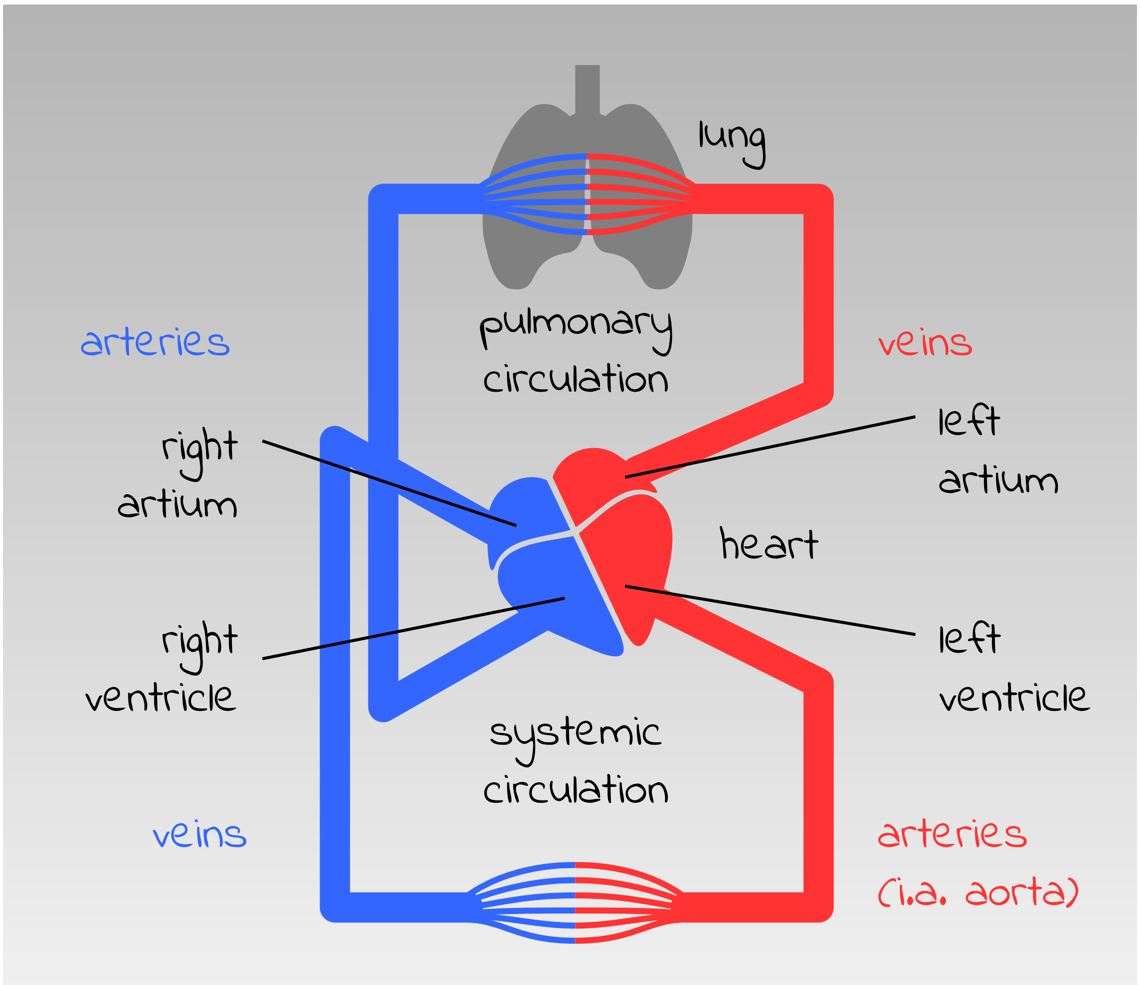


Figure 2.1: Schematic diagram of the circulatory system.

role: its vessel wall is highly elastic. Thus, it contributes to the so-called **Windkessel effect**, which **establishes a balance between systolic and diastolic blood pressure** and thus converts the blood intermittently ejected from the heart into a continuous bloodstream [9].

From the previous section it is clear to see that the aorta plays a highly important role in the human organism. Diseases affecting the aorta often lead to a patient's death. It is therefore of high interest to deeply **understand** these diseases and use this understanding to **improve treatment** strategies.

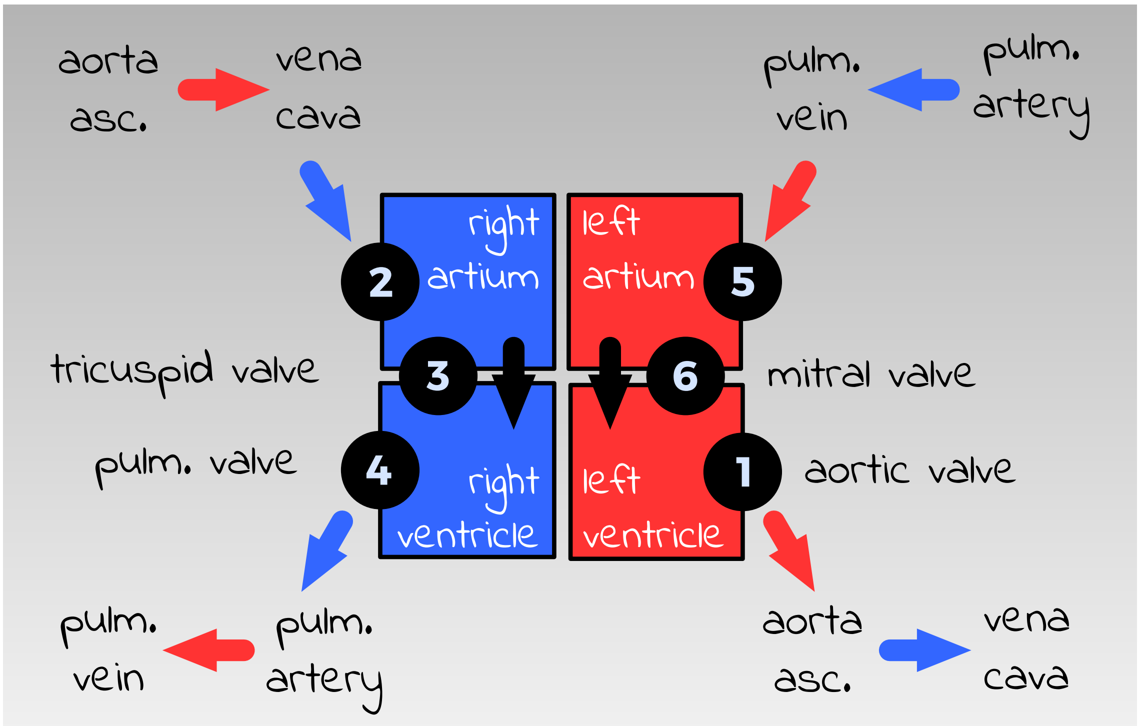


Figure 2.2: Schematic sectional view of the heart showing how blood enters and leaves the heart through different entries. Entries where high pressures occur are secured by valves.

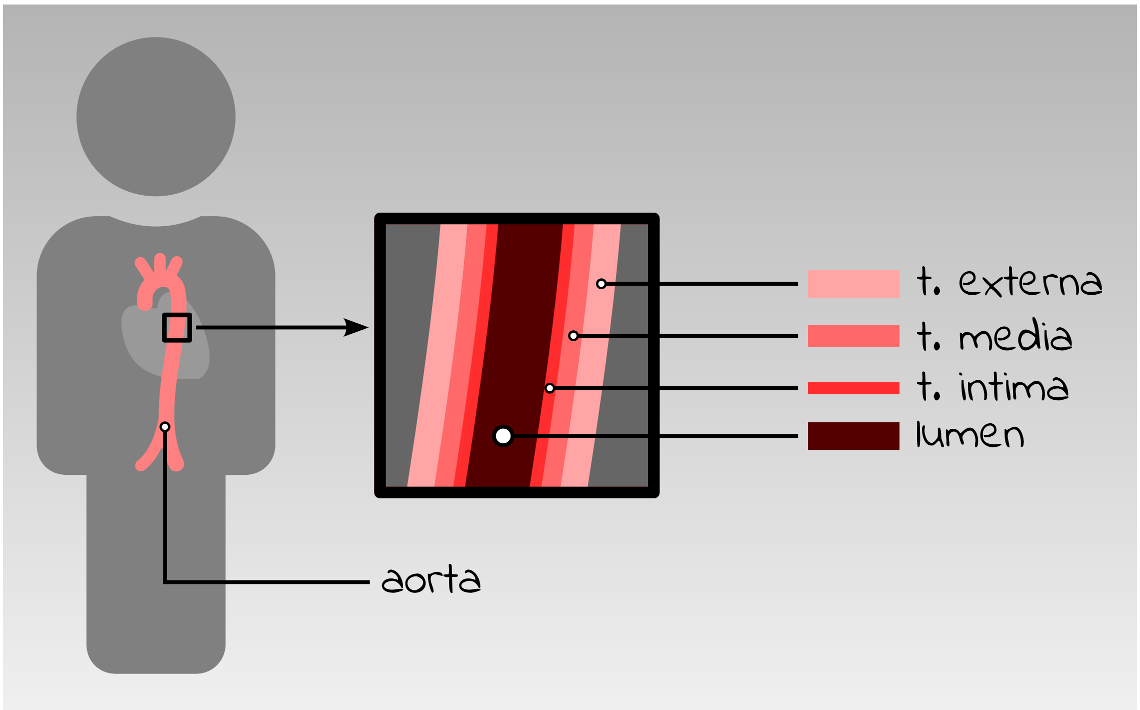


Figure 2.3: The aortic wall comprises three layers: tunica externa, tunica media and tunica intima.

## 2.2 Histology of the aorta

Understanding AD requires a profound knowledge of how the aortic wall is made up. This knowledge is subject to histology, the branch of medicine that studies tissues and their microscopic structure [10]. Understanding the structure of the inner wall of the aorta is further helpful when it comes to a reasonable visualization of the aorta.

The wall of large blood vessels comprise multiple layers (Figure 2.3). In medicine, these layers are called tunicae. The tunicae of the aortic wall are: the (i) outermost layer (tunica externa or adventitia), the (ii) middle layer (tunica media or media), and the (iii) inner layer (tunica intima or intima).

The **intima** comprises the **endothelium**, a **subendothelial layer** of loose connective tissue and the **lamina**, a layer separating intima and media. The **endothelium** is a unicellular layer. Its flat, polygonal-shaped cells are aligned lengthwise along the vessel. To ensure an optimal blood flow, the endothelial cells have a smooth surface. To repel solid components of the blood and thus prevent deposits on the wall, the nucleus of the endothelial cells bulges into the inner cavity of the vessel, the so-called lumen. The endothelium plays a crucial role in the exchange of substances between the blood and the surrounding tissue. In addition, it prevents the penetration of unwanted substances into the vessels. Moreover, the endothelium can release substances to communicate and interact with the surrounding layers. Thereby it contributes to the regularization of the vasotonus, the tension of the vessel. Finally, the endothelium plays a significant role in the control of blood clotting: certain active substances contained in the endothelium prevent thrombus formation inside the vessels. The endothelium is wrapped into a layer of connective tissue (**subendothelial layer**) which is again surrounded by the internal elastic **lamina** (membrana elastica interna). The lamina has small holes to allow substances to diffuse from the blood into the outer layers of the wall of the vessel. Beyond that, the lamina delimits the intima from the media.

The **media** is made up of muscular tissue, elastic fibers and connective tissue. The muscular cells are arranged in five to seven layers of circular smooth muscle. They give the vessel a high tensile strength and ductility. The media is responsible for toning the vessel, i.e. it contracts or dilates the vascular lumen in response to hormonal or nervous stimuli. The ability of the vessel to change its lumen enables the aorta to transform the thrusts of blood rhythmically pumped from the heart into a continuous bloodstream. This is called the Windkessel effect. The media is surrounded by the external elastic lamina (membrana elastica externa) separating it from the adventitia.

The **adventitia** is a coat of loose connective tissue that consists of a structural protein called collagen and elastic fibers [11]. It is bound to the stroma, the tissue that embeds the vessel in its environment. The adventitia isolates the vessel from neighboring structures. Further, traversed by fine veins called vasa vasorum (*vessels of the vessel*) the adventitia plays an important role in the supply of the vessel. Finally, vascular nerves (nervi vasorum) are found in the adventitia. They stimulate the media to contract or dilate the vascular lumen.

AD is a disease where the aortic wall gets damaged. Therefore it is important to understand how the aortic wall is made up to understand AD.

## 2.3 Anatomy of the aorta

Beyond understanding AD, it is important to know how it can be treated. The treatment of AD depends on the type of AD. Therefore, a proper classification of AD is highly important. Many classifications consider the exact origin of the AD. It is thus important to have a basic knowledge of the anatomy of the aorta.

Anatomy is defined as the branch of science concerned with the bodily structures of humans, animals, and other living organisms, especially as revealed by dissection and separation of parts [12]. Anatomy therefore describes the aorta itself and provides a **systematic subdivision of the aorta**. From a high-level point of view, the aorta can be separated into the thoracic and abdominal aorta depending on where the corresponding section is located. A more distinctive and commonly used [13, 14, 15, 4] separation divides the aorta into five different sections (Figure 2.4): the (i) ascending aorta (aorta ascendens), the (ii) aortic arch (arcus aortae), and the descending aorta (aorta descendens), which is further subdivided into the (iii) thoracic aorta (aorta thoracalis) and the abdominal aorta (aorta abdominalis) comprising the (iv) suprarenal abdominal aorta (aorta abdominalis suprarenalis) and the (v) infrarenal abdominal aorta (aorta abdominalis infrarenalis).

The **ascending aorta** originates from the opening of the aortic valve (valva aortae) in the left ventricle of the heart (ventriculus cordis sinister) [16]. The root of the ascending aorta is where the coronary arteries (arteriae coronariae), responsible for supplying the heart, originate from [17]. The ascending aorta is completely embedded into the pericardium (pericardium), has a total length of about five centimeters, and runs in cranial (towards the head) direction up to the second right costal cartilage. It makes a transition into the aortic arch at the pericardial reflection on the aorta [18].

The **aortic arch** connects the ascending aorta with the descending aorta. It lies within the mediastinum and runs to the left of the trachea [19] where it continues as the descending aorta [20]. There are three branches originating from the aortic arch [20]. The first and largest is the brachiocephalic trunk (truncus brachiocephalicus). It splits into the right subclavian artery (arteria subclavia dextra) and the right common carotid artery (arteria carotis communis dextra) supplying blood to the right arm, the neck and the head. The second branch is the left common carotid artery (arteria carotis communis sinistra) and the third one the left subclavian artery (arteria subclavia sinistra).

The aortic arch is followed by the thoracic aorta. The **thoracic aorta** is part of the **descending aorta** and runs caudal (towards the tail) down to the diaphragm [21, 22, 23]. There are several paired branches originating from the thoracic aorta. They supply

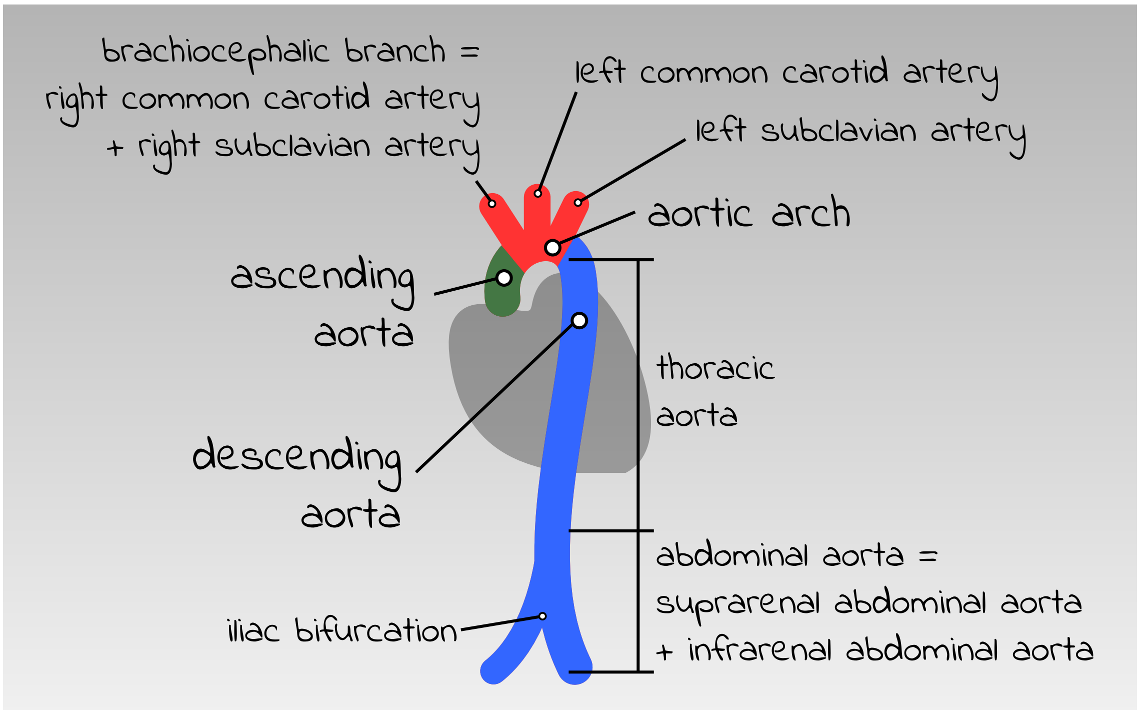


Figure 2.4: The aorta is anatomically divided into five different sections.

various structures and organs such as the chest wall, the thoracic vertebrae (vertebrae thoracicae), the spinal cord (medulla spinalis), the dorsal muscles, the lung (pulmo), and the esophagus (oesophagus). Finally, the thoracic aorta passes the diaphragm through the aortic hiatus (hiatus aorticus) and becomes the **abdominal aorta** which is starts at last vertebra of the thoracic spine (T12).

Between the first and the second vertebra of the lumbar spine (L1 and L2), the renal arteries (arteriae renalis) branch off the abdominal aorta [24]. The part of the abdominal aorta that is above these branches is called **suprarenal abdominal aorta** and the part below is called **infrarenal abdominal aorta** accordingly. The suprarenal abdominal aorta supplies the stomach (ventriculus), the spleen (lien), the duodenum (intestinum duodenum), the pancreas (pancreas) and the liver (iecur) while the infrarenal abdominal aorta supplies the lower sections of the gastrointestinal tract, the rectum (rectum) and some pelvic organs. At its end, the abdominal aorta leads into the **iliac bifurcation** which supplies blood to the lower limbs and the pelvis.

A basic knowledge of the anatomy of the aorta is essential for the classification of the AD.

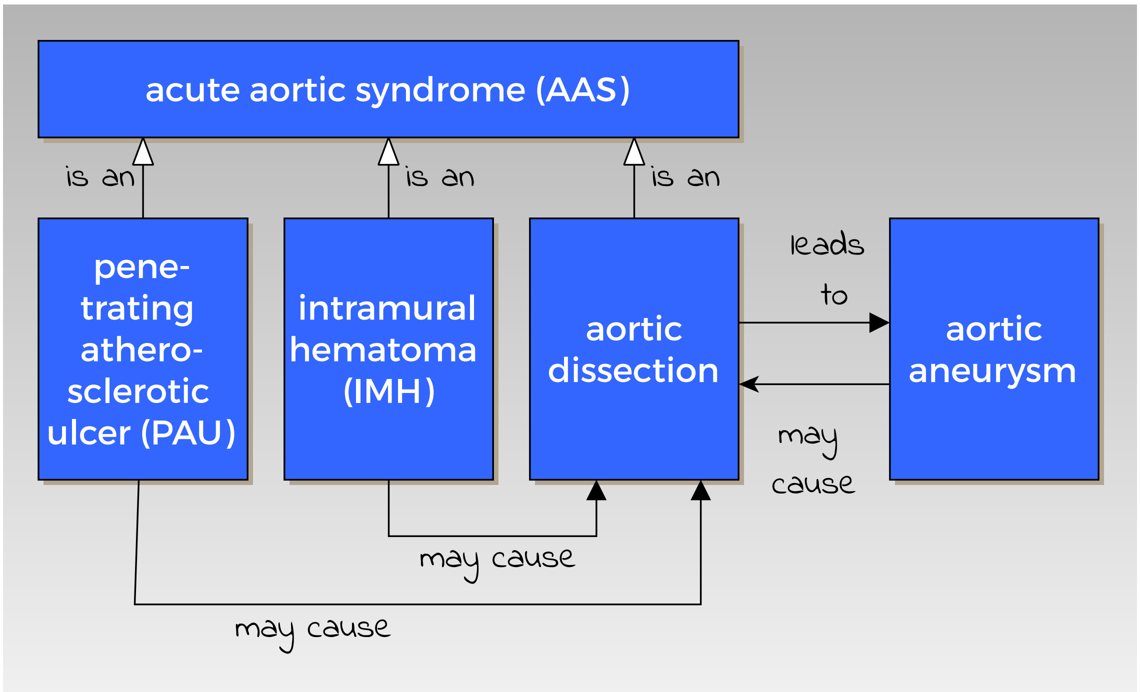

acute aortic syndrome (AAS)
is an
is an
is an
penetrating atherosclerotic ulcer (PAU)
intramural hematoma (IMH)
aortic dissection
leads to
aortic aneurysm
may cause
may cause
may cause

Figure 2.5: Aortic dissection is an acute aortic syndrome.

## 2.4 What is aortic dissection?

The AD is a highly lethal disease. It is described by a tearing of the intima allowing the blood flow to enter the aortic wall. The blood thereby creates a new, secondary channel referred to as the false lumen. The false lumen propagates and often compresses the true lumen leading to ischemic complications [1]. In most of the cases, these complications result in the patients death. In 2001, Vilacosta and Roman proposed to summarize aortic diseases with a similar clinical profile under the term acute aortic syndrome (AAS) [25]. These diseases comprise: the aortic dissection, the intramural thrombus (IMH) and the penetrating atherosclerotic aortic ulcer (PAU). Figure 2.5 shows how these diseases are related to each other. Both, the IMH and the PAU often cause an AD. The AD is mostly found in patients with a history of hypertension. Proper treatment of AD requires a correct and rigorous diagnosis which again depends on how ADs can be classified. However, the AD is hard to find and is often misdiagnosed. Epidemiological studies show that the AD is still a wide field of research.

## 2.5 Pathology of aortic dissection

Pathology is the study of diseases. It can further be subdivided into three parts: (i) direct formation and the temporal course (**pathogenesis**), (ii) causes (**aetiology**) and (iii) effects (**pathophysiology**) of diseases [26]. Understanding the pathology of AD shows the high risk implied whenever an AD occurs.

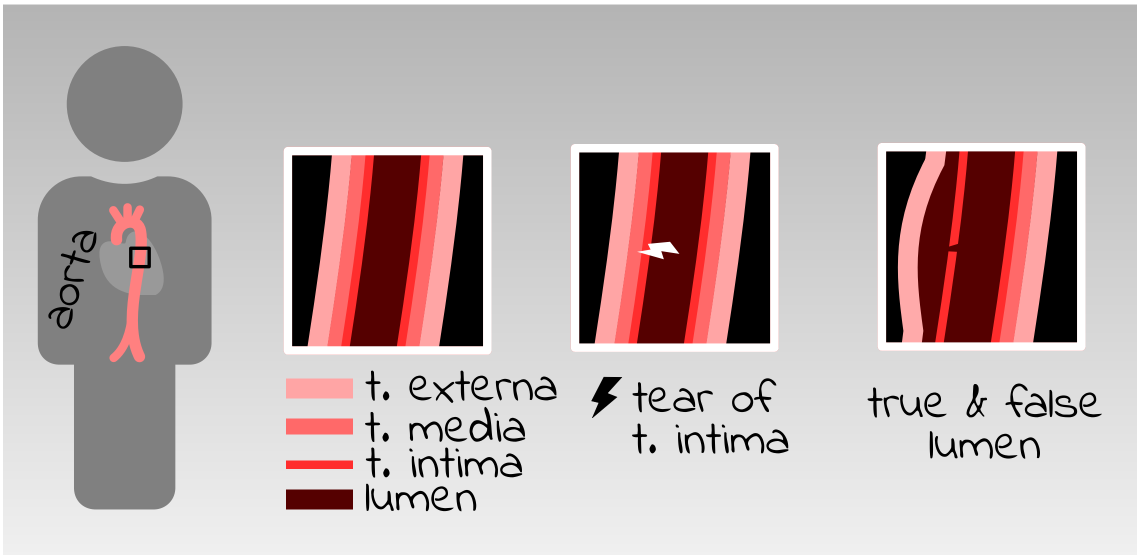


Figure 2.6: AD is initiated by a tear of the aortic wall allowing blood to enter the media dividing it. Blood propagates longitudinal and creates a false lumen.

### 2.5.1 Pathogenesis

Pathogenesis deals with the **direct formation and the temporal course of a disease**.

Figure 2.6 shows AD. It is usually caused by the **tearing** of the intima [27]. The tear is called **entry**. In most cases, it originates from the lumen. However, the tear can also be initiated by the rupture of the vasa vasorum bleeding into the media (IMH) and tearing the intima from the outside.

The most **common site of initiation is the ascending aorta** (50%) [28]. In 65% of all cases, the intima rips a few centimeters above the aortic valve, in the rising section of the aorta. 20% of ruptures occur in the descending section of the aorta immediately after the left subclavian artery. Less frequently the crack occurs in the aortic arch (10% of all cases) or in the abdominal aorta (5% of all cases) [29].

In any case, high arterial blood pressure causes blood to be squeezed out of the lumen and into the entry. As a result, the **blood increases the crack and propagates into the media**. It tears apart the tissue of the media and divides it into an outer one-third and inner two-thirds [30]. Finally, the **blood propagates in longitudinal direction**. Propagation can be both proximal (towards the heart) and distal (away from the heart). Blood may also propagate in both directions. When blood expands in the media, it **creates a cavity called the wrong lumen**. In contrast, the original lumen is called true lumen. The intermediate wall separating the two lumens consists of intimal tissue and is called an **intimal flap**.

Depending on the blood pressure and resistance of the media, the **wrong lumen can extend from a few millimeters to a few centimeters**. In extreme cases, the entire aorta is affected by bleeding. Arteries that branch directly from the aorta, such as pelvic arteries, kidney arteries, or neck arteries, can also be affected. This case is known as **circumferential dissection**. In rare cases, the false lumen tears again causing the blood to flow back into the true lumen. The second tear is called **re-entry**.

In any case, the change in physical conditions leads to a **change in local haemodynamics**. AD often causes a **thickened intima** while the **media is locally thinned and severely damaged**. Due to increased pressure, the **adventitia is often bloated** at an acute stage. In chronic stage, a thickened adventitia is more likely to be found.

The blood flow between true and false lumen is called **communication** in medicine. When there is no communication, the AD may be less obvious [31].

### 2.5.2 Aetiology

Aetiology studies the **origin of a disease**. In aetiology, an essential object of investigation is **predisposition**. Predisposition is the susceptibility to a particular disease. It is often possible to establish a relation to different characteristics, such as gender, genetic predisposition or psyche. The most common predisposing factors in case of AD are **hypertension**, a **degeneration of the media** or **dispositions that favor atherosclerosis**.

**A degeneration of the media** can come from congenital diseases associated with connective tissue changes. These include Marfan syndrome, Ehlers-Danlos syndrome, a bicuspid aortic valve or cystic media necrosis [32]. However, in a majority of cases, media degeneration is caused by an inadequately treated hypertension disorder.

**Atherosclerosis** refers to pathological storage of cholesterol and other fats in the intima. Atherosclerosis can be favoured by various factors. For example, men are far more likely to be affected by atherosclerosis than women. Other factors that favour atherosclerosis are: advanced age, increased cholesterol, consumption of cigarettes or increased blood pressure. It is believed that the fatty deposits of atherosclerosis may also be caused by endothelial dysfunction. Over many years, these deposits can lead to stain-shaped, inflammatory changes in the blood vessels. These changes are known as plaques. In the long term, plaques may lead to a narrowing of the lumen. As a result, plaques may tear. The rupture of plaques leads to penetrating aortic ulcer (PAU).

In addition to media degeneration and atherosclerosis, there are some rarer causes that can also lead to AD. These include (i) **intramural haematoma** (IMH), (ii) inflammatory diseases of the arteries (vasculitis) with the aortic involvement, or (iii) coarctation. Coarctation is a narrowing of the aorta, which occurs either alone or in conjunction with congenital heart defects.

In addition, cases of AD can often be diagnosed in relation to the **consumption of**

**cocaine or methamphetamine**. **Smoking** is another factor that increases the risk of AD.

Finally, external influences can be named as causes of AD. One of these influences could be **blunt chest trauma**, e.g. caused by car accidents. Another influence can be a medical intervention such as cardiac catheterization. AD can generally be **a late consequence of heart surgery**. About 18% of all cases of AD show at least one open heart surgery in the patient's history. The risk is particularly high for patients with **aortic valve replacements**.

### 2.5.3 Pathophysiology and direct complications

**Pathophysiology** studies abnormal functions of an organism as a result of the disease. A **complication** is an undesirable consequence of a disease. In practice, however, these concepts are difficult to separate.

In case of AD, different complications may occur depending on the situation. In the best case, a re-entry forms soon after the entry causing the blood to immediately flow back into the true lumen. This case maintains the systemic blood flow. If small enough, this case of AD **may remain symptomless and without complications**.

However, **serious consequences are expected** in more common cases. Often acute blood loss leads to **shock**: the body recognizes the lack of blood and therefore withdraws the remaining blood from the periphery (arms and legs) to ensure the supply of vital organs in head and torso. This process is called circulation. It works by narrowing arterioles and simultaneously widening peripherical venules. However, in case of AD this protective mechanism has serious consequences: blood liquid leaks into tissue surrounding the arterioles while microthrombi forming in the venules prevent the undisturbed reflux of blood. As a consequence, the shock leads to ischema of vital organs and often in a **multi-organ failure**, which is **lethal** in any case.

In addition to shock, further complications can occur: if there is a large pressure in the wrong lumen, it may compress the true lumen causing **stenosis** (narrowing) of the aorta. As with shock, aortic stenosis may also lead to **ischemia** of organs such as the brain, kidneys, liver or intestine. Depending on which organ or area is affected by ischemia, severe secondary complications such as **stroke** or other **neurologic deficits**, **acute renal failure** or **paralysis** may be the result.

Even the **heart may be affected** by a ischemia when blood flow is limited in the coronary arteries. In addition, AD can lead to a **dilation (extension) of the heart valves**, followed by **regurgitation (reflux) of blood into the heart** essentially **causing heart failure**.

As AD causes blood to be squeezed into the media, the blood is only held by the adventitia. The structure of the **aortic wall is apparently weakened locally**. This fact may cause **aortic rupture** with leads to **immediate death** without emergency surgery repair.

The above-stated complications are most likely to occur in case of acute AD, i.e. an AD whose origin dates back less than two weeks. Beyond the 2-week limit, the **risk of complications decreases if thrombosis occurs in the wrong lumen**. This case is known as **non-communicating dissection**.

Summing up, the pathology of **AD** characterizes this disease as a **life-threatening** one.

## 2.6 Classification of aortic dissection

An AD may behave different than another one. Some ADs have a higher risk of mortality than others. Different ADs require different strategies of treatment. It is therefore crucial to come up with a rigorous classification scheme for ADs.

Basically there are **different ways to classify** an AD. Various decision-making criteria are used. Classical approaches ask for **affected sections of the aorta**. Other approaches include related disease patterns and consider the circumstances that led to AD. Finally, an AD can also be classified according to **time criteria**.

In 1965, the American cardiac surgeon **DeBakey** divided AD into three different categories. In his classification, an AD has type I when both the ascending and descending aorta are affected. An AD of type II is given when only the ascending aorta is affected and type III accordingly if only the descending aorta is affected.

In 1970, this classification was simplified by **Stanford** University staff. In their classification scheme, Stanford A sums up the two types of DeBakey I and DeBakey II. Accordingly, DeBakey III became Stanford B. As a result, the classification according to Stanford reduces the classification to the question whether the entry is in the area of the ascending aorta or not. An AD of type Stanford A is also referred to as a proximal (going to the center) AD, while an AD of type Stanford B is also known as a distal (running from the center) AD.

Simplifying DeBakey on two types was motivated by the following idea: a given classification should allow to directly deduce a patient's risk and further therapeutic steps. The classification according to Stanford is still the most common way of classifying ADs. Figure 2.7 shows DeBakey and Stanford classification.

It remains controversial if AD classification schemes should be extended to the more general disease pattern of AAS. This would lead to classification schemes considering the AD-related disease patterns of PAU and IMH. Asian countries tend to strict separation, as there are incidents of spontaneous healing in case of PAU and IMH. As a matter of fact, they apply more conservative therapies to these diseases. In Europe, on the other

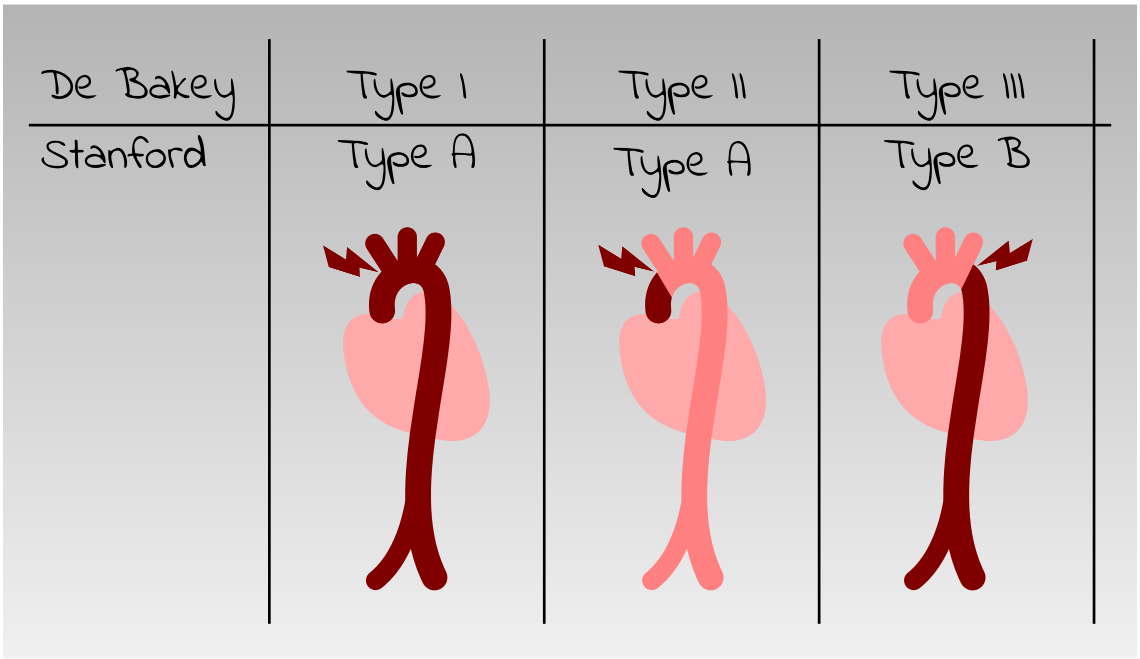


Figure 2.7: Aortic dissections are classified depending on where the initial tear occurs (i) and what parts of the aorta are affected (ii).

hand, a common classification scheme for AAS-related diseases is preferred [33]. A classification in this way was first proposed in 1989 and taken up by the **European Society of Cardiology** (ESC) in 2001 [34]. They differentiate between **five classes**. Class 1 is the classic dissection with an intimate flap between true and false lumen. Class 2 describes those cases where media separation is accompanied by intramural bleeding. Class 3 refers to cases of a discrete/subtle dissection without heamatoma. The ESC speaks of class 4 if there is a plaque rupture followed by a PAU. Finally, an AD is assigned to class 5 if it was either caused traumatic or iatrogenic.

In addition to the aforementioned classification schemes, a **temporal classification** can further help in risk estimation: an AD is called *acute* if its symptoms first occured within the last 14 days, otherwise it is called *chronic*.

## 2.7 Epidemiology of aortic dissection

Epidemiology is the branch of medicine which deals with the incidence, distribution, and possible control of diseases and other factors relating to health [35].

To conduct epidemiological studies regarding AD, an international consortium to evaluate incidents of AD has been founded in 1996. It is called *international registry of acute aortic dissections* (**IRAD**). The IRAD database currently contains more than 3000 records of acute AD [36]. Many studies relate to these records. One of these studies states that AD manifests in various ways and typical symptoms are rare or suggestive of a diverse range of other conditions leading to the fact that AD is often hard to diagnose [37].

In 1993, Spittell et al. have studied 236 cases of acute AD [38]. They concluded, that in 38% of all cases, the AD has not been found in initial diagnose and in 28% of all cases, it was only found post mortem [38]. It has therefore to be assumed that patient records used in various **studies are incomplete**. Due to this fact, various studies note that true incidence and prevalence remains unclear and may be underestimated [37, 39, 40, 41, 42]. Nienaber und Clough reason that used data is often taken from retrospective registers of specialized centers. This data depends on correct hospital coding and misses incidents leading to death before hospital admission [40]. Li and Liu supplement that in case of acute type A AD (TAAAD) already 20% of all patients die before being admitted to hospital [42]. They further state that AD is often misdiagnosed as ventricular arrhythmia or myocardial infarction. Olsson et al. discuss that knowledge on prevalence und incidence is often based on small studies that only consider short periods of time [43]. In contrast, they present a study based on findings over 15 years. In 2000, Meszaros et al. have published a long-term study researching 27 years of AD [44]. However, even though there are various studies, the aforementioned arguments reason that it is hard to present unambiguous results regarding the incidence rate of AD. Different **studies may diverge and deviate** between two to ten cases per 100.000 person-years.

For epidemiological studies, the IRAD database is used to derive the **typical AD patient**: he is male, in his seventh decade with a history of hypertension who presents with abrupt onset of chest pain [37]. However, there are rare cases with according medical history in which AD may occur much earlier [45, 46, 46, 47, 48, 49, 50, 51]. On average, man are about twice as likely to be affected from AD as women [37]. Around two thirds of all AD incidents are type A dissections [37]. Another study does research on cases of AD in iceland between 1992 and 2013 [41] and lists the **survival rate** of affected patients: 17.6% of all subjects die before hospital admission, 21.4% within 24 hours and 45.2% within 30 days. During the time period of the study, type A dissections were often treated using open surgery while the common treatment for type B dissections shifted from open surgery to endovascular repair. It has been observed, that the mortality rate dropped every year while the five-year survival rate increased within the last third of the study. An american study expects half of the patients to be dead by the third day, and almost 80% by the end of the second week [1]. According to IRAD, certain types (Type A) of dissections, if left untreated, kill 33% of patients within the first 24 hours, 50% of patients within 48 hours, and 75% of patients within two weeks [36].

Summing up, epidemiological studies show that AD is **hard to diagnose** on the one hand and has a **high rate of mortality** on the other hand. Diagnosis and therapy are therefore still fields of wide and intensive research.

## 2.8 Symptoms of aortic dissection

As already discussed, AD is often an acute disease with life-threatening complications. If not treated surgically within a short period of time, it will lead to death in many cases. To quickly decide the right therapeutic measures in an emergency, however, requires the correct diagnosis. In addition to knowledge of the medical history, an essential component in the process of diagnosis is the detection of the symptoms.

In the case of an AD, however, this proves to be difficult. There are **few signs that clearly indicate an AD**. In chronic cases, the AD may be completely asymptomatic. It is therefore often only discovered by accident when people are free of complaints. In the case of the acute AD, the symptoms are extremely versatile. Most of these symptoms are however signs of the complications of AD, not of the AD itself. The leading symptom of acute AD is a sudden, severe pain. Patients often describe it as a tearing, ripping, shearing or stabbing sensation. Sometimes patients also speak of a migratory pain. Such a feeling can be justified by the propagation of dissection along the aorta. The origin of the pain is to be found where the intima is tearing. As the most common form of AD is proximal, the pain most commonly originates from the chest in the area of the left shoulder blade. In contrast, a pain that can be found between the shoulder blades in the back area is a typical sign of distal AD.

So if there is a pain in the chest or back area, it can point to an AD. However, there are **still other diseases that can be attributed** to such pain. These include, for example, myocardial infarction, pulmonary embolism or acute coronary syndrome. As a matter of fact, the **pain does not allow a clear identification** of an AD. In any case, the high blood loss is accompanied by signs of shock: pallor, cold sweat, hypotension and tachycardia are the typical characteristics [52]. Furthermore, in a clinical examination a blood pressure difference $> 20mmHg$ between the arms can point to an AD [53]. Even a clouded awareness is not uncommon in patients with AD.

If the AD results in ischemia of one or more organs, further symptoms can be found.

In case of **myocardial infarction**, Horne et al. distinguish between typical and atypical symptoms [54]. Typical symptoms include: radiating chest pain and collapse. Frequently occuring atypical symptoms are: shortness of breath, nausea and vomitting, dizziness and fever.

If the AD causes ischemia of the brain, the symptoms of a **stroke** occur. Among these, the most common symptoms are: paralysis, sensory deficiencies, speech disorders, visual disturbances (hemianopia), headache and vertigo [55].

If the AD results in ischemia of the intestine, the symptoms of **acute mesenteriale ischemia** (AMI) occur. These include in addition to acute pain: nausea and vomiting, irritation of the peritoneum, fever, diarrhea, hematochezia, vomiting (hematemesis) and, in rare cases, the fecal vomiting [56].

Circulatory disturbances in the extremities often lead to attenuation or a complete loss of pulse. As can be seen, symptoms that are observed in case of AD are often symptoms of complications of AD. It is therefore **not easy to directly infer AD from given symptoms** in practice. Therefore, the **use of technical aids** for diagnosis is of great importance in the case of AD.

## 2.9 Diagnosis of aortic dissection

In general, the symptoms of an AD depending on the type of AD are of different severity and very versatile. Without technical aids, an AD is difficult to diagnose, as there are no typical clinical signs that clearly distinguishes AD from other diseases. However, the often life-threatening situation forces a rapid clarification of the suspected diagnosis in order to be able to determine further therapeutic measures. Using technical aids for diagnostic purposes is therefore particularly important in the case of AD. Advances in the field of **medical imaging** have established imaging techniques for the diagnosis of AD [57, 58, 59, 60, 61, 62, 63, 64, 65, 66].

The simplest method of investigation is a **echocardiography**. It is basically an ultrasound examination of the heart. There are two types of echocardiography: transthoracic echocardiography (TTE) and transesophageal echocardiography (TEE). At TTE, the heart is examined by placing an ultrasound sensor on the chest wall. However, studies have shown that this type of diagnosis is inadequately accurate [62]. For this reason, TEE is more common. An ultrasonic probe is pushed through the esophagus to the height of the heart. The procedure is simple and allows to identify complicated dissections with the patient in bed [67] within 15 minutes [66]. The results of the investigation are much more precise [62]. TEE is usually available and can also be performed in unstable patients or during surgery. In case of TEE, the patient is not exposed to any radiation exposure. The aortic sections nearby the heart as well as the thoracic aorta descendens are well visible on TEE. If the dissection is in this area, the true lumen is usually reliably distinguishable from the wrong lumen. Nevertheless, the use of TEE requires expertise in interpreting the results. In addition, changes in the aortic arch and the abdominal aorta are hard to find using TEE; these areas are also referred to as the blind spots of TEE. As a matter of fact, there are cases in which no abnormalities can be found in a echocardiography [37].

As a consequence, other imaging techniques are preferred. These include: **magnetic resonance imaging** (MRI), **computed tomography** (CT), or **angiography** [37]. They are more complex than TEE, but they allow a comprehensive and accurate representation of the entire aorta. In addition, they are able to show affected side branches and bleeding in the direct neighbourhood of the aorta. However, these procedures have some disadvantages over TEE: (i) they require the patient to be fit for transport, (ii) they allow only limited monitoring of a patient, (iii) the contrast agents used in some of these procedures may lead to allergic reactions, and (iv) a patient is exposed to high radiation in case of CT. As a consequence of (iv), MRI is usually preferred over CT from a health point-of-view.

The downside of MRI is that it is usually not available 24/7. This is why CT or CTA (**computed tomography angiography**) has established as today's **gold standard** in diagnosing AD.

Even though modern state-of-the-art imaging techniques provide high resolution images of the interior of the human body, **AD may be missed** in individual cases due to its various manifestations [68, 69]. In addition to the imaging methods, however, no investigation techniques have yet been found to help diagnose AD. Furthermore, there are often several hours between onset and diagnosis of AD [70]. So, although there might be a reasonable suspicion of AD, **diagnosis may not be early enough** to prevent catastrophic effects in time [71]. Therefore, a **more profound knowledge is required** to improve diagnose and treatment of AD. Comprehensive computer simulation and detailed visualization in Virtual Reality is an approach to help in finding these improvements.

## 2.10 Therapy of aortic dissection

In 2010, the American College of Cardiology (ACC) in cooperation with the American Heart Association (AHA) **published guidelines** for the diagnosis and the management of patients with thoracic aortic disease (TAD) [72]. In 2014, the European Society of Cardiology and the European Association for Cardio-Thoracic Surgery (ESC/EACTS) published similar guidelines together [73]. In 2017, the Japanese Association for Thoraic Surgery (JATS) revised and summarized the ACC/AHA-guidelines [74]. Figure 2.8 shows an overview on how to proceed with a patient when AD has been diagnosed.

All guidelines agree that **initial management focuses on the prevention of further complications**. Highly frequent monitoring is as essential as the decreasing of the aortic wall stress. The latter is achieved by controlling heart rate and blood pressure. To control blood pressure, the intravenous administration of beta blockers is recommended if there are no contraindications. Beta blockers are drugs that suppress the impact of adrenalin and noadrenalin. After the patient is successfully stabilized in the initial phase, further treatment depends on the type of AD. Therapeutic measures are thereby usually derived using the "Stanford" classification.

In case of an acute AD with affected aorta ascendens (AD **type A**) an **emergency surgery repair** is the most common recommendation. Such a surgery should be done as soon as possible, optimally within the first few hours after the AD has occured. Primary goal of the surgery is to eliminate the entry tear. To do so, the common procedure is the partial replacement of the aorta using a stent graft. A stent graft is an artificial vessel supported by a mesh of wire (Figure 2.9). A study proposes to simulate the placement of the stent graft before the surgery is actually carried out [75].

Replacing parts of the aorta using a stent graft is currently considered to be the fastest and safest solution for type A AD [76, 77]. In some cases, the surgery requires the additional replacement of the aortic valve. In case of aortic arch repair, the surgery requires further

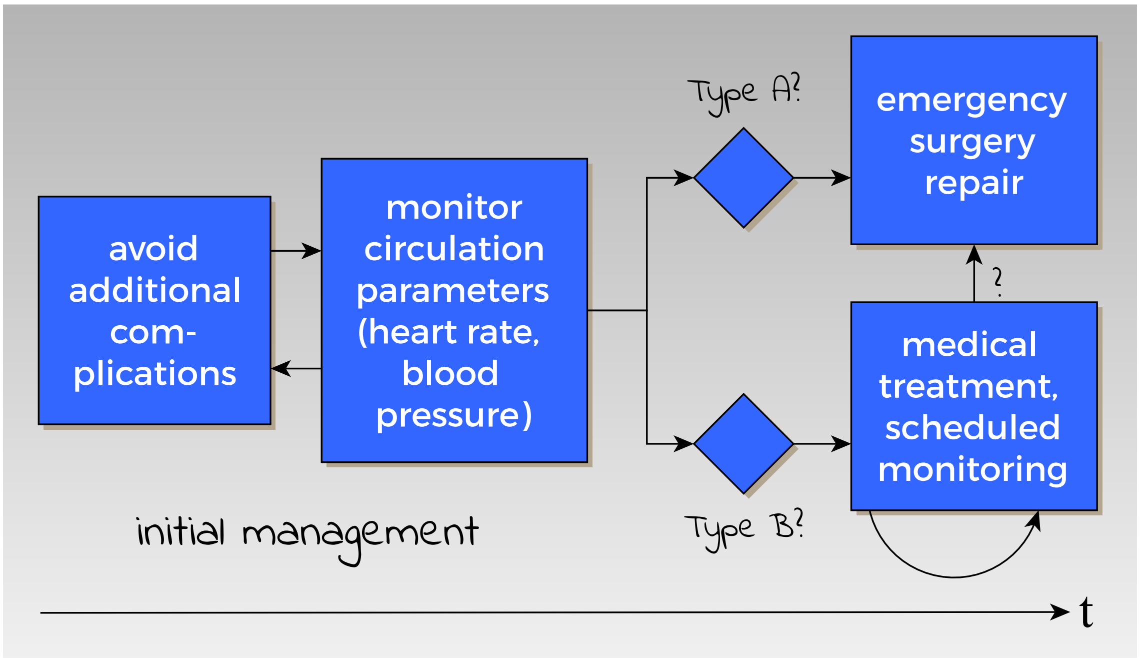

avoid additional com- plications
monitor circulation parameters (heart rate, blood pressure)
Type A?
emergency surgery repair
?
medical treatment, scheduled monitoring
Type B?
initial management
t

Figure 2.8: A diagram showing the therapeutic measures applied in case of AD.

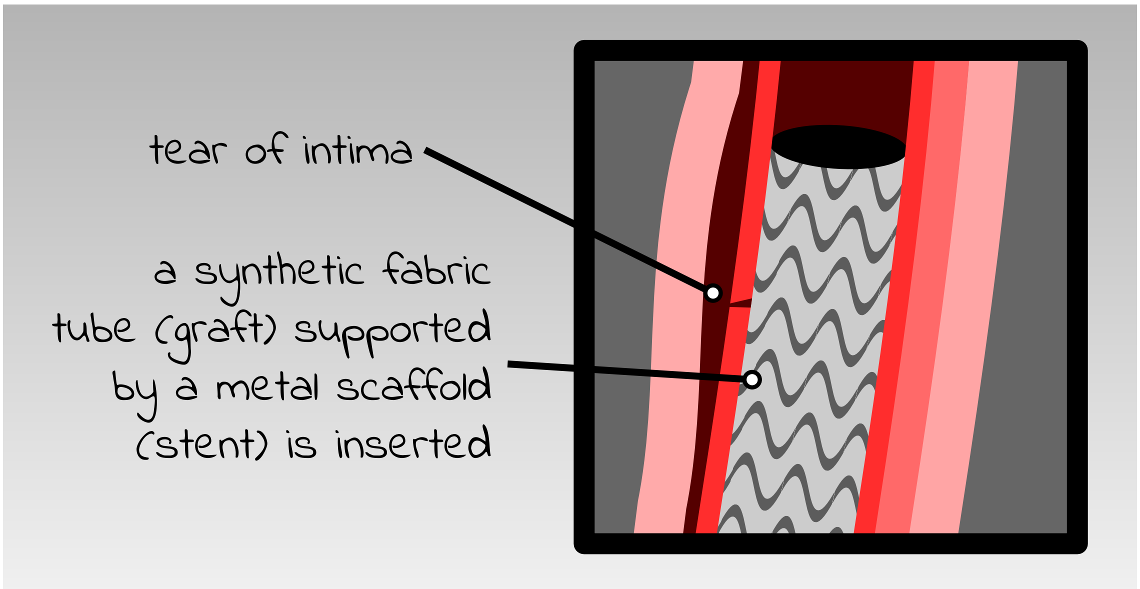

tear of intima
a synthetic fabric tube (graft) supported by a metal scaffold (stent) is inserted

Figure 2.9: A stent graft is used to prevent blood flowing through the entry tear.

measures sustain the function of brain, spinal cord or kidneys. The mentioned surgeries are open repairs in most cases.

If the AD affects the aorta descendens only (AD **type B**), a surgery is often tried to be avoided. In some cases, especially when the dissection is chronical, **medical treatment** is considered to be sufficient. Indications for surgery of type B dissections are: ascending aorta involvement, limb of mesenteric ischemia, aneurysm expansion, progression of dissection, ongoing pain, or uncontrolled hypertension. If type B dissections require surgery, endovascular repair (EVAR) is preferred in the abdominal area. This method is less invasive. Also, in the thoracic area, endovascular surgery (TEVAR) is becoming more and more debated [78, 79, 80, 81, 82].

After successful surgery of AD, **lifelong clinical monitoring** is necessary in every case. The ACC/AHA recommends computed tomography imaging or magnetic resonance imaging at a schedule of one-, three-, six-, and 12-months after dissection. The ESC/EATS speaks of CT at one-, six- and 12 month intervals and recommends a TEE in combination with a chest X-ray if a patient is unable to undergo CT because of severe renal dysfunction. In any case, if the patient proves to be stable, this procedure should be repeated every year [72].

The various consortiums that spread all over the world performing active research on how AD could be treated best prove that the fight against this life-threatening disease leaves **room for improvement**.

## 2.11 Summary

The aorta is the largest artery in the human body. It plays a crucial role in blood circulation which in turn is essential for metabolism, the sum of processes that keep the human organism alive. As a consequence, diseases affecting the aorta are often life threatening. One of these diseases is AD, the tearing of the inner layer of the aortic wall allowing blood to propagate between innermost and outermost layer of the aortic wall. Studying AD from various points-of-views reveals it's tragic truth: **AD is a dangerous disease often leading to a patients death**. Different attempts of classification are crucial for meeting the right decisions when it comes to treatment. However, results of diagnosis show that an AD is often hard to find. Epidemiological studies support that fact: AD is often missed or even only found post mortem. Therapy, while already performed over decades and steadily improved, is still at the beginning. A deeper understanding of AD is required to improve measures further helping to properly treat or even completely avoid this threatening disease. Comprehensive **computed simulations and immersive Virtual Reality visualizations are a crucial step towards gaining more knowledge on how to fight AD**.

# 3 Technical Background

Visualizing the simulation of aortic dissection is a complex work comprising several steps: medical images need to be retrieved, stored, preprocessed and modelled into an immersive visual experience. While chapter 4 covers chosen methods for implementing the visualization and chapter 5 presents the final solution, this chapter considers some selected topics of technical background required for implementing such a visualization.

What had been a challenge several decades ago is common practice today: thanks to **radiography**, we are able to gain insight into a human body in a non-invasive way. **Computed tomography** is one of the widespread modalities used for that. Besides mentioning physical prinicples of radiography, section 3.1 covers the methods used in **image reconstruction**.

After the image has been retrieved by the scanner, it needs to be stored digitally. When **storing a medical image**, certain requirements need to be fulfilled. The widely used open standard *Digital Imaging and Communications in Medicine* (DICOM) serves among other things the purpose of ensuring these requirements. It is discussed in section 3.2.

Stored images apparently contain a lot more than just the object of interest. Therefore, the object of interest needs to be segmented. **Segmentation** is a hard problem in computer vision and image understanding. Numerous algorithms exist to automate that task. Segmentation is covered in section 3.3.

In order to visualize segmented image data, it has to be **converted into a 3D model**. 3D models and how they are stored digitally is subject to section 3.4.

Eventually, the 3D model needs to be visualized in VR. Section 3.5 provides an extensive insight into **mathematical methods used for visualizing a 3D world in Virtual Reality**.

Virtual reality offers a new visual experience of computer graphics. Coming along with that visual experience is the wish for interaction with the Virtual Reality. Section 3.6 covers technical background for one important step towards the enhancement of user experience providing interaction: the **computation of a centerline** that can later be used as a path for flight control.

Finally, the chapter closes with section 3.7 discussing some methods that could be used for the **evaluation of VR applications**.

## 3.1 Medical imaging

### 3.1.1 The challenge of medical imaging

The evolution of photography has a long history reaching back to the discovery of the *camera obscura*. By the end of the 19th century, taking pictures of real world scenes had become a matter of course. On the contrary, **taking pictures of the inside of a solid body** with an opaque surface had seemed impossible back then. This suddenly changed in 1895 when the german physics professor Wilhelm Conrad Röntgen observed some *rays of mysterious origin* that were able to expose a photographic film even when optically shielded. Signifying their unknown quantity, he called them *X-rays*. His discovery has had an enormous impact on medicine and is considered to be the birth of medical imaging [83]. Medical imaging comprises various techniques facing the challenge of visualizing the interior of a body with no or minimal invasion.

### 3.1.2 The basic principle of radiographic image modalities

As a beam of X-rays passes through a solid body, some of it's initial energy is imparted to the volume. This is called **attenuation**. The attenuation of X-rays depends, among other things, on the material they pass through. This dependency is the key concept when it comes to radiography, an imaging technique based on X-rays. Given an X-ray beam with an initial intensity $I_0$, a material thickness of $\Delta x$ and a material-dependant constant $\mu$ called the *linear attenuation coefficient*, the remaining intensity $I_d$ of the beam after being shot through some material can be described by:

$$I_d = I_0 \cdot e^{-\mu \Delta x} \tag{3.1}$$

However, equation 3.1 is oversimplified as it ignores some significant facts. First, the beam is usually passing a non-homgeneous volume. This is obvious considering the beam being shot through a human body comprising different tissues. Second, attenuation does not only depend on the material but also on the energy of the ray (Figure 3.1). Third, the intensity of the beam is *polyenergetic*.

Taking these considerations into account, the intensity $I_d$ of the beam has to be formulated as:

$$I_d = \int I_0(E) \cdot e^{-\int \mu(x,E)\mathrm{d}x} \mathrm{d}E \tag{3.2}$$

Even though equation 3.2 is physically correct, it is intractable for image reconstruction from a mathematical point-of-view [83]. Therefore, a *monoenergetic* source with an effective energy $\overline{E}$ can be assumend and the remaining intensity $I_d$ is described by:

$$I_d = I_0(\overline{E}) \cdot e^{-\int \mu(x,\overline{E})\mathrm{d}x} \tag{3.3}$$

Given the initial intensity $I_0$ and a remaining intensity $I_d$, equation 3.3 can be rearranged to yield a basic projection measurement $g_d$:

$$g_d := -ln\left(\frac{I_d}{I_0}\right) = \int \mu(x,\overline{E})\mathrm{d}x \tag{3.4}$$

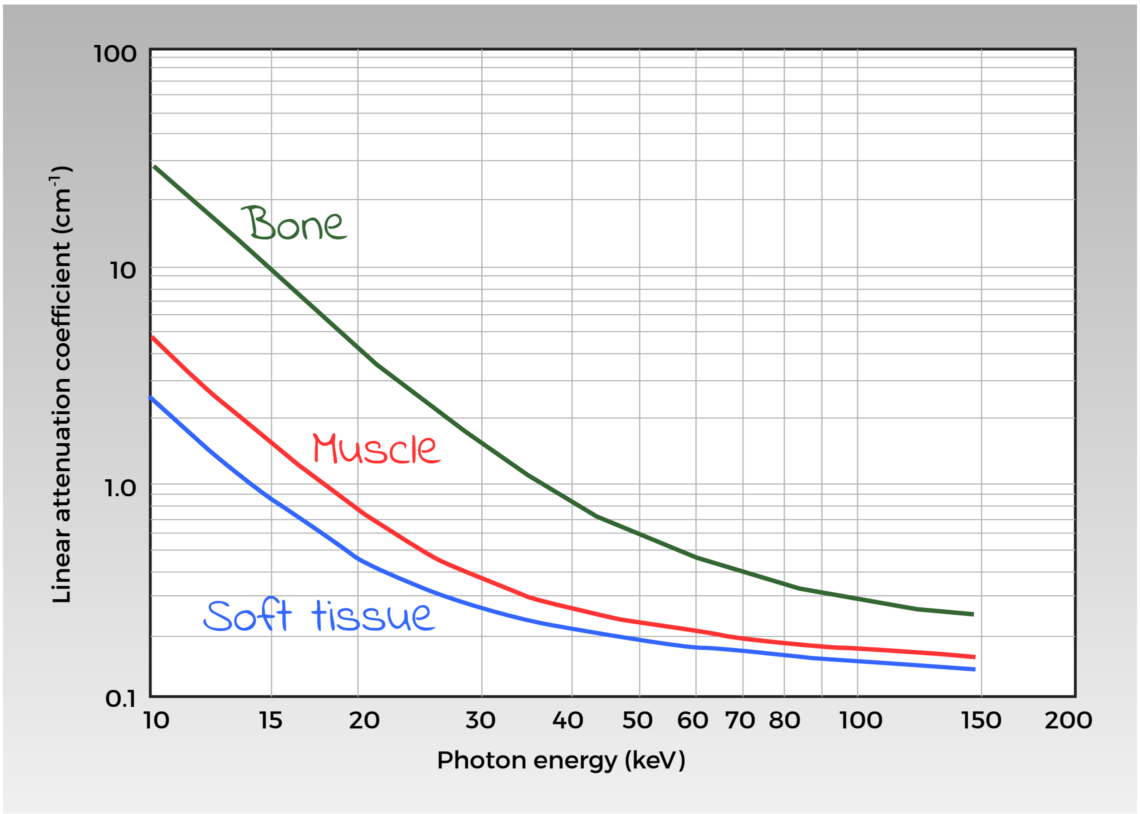


Figure 3.1: Linear attenuation depends on energy in a non-linear matter.

Equation 3.4 leads to a fundamental observation: **the measurements taken in radiographic imaging modalities are line integrals over the attenuation coefficient**. Figure 3.2 illustrates the meaning of this line integral. It depicts an X-ray beam being shot through a human body. The beam originates from an X-ray tube and it has an initial energy (intensity). The gray values of the (already reconstructed but preliminary unknown) cross-section correspond to the attenuation of the beam at a single point within the slice. The beam passes different layers of tissue until it's remaining intensity is finally captured by a detector. The remaining intensity is therefore related to the sum of the gray values along the beam. When this procedure is repeated with tube and detector moved into x direction, a 1D projection of the cross section at a certain angle is produced.

### 3.1.3 CT scanners

Medical radiography uses a wide range of different modalities to produce images. One of them is *projectional radiography.* It produces 2D images using X-ray radiation and is still widely used today. Another modality is **computed tomography** (CT). Taking X-ray measurements from different angles, tomographic images are obtained. By stacking these images on top of each other, specific areas of a scanned object can be reconstructed in three dimensions. Figure 3.3 shows the formerly mentioned 1D-projection of the cross section taken from different angles. The projections are apparently not the same for different angles. The different projections are used for later image reconstruction. Projections are taken from angles between 0°and 180°. Angles beyond 180° are not necessary as these would cause the beam to come from the opposite direction of an angle already used. Therefore projections beyond 180 degrees would result in yielding completely redundant information.

A tomographic image (*slice*) is a reconstruction of the value of $\mu$ at each pixel within a cross section. However, different scanners vary in their effective energy $\overline{E}$. Scanning a single object using different scanners would therefore lead to different numerical results. This makes the comparison of different scans impossible. To overcome this problem, so-called **CT-numbers** have been introduced. The CT number is defined as

$$h = 1000 \times \frac{\mu - \mu_{water}}{\mu_{water}}$$

and is expressed in *Hounsfield Units* (HU). These units are named after Godfrey Hounsfield who participated in the early development of computed tomography. $\mu_{water}$ is defined as 0 HU and is the radiodensity of destilled water at standard pressure and temperature. As air does not attenuate X-rays, $\mu_{air} = 0$ and therefore $-1000$ HU. Table 3.1 shows a list of some (organic) substances and their corresponding HU. It can be seen that the Hounsfield Units for different substances might vary and even overlap with the unit related to another substance. This gives an intuition of the segmentation problem described in section 3.3: while the lung can easily be distinguished from bones, blood and soft tissue are closer to each other in terms of radiodensity. As a matter of fact, vessel segmentation is not an easy task.

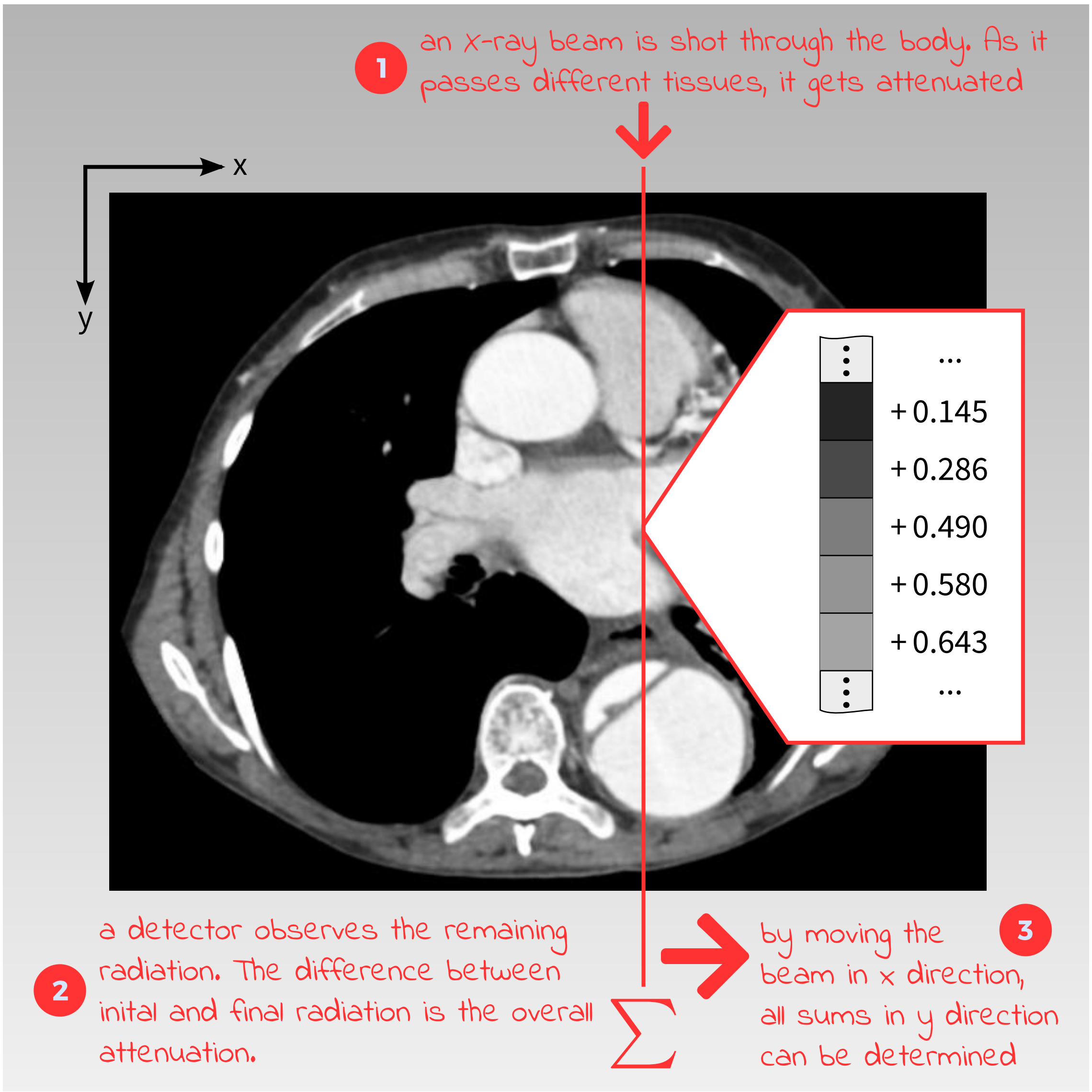
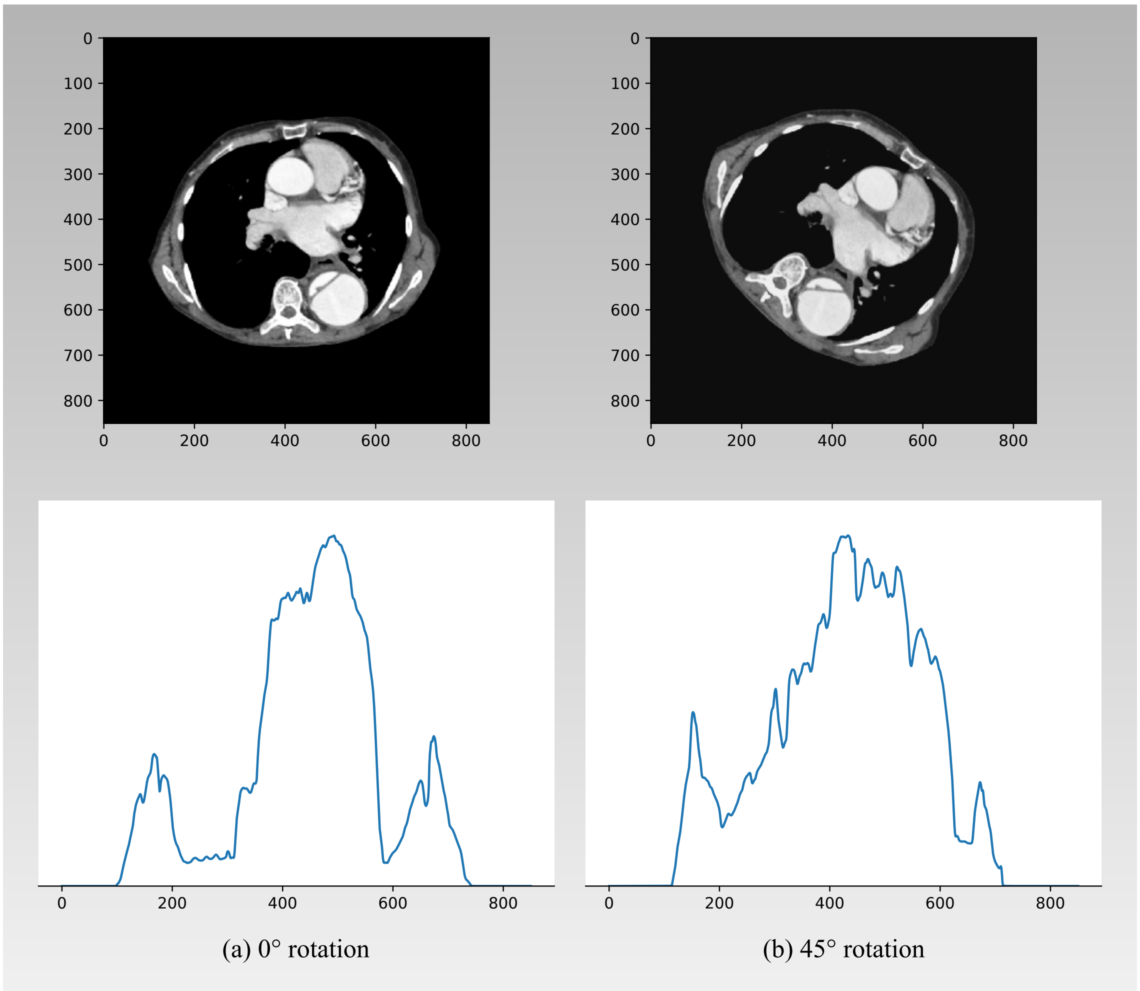

1
an X-ray beam is shot through the body. As it passes different tissues, it gets attenuated
x
y
...
+0.145
+0.286
+0.490
+0.580
+0.643
...
2
a detector observes the remaining radiation. The difference between inital and final radiation is the overall attenuation.
∑
3
by moving the beam in x direction, all sums in y direction can be determined

Figure 3.2: An X-ray beam is shot through the body. As it passes the body, it gets attenuated. The remaining intensity is captured by a detector.



(a) 0° rotation

(b) 45° rotation

Figure 3.3: As tube and detector rotate around the object of interest, different 1D projections are retrieved: for 0°(a) the projection is not the same as the one for 45°(b).

| **Substance** | **HU** |
|---|---|
| Water (reference) | 0 |
| Air | -1000 |
| Lung | -700 to -600 |
| Fat | -120 |
| Urine | -5 to +15 |
| Kidney | +20 to +45 |
| Blood | +13 to +50 |
| Muscle | +35 to +55 |
| Liver | +60 ± 6 |
| Soft Tissue, Contrast | +100 to +300 |
| Spongy bone | +700 |
| Compact bone | +3000 |

Table 3.1: Hounsfield Units for different (organic) substances

### 3.1.4 Image reconstruction using backprojection

Given a single cross section, a CT scanner produces a 1D projection for each angle between 0°and 180°(a certain stepsize is assumed). Stacking all these 1D projections on top of each other yields an image referred to as **sinogram**. A sinogram of a full cross section scan can be seen in figure 3.4. Each line of the sinogram depicts the 1D projection of the cross section at a given angle. Thus, the y-axis of the image relates to $\theta$. A simple algorithm to reconstruct the tomographic image given these 1D projections uses the sinogram. It is called **backprojection**.

Backprojection iterates over all horizontal lines of the sinogram. Each line is tiled in y direction resulting in a 2D image of the 1D projection. The height of the image matches the height of the sinogram, i.e. the 1D projection is repeated for the overall number of 1D projections available for the given scan. After generating the image, it is rotated by the angle $\theta$ it corresponds to. Finally, these images are **summed resulting in the reconstructed image**. Figure 3.5 illustrates the process of reconstruction given a simple 4x4 image with different intensities $a, b, c$ and $d$.

The method of **summation** can intuitively be explained as follows: given a 1D projection of the cross section, the overall sum is assumed to be the same on any point of the y-axis. Hence, a black line on the image indicates that there was (almost) no attenuation when the beam was shot through the cross section at the corresponding position given some angle $\theta$. When this assumption is made for all angles, overlapping them would **result in the original attenuation at any point** in $\mathbb{R}^2$. Due to the way the final image is reconstructed, it is referred to as **backprojection summation image**. Figure 3.6 shows some of the tiled 1D projections rotated by their corresponding angle. The projections are taken from the sinogram depicted earlier (Figure 3.4). Along with these titled projection, figure 3.6 shows the summation images obtained when summing up all titled projections up to the current angle.

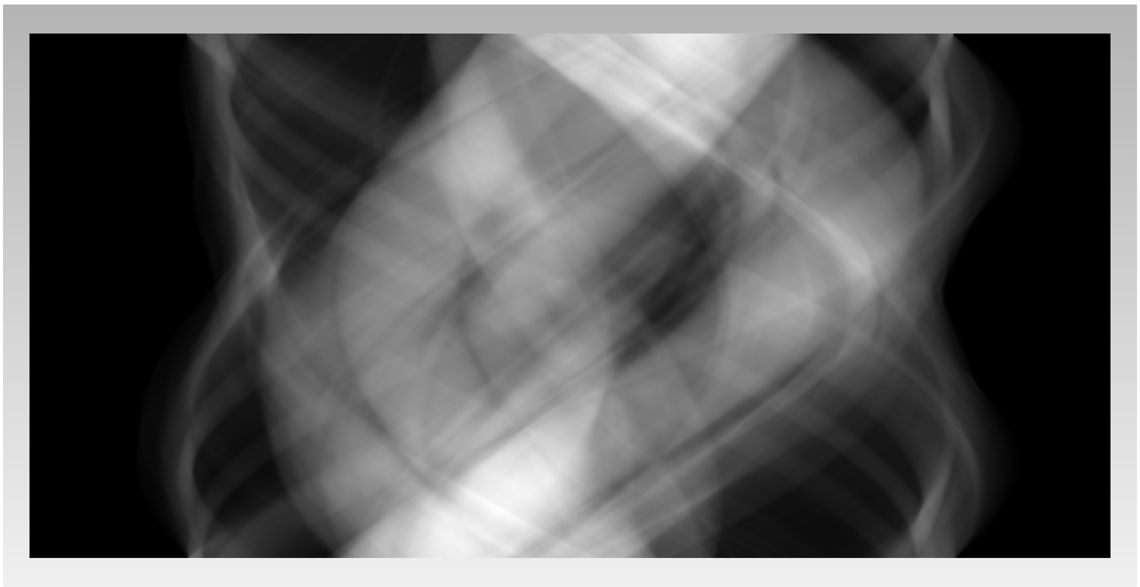


Figure 3.4: The unfiltered sinogram of a cross section scan. Each horizontal line of the sinogram depicts a 1D projection of the cross section taken at a specific angle.

In addition, figure 3.7 shows original cross section (left) and the final reconstructed summation image (right). It can clearly be seen that the summation image is far away from ground truth and very **blurry**. This is due to the fact that low frequent signals are highly overlapping in the sinogram. To **overcome this problem, filtered backprojection** is used.

### 3.1.5 Image reconstruction using filtered backprojection

Considering the sinogram depicted in figure 3.4, low frequent signals of the image are highly overlapping. To overcome this problem, the **sinogram is transformed into Fourier space**. After that, a **ramp filter** is applied to the transformed sinogram. Figure 3.8 shows the original signal (orange) and the modified signal (blue) after the ramp filter (green) has been applied. It can be seen that a modification of the original signal results in a new signal that suppresses low frequencies.

After the filter is applied to the sinogram in Fourier space, the sinogram is transformed **back into image space**. Figure 3.9 shows the sinogram after the ramp filter has been applied.

Using the filtered sinogram, **filtered backprojection works the exact same way the non-filtered version does**: each horizontal line of the sinogram is extracted, tiled and rotated. Finally, the sum of all tiled 1D projections is the reconstructed image. Figure 3.10 shows the tiled 1D projections and some intermediate summation images. When

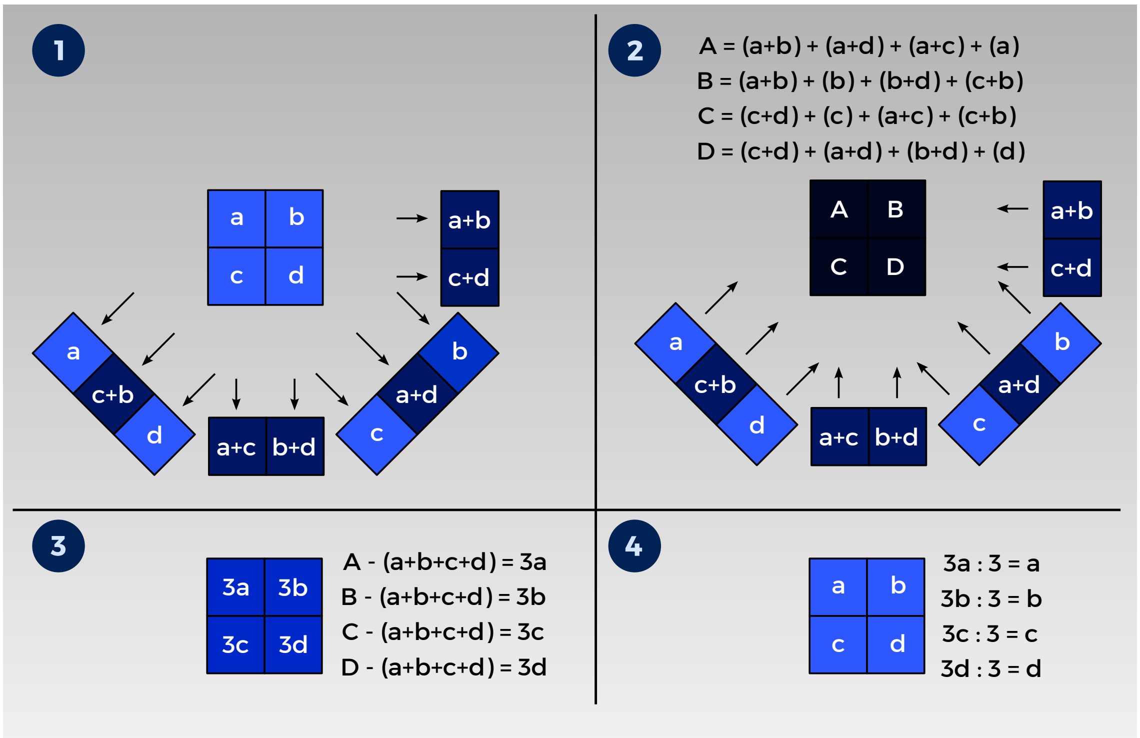


Figure 3.5: Given a (formerly unknown) image, 1D projections can be generated for different angles. The summation of these projections followed by a proper division yields the original image again (2-4).

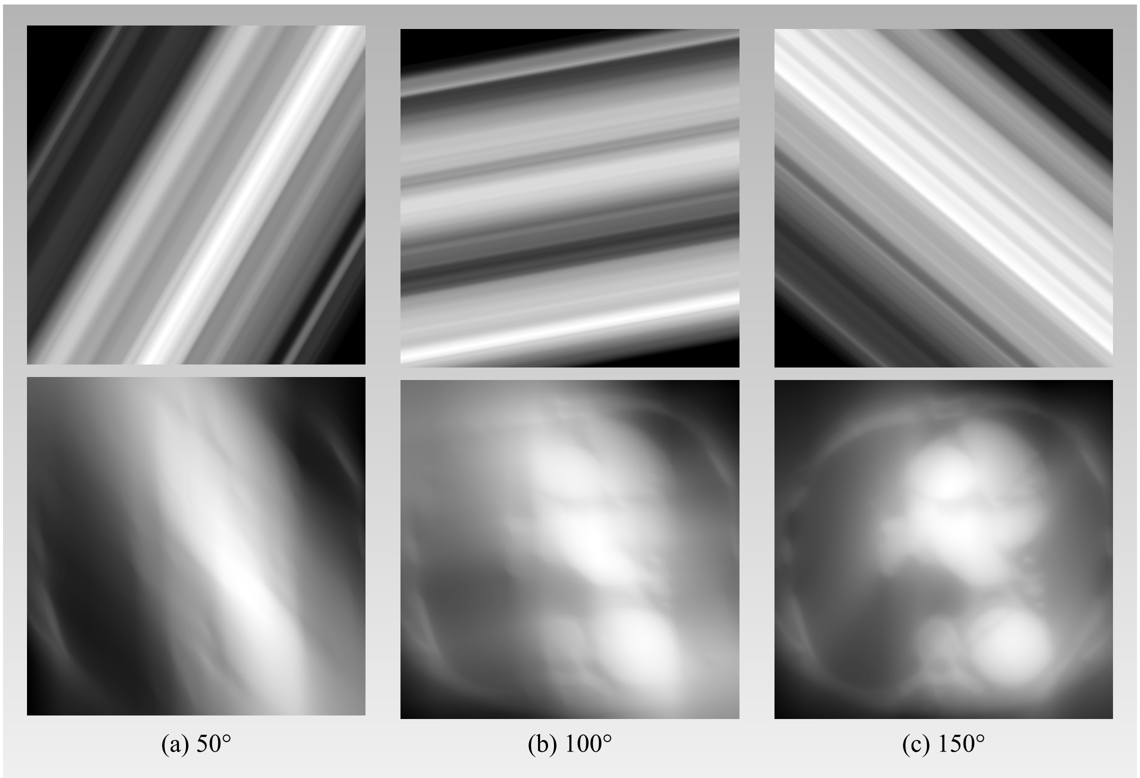


Figure 3.6: Tiled 1D projections and corresponding summation images for $\theta = 50$ (a), 100 (b), and 150 degrees (c). The more images are summed up, the more of the original image gets reconstructed.

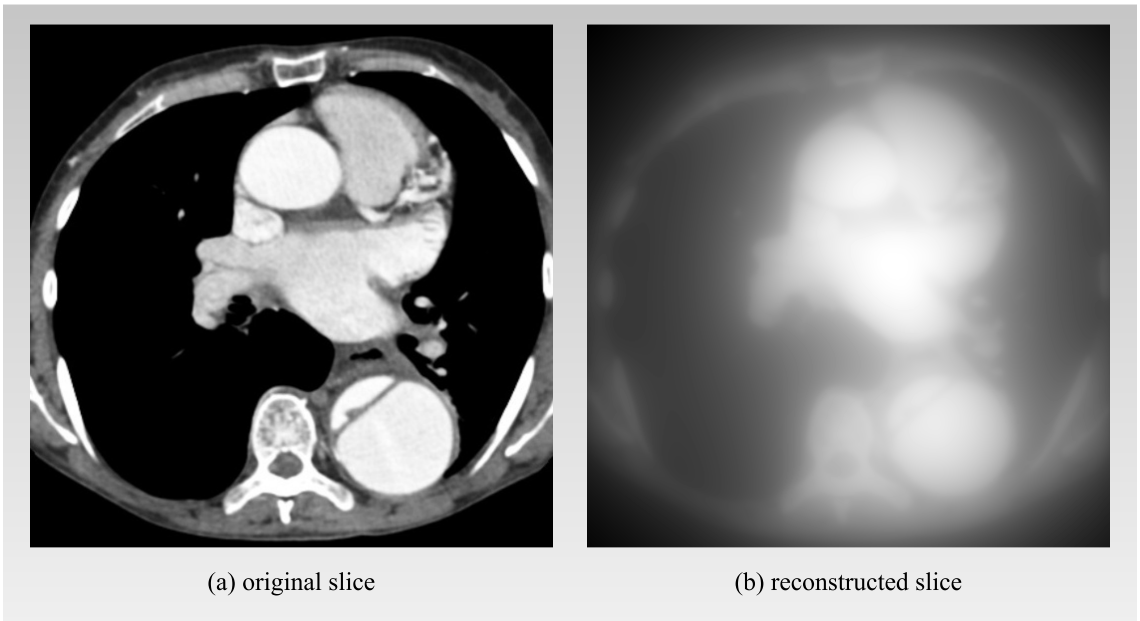


Figure 3.7: Using simple backprojection to reconstruct a tomography image (a) leads to blurry results (b) due to suppressed high frequencies in the sinogram.

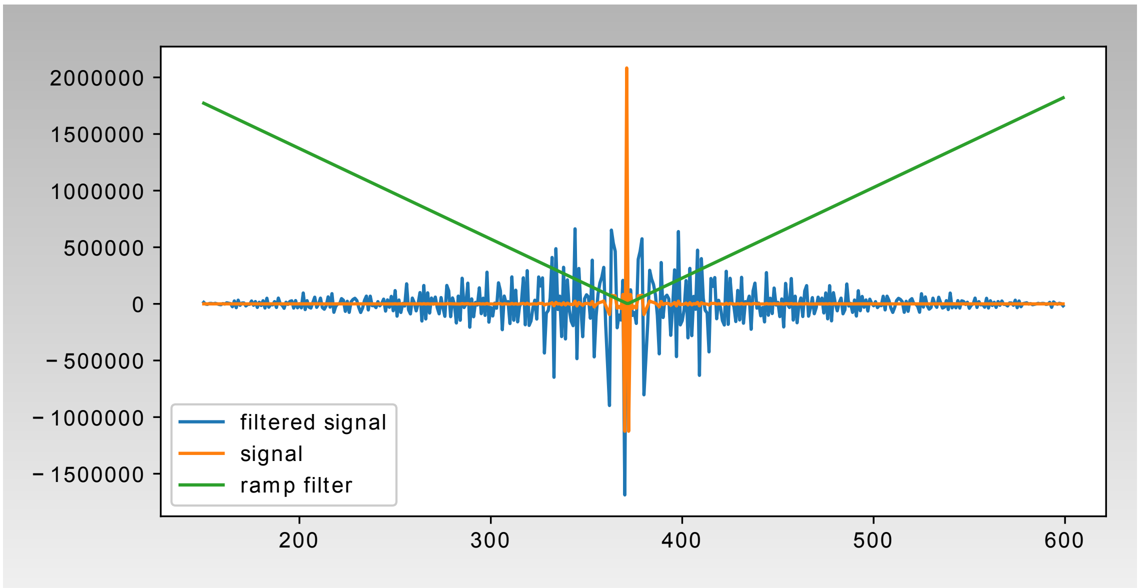


Figure 3.8: Translating the sinogram into Fourier space, it can be seen that low frequent signals are dominant. The application of a ramp filter supresses them in favor of high frequencies.

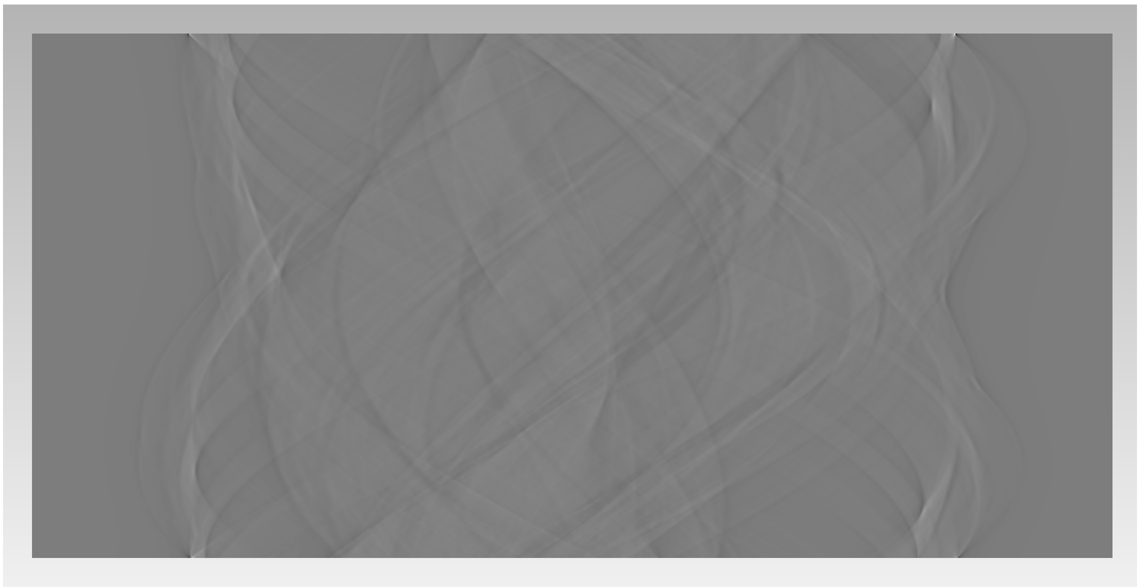


Figure 3.9: The filtered sinogram of a cross section scan. It is less blurry than the non-filtered one as low frequency signals of the image have been suppressed.

compared to figure 3.6, it can be seen that the intermediate results are far less blurry than the intermediate results that have used the non-filtered version of the sinogram.

The final result of the reconstructed image compared to the groundtruth of the cross section can be seen in figure 3.11. By applying the filter, low frequency signals where suppressed. The **reconstructed image is close to the (usually unknown) ground truth**.

New generation CT scanners provide some **technical improvements** to the process described above. One major improvement is that instead of using a single X-ray beam and a single detector, a so called cone beam and multiple detectors are used. In addition, a body is usually scanned while the table with the patient is constantly moving resulting in a helical scan that allows to rapidly collect multiple cross sections. These improvements allowed to highly increase the speed of image acquisition: while early clinical machines took up to several hours to scan a patient, modern scanners can do that job in under 100 milliseconds [83]. Even though these improvements have also made the mathematics of reconstruction more complex, the basic principle of reconstruction remains the same.

Whatever image reconstruction is finally chosen, a thoughtful way of **storing the retrieved data** has to be found. This issue is covered in the next chapter.

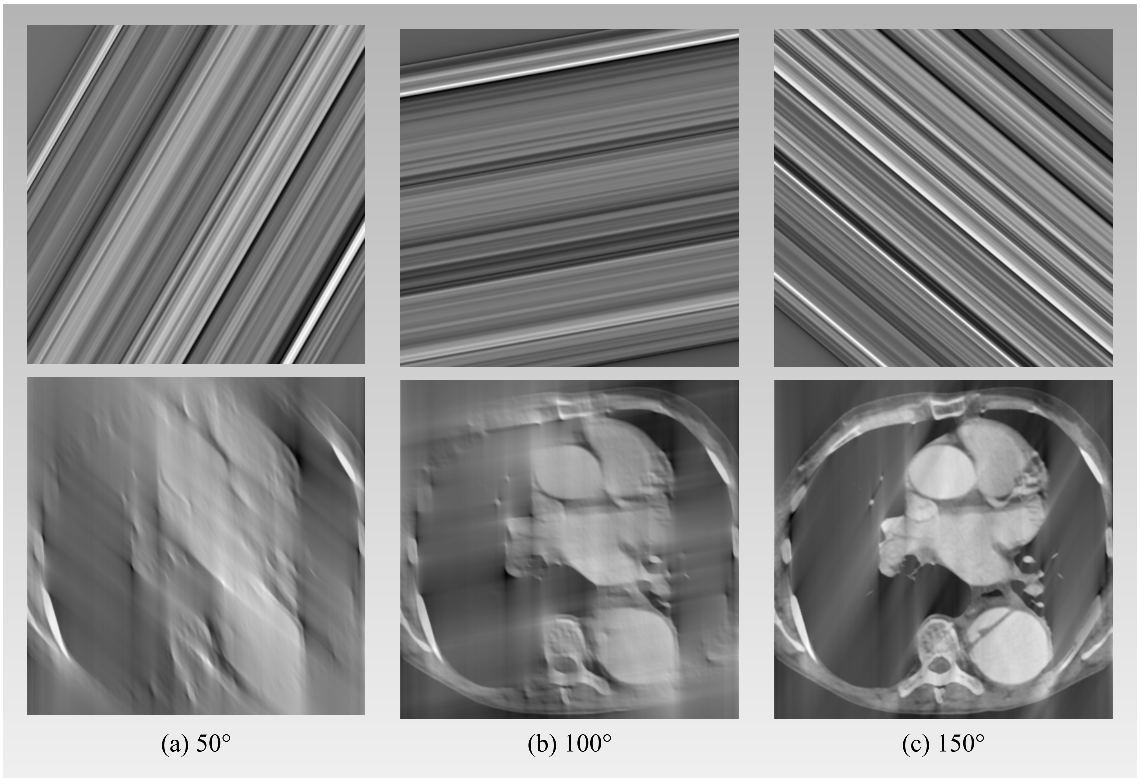


Figure 3.10: Above: Tiled 1D projections for $\theta = 50$ (left), 100 (center), and 150 degrees (right). Below: the more images are summed up, the more of the original image gets reconstructed.

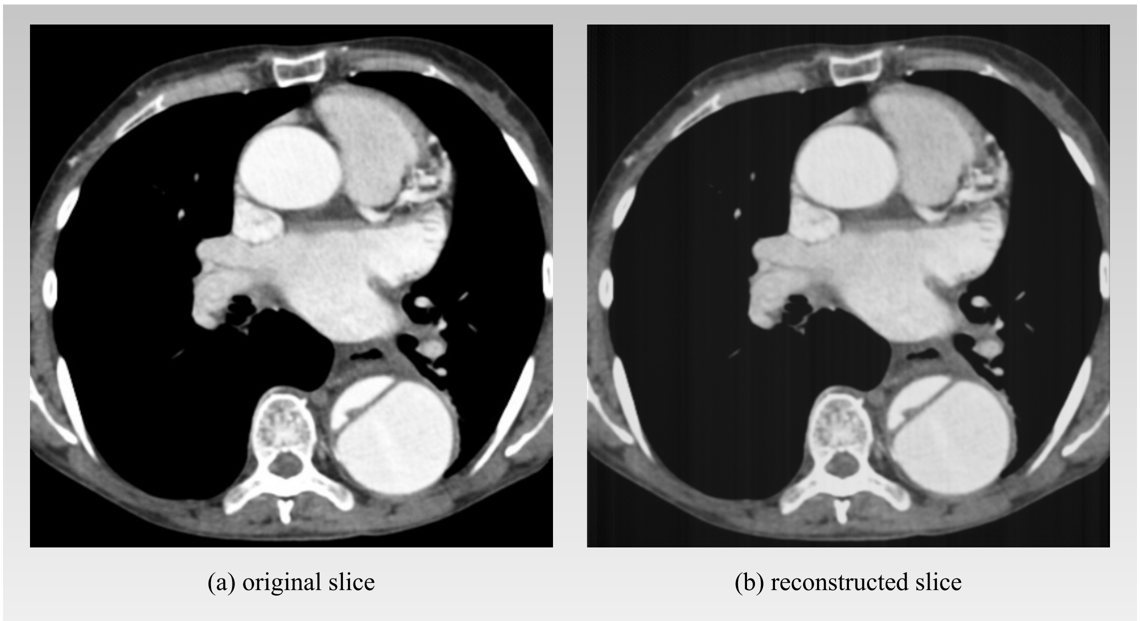


Figure 3.11: Filtering the sinogram in Fourier space enables a useful reconstruction (right) of the ground truth (left).

## 3.2 Storing medical images

After medical images are acquired by a device (e.g. a CT scanner), they have to be **stored digitally** for further processing or analysis.

Even though storing an image might seem to be a trivial task, the rapid expansion of telecommunication has contributed to the raise of the requirements for managing medical images in a thoughtful way. Müller describes the growing need for the standardization of storing and exchanging medical images resulting in **DICOM** (Digital Image and Communications in Medicine), an open standard which is well defined and widely spread today [84]. As already suggested by it's name, DICOM is far more than just a technical description on how to store images: it further defines a standard on how medical image data is supposed to be transferred and ensures that images are embedded in what DICOM calls the *real world context* [85]. Whenever possible, DICOM utilizes relevant parts of other standards such as JPEG or TCP/IP. Figures 3.12 - 3.14 show how DICOM defines the real world: a patient has one or more studies. Studies contain series comprised of different data.

In general, the DICOM model describes patients, studies, images, and other features of medical imaging. In addition, the relation between these entities is defined. The entity model connects real-world entities to the classes that model the corresponding DICOM information entities [86]. DICOM ensures, that these **additional information is always stored along with the corresponding pixel data**. Therefore, by using the file format proposed by DICOM, medical image data always comprises information about the patient,

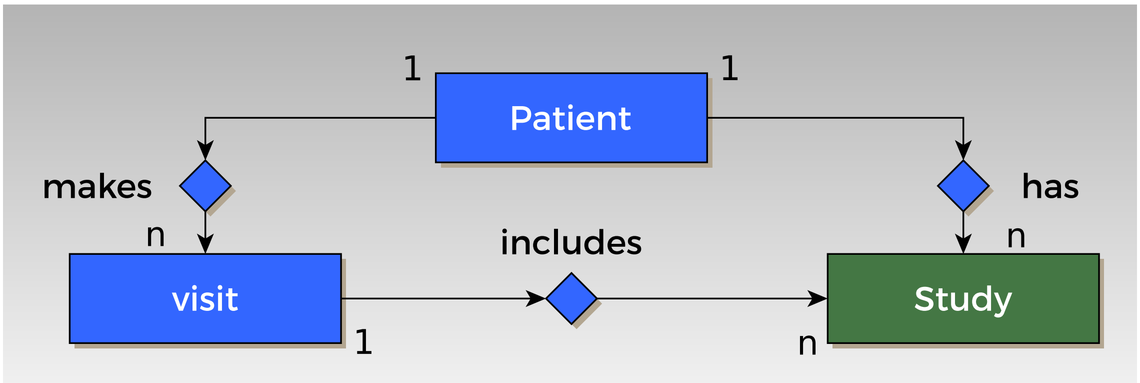


Figure 3.12: DICOM Model of the Real World: a patient and related studies

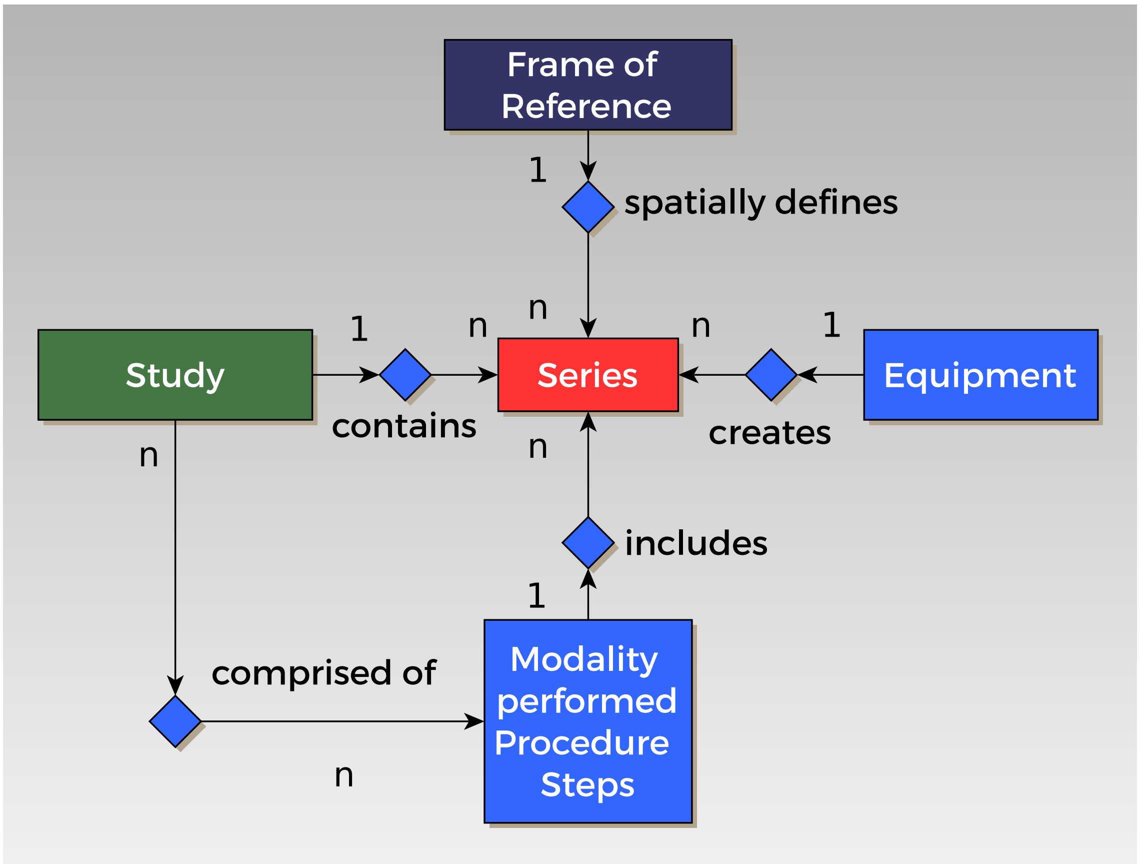


Figure 3.13: DICOM Model of the Real World: how to retrieve series

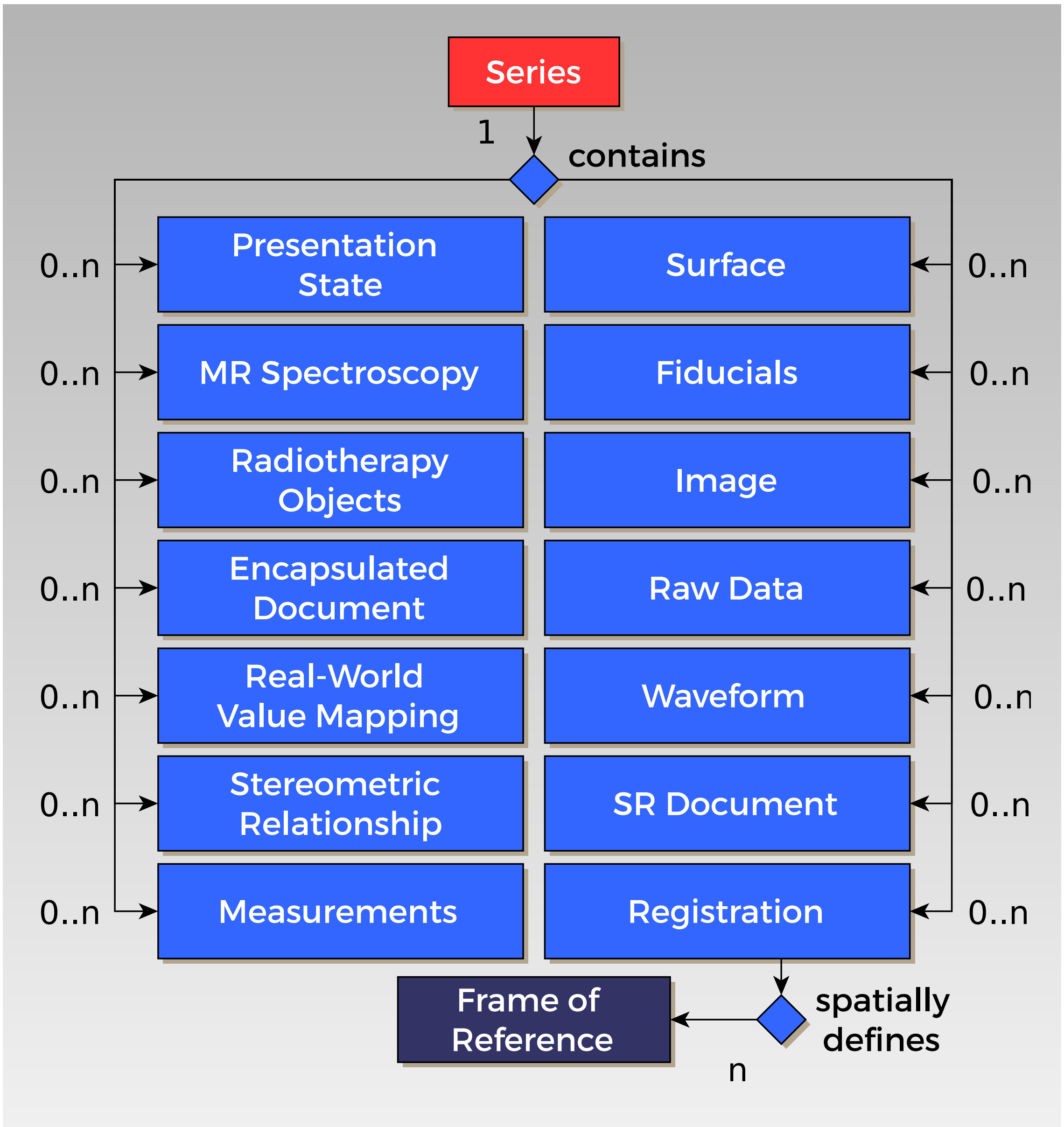


Figure 3.14: DICOM Model of the Real World: series contain various data

about when which diagnosis or treatment has been given, and about which devices have been used.

Technically, **a DICOM file acts as a container**. Even though it contains just a single node for pixel data, the concept of *frames* allows DICOM to not only store single images but rather three- or four-dimensional data. The DICOM file format is widely supported by medical imaging devices aiming towards vendor independent management of medical image data. In addition, DICOM is well understood by various software applications such as [87, 88, 89, 90, 91, 92, 93, 94, 95]. On top of that, many libraries and toolkits support DICOM, e.g. [96, 97, 98, 99]. DICOM is used in any field of medicine where imaging is prevalent, including: radiology, cardiology, oncology, radiotherapy, neurology, orthopedics, gynecology, dermatology, pathology, and others.

Summing up, there exists a variety of techniques for retrieving medical images. However, even though these techniques widely differ in how an image is acquired, the **storage of the image itself calls for a common standard** in order to exchange devices and visualization software. In addition, the medical domain requires the storage of images to be in such a way that relevant real world information is uniquely linked to the image or a series of images. With DICOM as a file format, these needs are covered.

## 3.3 Image segmentation

### 3.3.1 A definition of image segmentation

In 1978, Barrow and Tenenbaum described segmentation as the process of partitioning an image into semantically interpretable regions [100]. Approximately 10 years later, Leclerk distinguishes between *image* partitioning and *scene* partitioning and by defining image partitioning as the delineation of regions that have coherent attributes in the image, he considers it to be the basis of image understanding [101]. A few years later, Haralick and Shapiro come up with another definition that summarizes what has been thought before: they refer to segmentation as the partition of an image into a set of non-overlapping regions whose union is the entire image. They mention the purpose of segmenation seperately: to decompose the image into parts that are meaningful with respect to a particular application [102]. This definition is simplified by Shapiro and Stockman as they define segmentation to be the partition of an image into a set of meaningful regions that cover it [103]. Szeliski calls image segmentation the task of finding groups of pixels that go together [104].
Besides verbal definitions, there exist some formal definitions on image segmentation. A recent one is given by Salem et al. [105]: considering a digital image $I$ to be a function that maps a discrete image position $(m, n)$ to a color $C$

$$I : \{(m, n) : 1 \leq m \leq M, 1 \leq n \leq N\} \to C,$$

the segmentation can analogously be described as a function $S$ mapping each discrete image position $(m, n)$ to a class $K$:

$$S : \{(m, n) : 1 \leq m \leq M, 1 \leq n \leq N\} \to K$$

Even though there are various definitions on image segmentation, literature agrees on the fact that segmentation is an old and hard problem that is still under wide and active research. There are numerous algorithms that try to solve the problem of image segmentation. In 2006, the Ei Compendex listed more than 4000 segmentation algorithms [106]. They can basically be divided into four types of segmentation techniques: thresholding, boundary-based, region-based, and hybrid techniques [107].

### 3.3.2 Segmentation based on thresholding

Algorithms belonging to the family of thresholding techinques divide a gray-valued picture into background and foreground using a threshold. Given a threshold value $t$, the segmentation mask $M$ that assigns a 1 to each foreground pixel and a 0 to each background pixel can formally be written as:

$$M(u, v) = \mathbf{1}_{\{I(u,v)<t\}} = \begin{cases} 1 \text{ if } I(u, v) < t \\ 0 \text{ else} \end{cases}$$

Different thresholding methods differ when is comes to treating the problem of how to find the right threshold automatically [108, 109, 110, 111, 112, 113]. Even though thresholding is a simple concept, it has successfully been applied to CT scans when threshold values were derived using radiographs instead of reconstructed images [114, 115].

### 3.3.3 Boundary-based segmentation

Boundary-based algorithms assume that there is a significant change in pixel properties when regions change. These algorithms can further be subdivided into ridge-based and edge-based methods. Ridge-based methods try to follow local maxima in the original image. In contrast, edge-based methods try to track those peaks in the gradient space. The two dimensional gradient of a (discrete) image leads to an image showing the edges of the original as depicted in Figure 3.15.

Edge-based algorithms often lead to problems with missing, spurious or discontiuous edges. In addition, noise will cause the gradient operator to produce unwanted, artificial edges. Trying to overcome this problem by simply smoothing the image before edge detection might be suitable in simple cases (artificial images) but cause real edges to vanish in general (natural images). Many algorithms try to solve that problem by introducing new ideas based on simple gradient operations [116, 117, 118, 119, 120]. Another, more complex attempt to overcome this problem is to introduce more sophisticated frameworks such as active countour [121]. However, even though these approaches are frequently used today, the cannot solve the segmentation problem reliably in general [122].

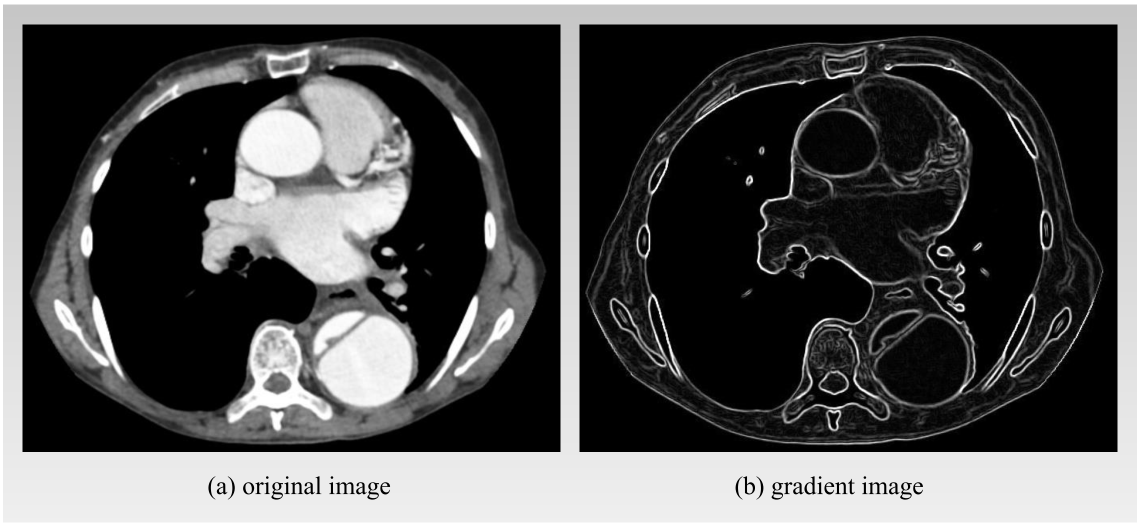


Figure 3.15: When the gradient operator is applied to an image, its edges get amplified. However, edges are sometimes missing, spurious or discontinuous.

### 3.3.4 Region-based segmentation

Region-based algorithms assume that regions in an image have similar properties. A formal definition of region-based segmentation is given by Zucker [123] and requires any pixel (3.5a) to be in one (3.5b) and only one region, (3.5c) to be connected to at least one other pixel of the same region, and (3.5d) to have some similarity to any other pixel within its region. Finally, no two adjacent regions are allowed to consist of pixels fulfilling the same predicate $P$ (3.5e) where $P$ asks whether a specific property holds for a given pixel and can be thought of as a similarity measurement.

$$R = \bigcup_{i=1}^{n} R_i \tag{3.5a}$$

$$R_i \bigcap R_j = \emptyset \quad \forall i, j \tag{3.5b}$$

$$R_i \text{ is a connected region } \forall i \tag{3.5c}$$

$$P(R_i) = TRUE \quad \forall i \tag{3.5d}$$

$$P(R_i \bigcup R_j) = FALSE \text{ for any adjacent region } R_i \text{ and } R_j \tag{3.5e}$$

Region-based segmentation is prone to error: due to limitiations in spacial resolution of CT or MR images, false bridges between two distinct regions might cause such algorithms to interpret the two regions as one. Another source of error could be a region that is partially occluded by another region with similar properties. There are several proposals on how region growing could be implemented [124, 125, 126]. The bottleneck of region-based approaches is to find a good predicate. Many solutions require manual guidance such as the definition of seed points or the preliminary knowledge of regions (k-means).

### 3.3.5 Hybrid segmentation approaches

Hybrid methods combine different segmentation techniques to improve their results. Some of these methods try to extract some knowledge from statistical features [127, 128, 129]. They often combine edge-based and region-based approaches [127, 128, 130, 131]. Many hybrid methods either require manual tuning of parameters [130] or initial seed points [132, 133]. Some solutions depend on mathematical morphology as morphology is considered to be attractive for this purpose [134]. These methods are said to be able to efficiently deal with geometrical features such as size, shape, contrast or connectivity, features, that are considered to be related to objects in an image [134]. Therefore, they are supposed to be well suited for image segmentation. One algorithm that is based on mathematical morphology and is implemented in various, freely available medical imaging software packages [87, 94] is watershed [135, 136]. The original concept of watershed is taken from geology: a source of water is placed in each local minima. As the valleys get flooded, watersheds develop according to geodesic saliency [137].

### 3.3.6 Segmentation in medical imaging

Regardless of the underlying image modality, image segmentation plays a crucial role in medical imaging. It is used in various applications such as the location of tumors [138], the measuring of tissue volumes, the segmentation of organs [139], the diagnosis of anatomical structures [140], surgery planning and many others. Modern approaches of image segmentation are often based on machine learning [141, 142, 143, 144]. However, image segmentation is a hard problem, especially when it comes to medical images: conventional algorithms such as methods based on threshold often struggle with the lack of distinct edges between tissues while machine learning algorithms often fail due to small, highly artificial or rarely classified training sets. Typical applications of image segmentation usually require an expert to enforce prior knowledge to the underlying algorithms, fine-tune parameters, and manually improve the segmentation proposed by any applied algorithm. This is especially the case when it comes to the segmentation of vessels as they are often hard to distinguish from surrounding tissue. Freely available medical imaging software such as InVesalius [94] or 3DSlicer [87] offer different segmentation algorithms. They deliver acceptable results for easy-to-solve segmentation tasks such as the segmentation of the lung but when it comes to complex problems such as the segmentation of a dissected aorta, manual correction is required to gain a usable result. For a complete automation of the segmentation, there is a need for further investigation on more advanced algorithms such as [145, 146, 147, 148, 149, 150].

## 3.4 Image modelling

### 3.4.1 Definition and types of 3D models

Rendering a 3D object in Virtual Reality requires a so-called **3D model** of the object. A 3D model is a digital (mathematical) representation of a three-dimensional object.

There exist various concepts how these models are described mathematically and - as a matter of consequence - stored digitally. These concepts can basically be divided into three types: wireframe models, solid models and surface models [151].

**Wireframe models** describe the outer hull of a 3D object using straight lines, conics and simple spline curves [151].

On the contrary, **solid models** describe the whole volume of a 3D object. They are composed of 3D primitives being merged by using different simple mathematical operations such as addition or subtraction. As a matter of fact, solid models provide informational completeness regarding their volume. Given that, they are often used in the CAD/CAM domain.

However, even though both, wireframe and solid models are a good solution for modelling "artificial" objects, it is hard to use them when it comes to the modelling of highly complex, "natural" 3D objects. This is due to the fact that both types of models use mathematical functions to describe a 3D object. As such, they can be considered simple and abstract and they lack of required accuracy when modelling natural objects in a true-to-detail manner.

A better suited model for that purpose is the **surface model**. Surface models, just as wireframe models, describe the hull of a 3D object. Unlike solid models, surface models do not allow holes or cracks. Moreover, they can be geometrically incorrect. They often approximate the hull of the 3D object by using **polygon modelling**. As convexity is a required property of a polygon to ensure a correct rendering in 3D [152], a widely used polygon is the triangle. Thus, the 3D object is finally described by a triangle mesh.

### 3.4.2 Data structures and file formats for digital 3D models

There are various data structures to represent and file formats to save digital 3D models. This work uses two of them: WEM and STL.

The **WEM** (Winged Edge Mesh) data structure initially described by Baumgart [153]. It is one way to digitally represent polygon models. The data structure explicitly links mesh-related primitives when surfaces meet at a common edge. These primitives are: vertices, edges and faces. A vertice is basically a point in 3D. Connecting two of these points results in an edge. A face is an area bounded by edges. In case of triangulated meshes, faces are triangles. By linking these primitives together, 3D models can be described in a way that allows important surface operations such as adjacency queries to be performed efficiently.

**STL** (abbrev. "stereolithography") is a widely spread and simple file format for storing 3D models. It is well understood by various 3D software applications. STL is restricted to surface descriptions, i.e. further model information such as color, texture, scaling or additional meta information is usually not preserved when saving files in this format. How-

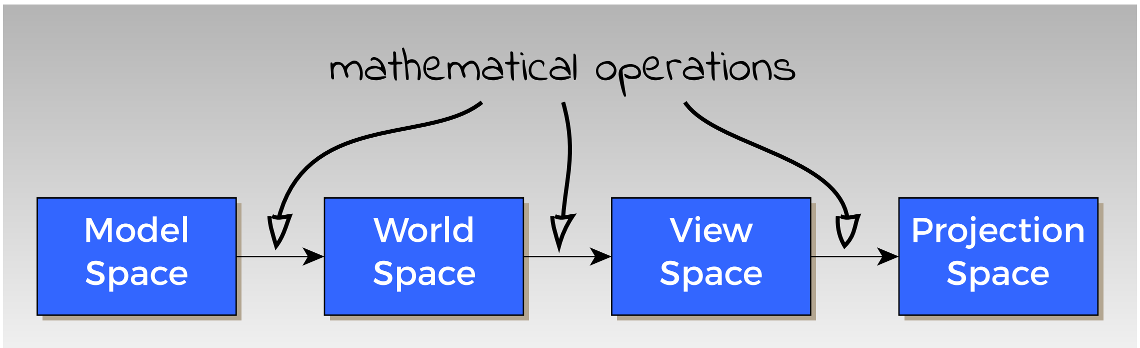


Figure 3.16: The application of mathematical operations allow to translate a 3D model into a 3D scene and project the 3D scene onto a plane.

ever, as CT scans do not contain any of these information, STL is a suitable intermediate output for the purpose of this work. In STL, a triangulated surface is described by a list of triangles. Each triangle is again described by it's unit normals and it's vertices. STL can be saved in either ASCII or binary format.

## 3.5 Mathematical Background of VR Visualization

### 3.5.1 The 3D model and different coordinate spaces

Rendering a scene to Virtual Reality is similar to conventional screen rendering. From a high-level point-of-view, it applies various mathematical operations on every vertex of the model until the final image is obtained. These operations basically **transform the model from one so-called coordinate space into another**. Figure 3.16 shows the the different coordinate spaces the model is transformed into. First, the model is represented in *model space*. When loaded into a scene, the origin of model space is set to the origin of world space. The first transformation translates the model into *world space* by applying scaling, translation and rotation to the model. After that, a second transformation translates the model into *view space* (sometimes called *camera space*). This allows the scene to be seen from the cameras (or eyes) point-of-view. Finally, the model is translated into *projection space* by skewing it in such a way, that a 3D effect is retained on a 2D plane.

The following section derives the **mathematical operations needed for the final translation**. First, a 2D model is considered. Then, this model is extended to 3D. Finally, the translation process is extended to fit Virtual Reality.

### 3.5.2 Placing a 2D model in a 2D world

Figure 3.17a shows a simple model. It can be described by the following *model coordinates*:

$$v_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, v_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, v_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, v_4 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{3.6}$$

The model can be placed anywhere in the world (i.e. $v_1$ is not at the origin any more) by **scaling**, **rotating** and **translating** it.

### Scaling the model

To scale the model, a **factor is multiplied** to each of it's vertices x and y coordinate. In Figure 3.17b the original model (gray) is scaled to half of it's size (blue) using a scaling vector $s = \begin{pmatrix}0.5 & 0.5\end{pmatrix}^T$. The scaled vertices are obtained by pointwise multiplication (denoted by $\circ$ here):

$$v_1 \circ s = \begin{pmatrix}0\\0\end{pmatrix}, v_2 \circ s = \begin{pmatrix}0.5\\0\end{pmatrix}, v_3 \circ s = \begin{pmatrix}0.5\\0.5\end{pmatrix}, v_4 \circ s = \begin{pmatrix}0\\0.5\end{pmatrix} \tag{3.7}$$

### Rotating the model

Figure 3.18 illustrates the principle of rotation. It works by applying **trigonometric functions** on a rotation angle. The coordinates of a single point $p$ originally rotated by $\alpha$ can be expressed as:

$$\begin{aligned} x(\alpha) = x &= l \cdot cos(\alpha) \\ y(\alpha) = y &= l \cdot sin(\alpha) \end{aligned} \tag{3.8}$$

$p'$ is the result of $p$ being rotated another $\theta$ degrees. The new coordinates are:

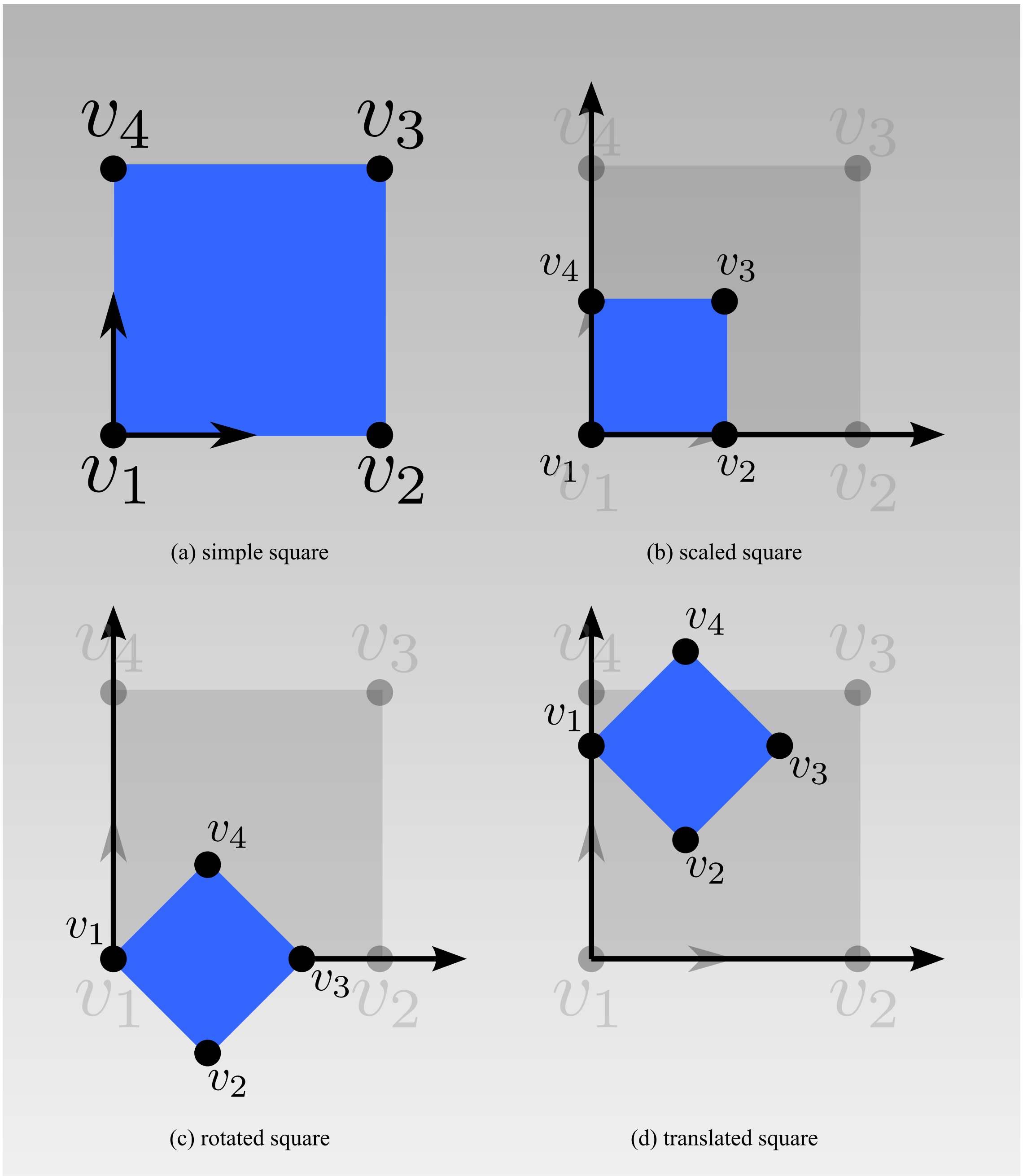
$$\begin{aligned} x'(\alpha,\theta) = x' &= l \cdot cos(\alpha+\theta) = l \cdot cos(\alpha)cos(\theta) - l \cdot sin(\alpha)sin(\theta) \\ y'(\alpha,\theta) = y' &= l \cdot sin(\alpha+\theta) = l \cdot sin(\alpha)cos(\theta) + l \cdot cos(\alpha)sin(\theta) \end{aligned} \tag{3.9}$$

Substituting 3.9 using 3.8 yields:

$$\begin{aligned} x' &= x \cdot cos(\theta) - y \cdot sin(\theta) \\ y' &= x \cdot sin(\theta) + y \cdot cos(\theta) \end{aligned} \tag{3.10}$$

Applying 3.10 to the model given in 3.7, a 45° rotation yields (Figure 3.17c):

$$\begin{aligned} v_1.x = v_1.x \cdot cos(45^\circ) - v_1.y \cdot sin(45^\circ) &= 0 \\ v_1.y = v_1.x \cdot sin(45^\circ) + v_1.y \cdot cos(45^\circ) &= 0 \\ v_2.x = v_2.x \cdot cos(45^\circ) - v_2.y \cdot sin(45^\circ) &\approx 0.35 \\ v_2.y = v_2.x \cdot sin(45^\circ) + v_2.y \cdot cos(45^\circ) &\approx -0.35 \\ v_3.x = v_3.x \cdot cos(45^\circ) - v_3.y \cdot sin(45^\circ) &\approx 0.71 \\ v_3.y = v_3.x \cdot sin(45^\circ) + v_3.y \cdot cos(45^\circ) &= 0 \\ v_4.x = v_4.x \cdot cos(45^\circ) - v_4.y \cdot sin(45^\circ) &\approx 0.35 \\ v_4.y = v_4.x \cdot sin(45^\circ) + v_4.y \cdot cos(45^\circ) &\approx 0.35 \end{aligned} \tag{3.11}$$



(a) simple square

(b) scaled square

(c) rotated square

(d) translated square

Figure 3.17: Different operations applied subsequently on a simple model: transformation from model coordinates into world coordinates

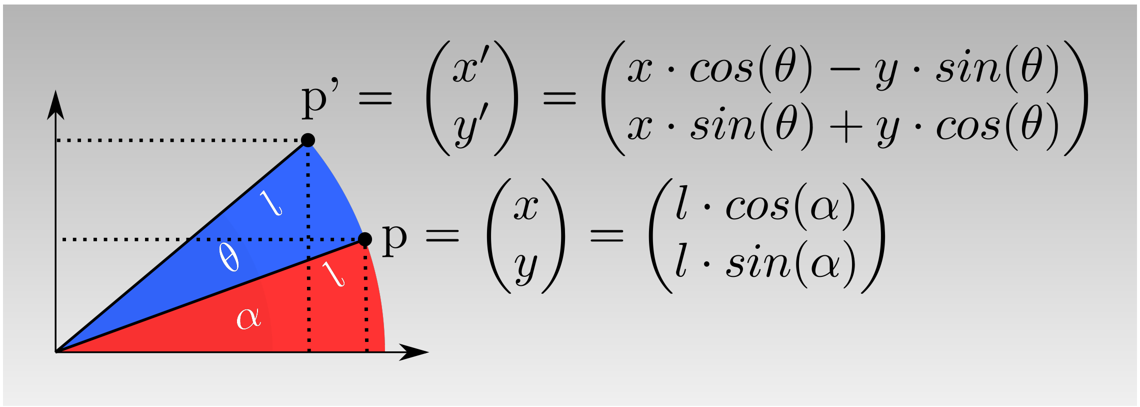

$p' = \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \cdot cos(\theta) - y \cdot sin(\theta) \\ x \cdot sin(\theta) + y \cdot cos(\theta) \end{pmatrix}$
$p = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} l \cdot cos(\alpha) \\ l \cdot sin(\alpha) \end{pmatrix}$
l
θ
l
α

Figure 3.18: A point originally rotated by $\alpha$ is now rotated by $\theta$

**Translating the model**

Translation can be obtained by **adding a translation vector** (e.g. $t = \begin{pmatrix} x_{off} & y_{off} \end{pmatrix}^T = \begin{pmatrix} 0 & 0.8 \end{pmatrix}^T$) to the models vertices:

$$v_1 + t = \begin{pmatrix} 0 \\ 0.8 \end{pmatrix}, v_2 + t \approx \begin{pmatrix} 0.35 \\ 0.45 \end{pmatrix}, v_3 + t \approx \begin{pmatrix} 0.71 \\ 0.8 \end{pmatrix}, v_4 + t \approx \begin{pmatrix} 0.35 \\ 1.15 \end{pmatrix} \tag{3.12}$$

Figure 3.17d shows the final position of the model in world space.

### 3.5.3 A framework for mathematical convenience

A combination of all translation operations yields:

$$\begin{aligned}
v_1.x &= (0.5 \cdot v_1.x) \cdot cos(45^\circ) - (0.5 \cdot v_1.y) \cdot sin(45^\circ) + 0 && = 0 \\
v_1.y &= (0.5 \cdot v_1.x) \cdot sin(45^\circ) + (0.5 \cdot v_1.y) \cdot cos(45^\circ) + 0.8 && = 0.8 \\
v_2.x &= (0.5 \cdot v_2.x) \cdot cos(45^\circ) - (0.5 \cdot v_2.y) \cdot sin(45^\circ) + 0 && \approx 0.35 \\
v_2.y &= (0.5 \cdot v_2.x) \cdot sin(45^\circ) + (0.5 \cdot v_2.y) \cdot cos(45^\circ) + 0.8 && \approx 0.45 \\
v_3.x &= (0.5 \cdot v_3.x) \cdot cos(45^\circ) - (0.5 \cdot v_3.y) \cdot sin(45^\circ) + 0 && \approx 0.71 \\
v_3.y &= (0.5 \cdot v_3.x) \cdot sin(45^\circ) + (0.5 \cdot v_3.y) \cdot cos(45^\circ) + 0.8 && = 0.8 \\
v_4.x &= (0.5 \cdot v_4.x) \cdot cos(45^\circ) - (0.5 \cdot v_4.y) \cdot sin(45^\circ) + 0 && \approx 0.35 \\
v_4.y &= (0.5 \cdot v_4.x) \cdot sin(45^\circ) + (0.5 \cdot v_4.y) \cdot cos(45^\circ) + 0.8 && \approx 1.15
\end{aligned} \tag{3.13}$$

Using matrices, 3.13 **can be simplified** to:

$$\begin{aligned}
v_1 &= T \cdot R \cdot S \cdot v_1, \\
v_2 &= T \cdot R \cdot S \cdot v_2, \\
v_3 &= T \cdot R \cdot S \cdot v_3, \\
v_4 &= T \cdot R \cdot S \cdot v_4
\end{aligned} \tag{3.14}$$

where $T$, $R$ and $S$ are translation-, rotation-, and scaling matrix:

$$T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.8 \\ 0 & 0 & 1 \end{pmatrix}, R = \begin{pmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}, S = \begin{pmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.15}$$

$T$ needs to be $3x3$; $R$, $S$ and all points are extended to allow a consistent mathematical framework. The extended points are called *homogeneous coordinates* and the extension $\begin{pmatrix} x & y \end{pmatrix}^T \mapsto \begin{pmatrix} x & y & 1 \end{pmatrix}^T$ is called a *transformation into projective space*. Finally $T \cdot R \cdot S$ can be precomputed. The result is known as the **model matrix** $M$:

$$M = T \cdot R \cdot S = \begin{pmatrix} 0.5 \cdot cos(\theta) & -0.5 \cdot sin(\theta) & 0 \\ 0.5 \cdot sin(\theta) & 0.5 \cdot cos(\theta) & 0.8 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.16}$$

Any point (e.g. $v_4$) can now be translated using $M$:

$$v_4 = M * v_4 = \begin{pmatrix} 0.5 \cdot cos(\theta) & -0.5 \cdot sin(\theta) & 0 \\ 0.5 \cdot sin(\theta) & 0.5 \cdot cos(\theta) & 0.8 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \approx \begin{pmatrix} 0.35 \\ 1.15 \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} 0.35 \\ 1.15 \end{pmatrix} \tag{3.17}$$

On top of that, matrix notation allows all model coordinates of an arbitrary model $V$ to be translated into world coordinates $V'$ at once:

$$V' = M \cdot V = M \cdot \left( \frac{v_1 \; v_2 \; ... \; v_{n-1} \; v_n}{\mathbf{1}} \right) = \left( \frac{v_1' \; v_2' \; ... \; v_{n-1}' \; v_n'}{\mathbf{1}} \right) \tag{3.18}$$

In case of the given model, this multiplication results in:

$$\begin{pmatrix} 0.5cos(\theta) & -0.5sin(\theta) & 0 \\ 0.5sin(\theta) & 0.5cos(\theta) & 0.8 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0.35 & 0 & -0.35 \\ 0.8 & 1.15 & 1.15 & 1.15 \\ 1 & 1 & 1 & 1 \end{pmatrix} \tag{3.19}$$

3.18 and 3.19 show that this method is apparently convenient in notation. However, more important and due to respective hardware design computational performance is increased using matrix multiplication.

### 3.5.4 An extension to 3D

Extending scaling and translation to three dimensions is rather trivial:

$$S = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, T = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.20}$$

On the contrary, rotation gets more complicated in 3D. While a point is simply rotated around it's origin in 2D, 3D rotation can occur around any of the axis (x, y, z). However,

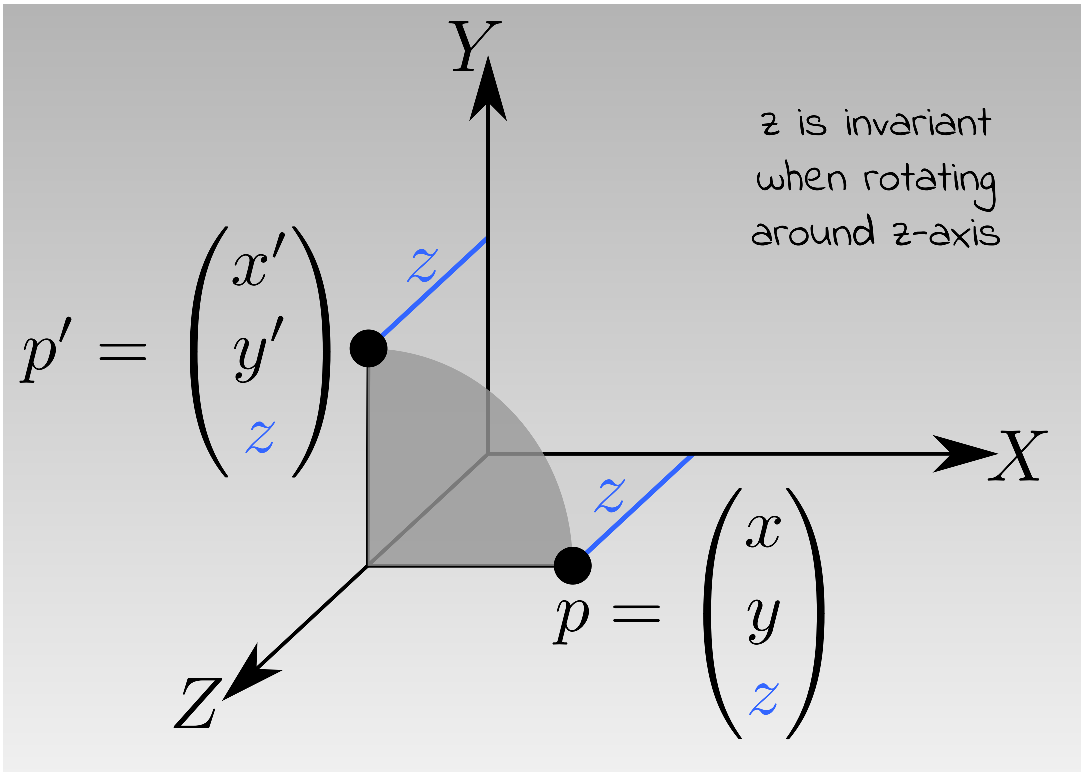

Y
z is invariant
when rotating
around z-axis
$p' = \begin{pmatrix} x' \\ y' \\ z \end{pmatrix}$
z
z
X
$p = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$
Z

Figure 3.19: Rotating a point around an axis. The coordinate of the axis is invariant. The point is just rotated in the plane spanned by the other two axes

it can be observed a point being rotated does not change the coordinate of the axis it is currently rotating around (Figure 3.19).
Considering a rotation around the z-axis, the rotation matrix $R_z$ is similar to the 2D case

$$R_z = \begin{pmatrix} cos(\theta) & -sin(\theta) & 0 & 0 \\ sin(\theta) & cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{3.21}$$

The same principle holds for $R_x$ (and $R_y$ respectively): the column corresponding to the rotation around $x$ (or $y$) is taken from the identity matrix (invariance). The other two columns describe the rotation in the plane spanned by the remaining axes:

$$R_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\theta) & -sin(\theta) & 0 \\ 0 & sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$R_y = \begin{pmatrix} cos(\theta) & 0 & sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -sin(\theta) & 0 & cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The final rotation is given by a rotation around all three axis: $R = R_x \cdot R_y \cdot R_z$.

### 3.5.5 From world space to view space

A scene can be populated with various models by applying different model matrices onto them. One important object is the **camera**. As any other object, it can be placed somewhere into the world using a model matrix. Then, to observe the scene from the cameras point-of-view, the coordinate system of the camera has to become the new origin, i.e. all world coordinates need to be translated into *camera coordinates* (or *view coordinates*). The camera model is designed for convenient use: it consists of two points, namely *position* and *target* (the point the camera is looking at). The (unit) vectors of the cameras coordinate system can directly be derived from these points. They are called *lookat*, *head* and *right*. *lookat* points from *position* into *target* direction:

$$lookat = ||position - target|| \tag{3.22}$$

*head* and *right* could be found using the cross product:

$$\begin{aligned} head &= \frac{lookat \times right}{||lookat \times right||} \\ right &= \frac{head \times lookat}{||head \times lookat||} \end{aligned} \tag{3.23}$$

However, the two equations in 3.23 are not solvable as they depend on each other. The solution to this problem comes from the knowledge that *head* is always parallel to the yz-plane. Choosing an arbitrary vector that fulfills this constraint (e.g. $tmp = (0, 1, 0)$) can be used to compute *right*:

$$right = tmp \times lookat \tag{3.24}$$

Finally *head* is:

$$head = lookat \times right \tag{3.25}$$

Using the cameras coordinate system, the following translation matrix could be used to move the camera into world origin.

$$T = \begin{pmatrix} 1 & 0 & 0 & -position.x \\ 0 & 1 & 0 & -position.y \\ 0 & 0 & 1 & -position.z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.26}$$

Being in world origin, the camera needs to be aligned: *lookat* and $-z$, *head* and $y$, and *right* and $x$ need to be colinear. The corresponding rotation matrix is given by:

$$R = \begin{pmatrix} right.x & right.y & right.z & 0 \\ head.x & head.y & head.z & 0 \\ lookat.x & lookat.y & lookat.z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.27}$$

$R{\cdot}T$ matches the cameras coordinate system with world origin and its inverse $V = (R{\cdot}T)^{-1}$ known as the **view matrix** translates world coordinates into camera coordinates.

### 3.5.6 Projecting the scene on a viewport

Figure 3.20 shows a pinhole camera observing a scene by capturing all objects that are visible inside the *viewing pyramid*. The viewing pyramid is defined by $\alpha$. When used in CG, the pyramid is additionally truncated by the *near plane* $n$ and the *far plane* $f$ resulting in the *viewing frustrum*.

Figure 3.21 shows the **principle of projection**: the frustrum is mapped to a cube. The cube is then mapped onto a 2D plane (screen).

Equation 3.28 shows a simple projection matrix:

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a & -1 \\ 0 & 0 & b & 0 \end{pmatrix} \tag{3.28}$$

Right multiplication of an arbitrary point $p = (x, y, z, 1)$ with $P$ yields:

$$p^T \cdot P = \begin{pmatrix} x & y & z & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a & -1 \\ 0 & 0 & b & 0 \end{pmatrix} = \begin{pmatrix} x & y & az + b & -z \end{pmatrix} = (p')^T \tag{3.29}$$

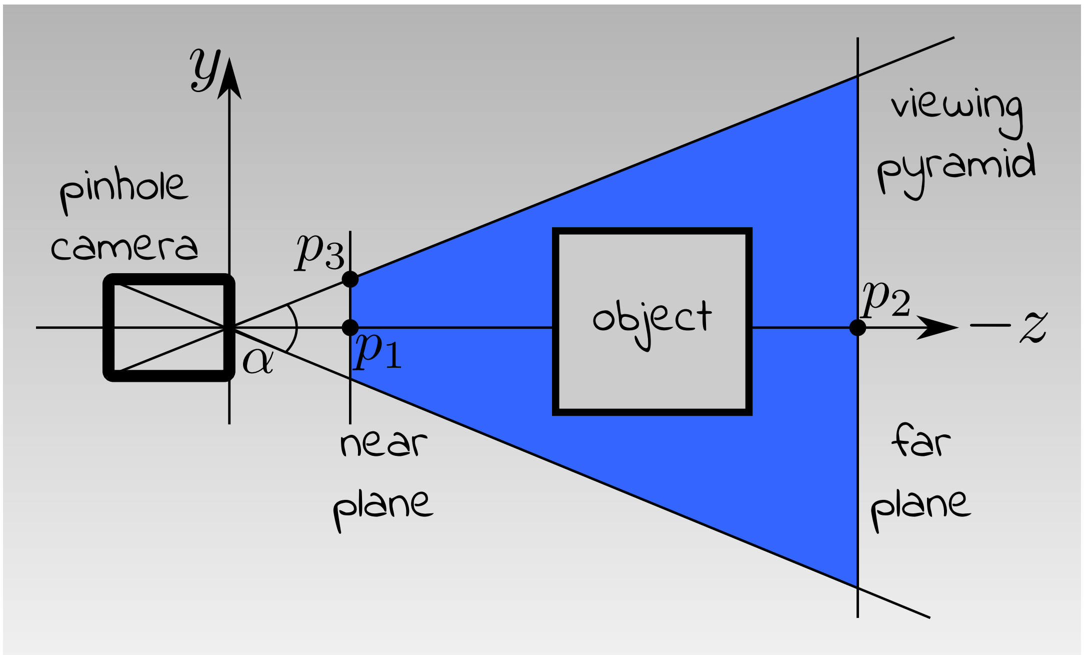


Figure 3.20: Basic Setup of a pinhole camera. Objects inside the viewing frustrum are captured and projected onto screen

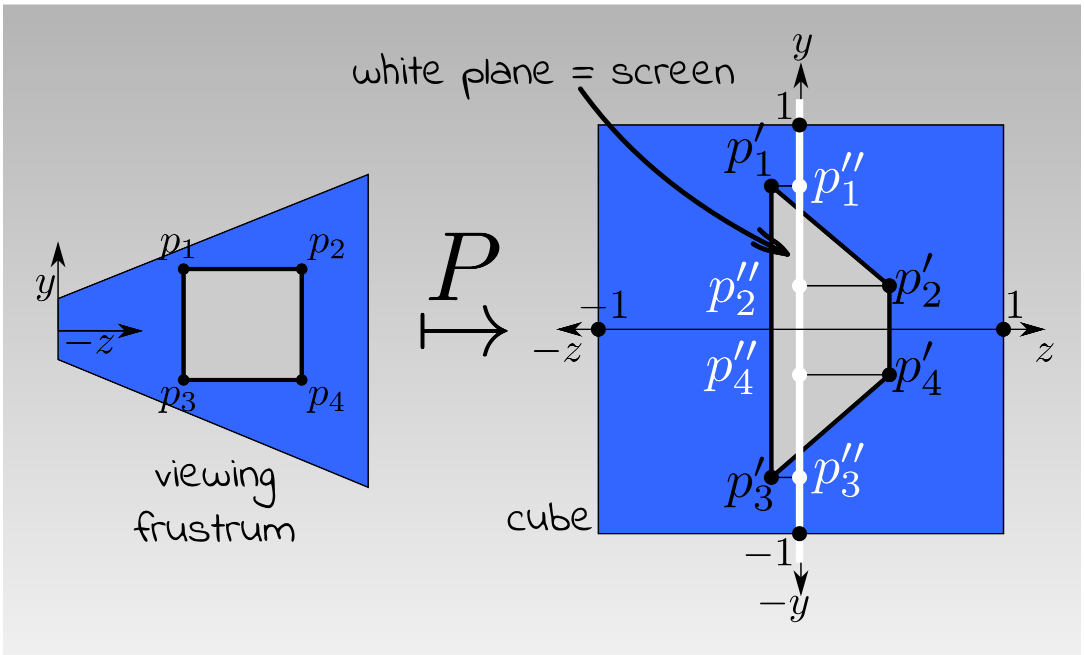


Figure 3.21: The principle of projection. The viewing frustrum is mapped to a unit cube using the projection matrix. Any point inside the viewing frustrum is then mapped using the same matrix. Finally, all points are mapped onto a screen plane

As the camera looks toward $z$-direction, $-z$ is a positive number. The back-transformation of the homogeneous coordinate $(p')^T$ yields $\begin{pmatrix} \frac{x}{-z} & \frac{y}{-z} & \frac{az+b}{-z} \end{pmatrix}$. This reveals perspective: far away objects are divided by a large $z$ becoming *small* while near objects are divided by a small $z$ staying *large*. Reconsidering equation 3.28, $a$ and $b$ need to be found. Two points (one at near plane and one at far plane) are inspected: $p_1 = \begin{pmatrix} 0 & 0 & -n & 1 \end{pmatrix}$ and $p_2 = \begin{pmatrix} 0 & 0 & -f & 1 \end{pmatrix}$. Multiplying them with $P$ yields:

$$\begin{aligned} p_1^T \cdot P = p_1' &= \begin{pmatrix} 0 & 0 & -an+b & n \end{pmatrix}^T \\ p_2^T \cdot P = p_2' &= \begin{pmatrix} 0 & 0 & -af+b & f \end{pmatrix}^T \end{aligned} \tag{3.30}$$

Back-transforming both points to euclidean coordinates yields:

$$\begin{aligned} p_1' &= \begin{pmatrix} 0 & 0 & \frac{-an+b}{n} \end{pmatrix}^T \\ p_2' &= \begin{pmatrix} 0 & 0 & \frac{-af+b}{f} \end{pmatrix}^T \end{aligned} \tag{3.31}$$

As shown in Figure 3.21, $p_1'$ has to map to $\begin{pmatrix} 0 & 0 & 1 \end{pmatrix}$ and $p_2'$ has to map to $\begin{pmatrix} 0 & 0 & -1 \end{pmatrix}$. This leads to:

$$-an + b = n \tag{3.32}$$

$$af - b = f \tag{3.33}$$

Subtracting 3.33 from 3.32 yields:

$$-a(n - f) = n + f \tag{3.34}$$

3.34 is now solved for $a$:

$$a = \frac{f+n}{f-n} \tag{3.35}$$

Substitution of $a$ in 3.32 yields:

$$\left(\frac{f+n}{f-n}\right) n + b = n \tag{3.36}$$

Now equation 3.36 is solved for $b$ and the result is simplified:

$$b = n + \left(\frac{f+n}{f-n}\right) n = n + \left(1 + \frac{f+n}{f-n}\right) = n + \left(\frac{f-n}{f-n} + \frac{f+n}{f-n}\right) = \frac{2nf}{f-n} \tag{3.37}$$

Finally, $P$ is:

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{f+n}{f-n} & -1 \\ 0 & 0 & \frac{2nf}{f-n} & 0 \end{pmatrix} \tag{3.38}$$

To derive the mapping of the $y$ axis, $p_3 = \begin{pmatrix} 0 & n \cdot tan(\frac{\alpha}{2}) & -n \end{pmatrix}$ is considered:

$$p_3 \cdot P = p_3' = \begin{pmatrix} 0 & n \cdot tan(\frac{\alpha}{2}) & \frac{f+n}{f-n}(-n) + \frac{2fn}{f-n} & n \end{pmatrix} \tag{3.39}$$

Back-transformation of 3.39 yields:

$$p_3' = \begin{pmatrix} 0 & tan(\frac{\alpha}{2}) & 1 \end{pmatrix} \tag{3.40}$$

$p_3'$ has to map to $\begin{pmatrix} 0 & 1 & 1 \end{pmatrix}$.
To derive the mapping of the $x$ axis, the $y$ axis procedure is repeated. The final projection matrix is given by:

$$P = \begin{pmatrix} cot(\frac{\alpha}{2}) & 0 & 0 & 0 \\ 0 & cot(\frac{\alpha}{2}) & 0 & 0 \\ 0 & 0 & \frac{f+n}{f-n} & -1 \\ 0 & 0 & \frac{2nf}{f-n} & 0 \end{pmatrix} \tag{3.41}$$

$P$ maps all objects inside the view frustrum into the unit cube. The cube is projected onto the plane by simple dropping the $z$-axis (not considering objects being partially outside the frustrum or occluded by other objects).

### 3.5.7 Extending MVP to fit VR

To take a model, place it somewhere into a scene, observe it from a cameras point-of-view and finally project it to screen, the matrices M, V, and P are used. However, Virtual Reality requires another consideration: the encoding of depth information. The human visual system **reconstructs depth information by viewing a scene from two points (eyes) being displaced from each other**. Figure 3.22 shows $P$ seen from two cameras described by their centers of projection ($O_l$ and $O_r$) their image planes ($\Pi_l$ and $\Pi_r$) and the direction they look into ($d$). The image points $p_l$ and $p_r$ are the intersections of the rays $\overline{PO_l}$ and $\overline{PO_r}$ with $\Pi_l$ and $\Pi_r$.
The similarity of $\triangle_{PO_lO_r}$ and $\triangle_{Pp_lp_r}$ allows:

$$\frac{T + x_l - x_r}{Z - f} = \frac{T}{Z} \tag{3.42}$$

Solving 3.42 for $Z$ yields:

$$\begin{aligned} & \frac{T + x_l - x_r}{Z - f} = \frac{T}{Z} \\ \Leftrightarrow \quad & T + x_l - x_r = \frac{T(Z-f)}{Z} = \frac{TZ - Tf}{Z} = T - \frac{Tf}{Z} \\ \Leftrightarrow \quad & x_l - x_r = -\frac{Tf}{Z} \\ \Leftrightarrow \quad & Z = -f\frac{T}{x_l - x_r} = f\frac{T}{x_r - x_l} \end{aligned} \tag{3.43}$$

where $x_r - x_l$ is called *disparity*. Equation 3.43 shows: the displacement of a projected point is indirectly proportional to the distance it's 3D point. Virtual Reality exploits

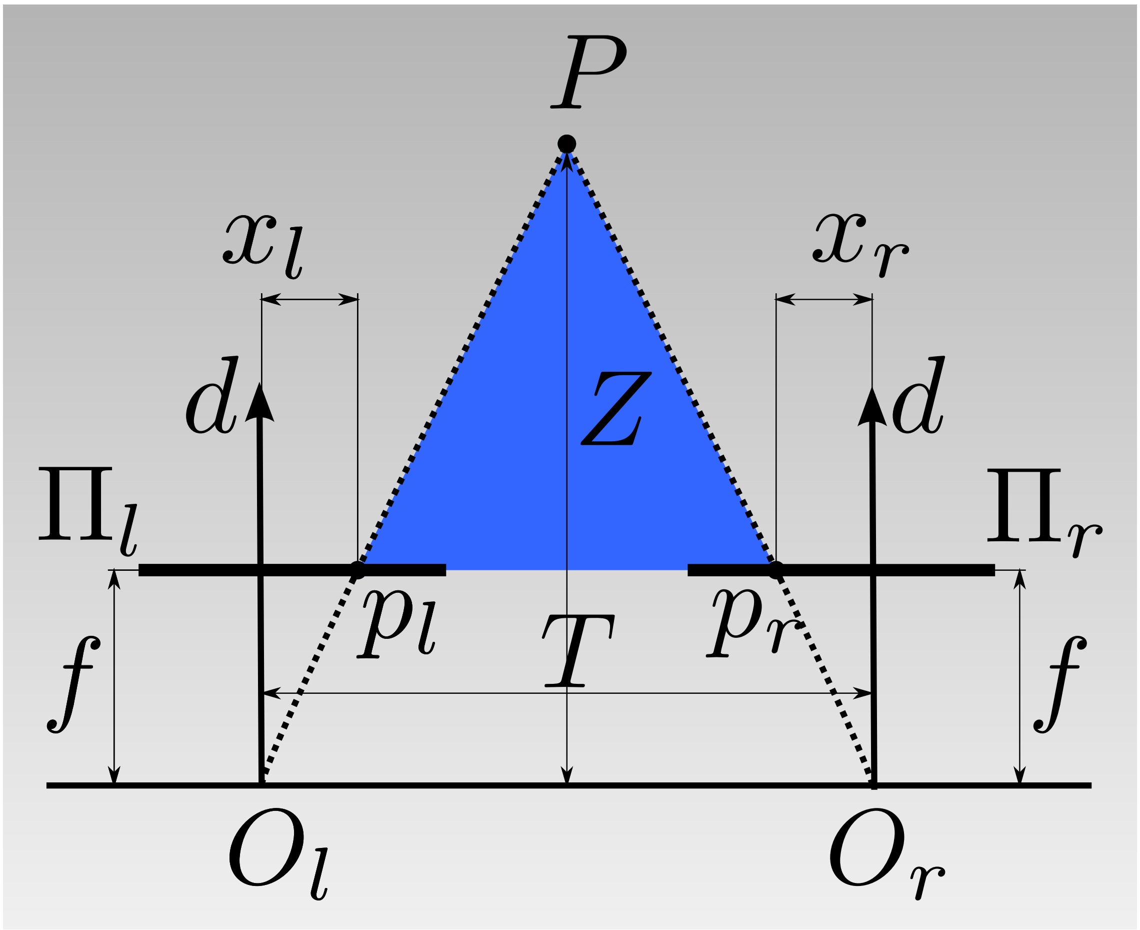


Figure 3.22: View a scene from two different points allows depth reconstruction.

that by rendering two images, one in front of each eye. In a common setup both displays show the same image and the images are slightly displaced from each other. Displacement is encoded using an *eye matrix* which is applied onto a scene after being transformed into camera coordinates. Therefore, to load a model into Virtual Reality, the so-called Model-View-Eye-Projection matrix $MVEP$ is used.

## 3.6 Centerline computation

The centerline of a volume is a line through the center of that volume [154]. Given such a volume, the automatic extraction of the centerline is a non-trivial task that is still under active research [155]. Different proposals have been made to solve that problem in medical or other applications [155, 156, 157, 158, 159, 160]. In their work on a clustering-based algorithm, Ferchinie and Wang [156] distinguish four classes of algorithms: topological thinning of an object (i), distance transform (ii), the "prairie fire" approach (iii), and methods based on Voronoi Diagrams (iv).

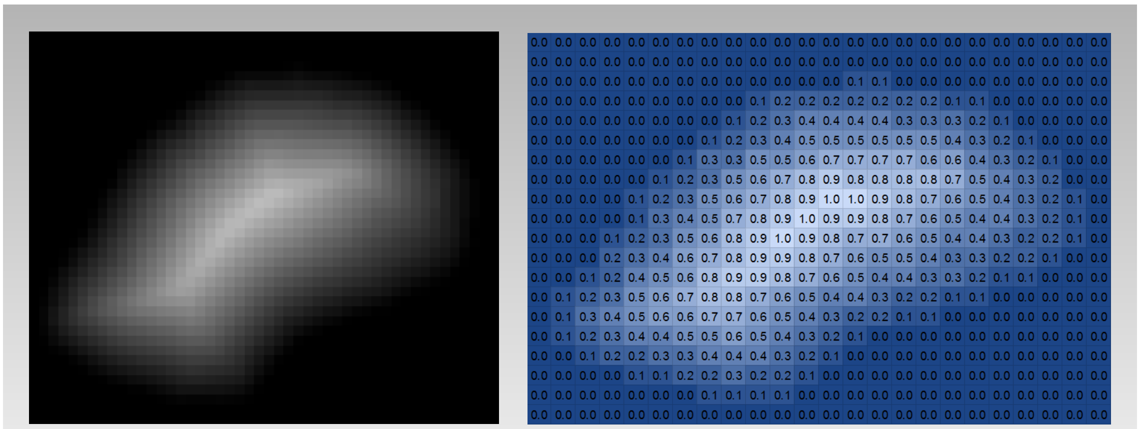
An algorithm using state-of-the-art concepts for centerline computation has been proposed by **Egger et al.** in 2007 [161]. Even though their algorithm could be used in general, it has been designed for the specific case of centerline computation in blood vessels. The algorithm requires minimal user interaction, namely the definition of a startpoint and an endpoint. Initially, it connects startpoint and endpoint - depending on the situation - either by the shortest path algorithm of Dijkstra [162] or by a straight line. After that, it aligns this path to the blood vessels using an active contour model (ACM) with polyhedra placed along it [161]. The ACM was initially described by Kass et al. [121] and aligns the centerline by minimizing the following energy functional:

$$E = \int_{s=0}^{1} E_{int}(v(s)) + E_{ext}(v(s))ds$$

where the smoothing is driven by the internal forces $E_{int}$ while the external forces $E_{ext}$ depend on the underlying image and ensure that the line sticks to the center of the vessel. Even though the user-defined endpoints are fixed and therefore excluded from the smoothing term of the energy, they have an impact on neighboring points. In their proposal, Egger et al. have applied the described algorithm to different blood vessels such as the iliac artieries or the aorta. They consider their solution to be fast and robust [161].

When less accuracy is needed, the algorithm can simplified in such a way that $E_{int}$ is set to zero. In this scenario, smoothing is negelected and the solution depends on the underlying image only. A simple centerline computation based on $E_{ext}$ only can be performed by using distance transform.

The distance transform is a 2D map where each position holds a value indicating a measured distance from the segments boundary to the segments center. To compute the distance between two points, different measures, such as euclidean distance, can be applied. Figure 3.23 shows the principle of a distance map given a single cross section of the



| | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.1 | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.4 | 0.4 | 0.4 | 0.3 | 0.3 | 0.3 | 0.2 | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.4 | 0.3 | 0.2 | 0.1 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 0.3 | 0.3 | 0.5 | 0.5 | 0.6 | 0.7 | 0.7 | 0.7 | 0.7 | 0.6 | 0.6 | 0.4 | 0.3 | 0.2 | 0.1 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 0.2 | 0.3 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 0.8 | 0.8 | 0.8 | 0.8 | 0.7 | 0.5 | 0.4 | 0.3 | 0.2 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 0.2 | 0.3 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 | 1.0 | 0.9 | 0.8 | 0.7 | 0.6 | 0.5 | 0.4 | 0.3 | 0.2 | 0.1 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 0.3 | 0.4 | 0.5 | 0.7 | 0.8 | 0.9 | 1.0 | 0.9 | 0.9 | 0.8 | 0.7 | 0.6 | 0.5 | 0.4 | 0.4 | 0.3 | 0.2 | 0.1 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.1 | 0.2 | 0.3 | 0.5 | 0.6 | 0.8 | 0.9 | 1.0 | 0.9 | 0.8 | 0.7 | 0.7 | 0.6 | 0.5 | 0.4 | 0.4 | 0.3 | 0.2 | 0.2 | 0.1 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.2 | 0.3 | 0.4 | 0.6 | 0.7 | 0.8 | 0.9 | 0.9 | 0.8 | 0.7 | 0.6 | 0.5 | 0.5 | 0.4 | 0.3 | 0.3 | 0.2 | 0.2 | 0.1 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.1 | 0.2 | 0.4 | 0.5 | 0.6 | 0.8 | 0.9 | 0.9 | 0.8 | 0.7 | 0.6 | 0.5 | 0.4 | 0.4 | 0.3 | 0.3 | 0.2 | 0.1 | 0.1 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.1 | 0.2 | 0.3 | 0.5 | 0.6 | 0.7 | 0.8 | 0.8 | 0.7 | 0.6 | 0.5 | 0.4 | 0.4 | 0.3 | 0.2 | 0.2 | 0.1 | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.1 | 0.3 | 0.4 | 0.5 | 0.6 | 0.6 | 0.7 | 0.7 | 0.6 | 0.5 | 0.4 | 0.3 | 0.2 | 0.2 | 0.1 | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.4 | 0.5 | 0.5 | 0.6 | 0.5 | 0.4 | 0.3 | 0.2 | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.1 | 0.2 | 0.2 | 0.3 | 0.3 | 0.4 | 0.4 | 0.4 | 0.3 | 0.2 | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 0.1 | 0.2 | 0.2 | 0.3 | 0.2 | 0.2 | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 0.1 | 0.1 | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

(a) a visualized distance map

(b) pixel intensities relate to distance

Figure 3.23: A distance map computed the distance of each pixel from it's closest border. It can be visualized by representing numbers by intensity values.

aorta: the visualized distance map (a) is a graphical representation of how distance values are represented by color intensity (b).
It can be interpreted as follows: the brighter a pixel is, the further away it is from the segments boundary and thus the closer it is to the segments center.
Given a distance transform, a suitable threshold can be applied on the map. After that, the maximum value of the distance transform corresponds to the center of the segment. Iterating over a set of slices therefore results in a non-smoothed centerline. When the final result does not need to be accurate in terms of anatomy, smoothing the given centerline yields an adequate result.

## 3.7 Methods of evaluation

By providing an immersive visual experience, VR technologies aim to yield the perception of being physically present in a world that does not exist. However, the impression of *being there* is subjective and cannot be measured easily. As such, evaluating VR applications is not an easy task and still a topic of active research. Elaborate strategies have to be used to quantify results.

### 3.7.1 Evaluating computer games

Considering different types of software, a VR application of any kind can best be compared with a computer game, especially when it comes to **measuring the quality** of the application. However, the quality of a computer game can **hardly be quantified** as it highly depends on user's preference and experience. Among others, this problem has been approached by Ramadan and Hendradjaya in 2014 [163]. They introduce their work by stating that measuring the quality of a game is mostly done in a user-centric way while other types of software often underlie quality models with a higher grade of objectiveness. They mention a first framework for the measurement of software quality introduced by Cavano and McCall in 1977 [164]. In addition, they name two modern models based on this framework [165, 166]. They further reference Fullerton stating different qualities that should be present in a game [167] and several others suggesting methods of game testing [168, 169]. They focus on these publications to derive a quality model for computer games. Finally, they conclude that the quality of a game consists of four factors: **user experience** (i) with subfactors of fun (a), balance (b) and usability (c), **functionality** (ii) with subfactors of feature richness (a), availability (b), performance (c), internal completeness (d), and service compatibility (e), **maintainability** (iii), and **portability** (iv).

### 3.7.2 Evaulating VR applications

Considering medical applications, some game-relevant factors, such as fun or balance, i.e. the games difficulty, can be neglected. Regarding Virtual Reality, the **emphasis of evaluation is often put on user experience** as this is what distinguishes Virtual Reality from conventional computer games considering a users point-of-view. Various studies [170, 171, 172] have done extensive research on technologically mediated experiences in general or Virtual Reality in particular. Barfield and Furness summarize some of these studies and list different characteristics of virtual environment experience [173]. One way a VR experience can be described is by evaluating the following factors: *immersion*, *interactivity*, and *intuition*. Lavroff extends that list by adding *manipulation* and *navigation* [170]. Zeltzer places any VR application into what he calls the *AIP cube* consisting of three axis: *autonomy*, *interaction* and *presence* [172]. Waterworth and Waterworth propose a model taking *focus*, *locus* and *sensus* into account [174]. Bowman and Hodges put their focus on the interaction with virtual environments [175].

### 3.7.3 Immersion, Presence and performance

When comparing different studies on how to evaluate VR applications, **immersion and presence** are the factors often being considered. However, Bowman and MacMahan note that studies concerning Virtual Reality often confuse these two concepts [176]. A strong distinction between the two terms is given by Slater as he considers *presence* to be the user's subjective psychological response to a VR system and *immersion* to be the objective level of sensory fidelity a VR system provides [177]. Presence is based on research done by Gregory [178] and Stark et al. [179], it can further be subclassified [180] and is subject to various studies [181, 182, 183, 184, 185, 186, 187] that try to measure it in virtual environments using different methods such as questionnaires [185, 188, 189] or the collecting of physiological measurements of users being abandoned in Virtual Reality [190]. Similar approaches exist to quantify immersion [191]. However, by building on the definition provided by Slater, immersion can be related to technical quantities. Even though the impact of immersion on presence cannot be stated quantitatively, it apparently has a high influence.

Therefore, by reconsidering the four factors of game quality given by [163], user experience is highly dependant on the applications functionality. In particular, the **performance** of the application makes a strong contribution to how a user experiences Virtual Reality. As a matter of fact, **evaluating the grade of immersion by measuring the systems performance is a huge step towards quality evaluation** of VR applications. However, to simplify the task of evaluating immersion, the overall level of immersion can be reduced to visual immersion when considering Virtual Reality applications. Bowman and MacMahan point out that visual immersion is related to how close the systems visual output is to real-world visual stimuli [176]. They further state that visual immersion depends on many factors. According to them, these include: field of view, field of regard, display size, display resolution, stereoscopy, head-based rendering, realism of lighting, frame rate and refresh rate. Some of these factors are an essential part of any VR system, namely stereoscopy and head-based rendering. Other factors such as field of view, field of regard, display size, display resolution, frame rate, and refresh rate are defined by the specific hardware being used. Finally, the realism of lighting or the overall realism of the scene itself is determined by application development.

### 3.7.4 Performance measures in VR

Regarding conventional computer games, performance is often related to frame rate measured in frames per second (fps). However, frame rate just counts the number of frames being generated in a second without taking into account that one frame might have taken longer to compute than another one. Therefore, high average frame rates could serve as a vague indication but they do not necessarily guarantee a jitter free, smooth user experience. **Frame time**, i.e. the time it takes for a frame to be computed, is another measure that can be used instead of frame rates.

Even though frame time is more reliable when evaluating the performance of conventional computer games, further considerations have to be made when it comes to the performance of VR applications. While conventional computer games can directly render a computed frame to the screen, VR applications have an intermediate step: a computed frame is not directly rendered on the HMD but rather passed to a runtime. In a simple scenario, the runtime basically applies some lens correction to the rendered frame before it is passed on to the HMD. However, there might be cases when the time to compute a frame exceeds the refresh interval. Therefore, the new frame is not available when needed. This is often referred to as a *dropped frame*, *app miss* or *app drop*. Simply redisplaying the last frame is unacceptable in VR as ignoring the head movement would cause a bad user experience. As a consequence, the job of the runtime becomes significantly more complex when dropped frames are encountered: it needs to generate a so-called *synthesized frame*. To do so, it uses a technique called Asynchronous Spacewarp (ASW). ASW estimates a new frame based on motion vector analysis of prior rendered frames.

# 4 Methods

Visualizing a dissected aorta in Virtual Reality comprises several steps. Figure 4.1 shows the basic workflow of the visualization process.
A high-level point-of-view might distinguish three steps in that workflow: image acquisition and storage, the conversion of the digital image into a 3D scene that can be experienced in Virtual Reality, and the extension of Virtual Reality to interactive Virtual Reality. However, this chapter presents a more-in-detail survey on different methods used for implementing the visualization framework.

There are many programs that fulfull certain tasks such as segmentation, model generation, model refinement and rendering. These **software** is introduced in section 4.1.

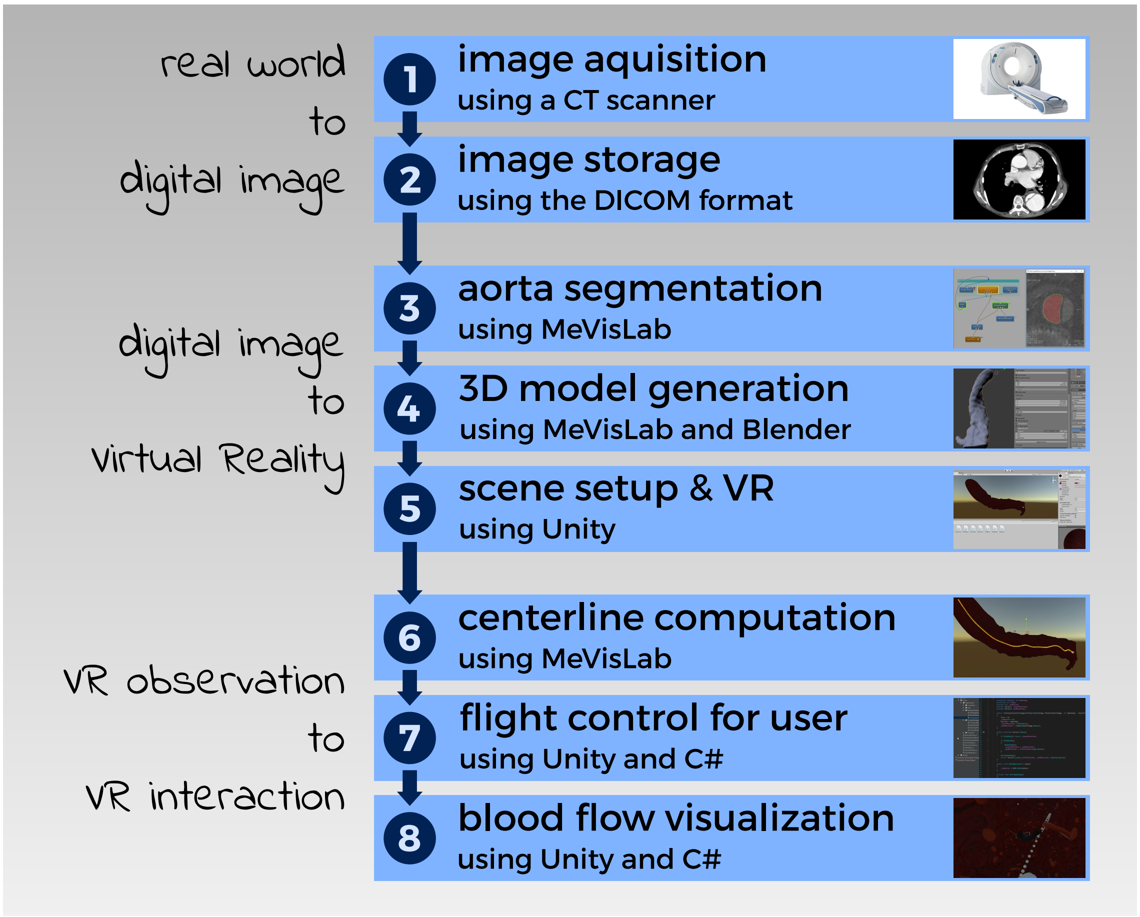
The process of visualization starts with an input file. Different **methods for retrieving a medical image** have already been discussed, but section 4.2 argues why a single file downloaded from the internet has currently been the only file used for this project.

Given the input file, section 4.3 discusses **how the aorta has been segmented** in the special case of this project. It introduces a MeVisLab network that has specifically been designed for that case.

Section 4.4 deals with different methods chosen for **modelling and rendering**. First, a MeVisLab network for generating a 3D model is introduced. Given the resulting model, the steps required for model refinement are listed and discussed. Finally, the model is prepared for being rendered in VR as it is described at the end of that section.

A VR application without the ability of interaction would not be sufficient for an immersive user experience. Therefore, section 4.5 covers one method used for implementing a simple flight control: the **centerline extraction**.

Finally, section 4.6 and section 4.7 describe some implementation details regarding the modelling of **blood flow** and **flight control**.



real world to digital image

1 image aquisition using a CT scanner

2 image storage using the DICOM format

digital image to virtual Reality

3 aorta segmentation using MeVisLab

4 3D model generation using MeVisLab and Blender

5 scene setup & VR using Unity

VR observation to VR interaction

6 centerline computation using MeVisLab

7 flight control for user using Unity and C#

8 blood flow visualization using Unity and C#

Figure 4.1: Workflow of visualizing the aortic dissection in interactive Virtual Reality

## 4.1 Used Software

Given a DICOM file serving as an input to the visualization pipeline, different tools need to be used to end up with a final visualization. This section describes them.

**MeVisLab** [95] stands for *Medical Visualization Laboratory.* It is a powerful modular framework for image processing and it has a special focus on medical imaging. It provides different levels of development. At a high level, a developer can implement so-called *networks*. These networks consist of modules that can visually be connected to gain new functionality. A more sophisticated way of development is using python. Finally, MeVisLab allows to implement own modules in C++. A network can be seen as a piece of software fulfilling some tasks. As such, the network is the program which is delivered to and used by an end-user. In this work, MeVisLab has been used for segmentation, the generation of the 3D model and the computation of the centerline.

**Blender** [192] is a free and open source 3D computer graphics software. It is widely used for modelling, texturing and animating 3D objects and scenes. It is written in C++ and functionality can be extended using python. It's GUI is highly configurable and automatically adapts to different tasks within a project. It is shipped with a vast amount of features and it provides different rendering engines for finalizing a project. When a 3D model is loaded into Blender, so-called *modifiers* can be applied on that model. Depending on the chosen modifier, different operations are performed on the model, e.g. smoothing or application of wall thickness. The concept of modifiers has a huge advantage: the original model is kept and modifiers can easily be activated or deactivated at any stage of design. Blender also allows the generation of texture maps, i.e. fitting a bitmap on the mesh of a model.

The GNU Image Manipulation Program (**GIMP**) [193] has been used for generating the bitmaps that have been used as texture. GIMP is a cross-platform, free image graphics editor which is often used for image retouching or editing. It allows an image to consist of various layers and provides different blending modes for these layers. It is shipped with different colour palettes, provides different tools for image manipulation and allows automated image processing using scripts that can be implemented in different programming languages, python being among them. One of GIMP's features is that it comes with different patterns an area can be filled with. In the process of visualizing the dissected aorta, this feature has been used for texture generation.

After the model has been generated and textured, it has to be visualized in Virtual Reality. For that purpose, **Unity** [194] has been used. Unity is a cross-platform game engine. It has been developed by David Helgason, Joachim Ante and Nicholas Francis and a first version has been launched on June 6, 2005 [195]. The goal of the project was to democratize the game development [196] by creating a game engine that is both, affordable and professional. As per today, Unity is an extremely popular, widespread product. They have around 770 million gamers all over the world playing games made by using the unity engine [197]. Creating a Unity scene comprises the import of 3D models, light and material setup, and the implementation of behaviour. After creation, a scene can be exported to different platforms. Unity supports more than 20 of these platforms, many popular Virtual Reality and Augmented Reality systems being among them.

## 4.2 Image retrieval and file format

The implemented framework aims to visualize aortic dissections. Given that goal, any digital medical 3D image could serve as an input for the framework. However, as discussed in section 2.9, CT scans are the most used modality in AD diagnosis. They are usually stored using the DICOM standard. When used in clinical routine, these images are supposed to be provided by medical doctors on-site. For this work, however, the *TRAGICOMIX* dataset provided by *OsiriX* has been used [198]. *OsiriX* is a software for inspecting medical images developed by pixmeo. In addition to their software, they have an online image library providing various data sets impersonated and publicly available. All their datasets are stored using DICOM standard and they are intended for research and teaching. *TRAGICOMIX* is one of these data sets. It is a thoracic CT scan of a patient with an aortic dissection status post surgical repair of the ascending aorta. After being downloaded, the dataset has further been preprocessed using MeVisLab and Blender.

## 4.3 Image segmentation

Various methods for image segmentation have already been discussed in Section 3.3. All of them have different advantages and disadvantages and there is no ”gold standard” that clearly outperforms the other methods. Current methods often require manual refinement, especially when it comes to the field of medical imaging, segmentation of vessels in particular. As manual refinement is a necessary post-processing step of segmentation and there has only been a single data set being used for the development of the visualization prototype, the given data set has been segmented manually in this work.

For that purpose, a MeVisLab network has been set up. It allows a DICOM file to be loaded in a first step. Then, it provides an interface allowing a user to step through all the slices of the scan. In each slice, a user can manually select the area of interest. The network supports the user by smoothing the freehand selection. In addition, the selection is closed automatically when the contour line gets close to the starting point of the selection.

The result of using this network is a so-called *CSO* (contour segmented object) file. It contains a list of curves describing the segmentation. While being a tedious work, manual segmentation also requires an expert's knowledge, especially when it comes to vessel segmentation: the fine structures of the vessels are sometimes hard to distinguish from surrounding tissue, not only for algorithms but for humans as well. However, as the current segmentation has served for testing purposes only, it has been performed by a computer scientist. As a matter of fact, only clearly distinguishable parts of the aorta have been segmented. In addition, the segmentation was not performed on each slide. Instead, each

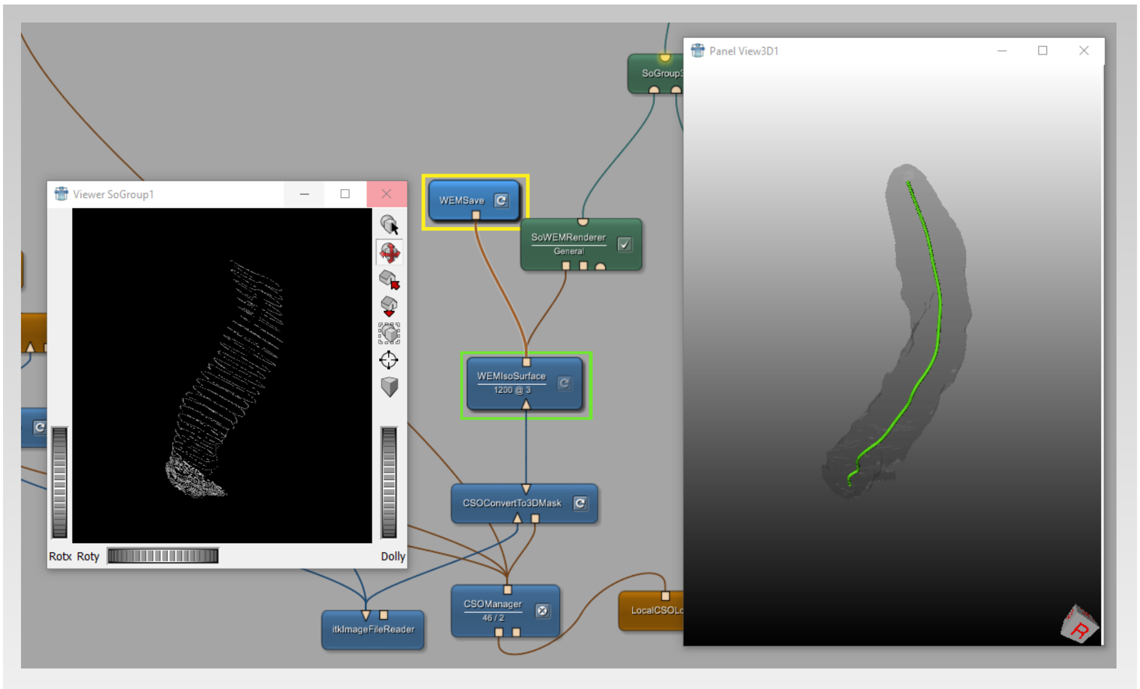


Figure 4.2: Using a MeVisLab network, the segmented aorta is converted into an iso surface.

tenth slice has been segmented. When the model has been generated in a later step, missing information has been interpolated. This decision has been met based on the fact that an inaccurate model is sufficient for implementation and testing of the visualization.

## 4.4 Modelling and rendering

### 4.4.1 3D model generation

Different ways to digitally store 3D models have been covered in section 3.4.1. As already discussed, a surface model is the best way to represent ”natural” objects such as the aorta. The surface model is stored using a triangle mesh. The generation of that mesh is performed using a **MeVisLab network** that uses the formerly generated CSO file. Figure 4.2 shows the network. The visualization on the left side of the image shows the segmentation curves. The right side shows the rendered model with a centerline inside. The network works by generating a 3D mask using all segmented curves. It enhances the mask by applying an interpolation strategy on the segmentation data. This strategy is described in [199].

After the mask is generated, it is converted into an IsoSurface digitally represented using a *WEM* data structure. The generated IsoSurface can finally be exported using the **STL** (abbrev. ”stereolithography”) file format. In this format, a triangulated surface is described by unit normals and vertices. Figure 4.3 shows **the exported STL** file. The red color is not part of the file information and has been added additionally to improve visibility.

The aorta, as it is exported by MeVisLab, is apparently not suited for an immersive visual experience in Virtual Reality. Beside the fact that is does not look realistic, it has some technical issues that need to be fixed before it can be rendered. For that purpose, it needs to be refined.

### 4.4.2 3D model refinement

**Model refinement** has been done using Blender. The following steps were made in Blender to improve the 3D model: normals have been inverted (i), a wall thickness has been defined (ii), smoothness has been increased (iii), and texture has been applied (iv).

The **inversion of the normals** is more than a visual enhancement: it is required for a proper rendering of the aorta. The reason for this is that the model generated by MeVisLab is a surface model. 3D rendering engines render such objects using so-called shaders, namely small programms compiled on the GPU. Shaders render surfaces by considering the normals of the faces. Therefore, considering a common shader, a hollow object modelled by a surface model can either be seen from the inside or from the outside. The rendered perspective depends on the direction the face normals point to. When a model has normals pointing outside of the object, the inside perspective is not rendered correctly and vice-versa. Usually, 3D modelled objects are seen from the outside. This is also true for the model of the aorta: MeVisLab generates it to be used for an outside view. However, the application of this work requires an inside view of the aorta. Therefore, blender has been used to invert all normals pointing into the inside of the object.

Surface models describe the hull of 3D objects. As such, they have **no real walls**. When considering objects being seen from the outside, this is usually not a problem. However, placing the user inside the object having no real walls caused the rendered scene to be inconsistent in some areas. To overcome this problem, a *Solidify* modifier has been applied to the model. Figure 4.4 shows the settings that were used to apply wall thickness to the wall of the aorta.

To **increase smoothness**, a modifier called *Subsurf* has been applied. It uses a method called *subdivision surface* taking the coarse mesh as an input refining it by subdividing it's faces into smaller faces resulting in a smoother overall surface. The algorithm used here is called *Catmull-Clark* and named after it's inventors [200]. It basically adds a *face point* and *edge points* to each face of the original surface. Then, it connects the face point with each corresponding edge point. Finally, it adapts the original points to fit together with the newly created points. By doing so, a smoother surface is obtained. To increase smoothness, the process can be repeated as often as desired. However, higher smoothness means more vertices and egdes and thus an increased memory consumption. In this work, the model of the aorta has been smoothed by repeating subdivision surface two times. This has increased the number of vertices by more than 20x. Table 4.1 shows blender

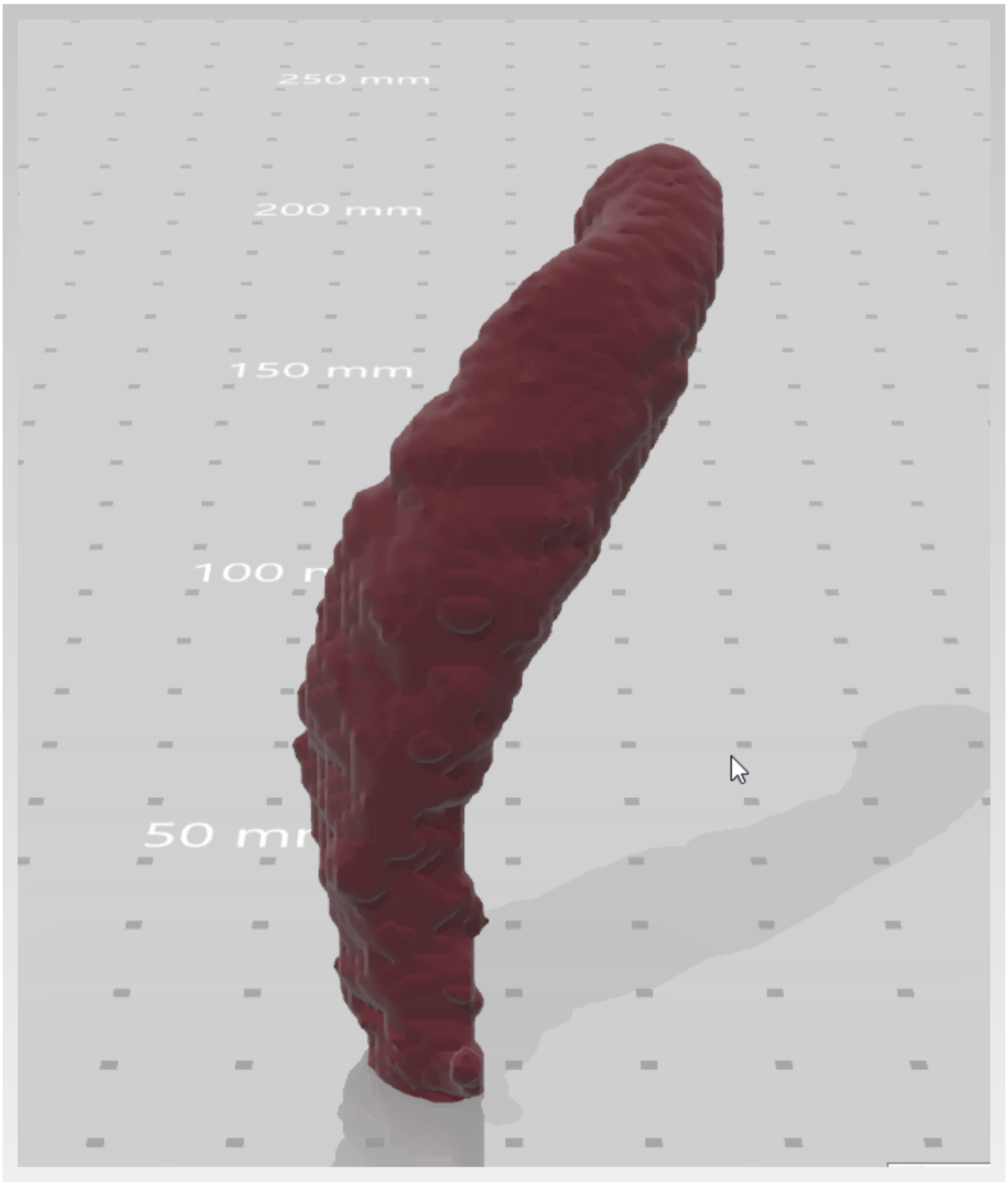
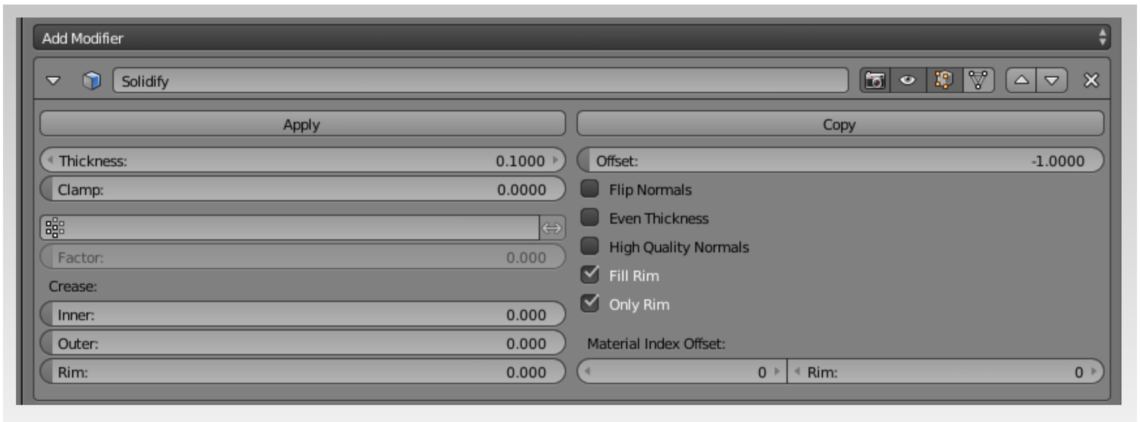


Figure 4.3: The segmented 3D aorta. The red color is not part of the STL file and is added for better visibility.



Thickness: 0.1000
Clamp: 0.0000
Factor: 0.000
Crease:
Inner: 0.000
Outer: 0.000
Rim: 0.000
Offset: -1.0000
□ Flip Normals
□ Even Thickness
□ High Quality Normals
☑ Fill Rim
☑ Only Rim
Material Index Offset: 0 Rim: 0

Figure 4.4: Using Blender's *Solidify* modifier, wall thickness has been applied to the model of the aorta.

| | **original** | **Catmull-Clark x 1** | **Catmull-Clark x 2** |
|---|---|---|---|
| **Vertices** | 10,551 | 63,286 | 253,132 |
| **Faces** | 21,094 | 63,282 | 253,128 |
| **Mem** | 36.29 M | 43.37 M | 82.90 M |

Table 4.1: Repeateadly running Catmull-Clark on the original model of the aorta rapidly increases the number of vertices and faces of the model.

statistics regarding the increase of vertices and faces when applying Catmull-Clark in detail. In addition to Catmull-Clark, the smoothness of the model has further been increased using a *Laplacian Smooth* modifier. According to Blenders documentation, this modifier is well-suited for objects that have been reconstructed from the real world [201]. It removes noise while preserving the original model.

In a final step, the refined model needs to be equipped with a **texture**. A texture is an image that is mapped onto a 3D model. It provides the model with a realistic looking surface. Textures are often available at various online resources. Some famous texture resources have been surveyed by Hossain and Serikawa in 2013 [202]. Even though these resources provide a vast amount of different textures, obtaining a texture for the inside of the aortic wall is not a common requirement in computer graphics. Thus, the texture used for this work had to be generated artificially using the *dried mud* pattern provided by GIMP. A red overlay had been applied to make the pattern more vessel-like. Considering the histology of the aorta, the resulting pattern is a good approximation for the flat, polygon-shaped cells of the endothelium.

### 4.4.3 Rendering the 3D model

To **render** the 3D model, it has to be placed in a 3D scene that is later rendered to VR. For this purpose, Unity has been used. The model has been imported into Unity's assets. It has not been transformed in any way as it was already properly scaled. The mesh renderer that is applied to any object being loaded into Unity has been left with it's default settings. Based on the formerly generated Unity texture, **a material has been created** and applied to the model of the aorta. The overall scene has been illuminated using a **directional light** being properly positioned in the scene. In addition, the **global light settings** (figure 4.5) have been left to default. Finally, the user is equipped with a **torch** allowing to illuminate certain areas of the aortic wall. For that torch, a spotlight has been chosen. It's range has been set to 150 meters and the spot angle is at 30 degrees. In order to render the scene to VR, Unity uses the SteamVR plugin which is shipped with a **predefined camera rig** consisting of the camera view that is later rendered to the head mounted display as well as prefabs for left and right controller. It has been used instead of the default main camera loaded when a new Unity scene is created. All camera settings have been left as the were predefined by the plugin.

## 4.5 Centerline computation

There are various ways how **flying through the aorta** could be implemented. This work uses a simple approach: a user placed inside the aorta is supposed to fly along a predefined path, namely the centerline. Even though the implementation of Virtual Reality interaction by binding the players flight to a predefined centerline is a simple approach from a players point-of-view, the automatic extraction of the centerline is a non-trivial task that is still under active research [155]. Different methods for the automatic computation of the centerline have already been discussed in section 3.6. As this work simply uses the centerline as a path for the player to fly along, the exactness of results is not as important as in other applications such as catheter simulation.

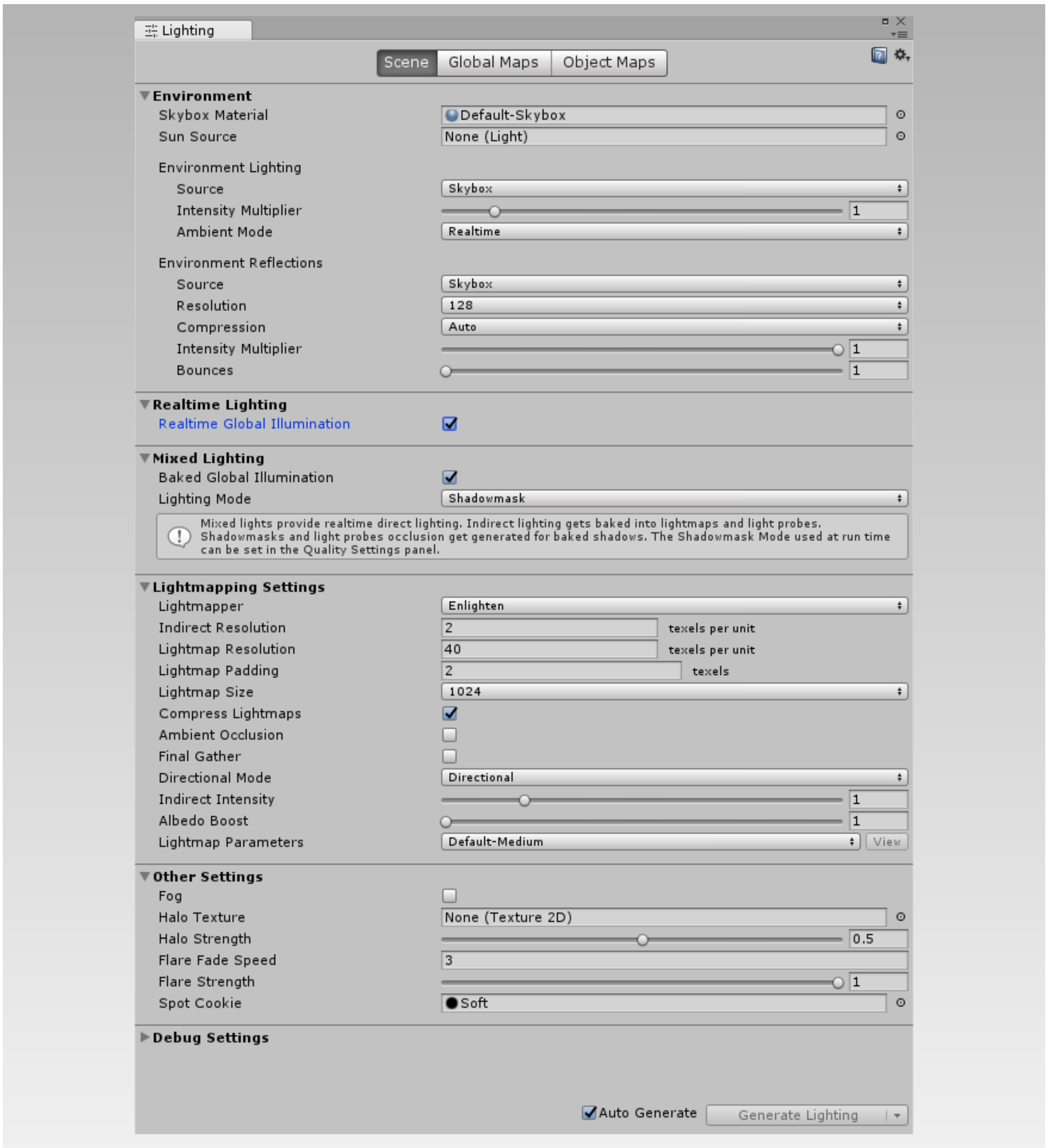
Therefore, the algorithm chosen in this work uses a simple approach, namely the **computation of the centerline based on distance transform**. The algorithm for centerline computation has been implemented using MeVisLab. Figure 4.6 shows the whole network.

The network is divided into **four parts**.

Given a set of masks derived from the previous segmentation, the **first part** computes a distance transform for each slice (figure 4.7).
Using the distance transform, the **second part** of the implementation determines a suitable threshold for each slice.

The threshold is further used to extract maximal values in each slice by the **third part** of the algorithm. As a matter of how the previous steps work together, the maximum value of the distance transform corresponds to the center of the segment. Iterating over all slices therefore results in a non-smoothed centerline as it can be seen in figure 4.8.

In a **final step**, the implementation smoothes out the centerline. The result of this step can be seen in figure 4.9. The list of points is finally ported to an XML-based file for



**Environment**

| | |
|---|---|
| Skybox Material | Default-Skybox |
| Sun Source | None (Light) |
| Environment Lighting | |
| Source | Skybox |
| Intensity Multiplier | 1 |
| Ambient Mode | Realtime |
| Environment Reflections | |
| Source | Skybox |
| Resolution | 128 |
| Compression | Auto |
| Intensity Multiplier | 1 |
| Bounces | 1 |

**Realtime Lighting**

| | |
|---|---|
| Realtime Global Illumination | ☑ |

**Mixed Lighting**

| | |
|---|---|
| Baked Global Illumination | ☑ |
| Lighting Mode | Shadowmask |

Mixed lights provide realtime direct lighting. Indirect lighting gets baked into lightmaps and light probes. Shadowmasks and light probes occlusion get generated for baked shadows. The Shadowmask Mode used at run time can be set in the Quality Settings panel.

**Lightmapping Settings**

| | |
|---|---|
| Lightmapper | Enlighten |
| Indirect Resolution | 2 texels per unit |
| Lightmap Resolution | 40 texels per unit |
| Lightmap Padding | 2 texels |
| Lightmap Size | 1024 |
| Compress Lightmaps | ☑ |
| Ambient Occlusion | □ |
| Final Gather | □ |
| Directional Mode | Directional |
| Indirect Intensity | 1 |
| Albedo Boost | 1 |
| Lightmap Parameters | Default-Medium View |

**Other Settings**

| | |
|---|---|
| Fog | □ |
| Halo Texture | None (Texture 2D) |
| Halo Strength | 0.5 |
| Flare Fade Speed | 3 |
| Flare Strength | 1 |
| Spot Cookie | ● Soft |

**Debug Settings**

☑ Auto Generate

Figure 4.5: Unity default light settings are used to configure global illumination.

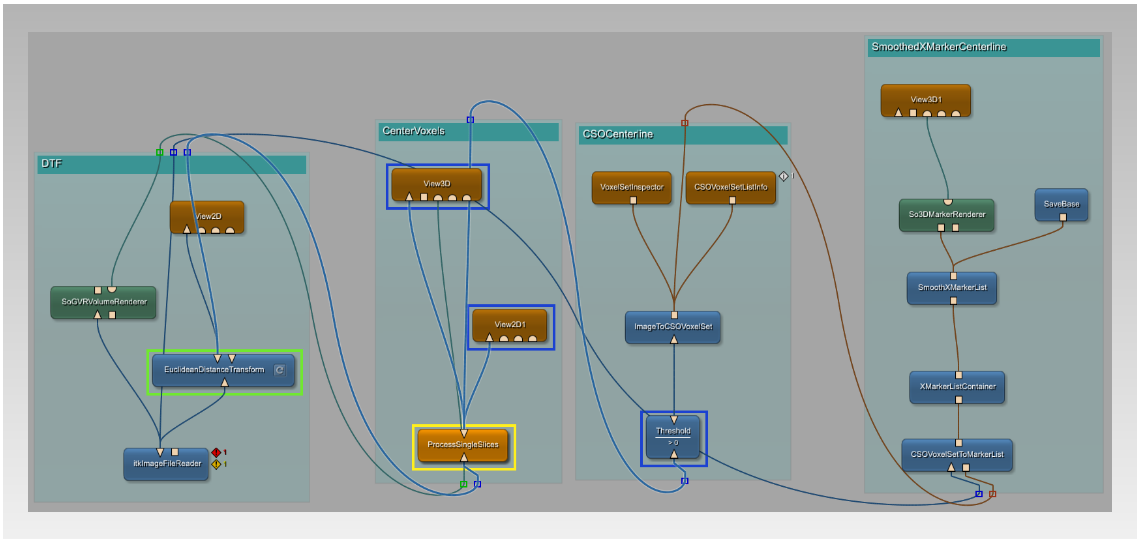
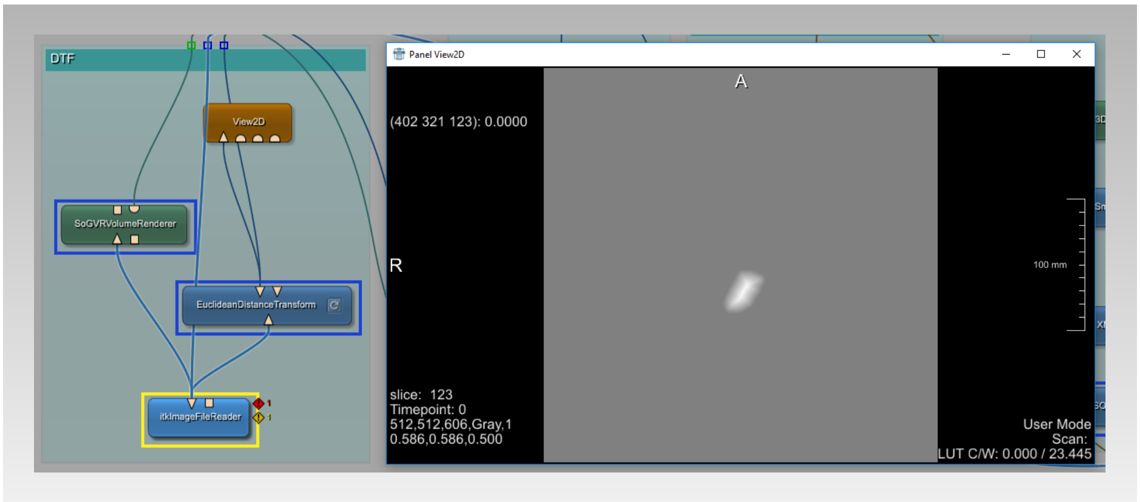


Figure 4.6: A MeVisLab network computing the centerline of the dissected aorta by implementing the algorithm proposed by Egger et al.



(402 321 123): 0.0000

slice: 123
Timepoint: 0
512,512,606,Gray,1
0.586,0.586,0.500

User Mode
Scan:
LUT C/W: 0.000 / 23.445

Figure 4.7: Computing the distance transform for each slice is the first part of the centerline computation.

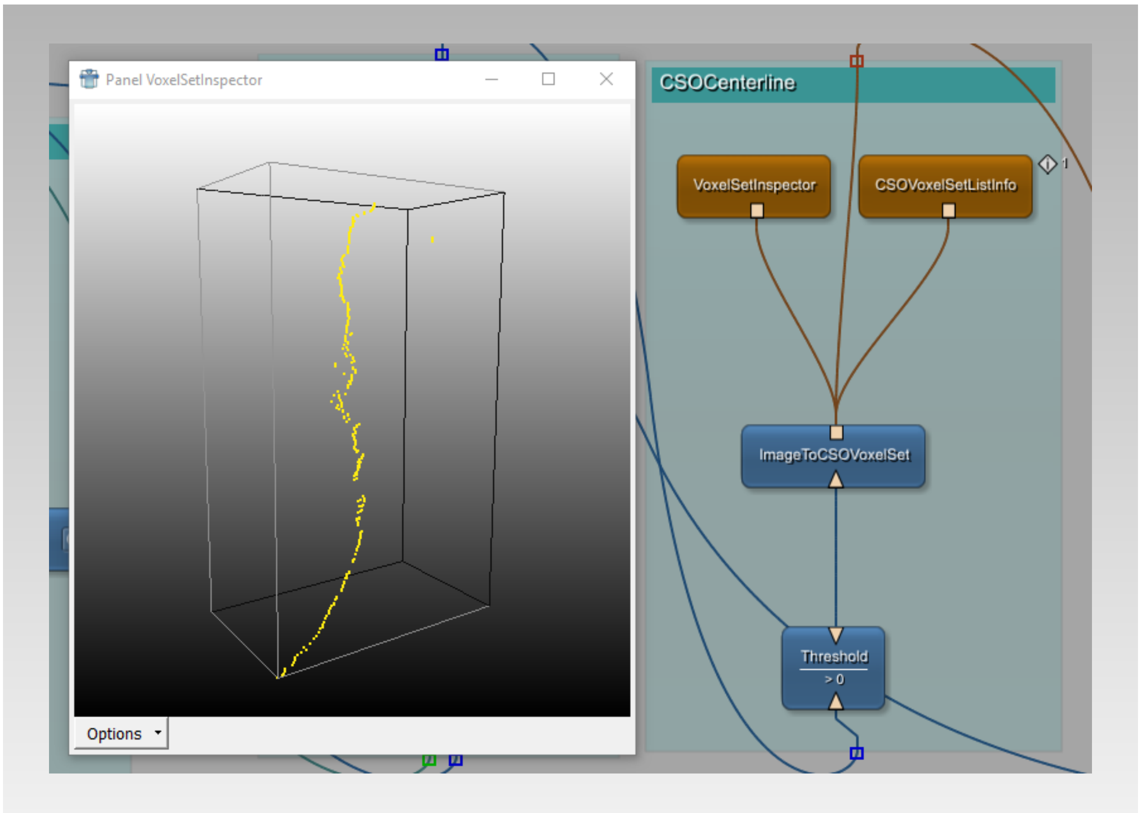
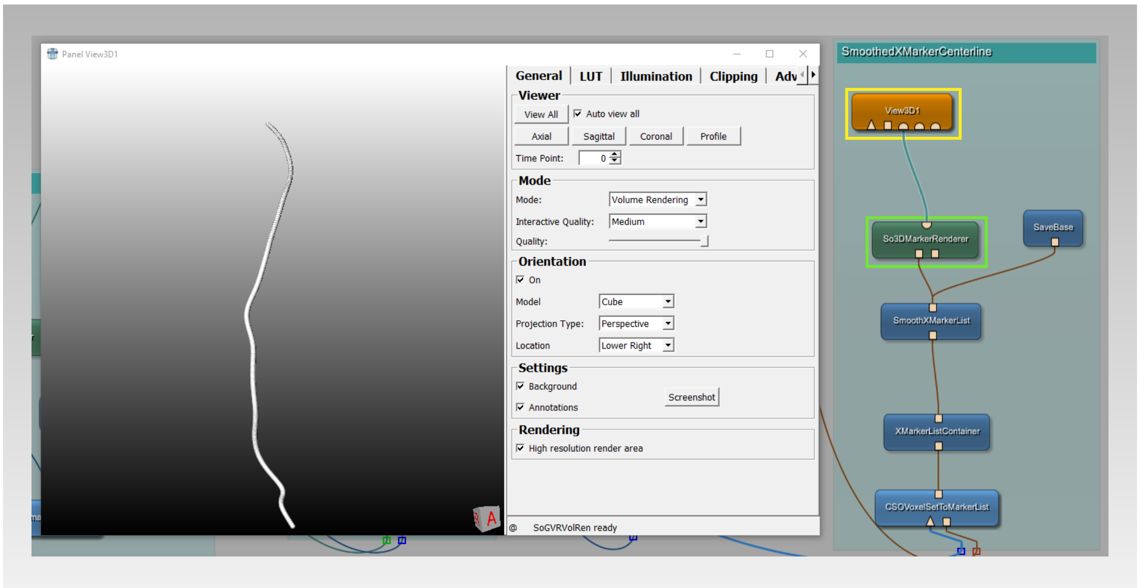


Figure 4.8: A non-smoothed candidate of the centerline based on previous thresholding results.



Viewer
☑ Auto view all
Time Point: 0

Mode
Mode: Volume Rendering
Interactive Quality: Medium
Quality:

Orientation
☑ On
Model Cube
Projection Type: Perspective
Location Lower Right

Settings
☑ Background
☑ Annotations

Rendering
☑ High resolution render area

@ SoGVRVolRen ready

Figure 4.9: The final centerline is smoothed out by the implementation.

further preprocessing.

In order to make use of the generated centerline in Virtual Reality, it has to be **imported into Unity**. A C# script (`PathEditor.cs`) parses the XML file. By iterating over all the given positions, a sphere is placed at each of the coordinates. The group of spheres is a Unity GameObject, namely the path a player can fly along.

## 4.6 Blood flow

The field of science that deals with making fluid flows visible is called **flow visualization**. It has been studied for centuries and is still an important field of research today. It has numerous applications in many different industrial branches. To name a few, the study of gaseous flows is a key issue in the development of cars, aircrafts, spacecrafts, and the design of machines such as turbines and combustion engines while liquid flows are studied in context with naval applications such as ship design [203]. In addition, flow visualization is widely used in civil engineering projects. Moreover, it plays an important role in chemistry.

Considering medical applications, **flow visualization is a fundamental basis for understanding how blood flows through vessels** [204, 205, 206, 207, 208, 209, 210, 211, 212] or how gaseous substances behave in the airways [213]. While experimental fluid dynamics visualizes flow by using methods such as the application of colored oil to the surface of an object placed into a wind tunnel or the usage of illuminated smoke, scientific visualization is based on texture advection or complex mathematical models such as the widely used Navier-Stokes equations [214].

Moreover, the fast increase of computational power has led to an extensive use of numerical methods following the emergence of a new branch of fluid mechanics referred to as **computational fluid dynamics** (CFD). As modern computers can easily produce a vast quantity of data, visualization has become even more important with respect to the understanding and the evaluation of simulated results. Computer-aided flow simulation and visualization can be considered especially important in medical applications as flow cannot easily be researched in living beings.

While this work is focused on implementing a prototype for visualizing a real patients aortic dissection in Virtual Reality, a practical use of the visualization requires it to be based on blood flow data produced by computational models. This is fundamental for providing a deeper understanding to how AD works and behaves. However, the implementation of CFD-frameworks modelling AD is still in progress. As a matter of fact, the final definition of interfaces communicating with these frameworks is a future issue. For the moment, **the prototype visualizes blood flow** by generating a predefined number of red blood cell objects traveling parallel to the centerline with constant velocity.

**Blood pressure could be modelled** in a simple manner: the area of the aorta cross sections could be derived from the results of segmentation. Given these areas the velocity of a single blood cell passing a specific cross section of the aorta could behave indirectly proportional to the respective area. In other words, the velocity of a blood cell may increase when the cell passes a narrow section of the vessel while it decreases when the aorta gets wider. However, since the visualization needs to consider more details such as the Windkessel effect or turbulence caused by the structure of the aortic wall, the simple approach is not sufficient for realistic visualization. As previously mentioned, this information has to be provided by underlying CFD-frameworks communicating with the visualization via well defined interfaces. As a matter of fact, this concept has not been implemented.

Summing up, **blood flow is one of the most crucial concepts of AD visualization**. It is supposed to help in gaining a deeper understanding on how AD behaves in real world. Data provided by underlying CFD frameworks is required to obtain a realistic visualization of blood flow. A proper definition of interfaces communicating with these frameworks is a future issue.

## 4.7 Flight Control

Flight control is relevant for both, the player and the blood cells. However, even though player and blood cells have a similar behaviour, there are some differences. The major difference is that while blood cells fly automatically, a player flight is supposed to be controlled by the user. Therefore, **a framework has been implemented to keep the flight control as abstract and extensible as possible**. Figure 4.10 shows the basic architecture of the flight control framework.

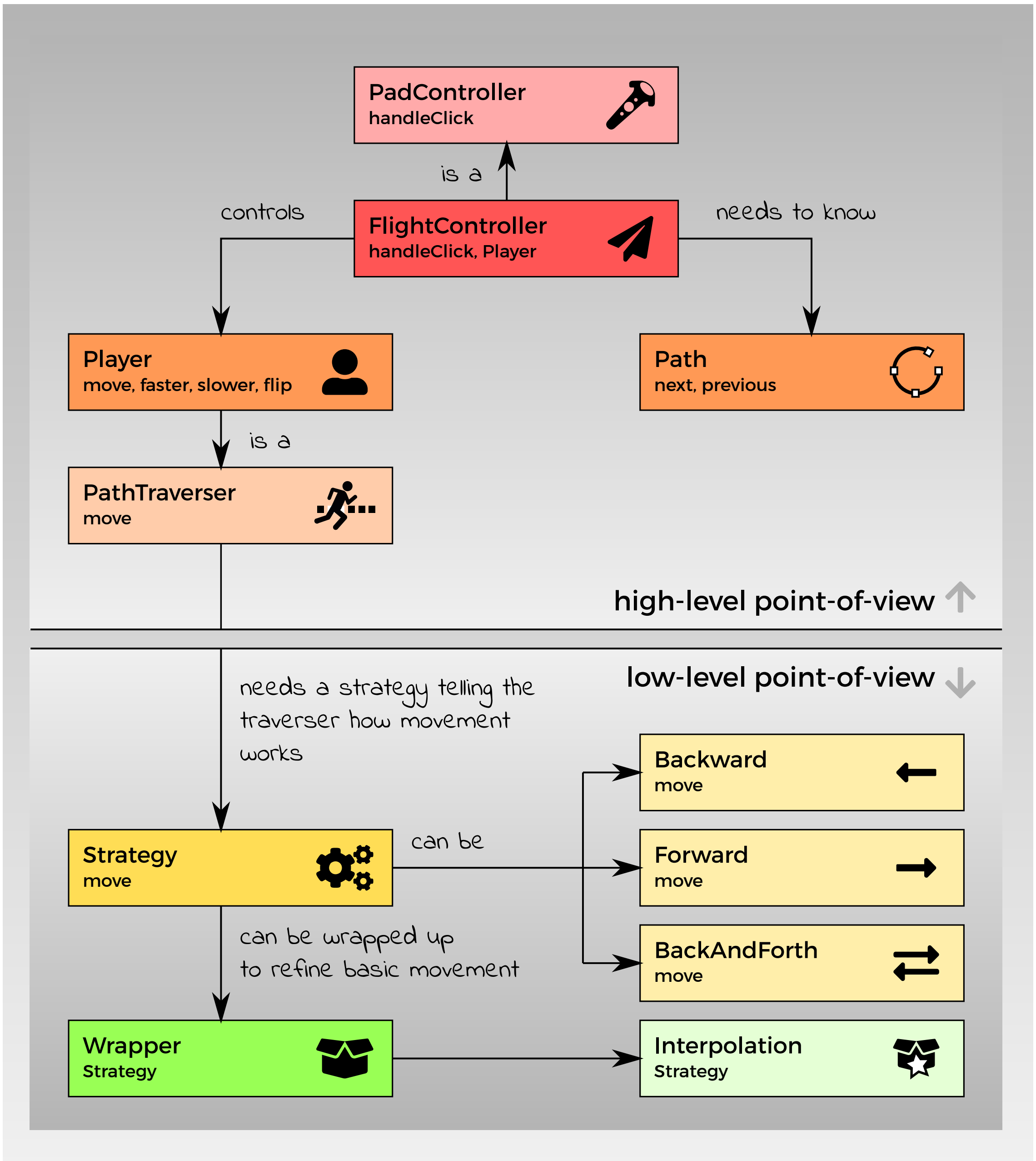


Figure 4.10: A framework for flight control

As the HTC Vive controller could be used for various applications, the framework implements a generic `PadController` that allows any subclassed controller to react to a users input. In addition, the `PadController` is the class linking the flight controller framework into the Unity scene. `FlightController` is a subclassed `PadController`. Besides being a `PadController` and therefore able to handle user input, the (player) `FlightController` needs to interact with two important objects: the player itself (i), and the path the player is supposed to fly along (ii).

The path (`Path`) is a simple class wrapping the centerline, a Unity `GameObject` consisting of multiple spheres. It extends the `GameObject` by the ability to hop back and forth between adjacent nodes (`next()`, `previous()`) and implements some more functions to provide information about the current node and whether it is the start node or the end node of the path.

The `Player` is the object that actually moves on a path. It is used by the aforementioned `FlightController`. It can perform operations such as moving (`move()`), increasing the speed of flight (`faster()`), decreasing the speed of flight (`slower()`) or flipping its flight direction (`flip()`). Flipping is a complex process as the handling of the controller gets inverted when the player changes its direction. A `Player` is just one out of many possible objects that could move along a path. There might be other objects, such as the blood cells. Therefore, an abstract class for traversing along a path is implemented (`PathTraverser`). The `Player` is a subclassed `PathTraverser`.

Any `PathTraverser` is able to move along a path. However, there are different algorithms of how movement works. To solve this issue in an extensible way, the strategy pattern has been used. The strategy pattern is a design pattern first described in *Design Patterns: Elements of Reusable Object-Oriented Software* written in 1994 by Gamma et al. [215]. The pattern suggests a way to decouple algorithms from the objects using it. It thereby enables a program to interchange algorithms at runtime. In case of the flight controller, `PathTraverser` is part of the strategy pattern as the way an object actually traverses the path is determined by the underlying strategy at runtime. Different strategies are implemented: one strategy allows an object to move in forward direction only (`Forward`) while a second one performs the same movement in backward direction (`Backward`). When one of these two strategies reach an end, the moving object simply stops there. A third strategy combines backward and forward strategy (`BackAndForth`). Using this strategy, an object moves in either direction until an end is reached. Then the algorithm is exchanged by the one not currently used which causes to object to flip direction. While blood has to flow in one direction only, the player uses a `BackAndForth` strategy for movement.

Besides the strategy, the framework is enriched by wrappers. A wrapper can be wrapped around any of the strategy implementations to modify its behaviour. In case of the `Player`, the `BackAndForth` strategy is wrapped by a wrapper implementing interpolations (`Interpolation`). This wrapper allows the `Player` to perform finer changes in speed as the behaviour is not bound to directly hopping from one node to the next but is rather able to do intermediate steps within a single hop. Summarized, the flight control framework allows the player to interact in its virtual environment, namely by controlling the speed and direction of the flight though the dissected aorta. In addition, the framework is designed using state-of-the-art patterns and with extensibility in mind.

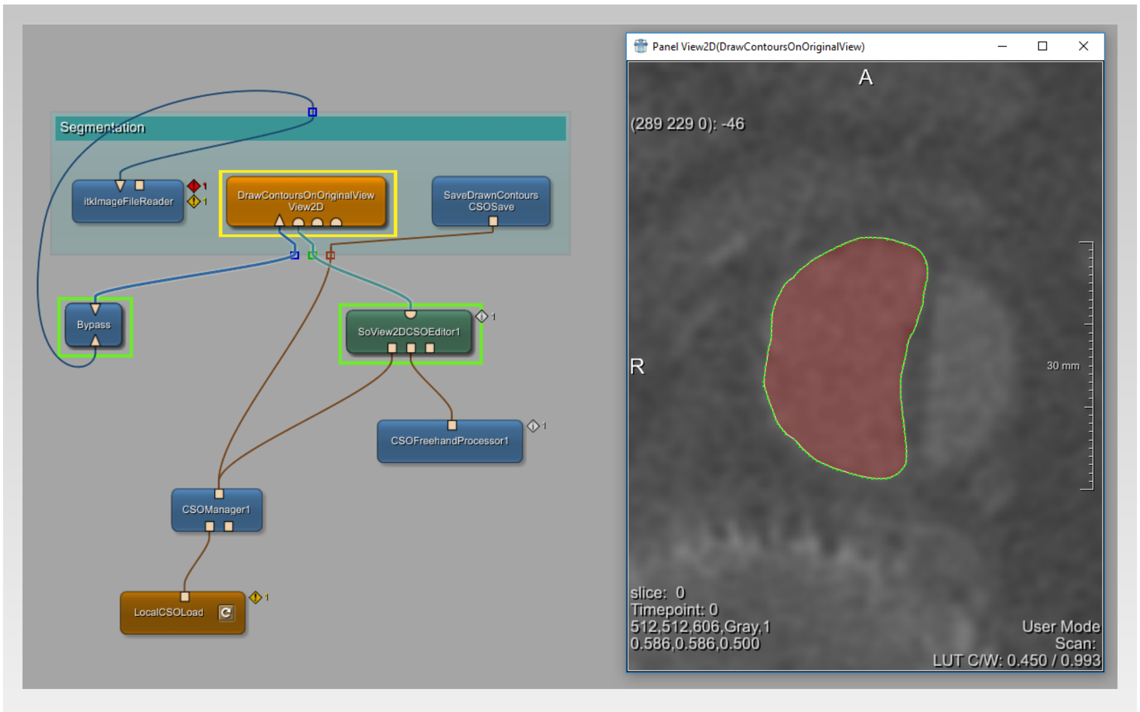
# 5 Results and evaluation

This chapter summarizes and evaluates the results of this work.

Section 5.1 describes the **final framework** and how a new CT scan could be visualized with the aid of that framework. It focuses on the usage of implemented MeVisLab frameworks and describes how the medical image is segmented, how the segmented image is used for model generation, how the model is refined, how the texture is generated and how the scene is set up in Unity. It also mentions some implementation-specific details relevant for extending the framework.

Section 5.2 lists **hardware and hardware requirements** used for development and testing of the framework. The VR system being used is briefly described and current versions of specific tools are mentioned.

Finally, section 5.3 presents the **results of performance tests** that have been applied on the visualization framework. The section concludes by pointing out that a smooth, jitter free VR user experience of the visualization of the dissected aorta is given up to a certain number of rendered blood cells using the aforementioned hardware.



(289 229 0): -46

slice: 0
Timepoint: 0
512,512,606,Gray,1
0.586,0.586,0.500

Scan:
LUT C/W: 0.450 / 0.993

Figure 5.1: Using a MeVisLab network, the true lumen of the aorta is segmented manually in an axial slice of a patients CT scan.

## 5.1 The final framework

This section describes the process of visualizing an aortic dissection in VR.

Given a medical image, may it come from a CT scan or any other source, has to be loaded into MeVisLab (using the `itkImageFileReader` module) using the **segmentation** network (`segmentation.mlab`, figure 5.1). If given, a (partial) CSO may be imported as well (using the `LocalCSOLoad` module). This may be the case if one wants to continue and unfinished segmentation or if a segmentation has to be refined or corrected. Given the image, the `View2D` module (`DrawContoursOnOriginalView`) may be used to segment the given slices. The `CSOManager1` module may be used to order/group the segmentations. Finally, the CSO can be saved using the `CSOSave` module (`SaveDrawnContours`). At the end of this step, the medical image and a corresponding cso file exist. As this step requires user interaction, it cannot be fully automated as long as the segmentation itself is not replaced by an algorithm that works without manual aid.

Given both, the medical image and the CSO file, the next step is **model generation**. Again, a MeVisLab network is used for that step (`dissection.mlab`). It requires the user to load the image (using the `itkImageFileReader` module) and the generated CSO (using the `LocalCSOLoad` module). A CSOManager allows realignment of segmentation contours if desired. Given that, the CSO can be converted to a 3D mask using the `CSOConvertTo3DMask` module. After that, MeVisLab converts the 3D mask into an iso surface (`WEMIsoSurface`), IsoSurface, namely a surface described by the implicit equation $F(x, y, z) = f$. The iso surface can finally be exported into an STL file (using the **WEMSave** module). Therefore, the resulting product of this step is a single STL file.

The STL file needs to be **refined** using blender now. Refinement steps and used have already been described in section 4.4.2 and comprise: inversion of normals (i), application of wall thickness (ii), and the generation of a texture map (iii). The latter is drawn using GIMP as described in section 4.4.2 as well. At a later stage of the project, this texture may be replaced by something more realistic. Some other ideas of visualizing the inner wall of the aorta exist. They are discussed in section 6.1.1. At the end of this step, a refined model and a corresponding texture exist.

The refined model and the texture are now needed for Unity scene setup. First, the model has to be loaded into unity. Using the texture, a material has then to be generated in Unity. This material has to be applied to the model in a next step. Camera and lighting settings are described in section 4.4.3. The result of this step is a static 3D scene modelled in Unity. It can already be rendered to VR.

The static Unity scene is now enriched by user interaction, namely the users ability to fly through the aorta. The centerline of the aorta has been chosen as a path to fly along. It is generated using MeVisLab. To do so, the segmentation has to be converted into a 3d mask (`3dMaskGeneration.mlab`). The responsible network loads the medical image (`itkFileReader`) and the previously generated CSO file (`CSOLoad`). These files are then used as an input for the `CSOConvertTo3DMask` module. Instead of converting the mask to an STL file, it is now saved directly (using the nrrd file format). The resulting file is then used as an input for centerline generation (`GenerateCenterline.mlab`). It is again loaded using the `itkImageFileReader` module. MeVisLab applies a distance transform on the mask, looks for a finds the center (maximum) in each distance transform, and smoothes the resulting voxels that represent the centerline. At the end of this step, a smoothed XMarker list can be exported using the XML file format (`SmoothXMarkerList` and `SaveBase`).

As already described, the static scene in Unity consists of the model and the texture. The centerline can now be loaded using `PathEditor.cs`. It parses the xml file and places a sphere game object on each position of the centerline. The path is then used by the flight controller described in section 4.7.

In a final step, the scene is enhanced by visualizing blood flow. As the underlying simulation of AD is still in development, interfaces for interaction are not yet defined. As a consequence, the blood flow visualization is currently solved using a simple approach: blood cells are Unity GameObjects that are generated "at the one end" of the aorta. The fly along the centerline with random offset and in opposite direction of the player. In a later stage of the overall project, when interfaces between visualization and simulation

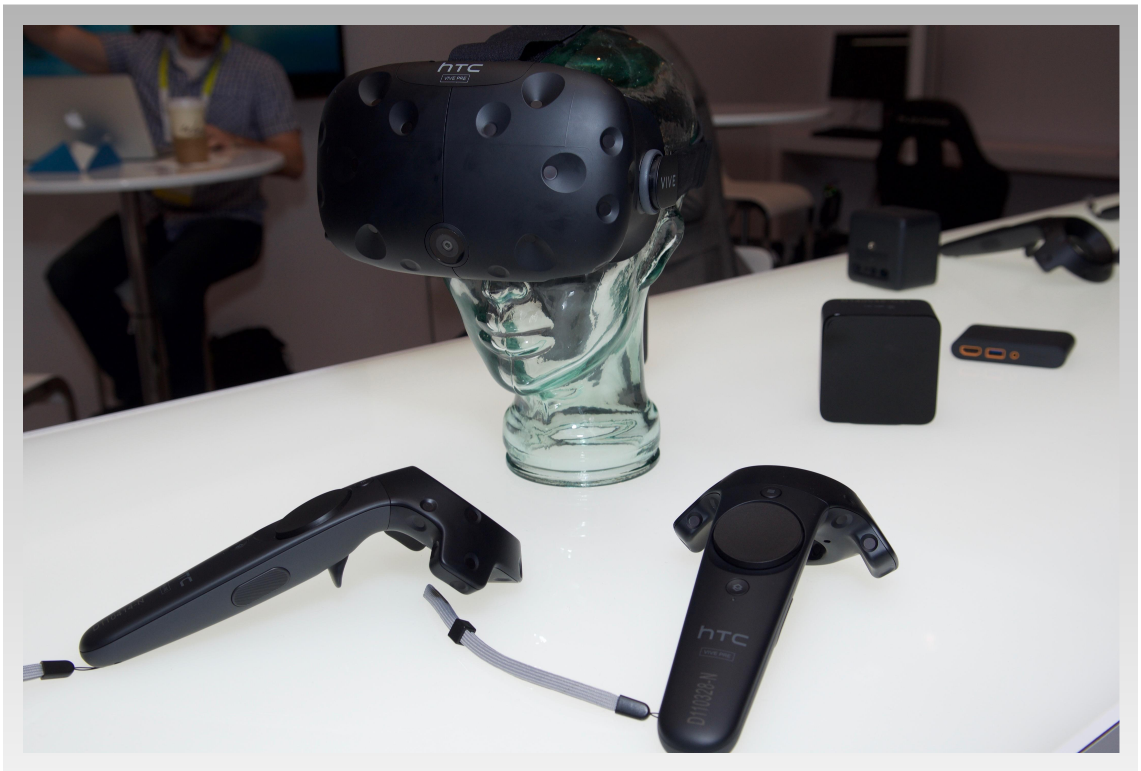


Figure 5.2: The HTC Vive consists of a Head Mounted Display (HMD), two handheld wireless controllers and two tracking stations. (source: https://commons.wikimedia.org/wiki/File:Vive_pre.jpeg)

are defined, the blood flow visualization may be replaced, e.g. by using Unity's particle system.

## 5.2 Environment of development

VR applications require **hardware with high performace**. Table 5.1 shows the system requirements recommended by VIVE™ [216] and the system used for implementation and testing this work. In addition, the HTC Vive has been the VR system being used. It is developed by HTC and Valve Corporation. It consists of a head mounted display (HMD), two wireless handheld controllers and two base stations called *Lighthouses* (see Figure 5.2).

The base stations are used for **tracking** purposes. HTC Vive is based on a VR design paradigm called *room scale*. It allows a user to freely move around in the physical world and transfers the user's movement to virtual reality by tracking position and orientation of HMD and controllers. When getting too close to physical walls, a piece of software called *chaperone* virtually reminds the user to stay inside the predefined space.

| | **recommended** | **used** |
|---|---|---|
| **Graphics** | NVIDIA® GeForce® GTX 1060,<br>AMD Radeon™ RX 480,<br>equivalent or better. | NVIDIA® GeForce® GTX 970 |
| **Processor** | Intel® Core™ i5-4590,<br>AMD FX™ 8350,<br>equivalent or better | Intel® Core™ i5-6600 |
| **Memory** | 4 GB RAM<br>or more | 32 GB RAM |
| **Video out** | HDMI 1.4,<br>DisplayPort 1.2<br>or newer | HDMI 2.0 |
| **USB ports** | 1x USB 2.0<br>or better port | USB 3.0 |
| **Operating system** | Windows® 7 SP1,<br>Windows® 8.1 or later,<br>Windows® 10 | Windows® 10 |

Table 5.1: Recommended and used system requirements.

Table 5.2 shows the **specifications of the different components** of the HTC Vive VR system as they are published on their official website [216].

In order to render a scene to the HMD, it has to be able to communicate with the application running on the PC. For this purpose, the aforementioned **runtime** is used. Valve distributes their runtime using STEAM™, an online distribution platform they have formerly developed for conventional games in 2002. For this work, STEAM® VR Version 2018-05-06 has been the runtime being used. The visualization of the dissected aorta has been developed using Unity Version 2017.3.0f3. In addition, the SteamVR Plugin Version 1.2.3 has been installed in Unity.

Given this environment, **framerates** have been measured using Unity's built-in profiler and stats-panel. For measuring **frametime**, **dropped frames** and **warp misses**, FCAT VR Capture Version 3.26.0.0, a tool provided by NVIDIA [217], has been used. FCAT VR Capture comes with a tool called FCAT VR Analyzer. It has been used to visualize the results yielded using FCAT VR Capture. The Analyzer is based on Python and requires pyqtgraph to be installed. For this work Anaconda 3.6.5 shipped with Python 3.6.3 has been used. The installed version of PyQtGraph has been 0.10.0.

## 5.3 Application performance

Considering the runtime being in between the computation and the final rendering of the frame, the following metrics have been measured in addition to the framerate to evaluate the performance of the visualization: frametime (i), dropped frames (ii), and warp misses (iii).

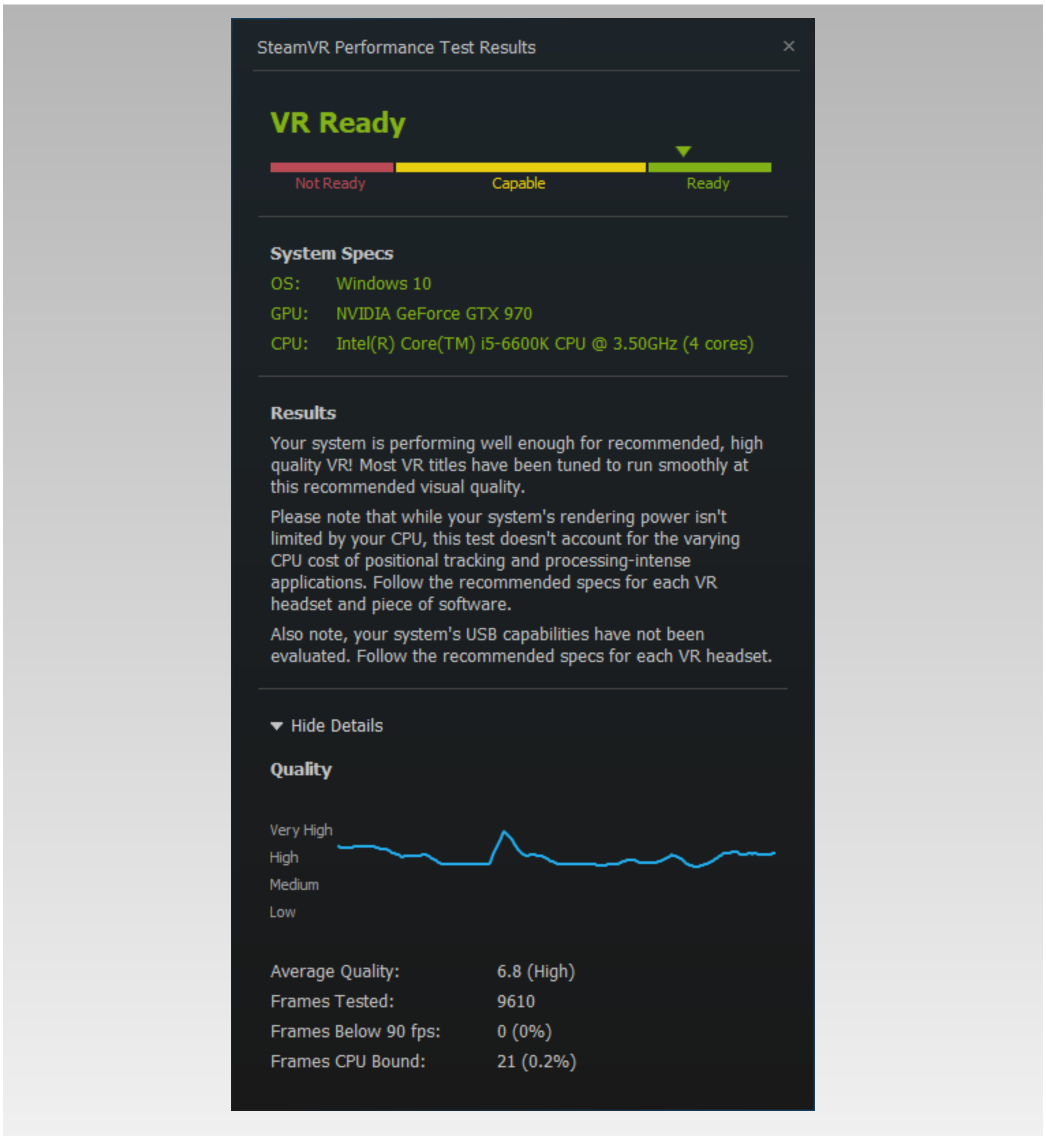
| **Headset Specs** | |
|---|---|
| Screen | Dual AMOLED 3.6” diagonal |
| Resolution | 1080 x 1200 pixels per eye (2160 x 1200 pixels combined) |
| Refresh rate | 90 Hz |
| Field of view | 110 degrees |
| Safety features | Chaperone play area boundaries and front-facing camera |
| Sensors | SteamVR Tracking, G-sensor, gyroscope, proximity |
| Connections | HDMI, USB 2.0, stereo 3.5 mm headphone jack, Power, Bluetooth |
| Input | Integrated microphone |
| Eye Relief | Interpupillary distance and lens distance adjustment |
| **Controller specs** | |
| Sensors | SteamVR Tracking |
| Input | Multifunction trackpad, Grip buttons, dual-stage trigger,<br>System button, Menu button |
| Use per charge | Approx. 6 hours |
| Connections | Micro-USB charging port |
| **Tracked area requirements** | |
| Standing / seated | No min. space requirements |
| Room-scale | 6’6” x 5’ min. room size, 16’4” max. between base stations |

Table 5.2: Specifications of the HTC VIVE Virtual Reality System.

In addition to their runtime, **Valve provides a performance test** checking whether the current hardware is capable of rendering to VR. Figure 5.3 shows the output of the results after running the performance test on the given hardware. One important result in this test is the number of frames below 90 fps as 90 fps is widely considered to be the minimal number of fps to provide a smooth user experience.

Frametimes, dropped frames and warp misses correlate with fps. All **metrics are influenced by the number of game objects placed into a scene**. In addition, the complexity of a game object contributes to the overall performance. Regarding the visualization of the aorta, the aorta itself is by far the most complex object in terms of mesh complexity given by it’s number of vertices or number of edges respectively. However, the aorta is a static object rendered only once while the blood cell is a moving object being cloned multiple times. Tests have shown, that performance is mainly affected by the number of blood cells used.

Evaluation has been done using different number of blood cells measuring overall memory consumption, average framerate, average frame timings, dropped frames, and warp misses for 60 seconds (Table 5.3). Figure 5.4 shows different frame timings for 0, 2200 and 5000 blood cells. Figure 5.5 shows the same plot with corresponding interval plots. As it can be seen in the interval plot for a setting with zero blood cells, all frames have successfully been generated while the interval plot for 5000 blood cells shows a high number of dropped frames. Plots show, that frametimes drop after initialization phase. This



SteamVR Performance Test Results

**VR Ready**

**System Specs**

| | |
|---|---|
| OS: | Windows 10 |
| GPU: | NVIDIA GeForce GTX 970 |
| CPU: | Intel(R) Core(TM) i5-6600K CPU @ 3.50GHz (4 cores) |

**Results**

Your system is performing well enough for recommended, high quality VR! Most VR titles have been tuned to run smoothly at this recommended visual quality.

Please note that while your system's rendering power isn't limited by your CPU, this test doesn't account for the varying CPU cost of positional tracking and processing-intense applications. Follow the recommended specs for each VR headset and piece of software.

Also note, your system's USB capabilities have not been evaluated. Follow the recommended specs for each VR headset.

**Quality**

| | |
|---|---|
| Average Quality: | 6.8 (High) |
| Frames Tested: | 9610 |
| Frames Below 90 fps: | 0 (0%) |
| Frames CPU Bound: | 21 (0.2%) |

Figure 5.3: The performance test provided by STEAM® VR checks the system's capability of rendering to VR.

| **number of blood cells** | **overall mem consumption** | **avg fps** | **avg frametime** | **dropped frames** | **warp misses** |
|---|---|---|---|---|---|
| 0 | 180 MB | 112.4 | 4.29 ms | 0 % | 0 |
| 2,000 | 250 MB | 110.3 | 7.37 ms | 0.03 % | 0 |
| 2,200 | 270 MB | 102.4 | 9.00 ms | 0.11 % | 0 |
| 2,500 | 280 MB | 72.9 | 11.56 ms | 0.39 % | 5 |
| 3,000 | 290 MB | 64.3 | 13.16 ms | 0.44 % | 3 |
| 5,000 | 350 MB | 22.7 | 26.11 ms | 0.63 % | 0 |

Table 5.3: Evaluation results with varying number of blood cells. Smooth results have been obtained with blood cells up to 2,200.

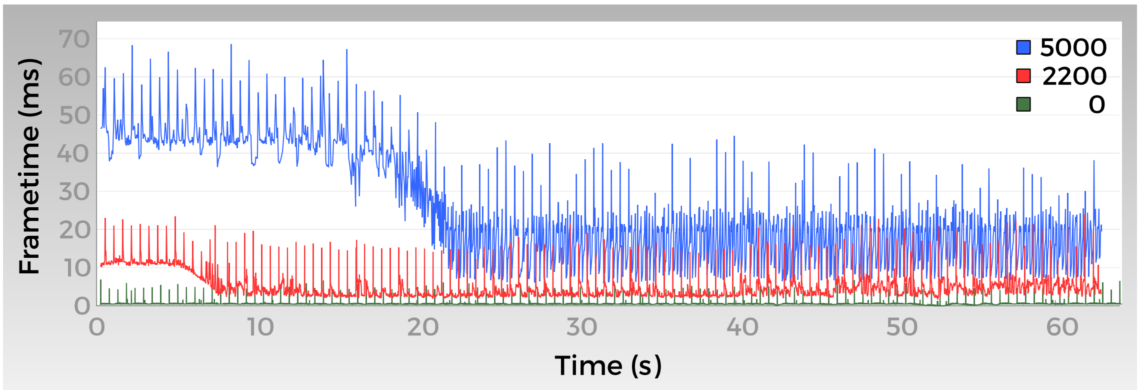


Figure 5.4: Frametime plots for three different settings with 0, 2000 and 5000 blood cells.

effect may be attributed to caching. In addition, the frametimes are oscillating at high frequency. Analysing the running scene using Unity revealed that the oscillation is caused by the garbage collector which runs at scheduled time steps. All in all, a good performance providing **a smooth user experience has been achieved with up to 2,200 blood cells** while 5000 blood cells had been far too much for rendering the scene smoothly.

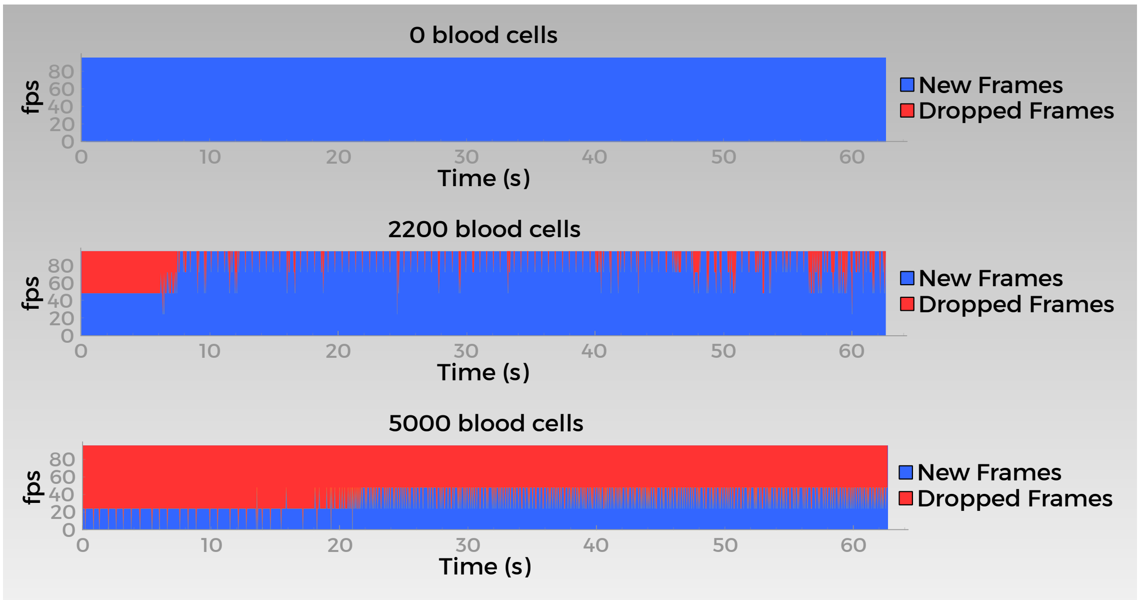


Figure 5.5: The interval plots show that all frames have sucessfully been generated for a setting with 0 blood cells while most of the frames have been dropped logging a setting with 5,000 blood cells.

# 6 Discussion and future outlook

This chapter focuses on **future outlook** as discussed in section 6.1. While the current framework proves that VR visualization of dissected aortas is feasible, various suggestions may lead to a final product that might eventually aid medicals in their daily clinical routine. Among these suggestions are some describing how a higher degree of automation may be achieved as this would be a desirable goal for a final product. Other suggestions deal with new features that might enrich the current solution, may it be a more sophisticated user interaction or the highly required development of interfaces for communication with underlying simulation frameworks. In addition to these suggestions, performance and portablity issues are briefly discussed.

The chapter ends with **concluding words** (section 6.2) highlighting the impact of the work that has been done: visualizing AD in VR is both, feasible and meaningful. As such, this work may be an influential contribution in fighting the tragedy of AD.

## 6.1 Improvements

### 6.1.1 Automation and functionality

While some improvements can be issued to simple software engineering, others require further research as they comprise problems that are not fully solved today. In particular, the following steps lack solutions not depending upon user interaction to some extent: segmentation (i), centerline computation (ii) and flow visualization (iii).

Regarding the **automationion of segmentation**, various proposals based on thresholding [108, 109, 110, 111, 112, 113], boundary [116, 117, 118, 119, 120] or region [107, 124, 125, 126] have been made. Modern approaches of segmentation often rely on deep learning [143, 155]. According to Chollett, deep learning is a subfield of machine learning [218]. While Chollet states conventional software development to be to process of generating answers given rules and data, he considers machine learning to be the process of generating rules given data and answers. In case of segmentation, real-life scans (data) and manual segmentation done by experts (answers) would lead to rules allowing the algorithms to segment vessels in formerly unknown datasets. However, as Tetteh states, the assembly of a properly labeled dataset for training such algorithms is the bottleneck for most medical applications [155]. Therefore, even though machine learning attempts look promising, a research in this direction would comprise research in the acquisition of large and meaningful datasets. Next to data acquisition, there are studies dealing with the generation of medical image data [219]. However, even though data generation resolves the aforementioned bottleneck, the question of artificial data being suitable training sets for machine learning is another subject that requires further investigation.

The problem of **centerline computation** highly relates to the problem of segmentation. When segmentation is solved reliably, centerline computation can be solved alongside [155]. In context of this application, two facts need to be pointed out: the centerline is just used as a path for flight control (i) and the aorta is relatively thick and scaled to a high extent when placed into the scene (ii). Since these two facts exist, a rough approximation of the centerline is fully sufficient in context of the given application. Therefore, when segmentation is solved, simple algorithms such as the previously mentioned distance transform could be applied to the segmentation for centerline computation. However, further development of the application might lead to the need of an accurate computation of the centerline. In this case, the computation of the centerline becomes subject to addditional research.

**Flow visualization** is another topic that offers multiple ways of improvement. At the current state, the blood cells are generated with random offset to the centerline following it's path at constant speed. A more realistic visualization might take blood pressure into account. From an intuitive point-of-view, the pressure increases when the radius of the aorta decreases and vice-versa. Pressure and velocity are related in a proportional way. Therefore, the velocity of a single blood cell could be modelled indirectly proportional to the area of the cross section it currently passes through. This area is known from segmentation and could be interpolated for missing positions. Even though this visualization would give a more realistic impression, it still disregards that blood pressure is not constant. A realistic flow visualization requires complex mathematical models. A goal for this work should be to provide an interface to such models so that they can feed the visualization with their computations. This interface is a key concept towards a practical use of this prototype in medical applications.

**Navigation** is another issue that needs to be considered. The current solution allows the user to fly along a predefined path. Different attempts might need to be found in coordination with medical staff using the final solution in clinical routine. The close examination of details might require the solution to allow a user to step aside the predefined path in order to take a closer look at certain areas of the aortic wall.

This examination might raise the need for realistic **texture**. Available texture databases might help in solving this issue. Texture could also be generated programmatically. This approach would allow the solution to visually encode relevant information provided by medical images or computed simulation.

One area that would require a programmatic solution of texture rendering is the **visualization of wall stress**. A common way of visualizing stress is by using heatmaps. These have to be generated using the information obtained from computed simulations. A heat map might be applied on top of the already existent texture. Users might also require the ability to toggle the display of additional informative textures such as heatmaps.

Another step towards realistic rendering could be a **dynamic model of the aorta**. At the current state of development, the solution provides a static model of the aorta. However, when considering time-dependant events such as the tearing of the aortic wall animations might be needed.

### 6.1.2 Performance and other technical issues

From a technical point-of-view, a higher resolution would be helpful when examining AD in VR. For this purpose, the currently used HTC Vive could be replaced by more modern VR systems such as **HTC Vive Pro** released on January 2018 [220]. When upgrading the VR system, better hardware might be required. This requirement might also arise when the complexity of the visualization increases.

Next to upgrading the hardware, hardware agnostic improvements should be considered, i.e. the **software performance** could be improved. A first step might be to find a proper frustrum: in it's predefined settings, Unity allows a user to look ahead for 1000m. As a matter of fact, the software needs to precompute objects within that range. However, as the aorta is a closed room, the frustrum could be scaled down to a reasonable size. The size highly depends on the scale of the aorta. Next to reducing the distance of the far plane, thoughtful considerations of removing game objects would highly increase performance. In particular, the removal of unseen blood cells has to be thought of. However, as a user can freely move in virtual reality, it is not sufficient to simply remove all blood cells the user has passed. A deeper analysis of bottlenecks might reveal more possibilities to improve performance. Some bottlenecks might be: the resolution of the segmentation in z-axis (i), the realtime computation of the centerline when using a dynamic model of the aorta (ii), and the smoothness of the model of the aorta as it is driven by the number of vertices and edges used (iii).

Another technical issue, next to performance, is that the solution might have to be adapted for usage with **different VR systems**. Unity offers support on that issue but there might be some yet unknown considerations to be taken into account.

### 6.1.3 Evaluation

In a final step, the **process of evaluation** has to be reconsidered. When the solution reaches the state of being suitable for clinical routine, user studies among medical stuff need to be arranged. Quantitative measures of user experience might need to be found. If used, they would offer a way towards evaluation strategies that are more objective than currently used ones, e.g. questionaires in user studies.

## 6.2 Conclusion

Within the last years, VR has gained high popularity. Also science is looking for practical use of VR: the visualization of scientific data in VR seems to be a tempting goal. Even though VR has lately been used in medical projects [221], a visualization of AD is, while useful, yet unavailable.

This work has shown that **medical image data can be ported to Virtual Reality** allowing a more profound understanding of the internals of a human body. In case of aortic dissection, an immersive visualization aims in better understanding the highly lethal disease that is still a topic of active research today. However, the current state of the solution does not offer a fully automated visualization and it is unlikely that when no further restrictions are given medical image data can be ported to VR without any user interaction. Nonetheless, increasing the grade of automation can be subject to further work, by improving the automation of intermediate steps on the one hand and by coupling these steps together more tightly on the other hand.

Summing up, the current solution shows that a VR **visualization of AD is feasible and meaningful**. However, to develop a visualization that can be used in daily clinical routine, some further work has to be done. In the end, a **realistic simulation and visualization of AD could be a crucial step** in fighting against a disease that often leads to tragic death: the aortic dissection. May this work be a milestone in this fight.

# Bibliography

[1] F. J. Criado. Aortic dissection: a 250-year perspective. *Texas Heart Institute journal*, 38(6):694–700, 2011. ISSN: 1526-6702. pmid: 22199439.

[2] E. Bianconi, A. Piovesan, F. Facchin, A. Beraudi, R. Casadei, F. Frabetti, L. Vitale, M. C. Pelleri, S. Tassani, F. Piva, S. Perez-Amodio, P. Strippoli, and S. Canaider. An estimation of the number of cells in the human body. *Annals of Human Biology*, 40(6):463–471, 2013. ISSN: 1464-5033. DOI: 10.3109/03014460.2013.807878. pmid: 23829164.

[3] R. Sender, S. Fuchs, and R. Milo. Revised Estimates for the Number of Human and Bacteria Cells in the Body. *PLOS Biology*, 14(8), 2016. ISSN: 1545-7885. DOI: 10.1371/journal.pbio.1002533.

[4] D. N. Shier, J. L. Butler, and R. Lewis. *Hole's Human Anatomy & Physiology*. McGraw-Hill Education, 2015. ISBN: 978-0-07-802429-0.

[5] L. A. Urry, M. L. Cain, S. A. Wasserman, P. V. Minorsky, J. B. Reece, and N. A. Campbell. *Campbell Biology*. Pearson Education, New York, NY, 11th edition, 2017. ISBN: 978-0-13-409341-3.

[6] merriamwebster.com. Blood. 2018. URL: https://www.merriam-webster.com/dictionary/blood (visited on 04/19/2018).

[7] merriamwebster.com. Circulatory System. 2018. URL: https://www.merriam-webster.com/dictionary/circulatory%20system (visited on 04/21/2018).

[8] D. Venes, B. G. Fenton, and J. Patwell. *Taber's Cyclopedic Medical Dictionary*. F.A. Davis Company, 23rd edition, 2017. ISBN: 978-0-8036-5904-9.

[9] G. Belz. Elastic properties and Windkessel function of the human aorta. *Cardiovascular Drugs and Therapy*, 9(1):73–83, 1995. ISSN: 0920-3206, 1573-7241. DOI: 10.1007/BF00877747.

[10] oxforddictionaries.com. Histology. 2018. URL: https://en.oxforddictionaries.com/definition/histology.

[11] A. Mescher. *Junqueira's Basic Histology*. McGraw-Hill Education, 14th edition, 2015. ISBN: 978-0-07-184268-6.

[12] oxforddictionaries.com. Anatomy. 2018. URL: `https://en.oxforddictionaries.com/definition/anatomy` (visited on 04/18/2018).

[13] K. M. Van De Graaff. *Human Anatomy*. McGraw-Hill Education, 2001. ISBN: 978-0-00-468701-8.

[14] G. J. Tortora and B. H. Derrickson. *Principles of Anatomy and Physiology*. Wiley, 2011. ISBN: 978-0-470-56510-0.

[15] G. J. Tortora and M. Nielsen. *Principles of Human Anatomy*. Wiley, 2013. ISBN: 978-1-118-34499-6.

[16] F. H. Netter. Thorax: Coronal Section Of Heart, Ascending Aorta. In *Atlas of Human Anatomy (Netter Basic Science)*, page 244. Grune & Stratton Inc., 2014. ISBN: 978-0-8089-2451-7.

[17] F. H. Netter. Coronary Arteries: Imaging. In *Atlas of Human Anatomy (Netter Basic Science)*, page 246. Grune & Stratton Inc., 2014. ISBN: 978-0-8089-2451-7.

[18] F. H. Netter. Left Atrium And Ventricle. In *Atlas of Human Anatomy (Netter Basic Science)*, page 248. Grune & Stratton Inc., 2014. ISBN: 978-0-8089-2451-7.

[19] F. H. Netter. Thyroid Gland and Pharynx: Posterior View. In *Atlas of Human Anatomy (Netter Basic Science)*, page 95. Grune & Stratton Inc., 2014. ISBN: 978-0-8089-2451-7.

[20] F. H. Netter. Arteries to Brain: Schema. In *Atlas of Human Anatomy (Netter Basic Science)*, page 157. Grune & Stratton Inc., 2014. ISBN: 978-0-8089-2451-7.

[21] F. H. Netter. Diaphragm: Abdominal Surface. In *Atlas of Human Anatomy (Netter Basic Science)*, page 192. Grune & Stratton Inc., 2014. ISBN: 978-0-8089-2451-7.

[22] F. H. Netter. Mediastinum: Left Lateral View. In *Atlas of Human Anatomy (Netter Basic Science)*, page 258. Grune & Stratton Inc., 2014. ISBN: 978-0-8089-2451-7.

[23] F. H. Netter. Esophagus in Situ. In *Atlas of Human Anatomy (Netter Basic Science)*, page 259. Grune & Stratton Inc., 2014. ISBN: 978-0-8089-2451-7.

[24] F. H. Netter. Arteries of Ureters and Urinary bladder. In *Atlas of Human Anatomy (Netter Basic Science)*, page 349. Grune & Stratton Inc., 2014. ISBN: 978-0-8089-2451-7.

[25] I. Vilacosta and J. S. Román. Acute aortic syndrome. *Heart*, 85(4):365–368, 2001. DOI: `10.1136/heart.85.4.365`.

[26] oxforddictionaries.com. Pathology. 2018. URL: `https://en.oxforddictionaries.com/definition/pathology`.

[27] J. S. Corvera. Acute aortic syndrome. *Annals of Cardiothoracic Surgery*, 5(4):188–193, 2016. DOI: `10.21037/acs.2016.04.05`.

[28] I. A. Khan and C. K. Nair. Clinical, Diagnostic, and Management Perspectives of Aortic Dissection. *Chest Journal*, 122:311–328, 2002. ISSN: 0012-3692. DOI: `10.1378/chest.122.1.311`.

[29] R. Häring and A. Anders. *Lehrbuch Chirurgie: Mit Repetitorium.* De Gruyter Lehrbuch. de Gruyter, Berlin, 2nd edition, 1988. 1014 pages. ISBN: 978-3-11-011280-1.

[30] P. R. Seidensticker and L. K. Hofmann. *Dual Source Ct Imaging: With 52 Tables.* Springer Medizin, Heidelberg, 2008. 303 pages. ISBN: 978-3-540-77601-7.

[31] M. A. Creager and E. Braunwald. *Atlas of Vascular Disease.* Current Medicine Group, 2003. ISBN: 978-1-57340-191-3.

[32] D. Hahn. *Kardiovaskuläres System.* J. Freyschmidt, editor. Handbuch diagnostische Radiologie. Springer Medizin, Berlin, 2007. 477 pages. ISBN: 978-3-540-41420-9.

[33] H. Eggebrecht, B. Plicht, P. Kahlert, and R. Erbel. Intramural Hematoma and Penetrating Ulcers: Indications to Endovascular Treatment. *European Journal of Vascular and Endovascular Surgery*, 38(6):659–665, 2009. ISSN: 10785884. DOI: 10.1016/j.ejvs.2009.09.001.

[34] R. Erbel. Diagnosis and management of aortic dissection Task Force on Aortic Dissection, European Society of Cardiology. *European Heart Journal*, 22(18):1642–1681, 2001. DOI: 10.1053/euhj.2001.2782.

[35] oxforddictionaries.com. Epidemiology. 2018. URL: https://en.oxforddictionaries.com/definition/epidemiology.

[36] iradonline.org. Information for Patients from the International Registry of Acute Aortic Dissection. 1996. URL: http://www.iradonline.org/irad.html (visited on 04/10/2018).

[37] P. G. Hagan, C. A. Nienaber, E. M. Isselbacher, D. Bruckman, D. J. Karavite, P. L. Russman, A. Evangelista, R. Fattori, T. Suzuki, J. K. Oh, A. G. Moore, J. F. Malouf, L. A. Pape, C. Gaca, U. Sechtem, S. Lenferink, H. J. Deutsch, H. Diedrichs, J. M. y Robles, A. Llovet, D. Gilon, S. K. Das, W. F. Armstrong, G. M. Deeb, and K. A. Eagle. The International Registry of Acute Aortic Dissection (IRAD) - New insights into an old disease. *JAMA*, 283(7):897, 2000. DOI: 10.1001/jama.283.7.897.

[38] P. C. Spittell, J. A. Spittell, J. W. Joyce, A. J. Tajik, W. D. Edwards, H. V. Schaff, and A. W. Stanson. Clinical features and differential diagnosis of aortic dissection: experience with 236 cases (1980 through 1990). *Mayo Clinic proceedings*, 68(7):642–651, 1993. ISSN: 0025-6196. pmid: 8350637.

[39] E. M. Isselbacher. Epidemiology of Thoracic Aortic Aneurysms, Aortic Dissection, Intramural Hematoma, and Penetrating Atherosclerotic Ulcers. In Kim A. Eagle, Ragavendra R. Baliga, Eric M. Isselbacher, and Christoph A. Nienaber, editors, *Aortic Dissection and Related Syndromes*, pages 3–15. Springer, 2006. DOI: 10.1007/978-0-387-36001-0_1.

[40] C. A. Nienaber and R. E. Clough. Management of acute aortic dissection. *The Lancet*, 385(9970):800–811, 2015. DOI: 10.1016/s0140-6736(14)61005-9.

[41] I. H. Melvinsdottir, S. H. Lund, B. A. Agnarsson, K. Sigvaldason, T. Gudbjartsson, and A. Geirsson. The incidence and mortality of acute thoracic aortic dissection: results from a whole nation study. *European Journal of Cardio-Thoracic Surgery*, 50(6):1111–1117, 2016. DOI: 10.1093/ejcts/ezw235.

[42] W. Li and P. Liu. Epidemiology of Acute Type A Aortic Dissection. In Ying-Fu Chen, editor, *Recent Advances in Acute Type A Aortic Dissection*, pages 3–29. Bentham Science Publishers, 2018. ISBN: 978-1-68108-089-5. DOI: 10.2174/97816810808881150100004.

[43] C. Olsson, S. Thelin, E. Stahle, A. Ekbom, and F. Granath. Thoracic Aortic Aneurysm and Dissection: Increasing Prevalence and Improved Outcomes Reported in a Nationwide Population-Based Study of More Than 14 000 Cases From 1987 to 2002. *Circulation*, 114(24):2611–2618, 2006. DOI: 10.1161/circulationaha.106.630400.

[44] I. Mészáros, J. Mórocz, J. Szlávi, J. Schmidt, L. Tornóci, L. Nagy, and L. Szép. Epidemiology and Clinicopathology of Aortic Dissection. *Chest*, 117:1271–1278, 2000. DOI: 10.1378/chest.117.5.1271.

[45] K. B. O'Dell and S. N. Hakim. Dissecting thoracic aortic aneurysm in a 22-year-old man. *Annals of Emergency Medicine*, 19(3):316–318, 1990. DOI: 10.1016/s0196-0644(05)82053-0.

[46] C. R. Fikar and S. Koch. Etiologic Factors of Acute Aortic Dissection in Children and Young Adults. *Clinical Pediatrics*, 39(2):71–80, 2000. DOI: 10.1177/000992280003900201.

[47] J. L. Januzzi, E. M. Isselbacher, R. Fattori, J. V. Cooper, D. E. Smith, J. Fang, K. A. Eagle, R. H. Mehta, C. A. Nienaber, and L. A. Pape. Characterizing the young patient with aortic dissection: results from the international registry of aortic dissection (IRAD). *Journal of the American College of Cardiology*, 43(4):665–669, 2004. DOI: 10.1016/j.jacc.2003.08.054.

[48] K. Ngan, C. Hsueh, H. C. Hsieh, and S. Ueng. Aortic dissection in a young patient without any predisposing factors. *Chang Gung medical journal*, 29(4):419–423, 2006. ISSN: 2072-0939. pmid: 17051841.

[49] P. Hountis, P. Dedeilias, and K. Bolos. Acute aortic dissection in a young patient without Marfan fibrillinopathy: a case report. *Cases Journal*, 2(1):7076, 2009. DOI: 10.1186/1757-1626-2-7076.

[50] M. Vallée, J. Pineault-Lepage, D. Ouimet, and V. Pichette. A case of an aortic dissection in a young adult: a refresher of the literature of this ”great masquerader”. *International Journal of General Medicine*:889, 2011. DOI: 10.2147/ijgm.s22906.

[51] J. Mathews. Rare Case of Acute Aortic Dissection in a Young Non-Hypertensive and Non-Pregnant Lady. *Journal of Cardiovascular Diseases & Diagnosis*, 02(03), 2014. DOI: 10.4172/2329-9517.1000154.

[52] H. Rössler, W. Rüther, P. Pitzen, and J. Steinhagen. *Orthopädie und Unfallchirurgie*. Elsevier, Urban & Fischer, München, 19th edition, 2005. 436 pages. ISBN: 978-3-437-42381-9.

[53] T. Ohara, M. Koga, N. Tokuda, E. Tanaka, H. Yokoyama, K. Minatoya, K. Nagatsuka, K. Toyoda, and K. Minematsu. Rapid Identification of Type A Aortic Dissection as a Cause of Acute Ischemic Stroke. *Journal of Stroke and Cerebrovascular Diseases: The Official Journal of National Stroke Association*, 25(8):1901–1906, 2016. DOI: `10.1016/j.jstrokecerebrovasdis.2016.04.005`.

[54] R. Horne. Patients' interpretation of symptoms as a cause of delay in reaching hospital during acute myocardial infarction. *Heart*, 83(4):388–393, 2000. ISSN: 00070769. DOI: `10.1136/heart.83.4.388`.

[55] S. S. Rathore, A. R. Hinn, L. S. Cooper, H. A. Tyroler, and W. D. Rosamond. Characterization of Incident Stroke Signs and Symptoms: Findings From the Atherosclerosis Risk in Communities Study. *Stroke*, 33(11):2718–2721, 2002. ISSN: 0039-2499, 1524-4628. DOI: `10.1161/01.STR.0000035286.87503.31`.

[56] J. Ritz and H. Buhr. Akute mesenteriale ischämie. *Der Chirurg*, 82(10):863–870, 2011. ISSN: 0009-4722, 1433-0385. DOI: `10.1007/s00104-011-2098-4`.

[57] T. Sommer, W. Fehske, N. Holzknecht, A. V. Smekal, E. Keller, G. Lutterbey, B. Kreft, C. Kuhl, J. Gieseke, D. Abu-Ramadan, and H. Schild. Aortic dissection: a comparative study of diagnosis with spiral CT, multiplanar transesophageal echocardiography, and MR imaging. *Radiology*, 199(2):347–352, 1996. DOI: `10.1148/radiology.199.2.8668776`.

[58] C. R. Wilbers, C. L. Carrol, and M. A. Hnilica. Optimal diagnostic imaging of aortic dissection. *Texas Heart Institute journal*, 17(4):271–278, 1990. ISSN: 0730-2347.

[59] J. E. Cigarroa, E. M. Isselbacher, R. W. DeSanctis, and K. A. Eagle. Diagnostic Imaging in the Evaluation of Suspected Aortic Dissection – Old Standards and New Directions. *New England Journal of Medicine*, 328(1):35–43, 1993. DOI: `10.1056/nejm199301073280107`.

[60] G. R. Caputo and C. B. Higgins. Advances in cardiac imaging modalities. Fast computed tomography, magnetic resonance imaging, and positron emission tomography. *Investigative radiology*, 25(7):838–854, 1990. ISSN: 0020-9996.

[61] S. Hamada, M. Takamiya, K. Kimura, S. Imakita, N. Nakajima, and H. Naito. Type A aortic dissection: evaluation with ultrafast CT. *Radiology*, 183:155–158, 1992. ISSN: 0033-8419. DOI: `10.1148/radiology.183.1.1549663`.

[62] C. A. Nienaber, R. P. Spielmann, Y. von Kodolitsch, V. Siglow, A. Piepho, T. Jaup, V. Nicolas, P. Weber, H. J. Triebel, and W. Bleifeld. Diagnosis of thoracic aortic dissection. Magnetic resonance imaging versus transesophageal echocardiography. *Circulation*, 85(2):434–447, 1992. DOI: `10.1161/01.cir.85.2.434`.

[63] R. S. Ballal, N. C. Nanda, R. Gatewood, B. D'Arcy, T. E. Samdarshi, W. L. Holman, J. K. Kirklin, and A. D. Pacifico. Usefulness of transesophageal echocardiography in assessment of aortic dissection. *Circulation*, 84(5):1903–1914, 1991. ISSN: 0009-7322.

[64] C. A. Nienaber, Y. von Kodolitsch, V. Nicolas, V. Siglow, A. Piepho, C. Brockhoff, D. H. Koschyk, and R. P. Spielmann. The Diagnosis of Thoracic Aortic Dissection by Noninvasive Imaging Procedures. *New England Journal of Medicine*, 328(1):1–9, 1993. DOI: 10.1056/nejm199301073280101.

[65] B. J. Holloway, D. Rosewarne, and R. G. Jones. Imaging of thoracic aortic disease. *The British Journal of Radiology*, 84:338–354, 2011. DOI: 10.1259/bjr/30655825.

[66] R. Erbel, W. Daniel, C. Visser, R. Engberding, J. Roelandt, H. Rennollet, and T. E. C. S. G. F. Echocardiography. Echocardiography in diagnosis of aortic dissection. *The Lancet*, 333:457–461, 1989. DOI: 10.1016/s0140-6736(89)91364-0.

[67] A. Hagendorff, K. Tiemann, G. Simonis, M. C. dell‘ Orto, and S. von Bardeleben. Empfehlungen zur Notfallechokardiographie. *Der Kardiologe*, 8(1):45–64, 2013. DOI: 10.1007/s12181-013-0531-2.

[68] J. T. Sherwood and I. S. Gill. Missed acute ascending aortic dissection. *Journal of cardiac surgery*, 16(1):86–88, 2001. ISSN: 0886-0440.

[69] L. G. Svensson, S. B. Labib, A. C. Eisenhauer, and J. R. Butterly. Intimal tear without hematoma: an important variant of aortic dissection that can elude current imaging techniques. *Circulation*, 99(10):1331–1336, 1999. ISSN: 0009-7322. pmid: 10077517.

[70] F. Alsous, A. Islam, A. Ezeldin, and S. Zarich. Potential pitfalls in the diagnosis of aortic dissection. *Connecticut Medicine*, 67(3):131–134, 2003. ISSN: 0010-6178. pmid: 12687786.

[71] K. Chandrasekaran, W. D. Edwards, J. F. Malouf, G. Gillman, J. K. Oh, L. A. Pape, W. F. Armstrong, and R. C. Bansal. Role of Echocardiography in the Diagnosis of Aortic Dissection. In *Aortic Dissection and Related Syndromes*, pages 61–85. Springer US, 2007. DOI: 10.1007/978-0-387-36001-0_4.

[72] L. F. Hiratzka, G. L. Bakris, J. A. Beckman, R. M. Bersin, V. F. Carr, D. E. Casey, K. A. Eagle, L. K. Hermann, E. M. Isselbacher, E. A. Kazerooni, N. T. Kouchoukos, B. W. Lytle, D. M. Milewicz, D. L. Reich, S. Sen, J. A. Shinn, L. G. Svensson, and D. M. Williams. 2010 ACCF/AHA/AATS/ACR/ASA/SCA/SCAI/SIR/STS/SVM Guidelines for the Diagnosis and Management of Patients With Thoracic Aortic Disease: A Report of the American College of Cardiology Foundation/American Heart Association Task Force on Practice Guidelines, American Association for Thoracic Surgery, American College of Radiology, American Stroke Association, Society of Cardiovascular Anesthesiologists, Society for Cardiovascular Angiography and Interventions, Society of Interventional Radiology, Society of Thoracic Surgeons, and Society for Vascular Medicine. *Circulation*, 121(13):266–369, 2010. DOI: 10.1161/cir.0b013e3181d4739e.

[73] R. Erbel, V. Aboyans, C. Boileau, E. Bossone, R. D. Bartolomeo, H. Eggebrecht, A. Evangelista, V. Falk, H. Frank, O. Gaemperli, M. Grabenwöger, A. Haverich, B. Lung, A. J. Manolis, F. Meijboom, C. A. Nienaber, M. Rof, H. Rousseau, U. Sechtem, P. A. Sirnes, R. S. von Allmen, and C. J. M. Vrints. 2014 ESC Guide-

lines on the diagnosis and treatment of aortic diseases. *European Heart Journal*, 35(41):2873–2926, 2014. DOI: `10.1093/eurheartj/ehu281`.

[74] S. A. Mokashi and L. G. Svensson. Guidelines for the management of thoracic aortic disease in 2017. *General Thoracic and Cardiovascular Surgery*, 2017. DOI: `10.1007/s11748-017-0831-8`.

[75] J. Egger, S. Großkopf, and B. Freisleben. Simulation of bifurcated stent grafts to treat abdominal aortic aneurysms (AAA). In M. I. Miga and K. R. Cleary, editors, *SPIE 6509*. Medical Imaging 2007: Visualization and Image-Guided Procedures, 65091N, 2007. DOI: `10.1117/12.709260`.

[76] Y. Chen, C. Luo, and D. Shum-Tim. *Recent Advances in Acute Type a Aortic Dissection*. Bentham Science Publishers, 2018. ISBN: 978-1-68108-089-5.

[77] F. Huang, L. Li, W. Qin, C. Su, L. Wang, L. Xiao, and X. Chen. Partial aortic root remodeling for root reconstruction in patients with acute type A dissection. *Journal of Biomedical Research*, 2016. DOI: `10.7555/jbr.30.20150097`.

[78] J. E. Bavaria, J. J. Appoo, M. S. Makaroun, J. Verter, Z. Yu, and R. S. Mitchell. Endovascular stent grafting versus open surgical repair of descending thoracic aortic aneurysms in low-risk patients: A multicenter comparative trial. *The Journal of Thoracic and Cardiovascular Surgery*, 133:369–377, 2007. ISSN: 0022-5223. DOI: `10.1016/j.jtcvs.2006.07.040`.

[79] D. Cheng, J. Martin, H. Shennib, J. Dunning, C. Muneretto, S. Schueler, L. V. Segesser, P. Sergeant, and M. Turina. Endovascular Aortic Repair Versus Open Surgical Repair for Descending Thoracic Aortic Disease. *Journal of the American College of Cardiology*, 55:986–1001, 2010. ISSN: 0735-1097. DOI: `10.1016/j.jacc.2009.11.047`.

[80] P. Narayan, A. Wong, I. Davies, G. D. Angelini, A. J. Bryan, P. Wilde, and G. J. Murphy. Thoracic endovascular repair versus open surgical repair — which is the more cost-effective intervention for descending thoracic aortic pathologies? *European Journal of Cardio-Throracic Surgery*, 2011. ISSN: 1010-7940. DOI: `10.1016/j.ejcts.2011.01.010`.

[81] R. S. von Allmen and R. M. Greenhalgh. What Can Thoracic Endovascular Aortic Repair Learn from Abdominal Endovascular Aortic Repair? In Robert S. Bonser, Domenico Pagano, Axel Haverich, and Jorge Mascaro, editors, *Controversies in Aortic Dissection and Aneurysmal Disease*, pages 363–375. Springer, 2014. ISBN: 978-1-4471-5621-5. DOI: `10.1007/978-1-4471-5622-2_33`.

[82] S. C. van Beek, A. P. Conijn, M. J. Koelemay, and R. Balm. Editor's Choice – Endovascular Aneurysm Repair Versus Open Repair for Patients with a Ruptured Abdominal Aortic Aneurysm: A Systematic Review and Meta-analysis of Short-term Survival. *European Journal of Vascular and Endovascular Surgery*, 47:593–602, 2014. ISSN: 1078-5884. DOI: `10.1016/j.ejvs.2014.03.003`.

[83] J. L. Prince and J. M. Links. *Medical Imaging Signals and Systems*. Pearson, Boston, 2nd edition, 2015. 519 pages. ISBN: 978-0-13-214518-3.

[84] H. Müller. *Konzeption Und Realisierung von DICOM Schnittstellen Für Ein Teleradiologiesystem.* Diplomica Verlag GmbH, 1997. ISBN: 978-3-8324-0204-4.

[85] DICOM Model of the Real World. URL: `http://dicom.nema.org/dicom/2013/output/chtml/part03/chapter_7.html` (visited on 05/25/2018).

[86] C. E. Kahn, C. P. Langlotz, D. S. Channin, and D. L. Rubin. Informatics in Radiology: An Information Model of the DICOM Standard. *Radiographics*, 31(1):295–304, 2011. ISSN: 0271-5333. DOI: `10.1148/rg.311105085`. pmid: `20980665`.

[87] 3DSlicer, version 4.8.1, 2018.

[88] GDCM, version 2.8.7, 2018.

[89] CinePaint — Open source deep paint software. URL: `http://www.cinepaint.org/` (visited on 05/25/2018).

[90] GIMP - GNU Image Manipulation Program. URL: `https://www.gimp.org/` (visited on 05/25/2018).

[91] Ginkgo CADx — Ginkgo CADx, Open Core DICOM Viewer + Dicomizer + CADx. URL: `http://ginkgo-cadx.com/en/` (visited on 05/25/2018).

[92] MicroDicom - Free DICOM viewer and software. URL: `http://www.microdicom.com/` (visited on 05/25/2018).

[93] OsiriX DICOM Viewer — The world famous medical imaging viewer. URL: `https://www.osirix-viewer.com/` (visited on 05/25/2018).

[94] InVesalius, version 3.1.1, 2018.

[95] MeVisLab: MeVisLab. URL: `https://www.mevislab.de/` (visited on 05/25/2018).

[96] Dicom in Python. URL: `https://github.com/pydicom` (visited on 05/25/2018).

[97] PixelMed Java DICOM Toolkit — Medical Free/Libre and Open Source Software. URL: `https://www.medfloss.org/node/344` (visited on 05/25/2018).

[98] ITK - Segmentation & Registration Toolkit. URL: `https://itk.org/` (visited on 05/25/2018).

[99] Elastix. URL: `http://elastix.isi.uu.nl/` (visited on 05/25/2018).

[100] G. H. Barrow and J. M. Tenenbaum. Recovering Intrinsic Scene Characteristics from Images. *Computer Vision Systems*:3–26, 1978.

[101] Y. G. Leclerc. Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision*, 3(1):73–102, 1989. ISSN: 0920-5691, 1573-1405. DOI: `10.1007/BF00054839`.

[102] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision.* Addison-Wesley Pub. Co, Reading, Mass, 1992. 2 pages. ISBN: 978-0-201-10877-4.

[103] L. G. Shapiro and G. C. Stockman. *Computer Vision.* Prentice Hall, Upper Saddle River, NJ, 2001. 580 pages. ISBN: 978-0-13-030796-5.

[104] R. Szeliski. *Computer Vision: Algorithms and Applications.* Texts in computer science. Springer, London ; New York, 2011. 812 pages. ISBN: 978-1-84882-934-3.

[105] A. Salem, A. Alaa, S. Alaa, and S. Marwa. *Computer Vision: Concepts, Methodologies, Tools, and Applications*. Critical exploration. IGI, Global, Hershey, PA, 2018. 4 pages. ISBN: 978-1-5225-5204-8.

[106] Y. Zhang, editor. Advances in Image and Video Segmentation. OCLC: ocm67383342. IRM Press, Hershey, PA, 2006. 457 pages. ISBN: 978-1-59140-753-9.

[107] F. Y. Shih and S. Cheng. Automatic seeded region growing for color image segmentation. *Image and Vision Computing*, 23(10):877–886, 2005. ISSN: 02628856. DOI: 10.1016/j.imavis.2005.05.015.

[108] N. Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979. ISSN: 0018-9472, 2168-2909. DOI: 10.1109/TSMC.1979.4310076.

[109] A. Wong and P. Sahoo. A gray-level threshold selection method based on maximum entropy principle. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(4):866–871, 1989. ISSN: 00189472. DOI: 10.1109/21.35351.

[110] N. R. Pal and S. K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26(9):1277–1294, 1993. ISSN: 00313203. DOI: 10.1016/0031-3203(93)90135-J.

[111] H.D. Cheng, C.H. Chen, H.H. Chiu, and Huijuan Xu. Fuzzy homogeneity approach to multilevel thresholding. *IEEE Transactions on Image Processing*, 7(7):1084–1086, 1998. ISSN: 10577149. DOI: 10.1109/83.701171.

[112] P. Saha and J. Udupa. Optimum image thresholding via class uncertainty and region homogeneity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(7):689–706, 2001. ISSN: 01628828. DOI: 10.1109/34.935844.

[113] B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146, 2004. ISSN: 1017-9909. DOI: 10.1117/1.1631315.

[114] K. Batenburg and J. Sijbers. Adaptive thresholding of tomograms by projection distance minimization. *Pattern Recognition*, 42(10):2297–2305, 2009. ISSN: 00313203. DOI: 10.1016/j.patcog.2008.11.027.

[115] K. Batenburg and J. Sijbers. Optimal Threshold Selection for Tomogram Segmentation by Projection Distance Minimization. *IEEE Transactions on Medical Imaging*, 28(5):676–686, 2009. ISSN: 0278-0062, 1558-254X. DOI: 10.1109/TMI.2008.2010437.

[116] A. Martelli. An application of heuristic search methods to edge and contour detection. *Communications of the ACM*, 19(2):73–83, 1976. ISSN: 00010782. DOI: 10.1145/359997.360004.

[117] V. S. Nalwa and E. Pauchon. Edgel aggregation and edge description. *Computer Vision, Graphics, and Image Processing*, 40(1):79–94, 1987. ISSN: 0734189X. DOI: 10.1016/0734-189X(87)90057-0.

[118] J. Basak, B. Chanda, and D. Dutta Majumder. On edge and line linking with connectionist models. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(3):413–428, 1994. ISSN: 00189472. DOI: 10.1109/21.278991.

[119] F. Y. Shih and S. Cheng. Adaptive mathematical morphology for edge linking. *Information Sciences*, 167(1-4):9–21, 2004. ISSN: 00200255. DOI: 10.1016/j.ins.2003.07.020.

[120] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Pearson, New York, NY, fourth edition, global edition edition, 2018. 1168 pages. ISBN: 978-1-292-22304-9.

[121] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988. ISSN: 0920-5691, 1573-1405. DOI: 10.1007/BF00133570.

[122] L. Wang, G. Chen, D. Shi, Y. Chang, S. Chan, J. Pu, and X. Yang. Active contours driven by edge entropy fitting energy for image segmentation. *Signal Processing*, 149:27–35, 2018. ISSN: 01651684. DOI: 10.1016/j.sigpro.2018.02.025.

[123] S. W. Zucker. Region growing: Childhood and adolescence. *Computer Graphics and Image Processing*, 5(3):382–399, 1976. ISSN: 0146664X. DOI: 10.1016/S0146-664X(76)80014-7.

[124] A. Tremeau and N. Borel. A region growing and merging algorithm to color segmentation. *Pattern Recognition*, 30(7):1191–1203, 1997. ISSN: 00313203. DOI: 10.1016/S0031-3203(96)00147-1.

[125] N. Ikonomatakis, K. Plataniotis, M. Zervakis, and A. Venetsanopoulos. Region growing and region merging image segmentation. In *DSP 97*. 13th International Conference on Digital Signal Processing Proceedings, volume 1, pages 299–302, Santorini, Greece. IEEE, 1997. ISBN: 978-0-7803-4137-1. DOI: 10.1109/ICDSP.1997.628077.

[126] S. Hojjatoleslami and J. Kittler. Region growing: a new approach. *IEEE Transactions on Image Processing*, 7(7):1079–1084, 1998. ISSN: 10577149. DOI: 10.1109/83.701170.

[127] J. Haddon and J. Boyce. Image segmentation by unifying region and boundary information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):929–948, 1990. ISSN: 01628828. DOI: 10.1109/34.58867.

[128] C. Chu and J. Aggarwal. The integration of image segmentation maps using region and edge information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1241–1252, 1993. ISSN: 01628828. DOI: 10.1109/34.250843.

[129] Y. Deng and B. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):800–810, 2001. ISSN: 01628828. DOI: 10.1109/34.946985.

[130] T. Pavlidis and Y. Liow. Integrating region growing and edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):225–233, 1990. ISSN: 01628828. DOI: 10.1109/34.49050.

[131] J. Fan, D. Yau, A. Elmagarmid, and W. Aref. Automatic image segmentation by integrating color-edge extraction and seeded region growing. *IEEE Transactions on Image Processing*, 10(10):1454–1466, 2001. ISSN: 10577149. DOI: 10.1109/83.951532.

[132] R. Adams and L. Bischof. Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–647, 1994. ISSN: 01628828. DOI: 10.1109/34.295913.

[133] A. Mehnert and P. Jackway. An improved seeded region growing algorithm. *Pattern Recognition Letters*, 18(10):1065–1071, 1997. ISSN: 01678655. DOI: 10.1016/S0167-8655(97)00131-1.

[134] P. Salembier. Morphological multiscale segmentation for image coding. *Signal Processing*, 38(3):359–386, 1994. ISSN: 01651684. DOI: 10.1016/0165-1684(94)90155-4.

[135] S. Beucher. Watersheds of functions and picture segmentation. In *I.C.A.S.S.P 1982 (IEEE International Conference on Acoustics, Speech, and Signal Processing)*, volume 7, pages 1928–1931. Institute of Electrical and Electronics Engineers, 1982. DOI: 10.1109/ICASSP.1982.1171424.

[136] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991. ISSN: 01628828. DOI: 10.1109/34.87344.

[137] L. Najman and M. Schmitt. Geodesic saliency of watershed contours and hierarchical segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1163–1173, 1996. ISSN: 01628828. DOI: 10.1109/34.546254.

[138] W. Wu, A. Y. C. Chen, L. Zhao, and J. J. Corso. Brain tumor detection and segmentation in a CRF (conditional random fields) framework with pixel-pairwise affinity and superpixel-level features. *International Journal of Computer Assisted Radiology and Surgery*, 9(2):241–253, 2014. ISSN: 1861-6410, 1861-6429. DOI: 10.1007/s11548-013-0922-7.

[139] A. Mostafa, A. E. Hassanien, M. Houseni, and H. Hefny. Liver segmentation in MRI images based on whale optimization algorithm. *Multimedia Tools and Applications*, 76(23):24931–24954, 2017. ISSN: 1380-7501, 1573-7721. DOI: 10.1007/s11042-017-4638-5.

[140] S. Kamalakannan, A. Gururajan, H. Sari-Sarraf, R. Long, and S. Antani. Double-Edge Detection of Radiographic Lumbar Vertebrae Images Using Pressurized Open DGVF Snakes. *IEEE Transactions on Biomedical Engineering*, 57(6):1325–1334, 2010. ISSN: 0018-9294. DOI: 10.1109/TBME.2010.2040082.

[141] J. Pont-Tuset, P. Arbelaez, J. T.Barron, F. Marques, and J. Malik. Multiscale Combinatorial Grouping for Image Segmentation and Object Proposal Generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(1):128–140, 2017. ISSN: 0162-8828, 2160-9292. DOI: 10.1109/TPAMI.2016.2537320.

[142] K. Maninis, J. Pont-Tuset, P. Arbelaez, and L. Van Gool. Convolutional Oriented Boundaries: From Image Segmentation to High-Level Tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):819–833, 2018. ISSN: 0162-8828, 2160-9292. DOI: `10.1109/TPAMI.2017.2700300`.

[143] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018. ISSN: 0162-8828, 2160-9292. DOI: `10.1109/TPAMI.2017.2699184`.

[144] D. Marmanis, K. Schindler, J. Wegner, S. Galliani, M. Datcu, and U. Stilla. Classification with an edge: Improving semantic image segmentation with boundary detection. *ISPRS Journal of Photogrammetry and Remote Sensing*, 135:158–172, 2018. ISSN: 09242716. DOI: `10.1016/j.isprsjprs.2017.11.009`.

[145] T. McInerney and D. Terzopoulos. Medical Image Segmentation Using Topologically Adaptable Snakes. In Nicholas Ayache, editor. Redacted by Gerhand Goos, Juris Hartmanis, and Jan van Leeuwen, *Computer Vision, Virtual Reality and Robotics in Medicine*. Volume 905, pages 92–101. Springer Berlin Heidelberg, Berlin, Heidelberg, 1995. ISBN: 978-3-540-59120-7. DOI: `10.1007/978-3-540-49197-2_11`.

[146] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 1(2):91–108, 1996. ISSN: 13618415. DOI: `10.1016/S1361-8415(96)80007-7`.

[147] M. Leventon, W. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000.* IEEE Conference on Computer Vision and Pattern Recognition, V_3_1–V_3_8. IEEE, 2002. ISBN: 978-0-7803-7507-9. DOI: `10.1109/SSBI.2002.1233989`.

[148] V. Grau, A. Mewes, M. Alcaniz, R. Kikinis, and S. Warfield. Improved Watershed Transform for Medical Image Segmentation Using Prior Information. *IEEE Transactions on Medical Imaging*, 23(4):447–458, 2004. ISSN: 0278-0062. DOI: `10.1109/TMI.2004.824224`.

[149] Y. Zhang, B. J. Matuszewski, L. Shark, and C. J. Moore. Medical Image Segmentation Using New Hybrid Level-Set Method. In *2008 Fifth International Conference BioMedical Visualization: Information Visualization in Medical and Biomedical Informatics*, pages 71–76. IEEE, 2008. ISBN: 978-0-7695-3284-4. DOI: `10.1109/MediVis.2008.12`.

[150] K. Chunhapongpipat, R. Boonklurb, S. Sirisup, and R. Lipikorn. Initial contour independent level set image segmentation method using synergetic vector flow fields. In *2012 6th International Conference on Signal Processing and Communication Systems*, pages 1–7. IEEE, 2012. ISBN: 978-1-4673-2393-2. DOI: `10.1109/ICSPCS.2012.6507987`.

[151] E. Dimas and D. Briassoulis. 3D geometric modelling based on NURBS: a review. *Advances in Engineering Software*, 30(9-11):741–751, 1999. ISSN: 09659978. DOI: 10.1016/S0965-9978(98)00110-0.

[152] E. Angel and D. Shreiner. *Interactive Computer Graphics: A Top-down Approach with Shader-Based OpenGL*. Addison-Wesley, Boston, 6th ed edition, 2012. 730 pages. ISBN: 978-0-13-254523-5.

[153] B. G. Baumgart. A polyhedron representation for computer vision. In page 589. ACM Press, 1975. DOI: 10.1145/1499949.1500071.

[154] oxforddictionaries.com. Centerline. 2018. URL: https://en.oxforddictionaries.com/definition/us/centerline (visited on 05/04/2018).

[155] G. Tetteh, V. Efremov, N. D. Forkert, M. Schneider, J. Kirschke, B. Weber, C. Zimmer, M. Piraud, and B. H. Menze. DeepVesselNet: Vessel Segmentation, Centerline Prediction, and Bifurcation Detection in 3-D Angiographic Volumes, 2018. arXiv: 1803.09340 [cs].

[156] S. Ferchichi and S. Wang. A Clustering-based Algorithm for Extracting the Centerlines of 2D and 3D Objects. In *Proceedings of the 18th International Conference on Pattern Recognition - Volume 02*, ICPR '06, pages 296–299, Washington, DC, USA. IEEE Computer Society, 2006. ISBN: 978-0-7695-2521-1. DOI: 10.1109/ICPR.2006.44.

[157] Q. Zhang and I. Couloigner. Accurate Centerline Detection and Line Width Estimation of Thick Lines Using the Radon Transform. *IEEE Transactions on Image Processing*, 16(2):310–316, 2007. ISSN: 1057-7149. DOI: 10.1109/TIP.2006.887731.

[158] S. Soleimanifard, M. Schär, A. G. Hays, R. G. Weiss, M. Stuber, and J. L. Prince. Vessel centerline tracking and boundary segmentation in coronary MRA with minimal manual interaction. *Proceedings. IEEE International Symposium on Biomedical Imaging*:1417–1420, 2012. ISSN: 1945-7928. DOI: 10.1109/ISBI.2012.6235834. pmid: 23032185.

[159] A. Sironi, V. Lepetit, and P. Fua. Multiscale Centerline Detection by Learning a Scale-Space Distance Transform. In pages 2697–2704. IEEE, 2014. ISBN: 978-1-4799-5118-5. DOI: 10.1109/CVPR.2014.351.

[160] G. Cheng, F. Zhu, S. Xiang, and C. Pan. Accurate Urban Road Centerline Extraction from VHR Imagery via Multiscale Segmentation and Tensor Voting, 2015. arXiv: 1508.06163 [cs].

[161] S. Großkopf, Z. Mostarkic, J. Egger, and B. Freisleben. A Fast Vessel Centerline Extraction Algorithm for Catheter Simulation. In *Twentieth IEEE International Symposium on Computer-Based Medical Systems (CBMS'07)(CBMS)*, pages 177–182, 2007. ISBN: 978-0-7695-2905-9. DOI: 10.1109/CBMS.2007.5.

[162] E. W. Dijkstra. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

[163] R. Ramadan and B. Hendradjaya. Development of game testing method for measuring game quality. In pages 1–6. IEEE, 2014. ISBN: 978-1-4799-8175-5. DOI: 10.1109/ICODSE.2014.7062694.

[164] J. P. Cavano and J. A. McCall. A framework for the measurement of software quality. *ACM SIGSOFT Software Engineering Notes*, 3(5):133–139, 1978. ISSN: 01635948. DOI: 10.1145/953579.811113.

[165] *1061-1992 IEEE Standard for a Software Quality Metrics Methodology.* IEEE, 1993. OCLC: 958658240.

[166] ISO 25000 STANDARDS. URL: http://iso25000.com/index.php/en/iso-25000-standards (visited on 05/16/2018).

[167] T. Fullerton, C. Swain, and S. Hoffman. *Game Design Workshop: A Playcentric Approach to Creating Innovative Games.* Elsevier Morgan Kaufmann, Amsterdam ; Boston, 2nd ed edition, 2008. 470 pages. ISBN: 978-0-240-80974-8. OCLC: ocn173659700.

[168] H. M. Chandler. *The Game Production Handbook.* Jones & Bartlett Learning, Burlington, MA, third edition edition, 2014. 462 pages. ISBN: 978-1-4496-8809-7.

[169] J. Novak. *Game Development Essentials.* Delmar Cengage Learning, Clifton Park, NY, third edition edition, 2012. 483 pages. ISBN: 978-1-111-30765-3.

[170] N. Lavroff. *Virtual Reality Playhouse.* Waite Group Press ; Distributed by Publishers Group West, Corte Madera, Calif. : Emeryville, CA, 1992. 143 pages. ISBN: 978-1-878739-19-3.

[171] W. Robinett. Synthetic Experience:A Proposed Taxonomy. *Presence: Teleoperators and Virtual Environments*, 1(2):229–247, 1992. ISSN: 1054-7460, 1531-3263. DOI: 10.1162/pres.1992.1.2.229.

[172] D. Zeltzer. Autonomy, Interaction, and Presence. *Presence: Teleoperators and Virtual Environments*, 1(1):127–132, 1992. ISSN: 1054-7460, 1531-3263. DOI: 10.1162/pres.1992.1.1.127.

[173] W. Barfield and T. A. Furness, editors. Virtual Environments and Advanced Interface Design. Oxford University Press, New York, 1995. 580 pages. ISBN: 978-0-19-507555-7.

[174] E. L. Waterworth and J. A. Waterworth. Focus, Locus, and Sensus: The Three Dimensions of Virtual Experience. *CyberPsychology & Behavior*, 4(2):203–213, 2001. ISSN: 1094-9313, 1557-8364. DOI: 10.1089/109493101300117893.

[175] D. A. Bowman and L. F. Hodges. Formalizing the Design, Evaluation, and Application of Interaction Techniques for Immersive Virtual Environments. *Journal of Visual Languages & Computing*, 10(1):37–53, 1999. ISSN: 1045926X. DOI: 10.1006/jvlc.1998.0111.

[176] D. A. Bowman and R. P. McMahan. Virtual Reality: How Much Immersion Is Enough? *Computer*, 40(7):36–43, 2007. ISSN: 0018-9162. DOI: 10.1109/MC.2007.257.

[177] M. Slater. Measuring Presence: A Response to the Witmer and Singer Presence Questionnaire. *Presence: Teleoperators and Virtual Environments*, 8(5):560–565, 1999. ISSN: 1054-7460, 1531-3263. DOI: 10.1162/105474699566477.

[178] R. L. Gregory. *Eye and Brain: The Psychology of Seeing*. Princeton University Press, Princeton, N.J, 5th ed edition, 1997. 277 pages. ISBN: 978-0-691-04840-6.

[179] L. W. Stark. Representation of human vision in the brain: How does human perception recognize images? *Journal of Electronic Imaging*, 10(1):123, 2001. ISSN: 1017-9909. DOI: 10.1117/1.1329895.

[180] C. Heeter. Being There: The Subjective Experience of Presence. *Presence: Teleoperators and Virtual Environments*, 1(2):262–271, 1992. ISSN: 1054-7460, 1531-3263. DOI: 10.1162/pres.1992.1.2.262.

[181] M. Slater and M. Usoh. Presence in immersive virtual environments. In pages 90–96. IEEE, 1993. ISBN: 978-0-7803-1363-7. DOI: 10.1109/VRAIS.1993.380793.

[182] H. G. Hoffman, J. Prothero, M. J. Wells, and J. Groen. Virtual Chess: Meaning Enhances Users' Sense of Presence in Virtual Environments. *International Journal of Human-Computer Interaction*, 10(3):251–263, 1998. ISSN: 1044-7318, 1532-7590. DOI: 10.1207/s15327590ijhc1003_3.

[183] K. Bystrom, W. Barfield, and C. Hendrix. A Conceptual Model of the Sense of Presence in Virtual Environments. *Presence: Teleoperators and Virtual Environments*, 8(2):241–244, 1999. ISSN: 1054-7460, 1531-3263. DOI: 10.1162/105474699566107.

[184] M. J. Schuemie, P. van der Straaten, M. Krijn, and C. A. van der Mast. Research on Presence in Virtual Reality: A Survey. *CyberPsychology & Behavior*, 4(2):183–201, 2001. ISSN: 1094-9313, 1557-8364. DOI: 10.1089/109493101300117884.

[185] W. A. Ijsselsteijn, J. Freeman, and H. De Ridder. Presence: Where Are We? *CyberPsychology & Behavior*, 4(2):179–182, 2001. ISSN: 1094-9313, 1557-8364. DOI: 10.1089/109493101300117875.

[186] M. Slater. Presence and The Sixth Sense. *Presence: Teleoperators and Virtual Environments*, 11(4):435–439, 2002. ISSN: 1054-7460, 1531-3263. DOI: 10.1162/105474602760204327.

[187] M. V. Sanchez-Vives and M. Slater. From presence to consciousness through virtual reality. *Nature Reviews Neuroscience*, 6(4):332–339, 2005. ISSN: 1471-003X, 1471-0048. DOI: 10.1038/nrn1651.

[188] W. Barfield and C. Hendrix. The effect of update rate on the sense of presence within virtual environments. *Virtual Reality*, 1(1):3–15, 1995. ISSN: 1359-4338, 1434-9957. DOI: 10.1007/BF02009709.

[189] B. G. Witmer and M. J. Singer. Measuring Presence in Virtual Environments: A Presence Questionnaire. *Presence: Teleoperators and Virtual Environments*, 7(3):225–240, 1998. ISSN: 1054-7460, 1531-3263. DOI: 10.1162/105474698565686.

[190] M. Meehan, B. Insko, M. Whitton, and F. P. Brooks. Physiological measures of presence in stressful virtual environments. *ACM Transactions on Graphics*, 21(3), 2002. ISSN: 07300301. DOI: 10.1145/566654.566630.

[191] R. Pausch, D. Proffitt, and G. Williams. Quantifying immersion in virtual reality. In pages 13–18. ACM Press, 1997. ISBN: 978-0-89791-896-1. DOI: 10.1145/258734.258744.

[192] Blender Foundation. Blender.org - Home of the Blender project - Free and Open 3D Creation Software. URL: https://www.blender.org/ (visited on 06/05/2018).

[193] GIMP - GNU Image Manipulation Program. URL: https://www.gimp.org/ (visited on 06/05/2018).

[194] Unity Technologies. Unity 2017.3, version 2017.3.0f3, 2018.

[195] J. Haas. A History of the Unity Game Engine. *Worcester, UK: Worcester Polytechnic Institute*, 2014.

[196] Unity Technologies. Unity Technologies Lands $12 Million in Series B Funding Led by WestSummit Capital and iGlobe Partners. 2011. URL: http://www.marketwired.com/press-release/unity-technologies-lands-12-million-series-b-%20%20funding-led-westsummit-capital-iglobe-partners-1540593.htm (visited on 05/04/2018).

[197] Unity Technologies. Unity - Fast Facts. URL: https://unity3d.com/public-relations (visited on 05/04/2018).

[198] OsiriX DICOM Viewer — DICOM Image Library. URL: https://www.osirix-viewer.com/resources/dicom-image-library/ (visited on 05/25/2018).

[199] G. Turk and J. F. O'Brien. Shape transformation using variational implicit functions. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 335–342, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co., 1999. ISBN: 0-201-48560-5. DOI: 10.1145/311535.311580.

[200] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10(6):350–355, 1978. ISSN: 00104485. DOI: 10.1016/0010-4485(78)90110-0.

[201] Laplacian Smooth Modifier — Blender Manual. URL: https://docs.blender.org/manual/en/dev/modeling/modifiers/deform/laplacian_smooth.html (visited on 05/30/2018).

[202] S. Hossain and S. Serikawa. Texture databases – A comprehensive survey. *Pattern Recognition Letters*. Smart Approaches for Human Action Recognition, 34(15):2007–2022, 2013. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2013.02.009.

[203] F. Post and T. Walsum. Fluid Flow Visualization. In pages 1–40, 1991. DOI: 10.1007/978-3-642-77165-1_1.

[204] B. K. Bharadvaj, R. F. Mabon, and D. P. Giddens. Steady flow in a model of the human carotid bifurcation. Part I—Flow visualization. *Journal of Biomechanics*, 15(5):349–362, 1982. ISSN: 0021-9290, 1873-2380. DOI: 10.1016/0021-9290(82)90057-4.

[205] S. Gaeta, P. Dyverfeldt, J. Eriksson, C. Carlhäll, T. Ebbers, and A. F. Bolger. Fixed volume particle trace emission for the analysis of left atrial blood flow using 4D Flow MRI. *Magnetic Resonance Imaging*, 47:83–88, 2018. ISSN: 0730-725X. DOI: 10.1016/j.mri.2017.12.008.

[206] S. Bonnin, V. Krivosic, E. Cognat, and R. Tadayoni. Visibility of blood flow on optical coherence tomography angiography in a case of branch retinal artery occlusion. URL: http://www.jovr.org/article.asp?issn=2008-322X;year=2018;volume=13;issue=1;spage=75;epage=77;aulast=Bonnin (visited on 05/07/2018).

[207] M. Isorni, S. Hascoet, A. Amato, and R. Roussin. Early insight of 4D flow imaging in patients with congenital heart disease. *Archives of Cardiovascular Diseases Supplements*, 10(1):140, 2018. ISSN: 1878-6480. DOI: 10.1016/j.acvdsp.2017.11.153.

[208] C. D. Malone, A. Banerjee, M. T. Alley, S. S. Vasanawala, A. C. Roberts, and A. Hsiao. Pelvic Blood Flow Predicts Fibroid Volume and Embolic Required for Uterine Fibroid Embolization: A Pilot Study With 4D Flow MR Angiography. *American Journal of Roentgenology*, 210(1):189–200, 2017. ISSN: 0361-803X. DOI: 10.2214/AJR.17.18127.

[209] G. Vikas, B. Mariana, F. Alexandru, C. Carl-Johan, and E. Tino. Improving left ventricular segmentation in four-dimensional flow MRI using intramodality image registration for cardiac blood flow analysis. *Magnetic Resonance in Medicine*, 79(1):554–560, 2017. ISSN: 0740-3194. DOI: 10.1002/mrm.26674.

[210] P. J. Wasicek, K. Shanmuganathan, W. A. Teeter, W. B. Gamble, P. Hu, D. M. Stein, T. M. Scalea, and M. L. Brenner. Assessment of Blood Flow Patterns Distal to Aortic Occlusion Using CT in Patients with Resuscitative Endovascular Balloon Occlusion of the Aorta. *Journal of the American College of Surgeons*, 226(3):294–308, 2018. ISSN: 1072-7515, 1879-1190. DOI: 10.1016/j.jamcollsurg.2017.12.005. pmid: 29248608.

[211] J. D. Briers and A. F. Fercher. Retinal blood-flow visualization by means of laser speckle photography. *Investigative Ophthalmology & Visual Science*, 22(2):255–259, 1982. ISSN: 1552-5783.

[212] S. Liu, P. Li, and Q. Luo. Fast blood flow visualization of high-resolution laser speckle imaging data using graphics processing unit. *Optics Express*, 16(19):14321–14329, 2008. ISSN: 1094-4087. DOI: 10.1364/OE.16.014321.

[213] J. Kim, N. L. Phuong, S. Aramaki, and K. Ito. Flow visualization through particle image velocimetry in realistic model of rhesus monkey's upper airway. *Respiratory Physiology & Neurobiology*, 251:16–27, 2018. ISSN: 1569-9048. DOI: 10.1016/j.resp.2018.02.007.

[214] A. J. Chorin and J. E. Marsden. *A Mathematical Introduction to Fluid Mechanics*. Springer-Verlag, New York, 1993. ISBN: 978-0-387-97918-2. OCLC: 26402878.

[215] E. Gamma. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley professional computing series. Addison-Wesley, Reading, Mass, 1995. 395 pages. ISBN: 978-0-201-63361-0.

[216] Vive™ — vive virtual reality system. URL: https://www.vive.com/us/product/vive-virtual-reality-system/ (visited on 05/16/2018).

[217] Free NVIDIA FCAT VR Benchmarking Application Available Now For Download. URL: http://www.geforce.com/whats-new/guides/fcat-vr-download-and-how-to-guide (visited on 05/17/2018).

[218] F. Chollet. *Deep Learning with Python*. Manning Publications Co, Shelter Island, NY, 2018. 361 pages. ISBN: 978-1-61729-443-3. OCLC: ocn982650571.

[219] D. Nie, R. Trullo, J. Lian, C. Petitjean, S. Ruan, Q. Wang, and D. Shen. Medical Image Synthesis with Context-Aware Generative Adversarial Networks. In *Medical Image Computing and Computer - Assisted Intervention - MICCAI 2017*, Lecture Notes in Computer Science, pages 417–425. Springer, Cham, 2017. ISBN: 978-3-319-66178-0. DOI: 10.1007/978-3-319-66179-7_48.

[220] VIVE Pro — The professional-grade VR headset. URL: https://www.vive.com/eu/product/vive-pro/ (visited on 07/24/2018).

[221] I. Scholl, S. Suder, and S. Schiffer. Direct Volume Rendering in Virtual Reality. In Andreas Maier, Thomas M. Deserno, Heinz Handels, Klaus Hermann Maier-Hein, Christoph Palm, and Thomas Tolxdorff, editors, *Bildverarbeitung Für Die Medizin 2018*, pages 297–302. Springer Berlin Heidelberg, Berlin, Heidelberg, 2018. ISBN: 978-3-662-56536-0. DOI: 10.1007/978-3-662-56537-7_79.