POLITECNICO MILANO 1863

SCHOOL OF DESIGN

LAUREA MAGISTRALE IN DESIGN & ENGINEERING

VIRTUAL REALITY VISUALIZATION OF COMPUTED TOMOGRAPHY FOR PRODUCT DEVELOPMENT PROCESS

AUTHOR: STANISLAW CETKOWSKI

ADVISOR: PROF. GIORGIO COLOMBO

CO-ADVISOR: PhD MARCO ROSSONI

Academic year 2020-2021

Sommario

La tomografia computerizzata (TC) è un processo che utilizza la tecnologia di rilevamento dell'emissione di luce per produrre immagini di scansione in sezione trasversale che mostrano la struttura interna di un oggetto scansionato. È una tecnologia molto utilizzata nel settore della medicina. Le scansioni TC vengono in genere utilizzate come raccolta di molte sezioni 2D. Tradizionalmente, la persona che esamina le scansioni TC deve creare mentalmente una rappresentazione 3D di un oggetto da immagini 2D. Inoltre, per cercare dettagli specifici può essere scomodo esaminare un modello 3D su uno schermo 2D: spostare il modello, ingrandire e rimpicciolire o guardare la struttura interna del modello utilizzando viste in sezione. Il modo più semplice ed efficiente per guardare il modello 3D sembra vederlo in realtà virtuale. Questa tecnologia consente all'utente di esaminare il modello 3D in un ambiente più autoesplicativo, più simile a come lo farebbe qualcuno in un ambiente del mondo reale, usando la mano per ruotare l'oggetto, affettare la parte per vedere cosa c'è al suo interno o il modo in cui avvicini l'oggetto per avere un aspetto migliore. Sebbene l'uso di una collaborazione tra scansioni TC e VR sembri promettente nel settore della medicina, non è possibile trovare molte o nessuna informazione sulla fusione di queste due tecnologie per il processo di sviluppo del design del prodotto o l'ispezione dei dati dell'ingegneria industriale. Questa collaborazione potrebbe avere un futuro altrettanto brillante in questi settori. Può aiutare con l'esame qualitativo della struttura interna delle parti in modo non distruttivo. Questa tesi mostra i possibili casi d'uso della fusione della tecnologia delle scansioni TC con le tecnologie VR per la progettazione e l'ingegneria. Per mostrare questi casi d'uso, è stata sviluppata l'applicazione di realtà virtuale. L'app è stata sviluppata nel software Unity con l'uso aggiuntivo di programmi come Materialise Mimics, 3D Slicer, ImageJ e Meshmixer. Il display hardware montato sulla testa per la visualizzazione dei dati tramite l'applicazione è Oculus Quest 2. Nell'applicazione sono presenti due set di dati CT di case study. Il primo set di dati è un insieme di 1098 sezioni di una struttura reticolare realizzata per i test meccanici di compressione. È stato realizzato dalla produzione additiva SLM in acciaio inossidabile 316L. Il secondo set di dati è un insieme di 216 sezioni di una lampada progettata per la produzione additiva. Questa relazione di tesi mostra e descrive in dettaglio l'applicazione con tutte le sue funzionalità. A parte questo, l'intero processo di sviluppo dell'applicazione, compreso il processo di preparazione dei dati CT per l'applicazione.

Abstract

Computed tomography (CT) is a process that uses the light emission-detection technology to produce cross-sectional scan images showing the internal structure of a scanned object. CT scans typically generate a collection of 2D slices of the object and a 3D model of the object can be derived out of that. However, interpreting specific details of 2D images on a computer screen may be difficult and inconvenient, especially for non-experts without a direct correlation with the original 3D model. Virtual Reality (VR) is a technology that have been explored to visualize CT scans datasets, and their corresponding 3D models, allowing the user to examine the 3D model in a more self-explanatory environment, more similar to how someone would do it in a real-world environment. Despite the fact that Virtual Reality has been deeply investigated in the biomedical field for visualizing CT related data, the same for data coming from industrial CT has not been done yet. This thesis aims at investigating how using VR to visualize product related data, derived from CT scans, can support the product development process. To achieve this objective, a virtual reality application has been developed and tested with different sets of CT scans.

After reviewing the state of the art, the VR application with all its assets is presented, showing in detail how every functionality can help to inspect CT data in order to enhance product development process. After that, the pipeline to prepare the CT data to be visualized in a VR environment is investigated and the development process of the application is detailed. The app has been developed in the *Unity 3D* software with the additional use of programs such as *Materialize Mimics*, *3D Slicer*, *ImageJ* and *Meshmixer* to help with the preparation of the CT data. The application is compiled to work with the *Oculus Quest 2*, a low-cost VR head-mounted display. Two case studies have been used to validate the proposed pipeline. The first dataset refers to a set of 1098 slices of a lattice structure made for compression mechanical tests: to this extent, visualizing the data in a VR environment is helpful for checking internal defects or possible deviation from the desired geometry. The second dataset is about of 216 slices of a lamp designed for additive manufacturing: in this case, the focus is to provide a designer with an intuitive tool for checking internal structures without having to become an expert on reading CT scans.

Table of Contents

1 Introduction

Figure 1. Man wearing VR headset

Computed tomography (called CT later throughout the dissertation) is a process that uses the light emission-detection technology to produce cross-sectional scan images portraying the external and the internal structure of the scanned object. It is a very commonly used technology in the biomedical sector. Nonetheless, its industrial use has also matured, but not in such major scale yet. CT uses the same beams as the X-ray technology but is able to depict soft anatomical structures way more precisely. CT scans are typically used as a collection of many 2D slices. Traditionally, the person reviewing CT scans must create a 3D representation of the object from the 2D images in their mind. Taking into account that human beings are not robots and tend to make mistakes, plus the complicated structure of the object being inspected, even for someone with an excellent spatial intelligence it might become difficult to recreate and there might occur some errors while building a 3D model from 2D images in someone's mind. However, two-dimensional CT slices can also be used to reconstruct a three-dimensional model digitally. Nonetheless, for someone that is not familiar with any 3D modelling software it may be inconvenient to examine the 3D model on a computer screen - to move the model around, zoom in and out or to look at the internal structure of the model using section views. The easier and more efficient way to look at the CT datasets seems to be viewing it in Virtual Reality. This technology allow the user to examine the 3D model in a more self-explanatory environment, more similar to how someone would do it in a real-world environment, by using hand to rotate the object, slice the part to see what is inside of it or the way you bring the object closer to have a better look. The use of a collaboration between CT scans and VR is constantly growingin a biomedical sector. For example, thanks to it surgeons can have the ability to examine patients' organs in a more spatial way and prepare for an operation or look for obstacles they can come across during an operation. Apart from the prospect of using it in medicine sector, this collaboration might have just as bright future in the design and engineering sector, despite the fact that this seems to be a total novelty as not much information about it can be found in the Internet. It can help with examining the internal structure of the parts in a non-destructive way, what can enhance the product development process to a large extent. For instance, it may be a great match for 3D printed objects. With the assist of the VR visualization of CT scans their partly empty internal structure can be thoroughly examined without the need to cut the part open to verify the quality or design features during the prototyping phase or the development phase of the product design process. It can also be a great match for inspecting already developed products during their lifecycle.

The same benefit of using VR could be gained with the datasets from magnetic resonance imagining (MRI) scans. MRI produces similar type of images as CT while using a similar device working principle but different technology. The technology, distinctly from CT, uses the magnetic field and the radio waves. The resulting images coming from CT and MRI are different in some ways and the different method is chosen according to what must be scanned. Generally, for the industrial application the CT technology is used. As the case study for this dissertation is done using the CT scan images let just keep in mind that same visualization methods developed for this project could as well be successfully adapted to MRI scans.

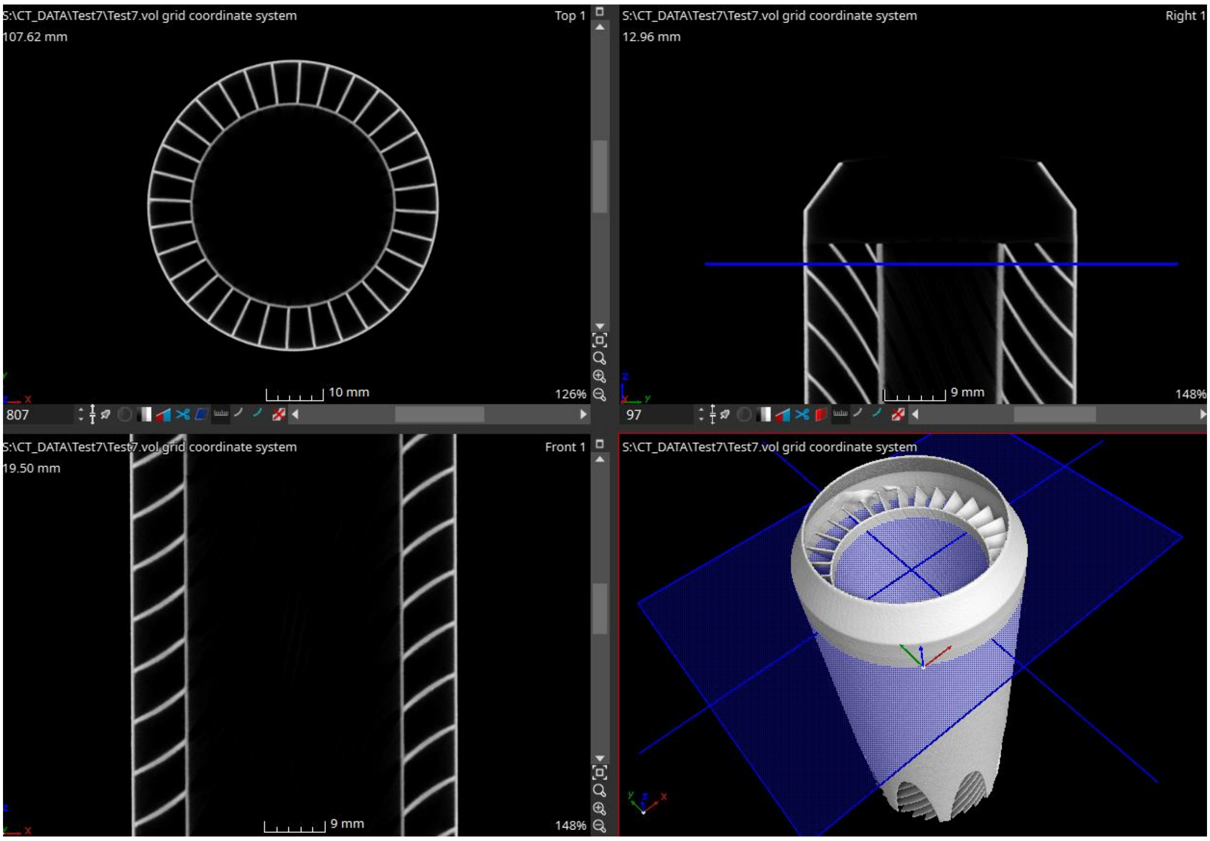

Figure 2. CT scan visualization of 3D printed diesel engine urea mixer on computer screen

As presented before, there are noticeable premises that combining the technologies of VR and CT scans for visualizing the data of the latter can bring brilliant added value to the process of examining the product majorly during the conceptual and development design phases, and additionally during the whole lifecycle to check how the product withstands over the time. The main aim of this thesis is to propose a virtual reality application suited for visualizing and analyzing industrial computed tomography data and only after this to conclude whether such an application introduced an added value to the process of examining CT data for the purpose of enhancing the product development process. The need to develop a completely new application is not a pure invention made up just for the purpose of this thesis, it has the rationale behind it. The issue has its foundation in the fact that the size of an industrial CT scans dataset size is much bigger if compared with a medical dataset. The resolution of a medical CT scan is in the best case about 0.5 mm, while the resolution of an industrial one is in the worst case about 0.1 mm. This is what causes troubles when there is an attempt to extend the approaches used for the medical CT datasets to industrial CT datasets.

The app is developed in *Unity 3D* software with the additional use of programs such as *Materialize Mimics*, *3D Slicer*, *ImageJ* and *Meshmixer* to help with the preparation of the CT data.

The VR head-mounted display (called HMD later throughout the dissertation) that the app is compiled to work with is *Oculus Quest 2.* It is a low-cost device and was chosen because of its common availability. There are two industrial CT datasets used for the case study of this dissertation. The first industrial computed tomography dataset used in the application is a set of 1098 slices of a lattice structure specimen made for compression mechanical tests. It was made from stainless steel 316L and manufactured using the SLS 3D printing method. This case is presented to show how visualizing the data in a VR environment is helpful for checking internal defects or possible deviation from the desired geometry. The second dataset is made of 216 slices of a lamp designed for additive manufacturing. This case, the focus is to provide a designer with an intuitive tool for checking internal structures without having to become an expert on reading CT scans.

The elaborated justification for the used software, hardware and the use of these particular datasets can be found in the chapter *5 Process of creating the app*. It is worth stressing that the application may have brought even more added value if developed also for the augmented reality environment. The constraint that limited the app only to the virtual reality was the use of *Oculus Quest 2*, which is a HMD rather exclusively for VR. More accurate comment about this matter can be found in the chapter *6 Conclusions*.

In short, the dissertation consists of:

- current state of art of VR, AR, CT and of combining preceding fields,
- presentation of the developed VR application with all its assets showing in detail how every functionality can help to inspect CT data in order to enhance product development process,
- detailed description of the process of preparing CT datasets and the process of creating the application,
- conclusions.

2 State of art

2.1 Introduction to CT, VR, AR

As this dissertation is talking about issues associated with the technologies beforenamed in the subtitle above, the short introduction will be given to each of them. It should help the reader understand the case study of the dissertation and the further reflections on it.

2.1.1 Computed tomography

The computed tomography is a technology primarily used in the medicine sector. It is used to produce cross-sectional images, usually of a human body. It was a revolutionary invention. The *Nobel Prize in Physiology or Medicine* in 1979 was awarded jointly to Allan M. Cormack and Godfrey N. Hounsfield "for the development of computer assisted tomography" (1) and the CT has been in use in the medicine sector ever since. Nowadays, CT scans help medical practitioners all around the world in saving people lives and treating their injuries. It uses the same technology as the X-ray technology, but the working principal of a CT device is different. The CT scans can depict more than a regular X-ray image in a more precise way, that is soft tissues, blood vessels but at the same time not excluding bones. For the better quality of acquired scanning datasets, images go through the image segmentation process (2). It accentuates the differences between the separate neighbouring image segments. That makes the image more distinctive for further qualitative as well as quantitative inspecting.

Figure 3. Photo of doctor doing CT scanning on their patient

For the aim of this dissertation, it appears accurate to introduce the basic working principle of a CT scanner. Through a whole scanning operation, a scanned object is being moved across a rotating hoop in a constant direction. In a hoop there is a light source emitter set sending X-rays through a moving object and on the other side of a hoop there are light detectors collecting light beams (3). X-ray beams are being attenuated as they are passing through an object. The amount of attenuation differs depending on the type of tissue that a beam passes through. Therefore, a bone structure is depicted as white and more soft structures are being depicted as darker in the image. This results in a contrast in a scan that can be seen between different anatomical structures of a body. This helps an examining person to distinguish between them.

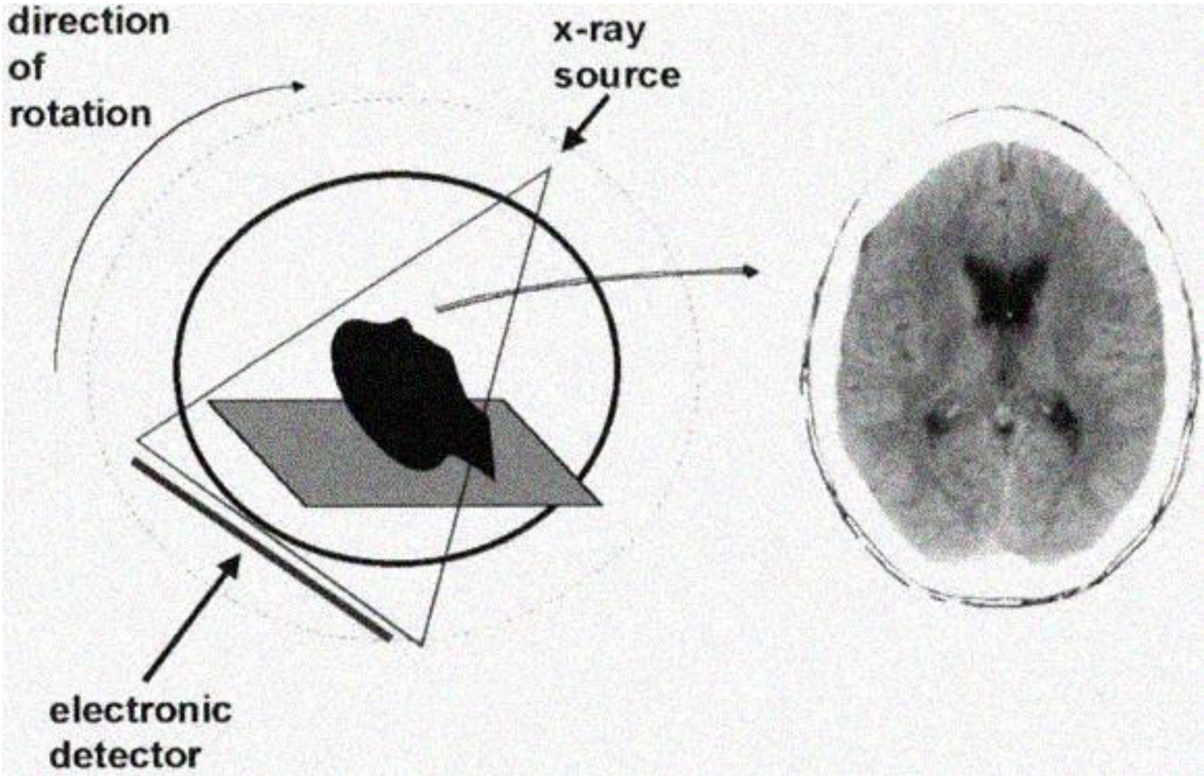

Figure 4. Diagram of CT scanner (3)

Every computed tomography section scan has a finite thickness, thus each pixel of the image represents a voxel – a tiny volume element. The smaller the voxel is, the better the resolution of the image becomes (4). The spatial resolution of current medical CT scanners is about 0.5 mm (5). The resolution of industrial CT scanners is significantly superior as it can reach 0.4 μm (or even lower) (6) (7).

Every medical CT scanner produces a set of 2D images in a normalized format called DICOM. This normalization was made in order to make the exchange and the interpretation of diagnostic images easier for all people working in the medicine sector (8). DICOM data is coded in a binary-text format. Text is used to save data of names, dates, IDs or other text sequences. Binary format is used to save numeric data, for example, pixels. This developed form of saving data enables to virtual trouble-free exchange of information between equipment of different manufactures. What is more, DICOM format is commonly used also for MRI, positron emission tomography (PET) and many other technologies of digital examination that produce high resolution images.

After the data is collected from a CT scanner, the usual way of visualizing it is to show 2D cross-sectional scans on a computer screen. A medical practitioner can control displaying program by a standard computer keyboard and a mouse. The user can determine the depth of the object section, but usually only in three directions. This limitation can be omitted when a 3D model is constructed from the 2D scans. In this case, the user can set a section plane at every angle possible. However, this is not yet a standard procedure. The objective of this work is to explore the possible ways of visualization of CT scans with the help of the virtual reality. The subject of 3D reconstruction of CT scans is broad and will not be introduced as it goes beyond the range of this dissertation.

2.1.2 Virtual Reality and Augmented Reality

It is regarded that the term *Virtual Reality* was created by Jaron Lanier in the early 1990s. The definition of the VR that can be found in the *Cambridge Dictionary* is: *"a set of images and sounds, produced by a computer, that seem to represent a place or a situation that a person can take part in".* This explanation is very plain and general, and it might take away a little bit of the magnitude of that technology. More accurate definition seems to come from Steve Bryson from *NASA*: *"Virtual reality is the use of computer technology to create the effect of an interactive three-dimensional world in which the objects have a sense of spatial presence"*.

Augmented Reality according to *Oxford Languages Dictionary* is: *"a technology that superimposes a computer-generated image on a user's view of the real world, thus providing a composite view*". One can say that the AR is a refined real-world environment that we currently are in. The AR is commonly used in smartphone applications. Social media applications like *Snapchat* or *Instagram* use the AR for their *filters*, which are objects or distortion put over the camera pictures and videos. There are also smartphones games like *Pokemon GO* that use the AR technology.

People who are not very up to date with the latest technologies might find it difficult to differentiate between the virtual and the augmented reality. What is more, there are applications in which the frontier between them is not that apparent. To make it marginally clearer, the AR is a technology that provides interactive moments by enhancing the real-world environment with extra visuals, audio, haptic feel or olfactory sensation while the VR aims to provide experience in a completely distinct world or tries to copy the real-world in a most lifelikeness form.

Although the VR beginnings are dating back to the 1990s it was not until the last few years that this technology's growth has stimulated. According to the *Forbes* magazine, the coronavirus pandemic has been boosting this growth even more (9). It is mainly caused by the companies shifting to remote working. The shift from having a meeting in person to having a meeting via online video applications was smooth and adding the VR technology to meetings might have a bright future. The global spending on VR/AR hardware and software purchased by consumers in 2020 rose to $12 billion, what is 50% more than in 2019. Moreover, it is forecasted that in the retail market the sales will grow at a rate 68.5% in the period from 2020 to 2027.

There are many ways of use VR/AR in the consumer market, so it seems reasonable to bring up at least a few. First and most developed application is for gaming industry. There are many games available for computers and consoles compatible with VR headsets and movement controllers. The player can immerse into the virtual environment to enhance the experience of gaming. Apart from the goggles tracking head movements which let the user feel like they are in the game environment, the movement controllers enforce the specific motions of the user body, all that in order to make the user feel more merged with the virtual world. Examples can be games where the player has a boxing match or play tennis. One of the first consoles using movement sensor was *Nintendo Wii* introduced to the world in 2006. As a mention before, there are also smartphone games that use AR like *Pokemon GO*. These types of games use a smartphone camera and overlay game objects over the real world environment.

Another interesting case is the online shopping. Thanks to the AR the user could see how they would look in a certain piece of apparel without the need of going out of their house. This is not a mature application yet, but the fashion sector companies are working intensively on its development. An app would overlap a piece of clothing 3D model on a body of the user as they are moving and the user could see how it would fit them. The principal of working is similar to *Snapchat* or *Instagram* filters. The coronavirus lockdowns in different parts of the world might have had a significant impact on the path of growth for this as people were not able to go and see the clothes in stores in reality. The *Shopify* research claims that introducing this technology will decrease the companies' return rates by 40% (10). This application does not restrict only to the clothes sector. It can be applied to any other retail sector, e.g. the home furnishing sector. *IKEA Place* is the smartphone application that let you put furniture in your home and see how it would fit the interior before purchasing it online. Summing up, the AR technology use seems to be a tremendous way to check the product combination with the environment or other objects it is meant to be located with.

Last example of the VR/AR application in the customer sector is for education and learning. In the Internet era that we are now in, the most common way of learning new things is being fed information from digital screens or audio voice, often both simultaneously. One can always try to find a teacher of a certain skill in their local area but learning this way is usually more time consuming or more expensive. Plenty of tutorials about almost everything can be found online and very often free, for example on *YouTube* web platform, what is very convenient for a learner. The situation is more complicated if the skill to be learned requires some additional physical object. The example of learning how to play piano illustrates this fairly. Buying a piano can be costly, but with AR you can put a fully function piano in your room and learn how to play without the need of acquiring the instrument. With that being said, with VR/AR the learner can completely participate in the process of learning in a real time, without losing their focus to follow what is being shown on the screen in a conventional way of learning and without the need of communing with physical objects that might not be in their possession.

Apart from the consumer market, the VR/AR technologies have been successfully implemented into the business market. Many sectors have already been using the VR/AR, nevertheless there is still a huge space for launching and improvements. The subject of application in medical sector will be presented more thoroughly later in the paper, so for now let introduce using the VR/AR in other business sectors.

The application that already exists and is predicted to significantly grow is online events and meetings. The AR and the VR are valuable technologies to enhance the nature of online meetings because they provide the ability to stimulate the lifelike experience of the in-person encounters. This is more than only watching the screen as in the conventional online calls (11). As a VR headset price is not that expensive anymore as it was 10 years ago, this gives the chance to attend VR environment meetings by employees working remotely from any part of the globe. Furthermore, it is possible to implement 3D models of objects into the VR environment where people from different part of the world can join in. This generates a great opportunity for the design and engineering sector, giving a better capability to examine a product via an online meeting. All this being said, introducing the VR environments meetings in the business can lead a company to relevant cost savings by eliminating travel and venue costs while also saving time.

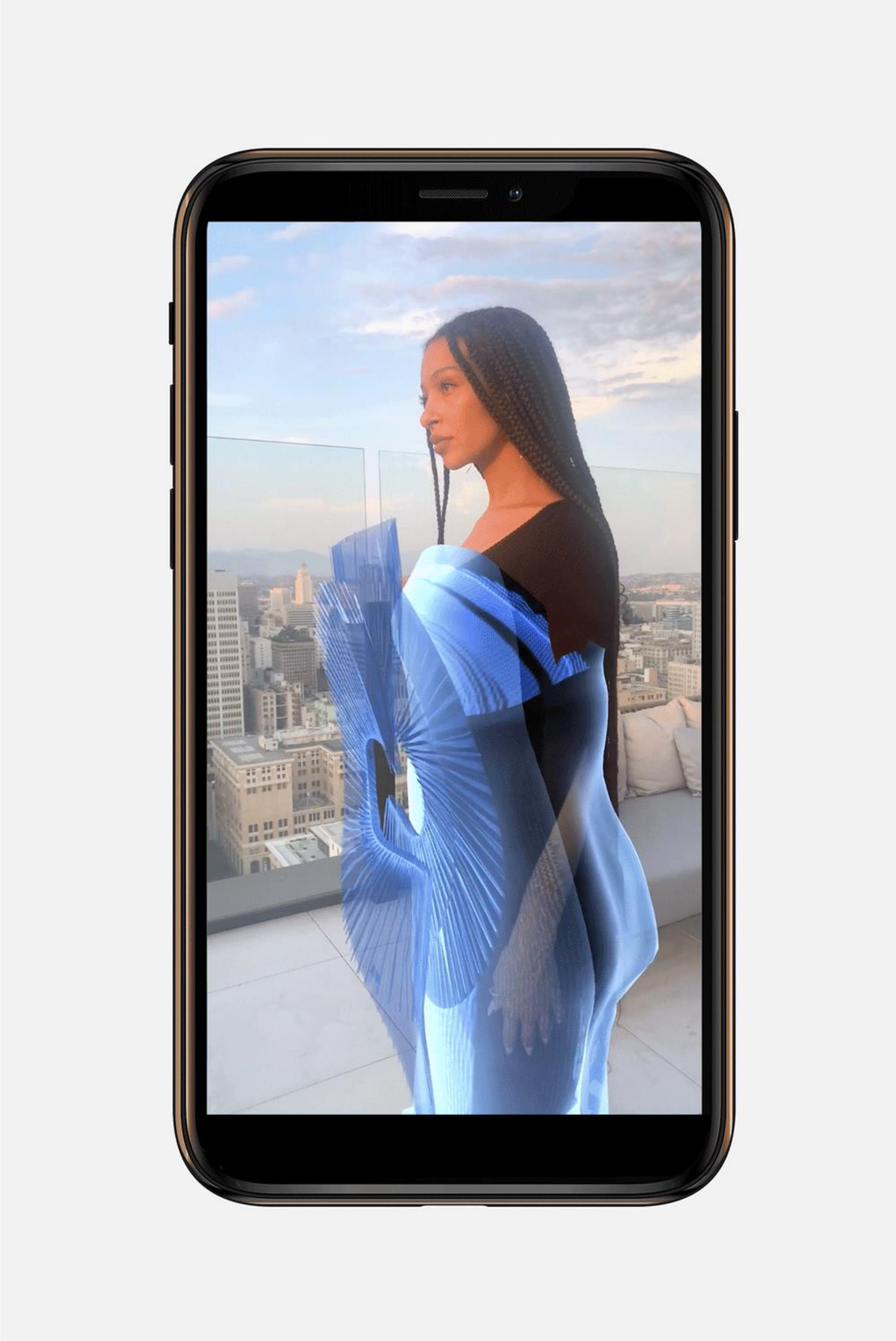

Figure 5. Woman wearing AR dress

Another valuable application of the VR/AR in business is education and courses. The idea and the benefits resemble similar to the ones that were mentioned for the consumer sector, but here the scale is bigger. It can bring more savings and added value to a company. The great example is the airline sector, where pilots have been practising in VR environment since decades (11).

Last but not least, the business case that shall not be ignored is a virtual reality tour. Thanks to the virtual reality we are now able to take a walk in an apartment or a hotel room that we would like to rent. With hotel rooms this might be seen just like an enhancement, but when it comes to renting a room or a flat it is kind of different. It can save a person looking for a place to stay a lot of time that they would spend visiting each accommodation as a VR tour can give much more information than the photographs. Moreover, VR tours give opportunity to visit cultural or tourist attractions from any place of the world in any time. It cannot equal to the impression made by being somewhere in person, but it can be a profitable way of marketing. Visiting a museum in the VR might make someone more curious about the venue and attract them to come there in person.

2.2 VR/AR for visualizing CT

In today's medicine virtual reality or augmented reality can be seen to be used as a supplementary help to the standard way of examining 2D CT scans before the procedure. It is also used to show patients what is inside their body parts or to show them the planned course of the surgery. Whereasmany scientists are working on efficient introduction and good practices of merging these two technologies, there are also clinics that are already using VR to visualize CT scans (12). Despite the fact that the VR and the AR have been deeply investigated in the biomedical field for visualizing CT related data, no trace of using VR nor AR for visualizing industrial CT scans can be found in the Internet.

2.2.1 Short introduction to beginnings of VR/AR with CT scans in medical sector

The medical scientists have been trying to find a way to explore virtual reality potential for surgical preparation and medical education since the late 1980s. The forecasts from the 1990s researchers about how virtual reality in medicine would look like in the future were not a total science-fiction. In 1998, Satava and Jones were already talking how in the future a surgeon will be able to superimpose a scanned image of a patient's organ over a real organ during an operation (13). This future projecting turned out to be fairly accurate. The trend to use augmented reality to for this type of use can be seen nowadays in the medicine research. Nonetheless, these predictions had to be created based on something. To illustrate better at what level of development merge of VR/AR with CR scans was at its beginning a few of the early examples will be presented hereafter. One of the first VR approach came from Delp and Rosen who proposed a method to test variant surgery formulas of the leg (14). The screenshot of their VR simulator to evaluate lower limb tendon transplantation can be seen in Fig. 5.a). Satava introduced a surgery simulation that used 3D modelled organs in 1991. This work is depicted in the Fig. 5.b). The simulator was not the most lifelike as it comes down to the anatomy of the organs, nevertheless it did not fail to provide an opportunity to practise in the virtual reality (15). Later, in 1993 Merrill introduced a realistic model of human torso with the realistic anatomy that could be examined that is shown in the Fig. 5.c). Probably the first 3D model of the full human body was created in 1994 by Spitzer and Whitlock from 1871 CT and MRI slices of the 1 mm thickness (16). The knee fragment is portrayed in the Fig. 5.d). For those times it was a very high resolution for this type of medical image. The last example of this short history introduction combines the use of VR and CT scans. W. E. Lorensen, F. A. Jolesz, and R. Kikinis conducted a virtual colonoscopy by the use of computed tomography scans (17). Virtual reality enabled to have a "flight" through the whole length of the colon. The screenshot of the application is depicted in the Fig. 5.e). This brief characterization does not show a full history timeline of the events and should be treated only like an outline of the beginnings of merging VR and CT scans.

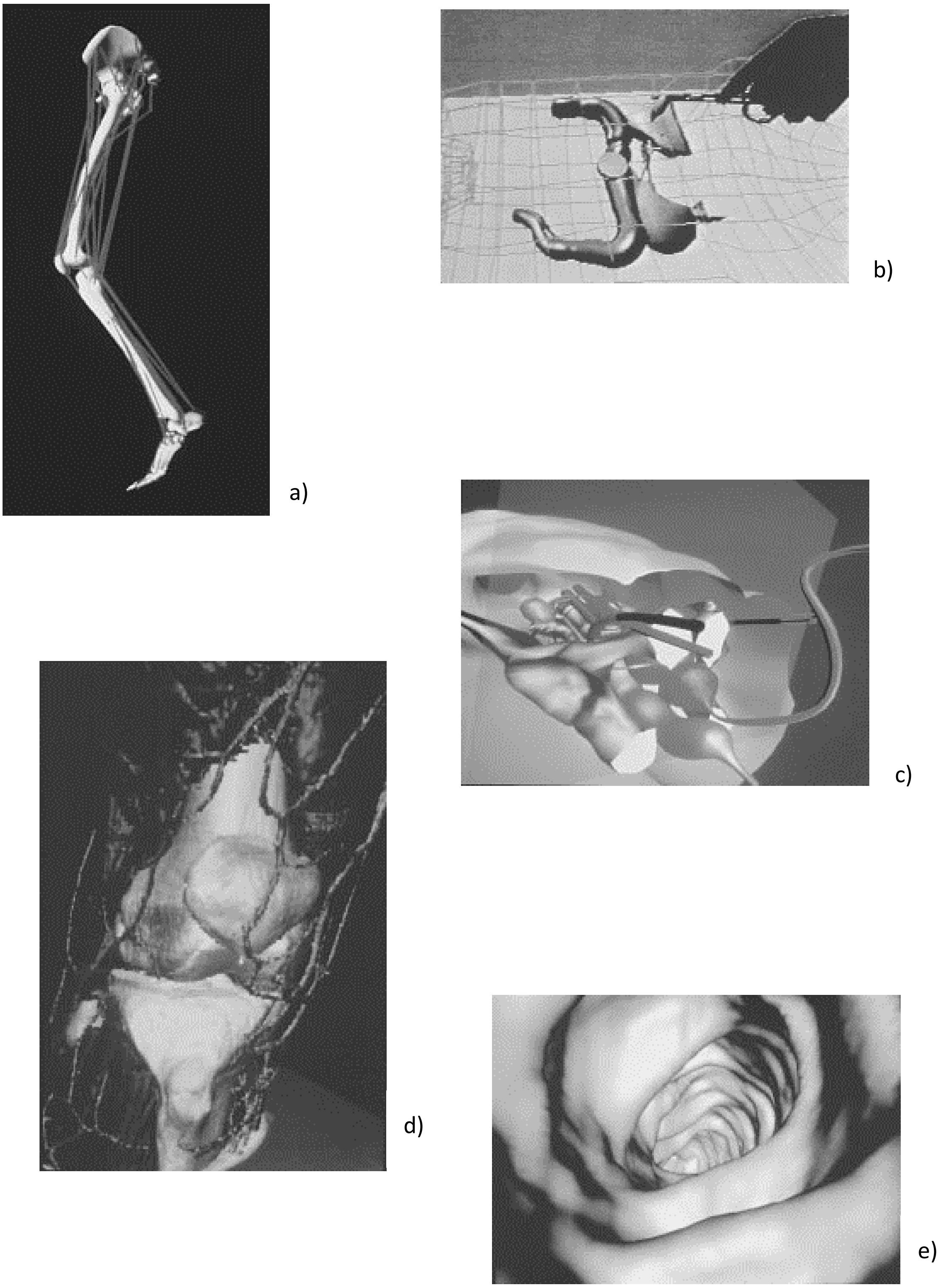

Figure 6. Illustrations of early virtual reality works in medicine (13)

2.2.2 Current application in medical sector

The VR or AR methods to visualize computed tomography scans for the medicine are now rapidly growing popularity. Plenty of science journal articles that cover this specific subject can be found and are continuously being issued. The size of the VR in healthcare global market was estimated to reach USD 885.7 million in 2020 (18). It is supposed to rise from USD 1206.6 million in 2021 to USD 11657.8 million in 2028. Respectively, the CAGR rate for this is 38.3%. The start of Covid-19 pandemic boosted the market, mostly due to the use of the technology for the therapy sessions for patients suffering from Post-Intensive Care Syndrome and the increased need for the remote and interactive learning sessions for the medical students and young professionals. Assuming from the citations that are appearing hereafter, the level of development of this technology seems to be maturing. It can no longer be called a novelty. The cost of VR/AR hardware has significantly decreased, and this equipment is widely available nowadays. The U.S. Food and Drug Administration supports virtual reality implementation into the medical sector (19).

When it comes down to using VR/AR methods of visualization for CT scans, according to various scientific research - doctors associated with CT scans are sympathetic towards new opportunities that VR/AR brings (20) (21) (22). First surgeries using the augmented reality technology have already been performed on patients successfully in 2020, one of them was placing six screws in a patient's spine for spinal fusion surgery and the second was removing a cancerous tumour known as a chordoma from the spine of a patient (23). All these factors combined, it begs the question why merge of AR/VR with CR scans is not yet commonly introduced in the medical sector and why the shift from conventional examining CT scans in two dimensions to VR visualization is arriving so slowly. It is difficult to find a satisfactory answer to these questions. Introducing a new technology is not always rapid and many independent factors may concur for this slow-paced change. The fact that medicine is an enormous and full of legislation pieces sector might have a key importance here (22). Researchers are working on this subject continuously since the 1990s so it leaves an impression that there must be a lot of potential in it. In their works they try to present how it can make everyday job of doctors and surgeons safer and simpler. This creates a lot of hope that merge of these technologies will be broadly used sooner than one can think. A few of examples of potential benefits of introducing the VR/AR visualization of CT scans in the medicine sector will be presented now.

To start with, VR/AR visualization can bring much help to prepare a surgery plan (20) (21) (18). It can assist with determining the best location for the surgical access and examining spatial orientation of anatomy structures. In addition to that, it may show the things that might not be noticed in 2D by the examining person. All these factors can have a crucial impact on the course of operation. It can lead to a better choice of body cuts and actions during a surgery.

Moreover, medical practitioners can make great use of VR/AR training (18) (24). It gives a learning possibility to medical students and young practitioners to learn their craft in a remote and interactive way. It has become even a more valid point since the beginning of the Covid-19 pandemic. It has boosted the remote learning around the world, also including the medical studies.

What is more, using VR/AR to visualize CT scans gives an ability to examine the organs by a few users who are not in the same place at a given time thanks to the multi-user environment function (22) (25). This creates a great opportunity to discuss a surgery or simply an organ condition remotely, having the object of the discussion in front of themselves, by people who are not able to meet in person due to a certain condition. It is possible to point, draw and mark on the 3D model in the virtual reality, what makes for great collaborative meeting possibilities.

Another use for the VR and AR in medicine is patient education (24). It gives the possibility to show and explain certain points of a particular treatments in a more clear way to a patient. A surgeon can describe the planned surgery to the patient, introducing the possible impediments and offering them options.

Just a few of potential uses of VR/AR visualization of computed tomography for medicine were mentioned above in order to picture the huge potential that lays in it. Although there are many advantages of joining these two technologies for the use of the medical sector, there might be also some drawbacks hidden in it, for example the motion sickness coming that is characteristic for the too long use of a HMD.

Figure 7. Surgeon using VR goggles

2.2.3 CT in non-medical industries

Figure 8. Fused filament fabrication 3D printing production

The industrial computed tomography usability has been already acknowledged and established for the purpose of object inspections (26) (6) (7) (27) (28) (29) (30) (31) (32). It should bring no surprise as there are a few significant asset that it provides (26). First of all, it offers a non-destructive way of qualitative inspecting internal areas of an object's geometry, e.g. looking for cracks and voids. Before that companies would have to dissemble the object or cut it through to have an access to its internal structure (27). Suitable use for this are, for example, 3D printed elements which have not homogenous internal structures. Secondly, it can bring much help with verification of assemblies of parts in assembled state. Thanks to this, complex assemblies no longer have to be disassembled to in order to examine hard-to-reach areas of assemblies. Thirdly, CT scanning of industrial objects provides precise way of inspecting dimensional as well as material quality control for both internal and external structure of an object at the same time. Apart from 2D slices that can be examined, CT scanning for industry is able to generate also a high density 3D point cloud or a precise 3D mesh for further inspection purpose. From this quantitative dimensional and tolerance verification to a 3D CAD model can be calculated by a computer, although yet not every CT scanning procedure has a permissible error of length measurement valid with the ISO non-destructive metrology norm, even for the devices and experts from research institutions and companies around the world working actively on CT dimensional metrology (6). In the future, when it complies with every necessary norm, this can be used as a substitute for tactile co-ordinate (CMM) and optical metrology or even potentially replace them. Such 3D representation of the industrial CT data can also successfully be used for qualitative visual inspections. What is more, a 3D mesh model generated from CT scans can be used for reverse engineering. A model built like that identically represents the real object. Another use of such model is for FEM analysis. Thanks to a model acquired like that, finite element method analysis can be done a mesh that represents the real object more accurately than the mesh built upon a 3D CAD model. The variety of use CT scanning for the industry is impressive. Furthermore, a huge asset is that it all practically can be done from one measuring dataset.

In 2016 it was estimated that there were around 2000 - 3000 CT systems working for non-medical applications worldwide. Moreover, it was estimated that there were more than 30 suppliers of such systems and more than 10 software provides for such systems worldwide (29). In 2020 the global industry CT market was valued at USD 441.43 million. In 2021 it was said to be growing at the 7.5% rate each year from 2021 to 2028 (28).

The building structure of an industrial CT scanner differs from the medical CT scanner. For the industrial machine usually the examined object is fixed to a rotating base while both the X-ray emitter and the X-ray detector remain in a steady position (27) (6). In this case an examined object is no longer constrained by the ring that is used in the medical CT system. It means that it is possible to inspect bigger object and in not standard shapes.

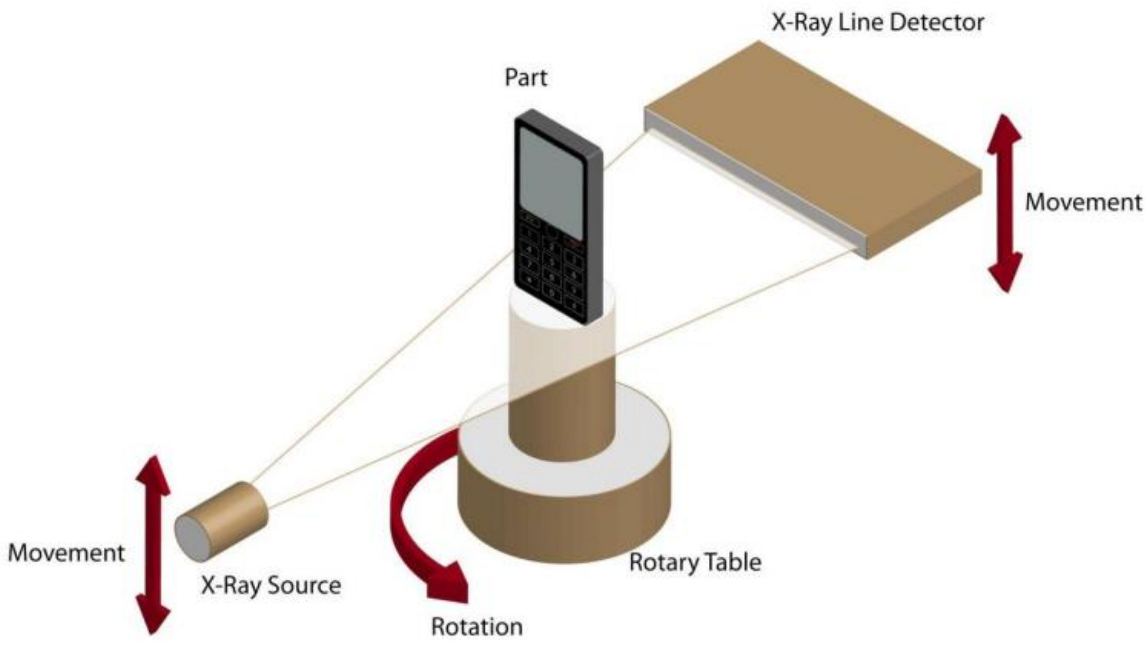

Figure 9. Working principle diagram of industrial CT system (27)

There are a few industries which actively benefit from using CT systems for object inspection. The manufacturing industry has started showing interest in it in 2005 (6). It finds it use for the objects that were produced by the manufacturing processes including: casting and forming, injection molding, subtractive manufacturing or additive manufacturing. CT system examining can bring a great value for this branches particularly because the characteristics of these objects. It can be successfully used also for every object that is being assembled, especially if it has a not-easy accessible internal structure or is a complex assembly. Among other industries that have started using computed tomography systems for their benefit are electric and electronic devices, inhomogeneous materials or food industry. Let an example for the electronic industry be inspecting semiconductors' bonds for the detection of imperfections. For this purpose CT systems with nano or micrometers accuracy are used. As far as inhomogeneous materials are considered the great example is the wood industry, where on line CT scanning can be used for cutting optimization based on the wood defect detection. For the food industry CT systems are able to detect any unwished objects in food that are unacceptable to reach the final customer.

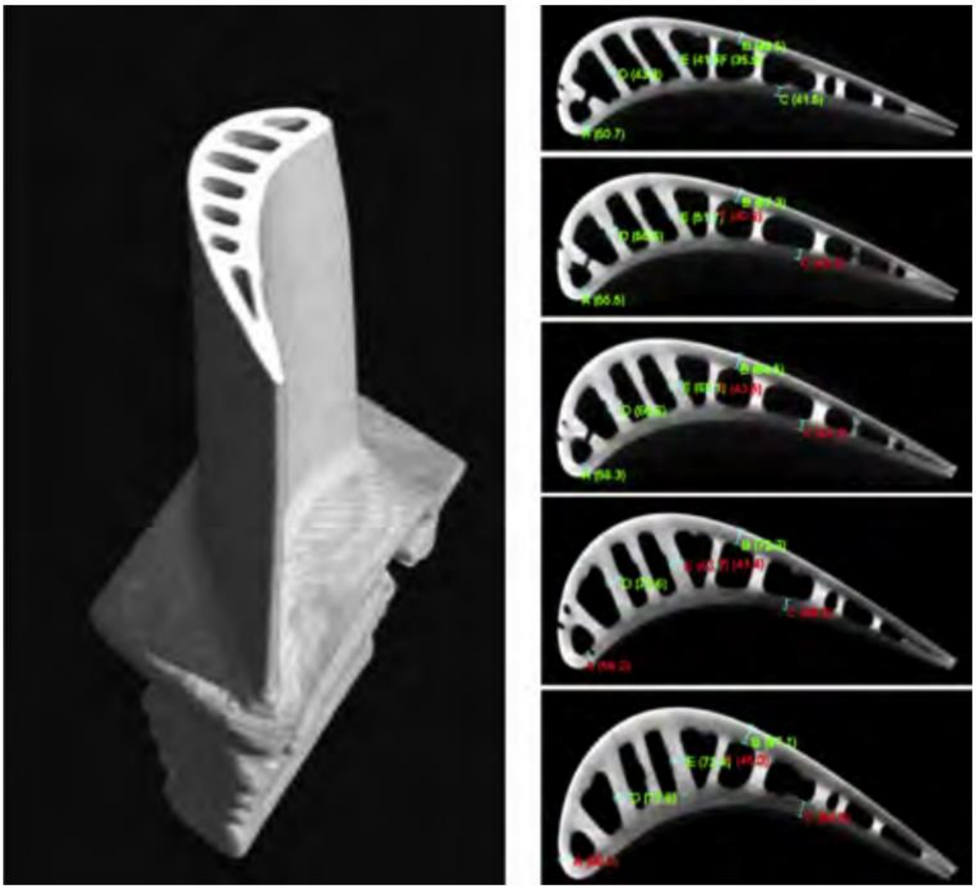

Figure 10. Dimensional quality check of turbine blade (6)

The key trends that are currently visible in the industrial computed tomography market (7) (32):

- the aerospace industry is said to remain the key contributor to the industrial CT market,
- Europe to hold a major industrial CT market share, mainly thanks to the aerospace and automotive industries – the strict safety protocols and preventive maintenance of industrial systems are intensifying due to several countries' governments; the United Kingdom holds 17% share of the worldwide aerospace industry, only second to the United States,
- the tendency to replace production line inspection workers with automated detection devices which boosted at the beginning of the Covid-19 pandemic might birth a great market for the industrial CT scanning.

Figure 11. Aerospace industry to remain dominating industrial CT market

The future research focus when it comes to the industrial computed tomography is put mainly on (6) (7):

- higher resolutions and higher accuracies - down to 0.4 μm (or even lower), voxel sizes down to 50 nm possible currently,
- faster scanning,
- improving and introducing reconstruction algorithms and scanning strategies,
- drifting towards more accurate material simulation – aiming to simulate microstructure of materials more precisely,
- scanning bigger parts and assemblies,
- improving and introducing new methods for reducing measurement errors,
- improving and introducing new ways of combining X-ray CT with other imaging methods,
- new visualization methods.

Although CT scans for manufacturing industry topic comes up in scientific papers to some extent, there seems to be no research done yet about introducing the VR nor AR methods of visualization of computed tomography scans for industry purposes. Nevertheless, combining the assets of VR and AR visualization with the assets of examining industry parts via computed tomography may give great results and have a promising feature. When it comes to qualitative visual inspection this can bring tremendous benefits.

2.2.4 VR for product development process

Although there is no information traceable about using VR specifically for visualizing industrial CT datasets for the product development process nor engineering, plenty of scientific articles can be found on the topic of using VR for various engineering purposes, e.g. the civil engineering or the energy engineering, and the broad areas of design, e.g. the interior design. The most relevant VR use cases for this dissertation that could be found are the product design and engineering. The primary applications, among many others, of the VR for the product development process are 3D modelling, virtual prototyping for simulation of using an object and product design evaluation (33).

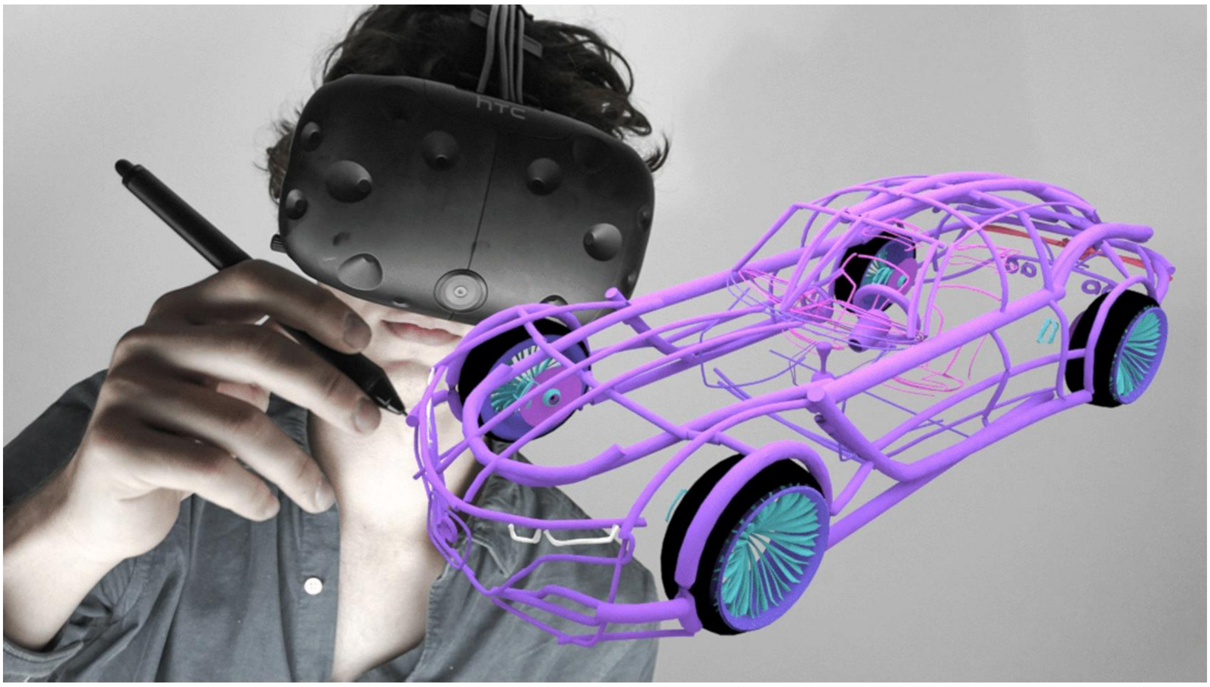

Figure 12. *Gravity Sketch* software

The virtual reality 3D modelling is not yet a common practice in the product design and engineering sector, where the 3D modelling is traditionally being done in the Computed Aided Design (CAD) systems with the help of a computer screen and a mouse. The VR 3D modelling applications such as *Gravity Sketch* are bringing innovation to the conceptual design phase. They enable designer to work in a 3D environment in every desirable scale, what can remarkably help with understanding the proportions and shapes of the designed objects, reducing the risk of the mistakes in the early development phases that can results in fatal effects during the subsequent phases of the product development process. The Gravity Sketch application has been trusted by big corporations from various industry sectors as e.g. *Ford*, *Nissan*, *Reebok* or *Hewlett-Packard Company*.

As far as virtual prototyping is concerned, VR can simulate the way that some things look, run and sound in a faster, less expensive and sometimes way safer than the physical prototypes (34). Consequently, it can bring the financial savings and reduce the time of the product development process by locating defects and errors early. Virtual prototyping with the help of VR is an excellent way to examine usability of products and the assembly/disassemble process. VR can give great visual and auditory imitation of the real world, however it is not yet matured enough to provide the user with the haptic, olfactory, gustatory sensations in every case. Thus, it may not be efficient enough to use VR for visual prototyping simulation in many cases.

Another use case of the VR for the product development process is VR design reviews. The VR technology makes it possible to hold a remote meeting in a virtual room via a multi-user application. This provides a great opportunity to showcase a product possibly during every stage of the design process to co-workers located in any place of the world in a more transparent way than via casual computer application meeting (34). This way of holdings review sessions can also be superior in cases when the access to a computer screen is constricted or impossible, e.g. on the site where the product is meant to be used or on the premises of a manufacturing plant. VR design reviews are being used by the automotive company *Ford* (35).

Figure 13. Inside *Ford* VR lab (35)

Lastly, the VR can be utilized for viewing and exploring the results of the Computer Aided Engineering (CAE) analysis such as the Computational Fluid Dynamics (CFD) simulation or the Finite Element Method (FEM) simulation (36). In this case, the VR enables the user to examine the simulation results in the environment more similar to the real-life mechanical, hydrodynamic or aerodynamic testing, making it more intuitive for the user to examine. It may have a significant role in correct reading of the simulation outcome.

3 The app and its assets

3.1 Introduction to the app

The application case study is proposing a new way of examining the industrial parts via visualizing computed tomography scans in the virtual reality environment. The starting postulate is that viewing engineering data in virtual reality can be more beneficial in comparison to viewing such data on a two-dimensional screen. The added value of the application is coming from many factors and it can be understood better from the following statements. To start with, examining a 3D model or a spatial intersection of 2D slices is way more natural for a human in the virtual reality than on the computer screen, as the user can see it the same way as they see it in a real world space. Secondly, in the created application the user is capable to view the scanned data in a 1:1 scale. In many cases it is not possible to execute on a computer screen, as the scanned object dimensions may go out of a monitor somtimes. These and more advantages of the application are to be shown in the subsequent part of this chapter, where every desired functionality of the application will be presented. The app has two primary modes – the 2D slices mode and the 3D model mode. They will be presented in detail in subchapters*4.3 2D slices mode* and *4.4 3D model mode*. The case study is based on two industrial CT datasets, the lattice structure specimen and the lamp, and every functionality of the app is accessible for both of them. The application does not require any additional hardware controllers as it is founded on the hand tracking usability. There are two ways to activate every button found in the app. It can either be pushed with the index finger or activated by the thumb-index finger pinch gesture after targeting it with the ray pointer. How to manipulate other functionalities will be explained along with introducing each of them.

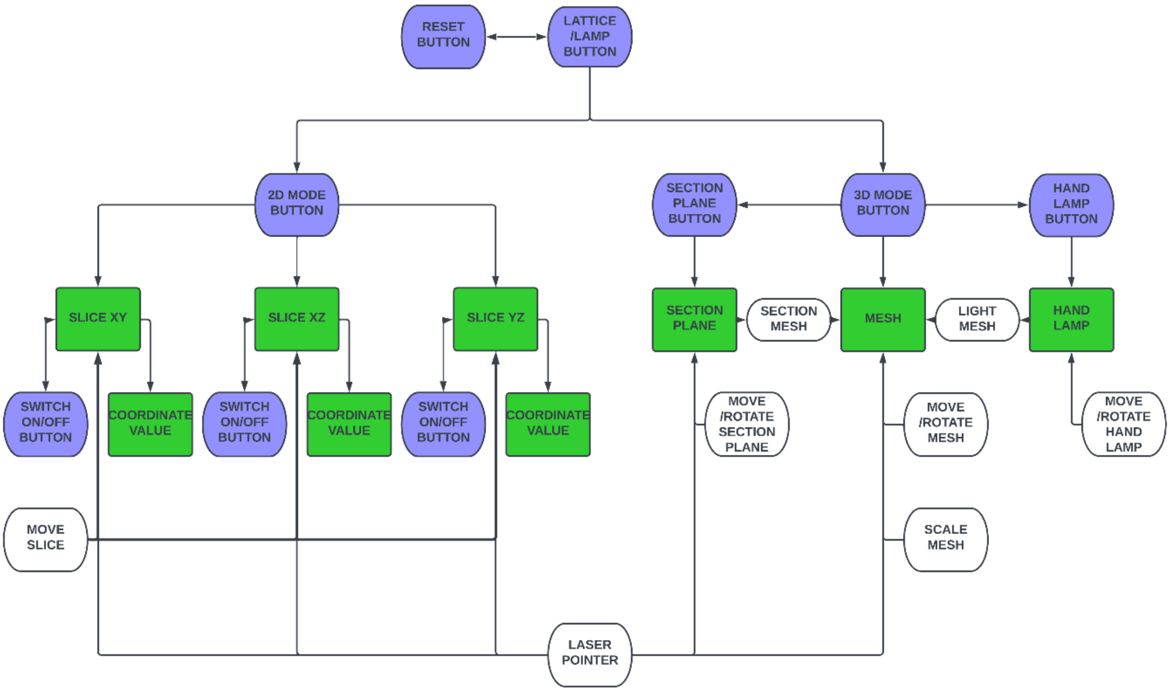

Figure 14. Application flow chart

The flowchart shown in the Fig. 14. is illustrating the working principle of the application. In the beginning the user specifies whether they want to examine the lattice structure of the lamp. At every time the user is capable of pushing the RESET BUTTON, what immediately takes them to the starting point of the application and brings everything back to the default settings.

After choosing the object of interest, the user decides whether they want to examine the 2D slices or the 3D model by pushing either the 2D MODE BUTTON or 3D MODE BUTTON.

When the user enters the 2D mode, the three perpendicular slices appear, each of them having its own SWITCH ON/OFF BUTTON and coordinate value shown as a form of text. With the SWITCH ON/OFF BUTTON the user can exclude the slices they do not need in a moment. Each slice can be moved along its axis what enables the user to examine particular areas of the object. The laser point enables the user to point to the area of interest.

When the user enters the 3D mode, the 3D model appears in a form of mesh. The 3D model can be moved, rotated and scaled. Along the 3D model the two buttons appear – the SECTION PLANE button and the HAND LAMP button. Clicking the first one causes the section plane to appear, which enables the user to section the 3D model with it. Clicking the HAND lamp button causes the hand lamp to appear, which enables the user to light the mesh in a desirable areas. The laser pointers also here makes the user able to point to areas.

3.2 Starting view

After opening the app the starting view is displayed. At this point is it up to user to switch on with a respective button either the lattice structure specimen of the lamp to be examined. There is also a third button to reset the whole application to the initial state. Only after this it is to be decided whether 2D slices mode or the 3D model mode to be switched on or to have both turned on simultaneously. Both of the modes can be easily switched off with the same button they were turned on. Apart from the buttons there can be seen also the coordinate system XYZ. The central point of the both objects under examination is the starting point of the coordinate system. The coordinate system has a crucial importance for the user to understand how the 2D slices are arranged in space. It can also be useful for communicating a direction of some detail in the 3D model mode.

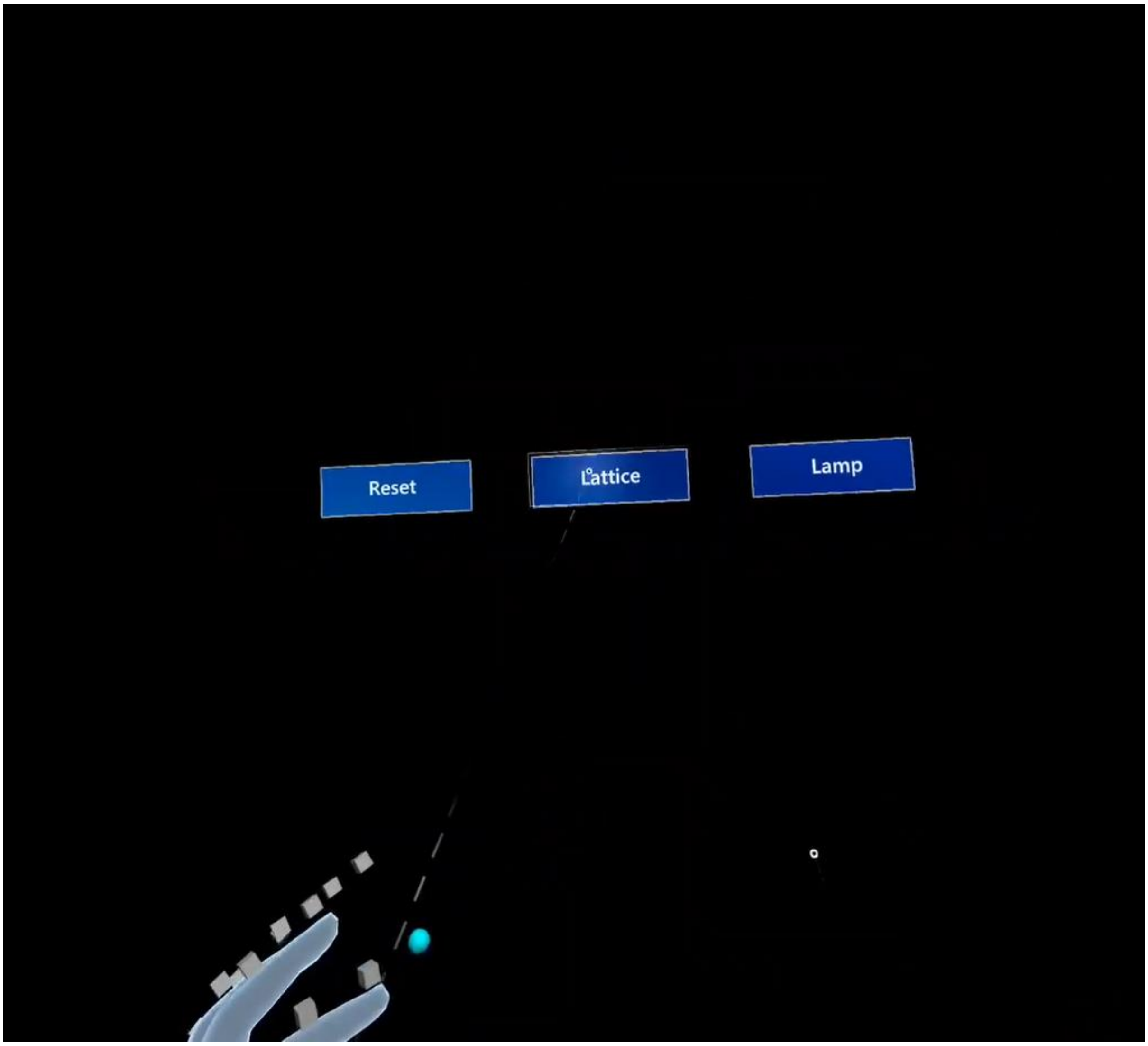

Figure 15. Starting view of the application

3.3 2D slices mode

After switching on the 2D slices mode button the intersection of three two-dimensional computed tomography slices appears in the scene. Each one of them belongs to respectively XY, XZ and YZ planes. At the same time, in the right part of the scene three buttons emerge – Slice XY, Slice XZ and Slice YZ, every with the number value right next to it. 2D slices are more accurate representation of the computed tomography data than the 3D model generated from it and are very precious when looking for a specific area of the object under examination and some details existing there.

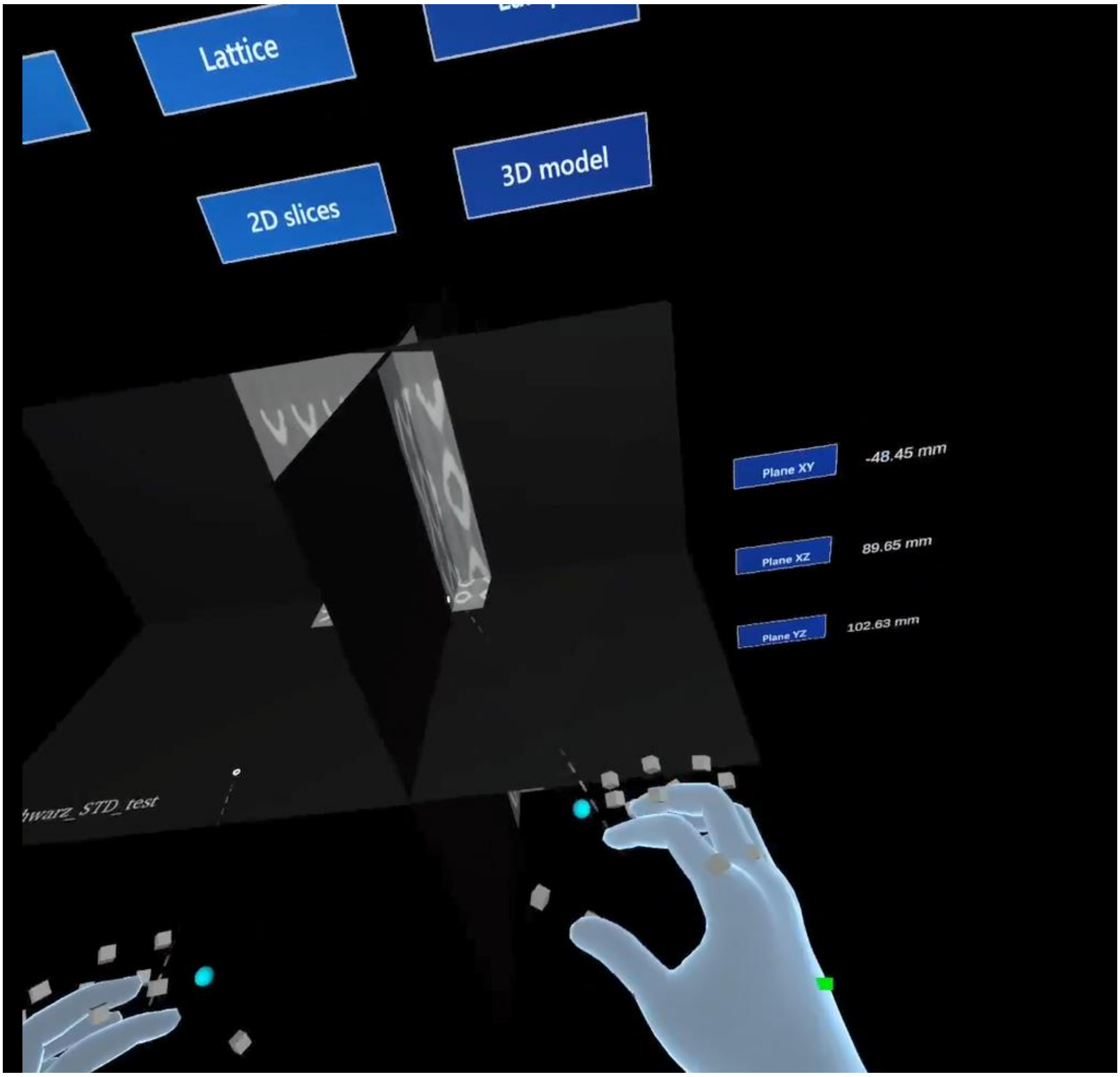

Figure 16. 2D slices mode of the application

3.3.1 Slice switch on/off

The buttons Slice XY, Slice XZ and Slice YZ are responsible for turning on and off each of the three slices. This is a handy functionality when the user wants to focus only on a specific slice plane or a pair of slice planes. In such hypothetical situation a third, for the moment unnecessary, slice plane can disturb the process of examining the data so there is an easy way to temporarily switch it off.

3.3.2 Pointer

Of each hand of the user comes out the ray pointer. With this pointer the user is able to show in an intuitive way a certain part of a slice. This can be very helpful while screen recording the application in order to show some data issue to a dedicated person.

3.3.3 Move slice

The user can grab each slice either with thumb-index finger pinching the slice itself or with the thumb-index finger pinch gesture from far. During the grab, the position of the slice is changing according to the movement of the grabbing hand. For example, the more left the user moves their hand while grabbing the slice YZ, the more the slice moves backwards in the direction of the X axis. All slices can be brought back to their initial position (0,0,0) by the reset button.

3.3.4 Coordinate value

The value number right next to each slice button is representing the slice position in the coordinate system. As stated before, the central point of the object under examination is the starting point of the coordinate system. When the slice position is changed, the coordinate value changes accordingly. This functionality helps to specify very accurately the exact position of a certain slice.

3.4 3D model mode

When the user is switching on the 3D model mode, the 3D model of the examined object appears. Initially, the model is in the same scale as the 2D slices in the 2D slices mode. Right next to the model two buttons appear – the section plane button and the hand lamp button. Every functionality of this mode will be accurately described in what follows.

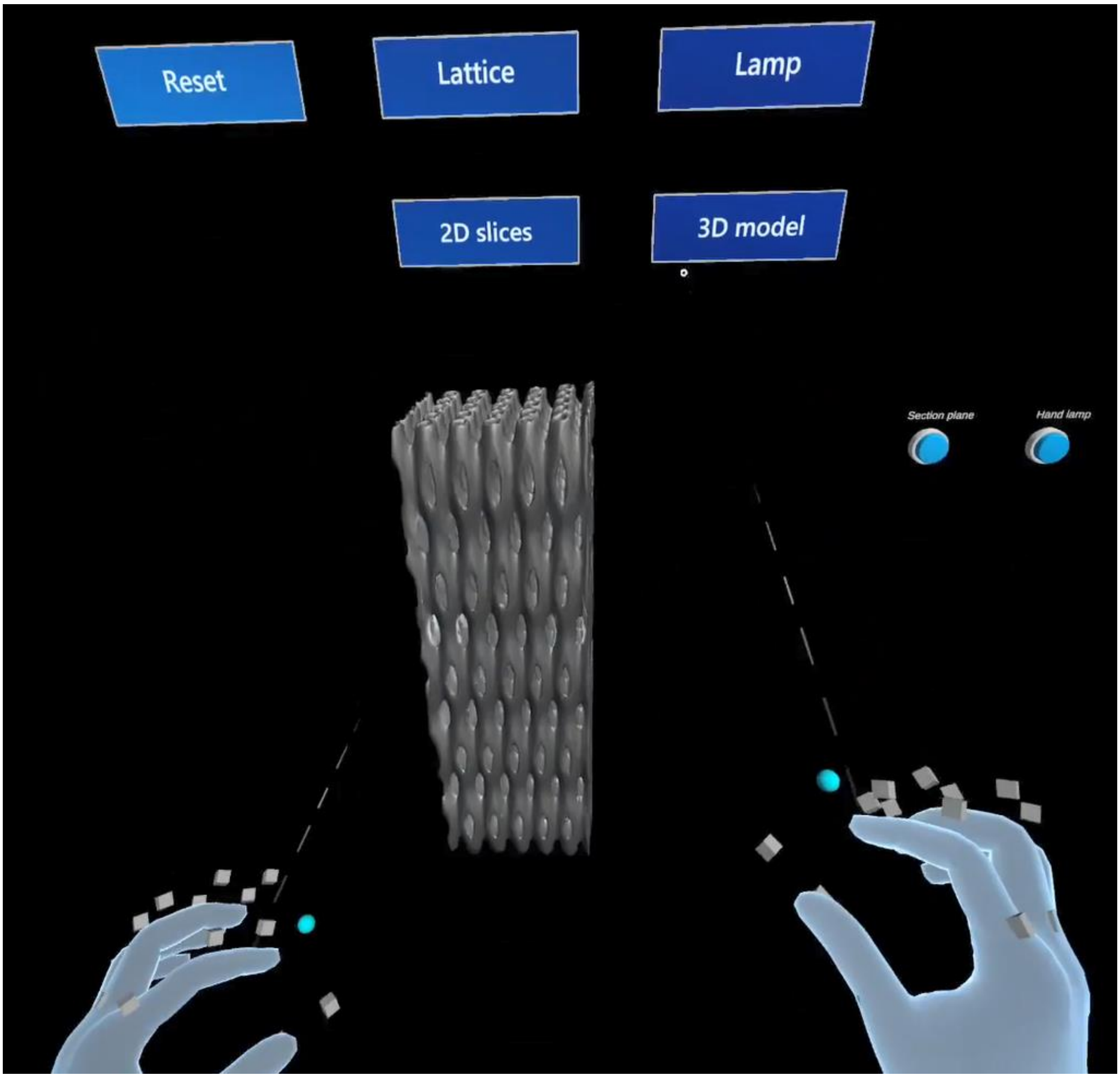

Figure 17. 3D model mode of the application

3.4.1 Pointer

The mode of use and the functionality of the pointer in the 3D model scene are the same as in the 2D slices scene. The idea is just to be able to show a certain point of the 3D model in the intuitive way.

3.4.2 Move/rotate object

The 3D model can be moved and rotated in every axis in a way that the user wants. To perform this the user has to grab the 3D model. It is done in the same way as grabbing the 2D slice. While the grab, the user hand position dictates the position of the object. The functionality to freely tailor the position of the object enable the user to view the object from the perspective that suits them the most.

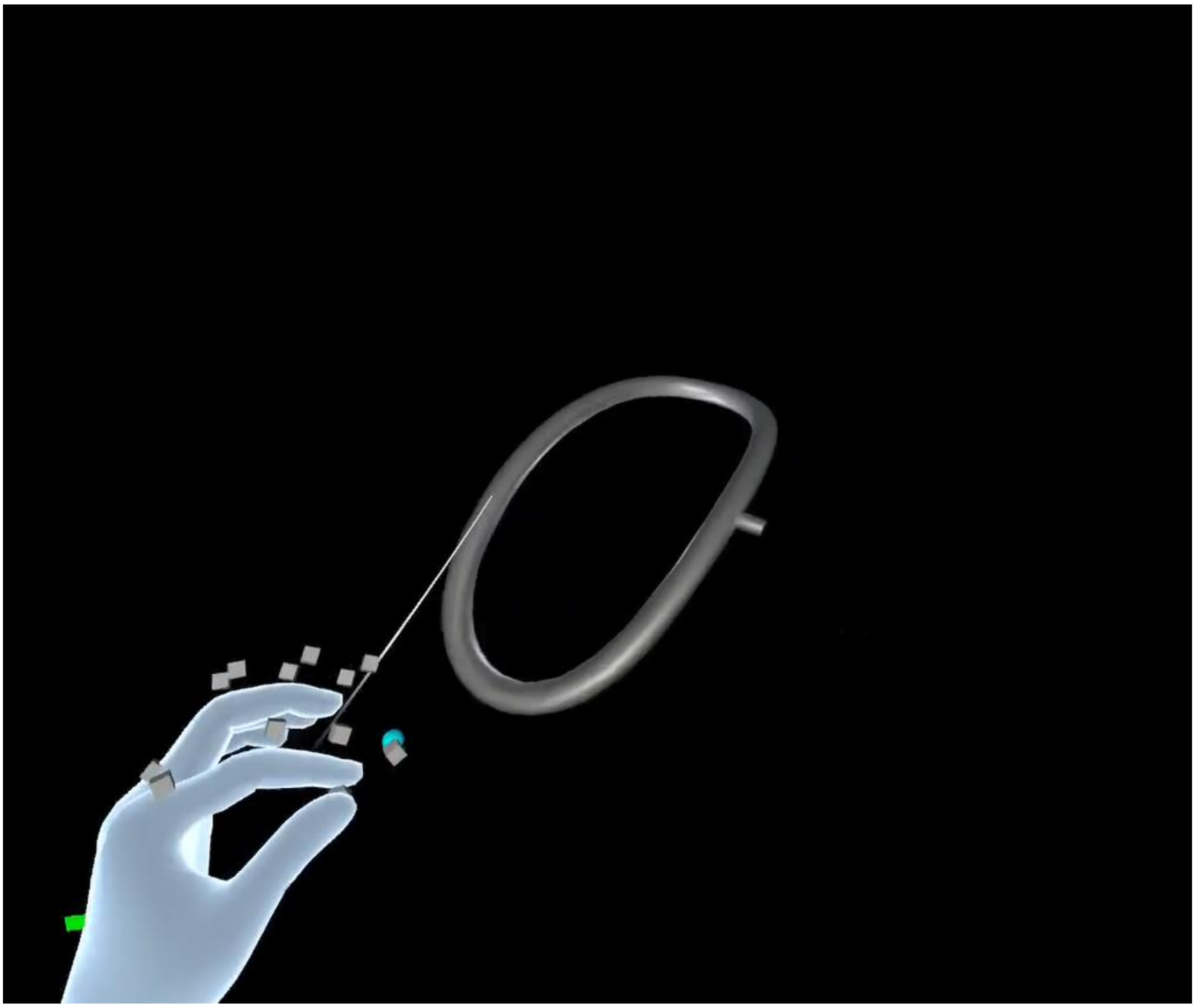

Figure 18. Moving and rotating 3D model in the application

3.4.3 Scale object

To scale the object the user has to grab it with both hands. Intuitively, bringing the hands closer to each other makes the object smaller and pulling the hands further away from each other makes the object bigger – as it happens with compressing and stretching an object in real-life. There is no constraint put on scale so the user is free to make the 3D model any size they want. This idea can help the user to enlarge small details of the object and respectfully minimize some big parts of the object to capture the bigger picture. The scale and the position can be brought to the initial state any time by pressing the reset button.

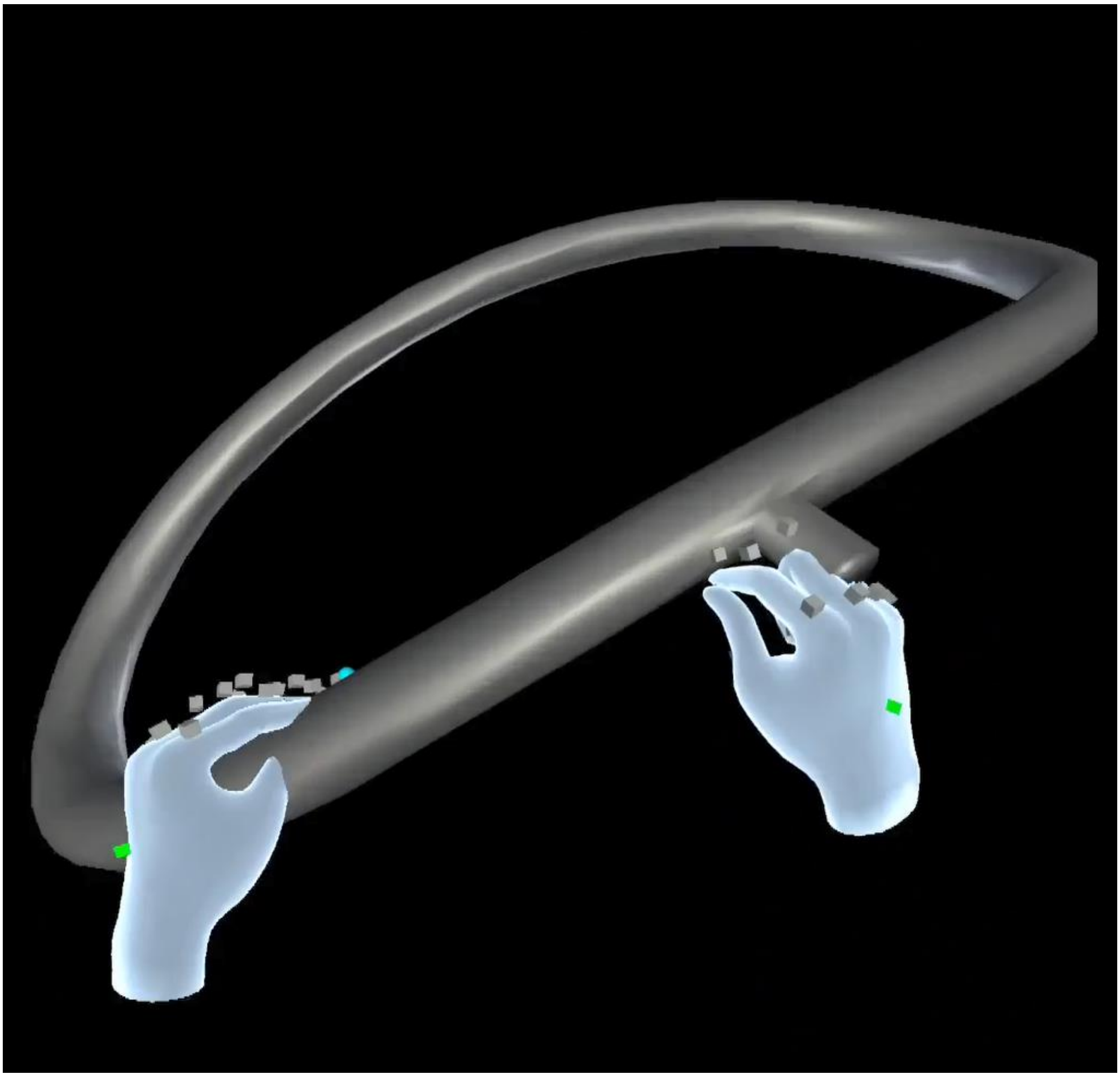

Figure 19. Scaling 3D model in the application

3.4.4 Section plane

To activate the section plane the user has to press the section plane button. When it is done the section plane appears in the space. The user can easily grab the section plane and freely move it across the 3D model to section in the exact plane they need. Thanks to that functionality the user can view the internal structure of the object in every area in a simple way. As the 3D model comes from the .OBJ mesh and it represents only the outside surface of the object that was computed tomography scanned, while cutting the 3D model with the section plane the inside area of the object is capped and represented with a solid color.

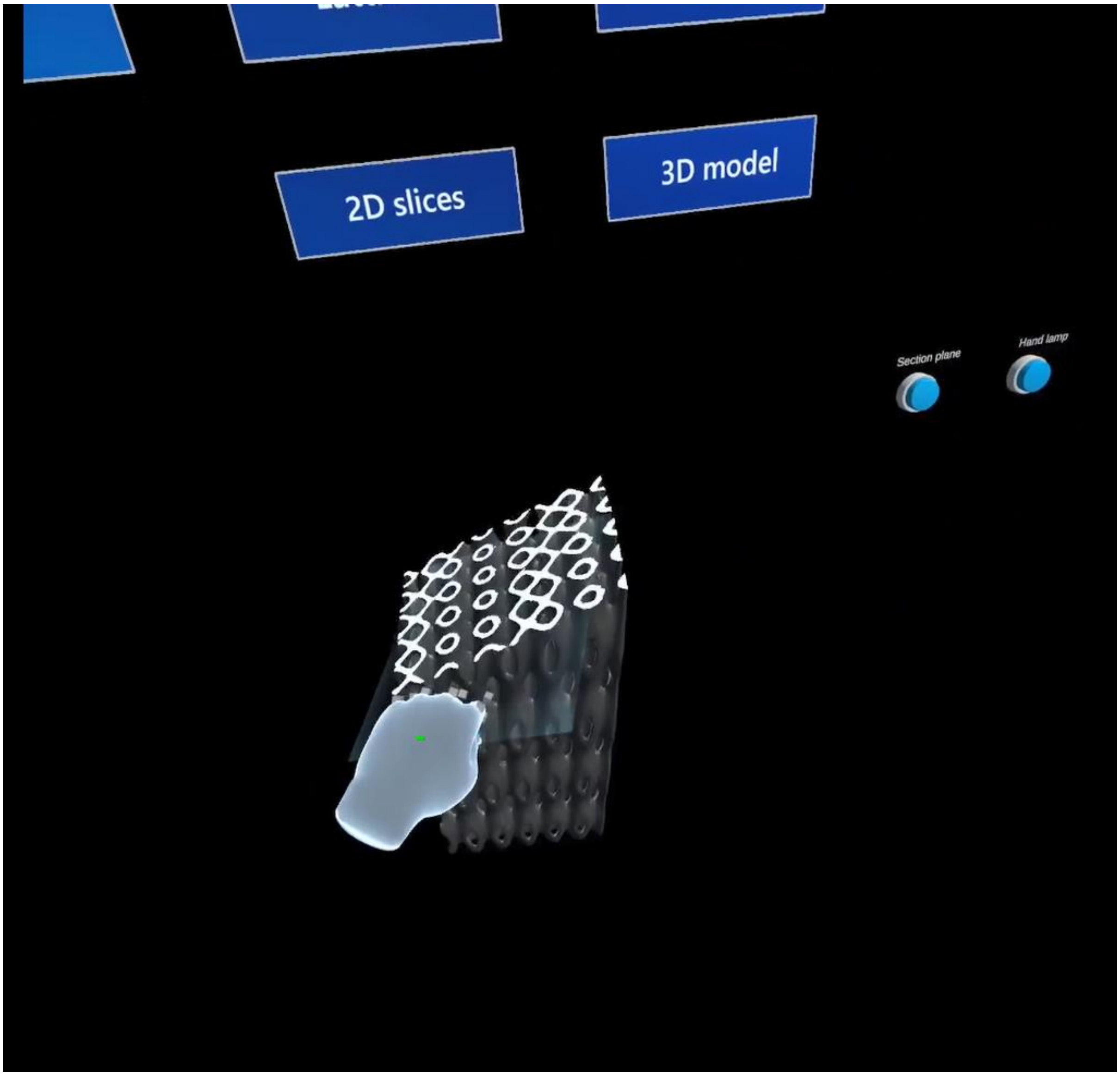

Figure 20. Section plane functionality in the application

3.4.5 Hand lamp

The hand lamp is a useful feature for the situation when more accurate lightning of a certain area is required. Pressing the hand lamp button activates the cylinder with a light source at the end of it. This lamp can be grabbed and moved around the scene in a way that suits the most the user and their need for a light in a desired spot.

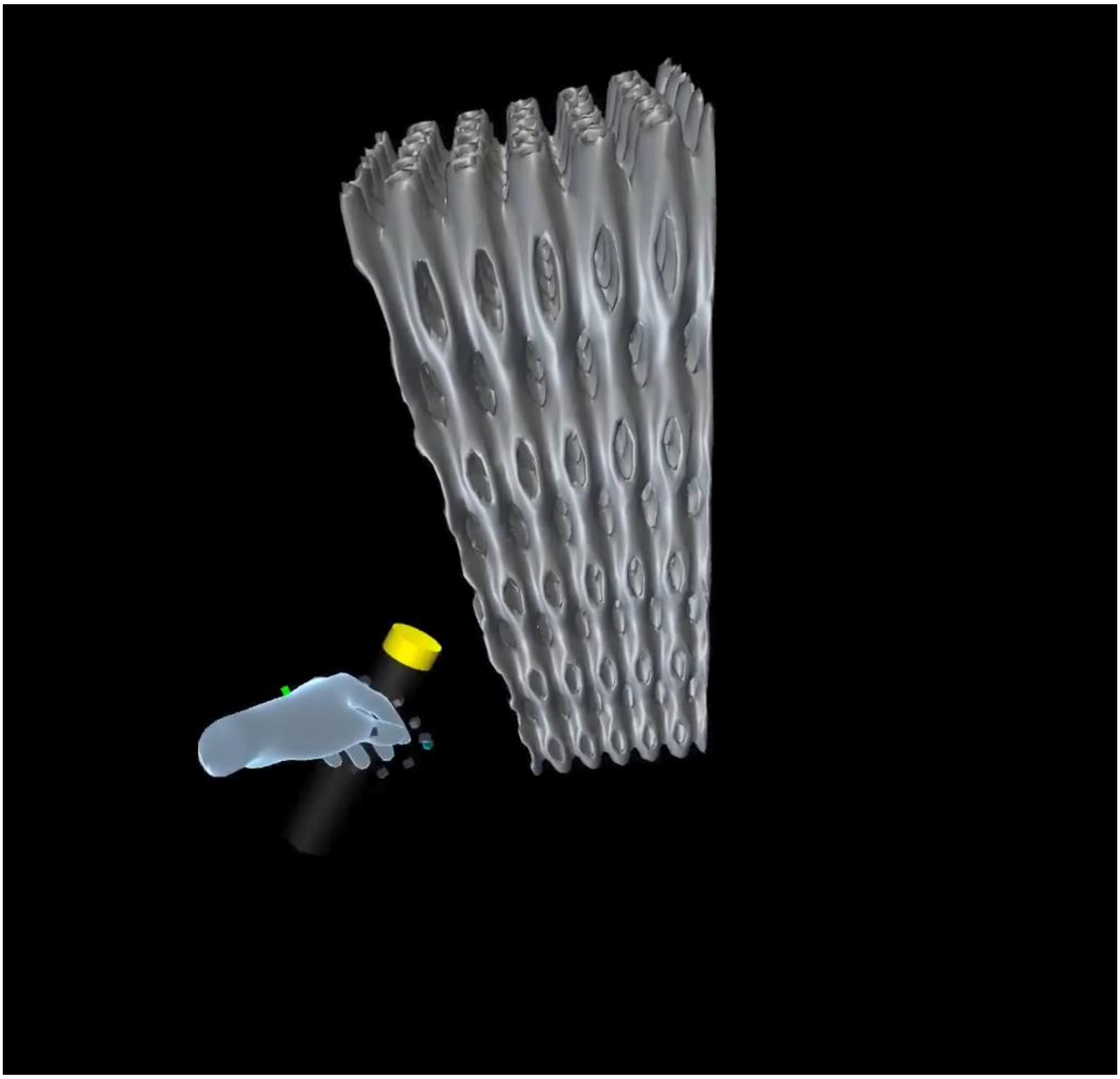

Figure 21. Hand lamp functionality in the application

3.5 Volume rendering

For the application it was chosen to use 3D mesh, but it is also possible to generate a 3D model as a volume rendering voxel structure. Neither of these two methods of showcasing CT scans data in 3D is superior. Each of them is preferable when it comes to different scenarios. This working principle of the volume rendering method, in simple terms, is to visualize 2D images directly in the 3D environment. This subject is very broad and could easily make for another thesis if extended, thus is only modestly introduced here. Moreover, the volume rendering technology is used primarily only on the computer screen, it is commonly used in the computer graphics, and until now is a cutting-edge processing method for virtual reality. It would take a special sophisticated *Unity* shader to display a volume rendering texture via the *Oculus Quest 2* HMD, which is not that easy to develop and this is not in the extend of this dissertation.

On the one hand, a vertices mesh is favored when the accurate representation of the external surface of an object is needed and the internal structure is not of the major significance, e.g. machined objects that are made from homogenous materials. This method of representation is excellent for quantitative metrology inspections when it is of a great importance to measure the exact dimensional deviations from the CAD model. It is also appropriate to generate a reverse engineered 3D mesh of an object that can later be used for modifying it for a CAD model or for FEM analysis. This method of generating a 3D model is applied in the application developed and shown in this dissertation.

On the other hand, a volume rendering voxel structure is excellent to showcase in detail internal structure of an examined object. This is desirable as far as qualitative inspection of the material internal structure is concerned. This representation method is suitable for visual detection of internal defects and for visual material characteristics inspection. For the purpose of the thesis this method is only introduced shortly and not being the main choice.

Nevertheless, the volume rendering texture from the same dataset used in the application was produced in *Unity*, but only to be showcased on a computer screen. The scripts for generating a volume rendering texture was taken from the free source software GitHub from the developer called Matias Lavik (37). The scripts generate 3D voxel data from 2D image sequence data. The volume rendering is based on the ray marching technique. This is a technique to transform volumetric voxel data to pixels that can be seen on a computer screen in the most efficient, low-memory using way. This technique is especially dedicated for displaying very complex bodies like, e.g. fire flames or smoke. Below in the Fig. 22. And Fig. 23, there can be seen illustrations showcasing the volume rendering texture generated from the lattice structure specimen image sequence CT dataset and the section view of it.

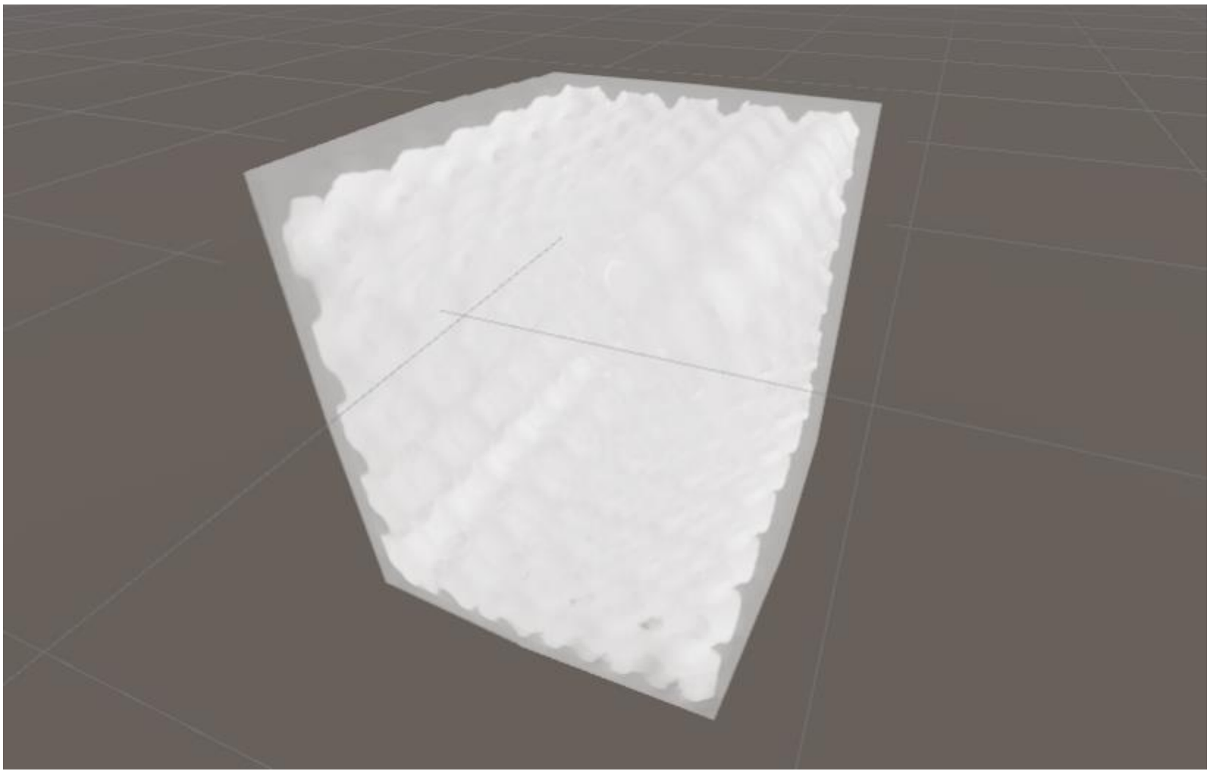

Figure 22. Volume rendering texture from the lattice structure specimen dataset

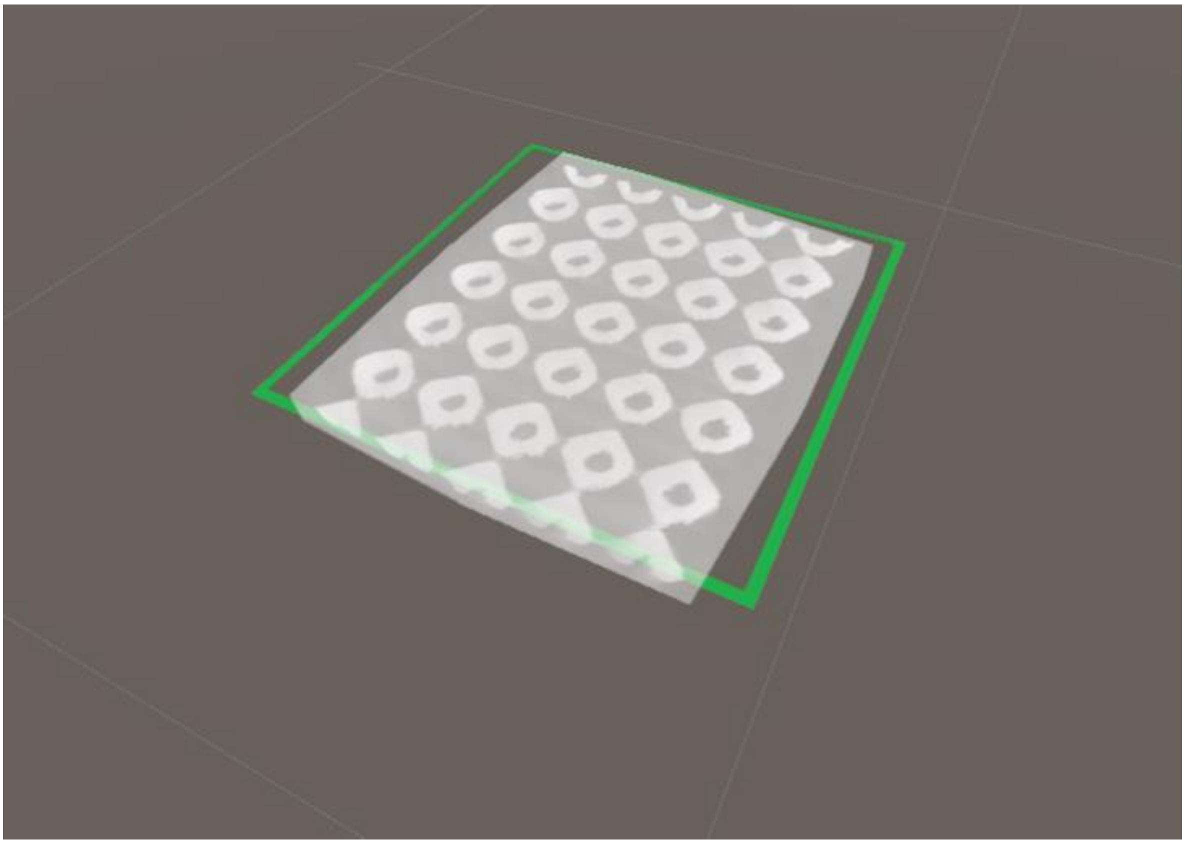

Figure 23. Section view of volume rendering texture from the lattice structure specimen dataset

4 Process of creating the app

4.1 Software and hardware selection

The major part of development of the application was performed in *Unity* software. *Unity* is one of the world's most leading platforms for creating and operating interactive, real-time 3D content. Unity offers a free of charge personal license programme for "individuals, hobbyists, and small organizations with less than $100K of revenue or funds raised in the last 12 months" (38). *Unity* also has built over the years a considerable community online which shares their solutions to various developing obstacles and is willing to help each other.

The application was developed in the *Unity*, but the raw CT data had have to be processed before importing it into the *Unity*. The additional programs that were used for the data preparation are *Materialize Mimics*, *3D Slicer*, *ImageJ* and *Autodesk Meshmixer*. Originally, the CT dataset is an image sequence sectioning the scanned object in parallel along one direction. To generate three perpendicular sets of slices for the purpose of the application, every in different direction, the *3D Slicer* software was used. The additional processing of the 2D images was made in the *ImageJ* software. The *Materialize Mimics* was used to generate the 3D meshes in .OBJ format from image sequences in the .TIF format. Later, the 3D mesh size was reduced using the *Autodesk Meshmixer* software.

The *Oculus Quest 2* is one of the most common and accessible head-mounted displays on the market right now. Since its launch on October 13th 2020 there was an estimated 1.87 million units sold throughout the world till the end of the first quarter of 2021 (39). This number must have grown significantly till now since moment the *Facebook* CEO Mark Zuckerberg presented *Metaverse*, a social virtual reality world, in October 2021. Moreover, these goggles are completely compatible with Unity. There is no special initial activity needed apart from arranging proper settings in *Unity* and the development can start immediately from connecting the display to a computer via USB cable.

The factors enumerated in the two paragraphs above were the encouragement and influenced the choice of *Unity* and *Oculus Quest 2* as the set for the development of the app. Nonetheless, there are some drawbacks involved in the choice of the hardware and it should not be left unsaid. To use and develop within *Oculus Quest* 2 having a *Facebook* account is a requirement which could be unpleasant for some that do not feel the need to own such an account. This headset also limited the application to be only in VR, as the AR development is not much extended within the goggles. Assets that could be gain from extending the application environment to AR can be found in chapter *6 Conclusions*.

4.2 Datasets

The first dataset is a .TIF image sequence of 1098 slices of 1276 x 1109 pixels of 1.44 GB size. It was generated using the computed tomography technology. The object that was scanned is a lattice structure specimen made for compression mechanical tests. It was made by the SLS additive manufacturing from the stainless steel 316L. To this extent, visualizing the data in a VR environment is helpful for checking internal defects or possible deviation from the desired geometry. The fragment of the .TIF CT dataset of the lattice structure specimen is shown in the Fig. 24.a) and the 3D mesh derived from its CT data is shown in the Fig. 25.

The second dataset is a .TIF image sequence of 216 slices of 1440 x 2560 pixels of 0.80 GB size. It was artificially generated from the CAD data. The object that was scanned is a lamp designed for additive manufacturing. In this case, the focus is to provide a designer with an intuitive tool for checking internal structures without having to become an expert on reading CT scans. The fragment of the .TIF CT dataset of the lamp is shown in the Fig. 24.b) and the 3D mesh derived from its CT data is shown in the Fig. 25.

The application could equally be used to examine every other industrial elements that can be CT scanned.. Nevertheless, the chosen objects fully qualify showcase every functionality of the app in a complete way.

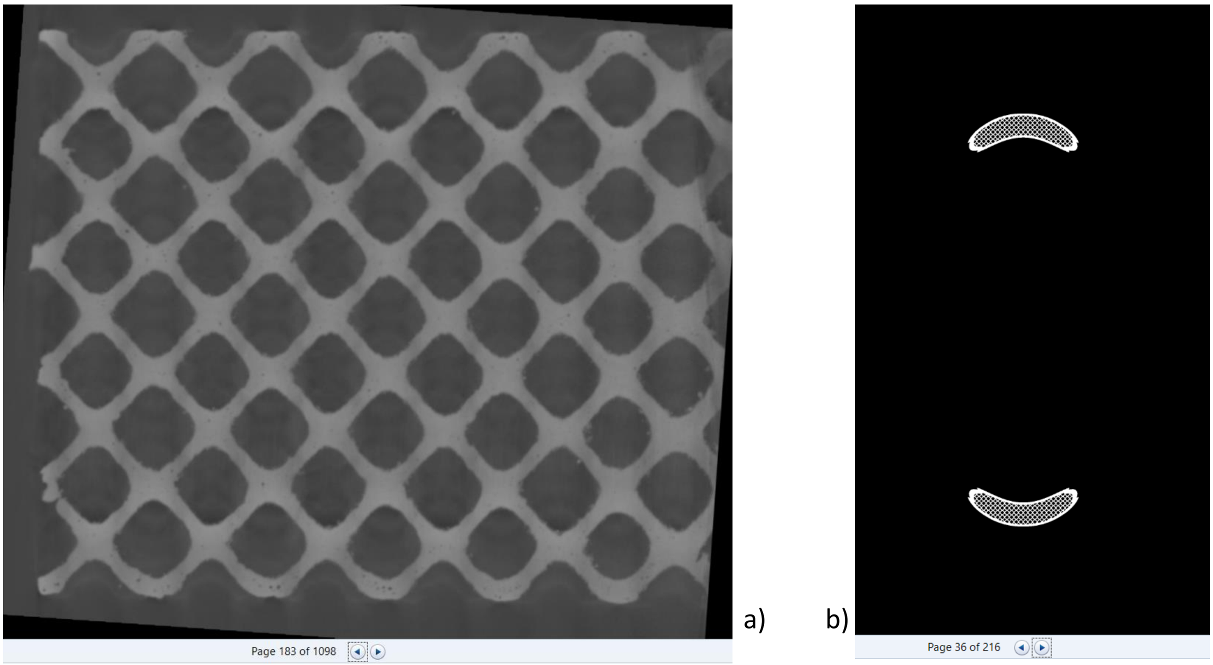

Figure 24. Image sequence datasets used in the application

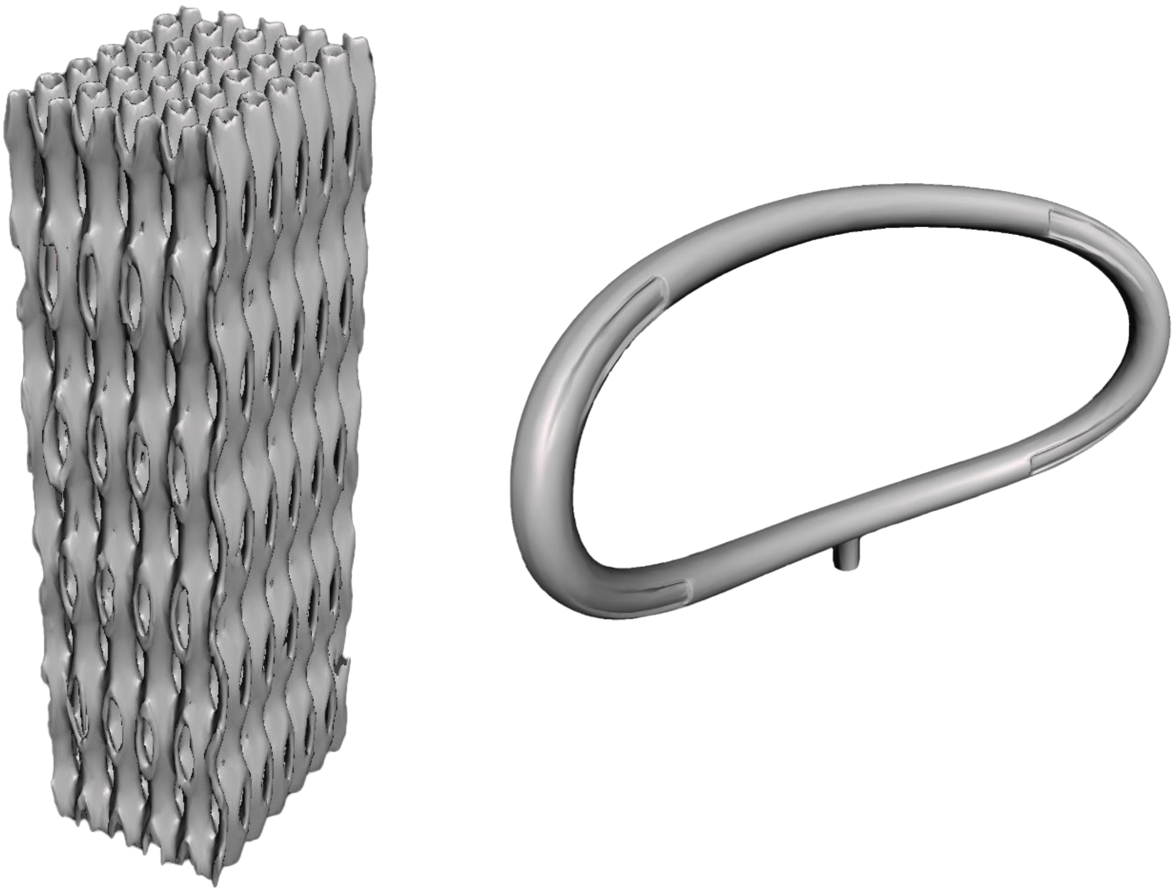

Figure 25. 3D meshes derived from CT datasets – the lattice structure on the left and the lamp on the right

4.3 Unity

The application is generated in 2019.4.32f1 version of *Unity*. There are many factors that have an impact on which version to use for developing. The main reason for using this particular version is that it is perfectly compatible with the *Microsoft* add-in *MRTK*, that plays a significant role in the application.

Microsoft MRTK for Unity (Mixed Reality Tookit) (40) is a project add-in that provides developers with a set of components and features which facilitate their work in VR and AR environments. Most of MRTK features are intended for *Microsoft Hololens* headsets but the ones that are used in the application work just as well with *Oculus Quest 2*.

In order to fully synchronize the *Oculus Quest 2* goggles with *Unity* the package *Oculus Integration* (41) is loaded into the application. This package is necessary for the application because it provides a correct operation of the *Oculus* headset in *Unity* and furthermore is required for the collaboration of the *MRTK* and the *Oculus* headset within *Unity*.

The application is built with the intention of controlling it via hand tracking. The application uses the hand tracking asset from the *MRTK* package. To activate it the hand tracking support option has to be also enabled in the *Oculus Project Configuration manager*. In the beginning of the development it was intended to use the *OVRHandPrefab* with the *OVRCameraRig* that comes from the *Oculus Integration* package for the purpose of the hand tracking. However, it would cause many errors during building the app. The *MRTK* hand tracking turned out to be remarkably more developer-friendly. Moreover, every feature of the *MRTK* package that is used in the application is dependent on the *MRTK* hand tracking – interactable buttons, and in the first place moving, rotating and scaling objects. In a nutshell, it would not be possible to use the *MRTK* functionalities crucial for the app operation if it was not for the *MRTK* hand tracking what makes for the great importance of it for developing of the application.

Figure 26. Hand tracking in *Unity*

The supportive scripts for all the buttons and for the movable slices in the 2D slices mode are coded in the C# programming language. The software used for the coding is *Microsoft Visual Studio 2019*.

In the starting view of the application three buttons with the following text can be seen – “2D slices”, “3D model” and “Reset”. The working principle of the first two buttons is the same. Clicking a button activates an appropriate empty game object in the scene which includes all the elements that belong to a respective mode. Clicking the same button again deactivates the game object. The working principle of the button scripts is quite simple. All of the buttons in the application apart from the ”Reset” button are used for turning on and off a dedicated object or mode, thus all the scrips for the buttons are similar and built around a function that either hides or shows a dedicated thing. The scrips are universal and use global variables, what means that a dedicated object in the scene has to be assigned to a particular script that has to be assigned to a button. The scrip that is used for both "2D slices" and "3D model" buttons is named *hideShow2*. The code can be found at the end of the report.

The "Reset" button is founded upon the script named *reset*. The working principle of the code is to bring every objects and settings to the initial state of the application. The script code is shown at the end of this dissertation.

Apart from the three buttons the coordinate system XYZ visual representation can be seen in the application starting view. The 3D model of it was entirely created inside unity using basic geometry objects as cylinders, sphere plus a TextMeshPro text. Although making it was very simple the function of this element has a crucial significance for using the app correctly.

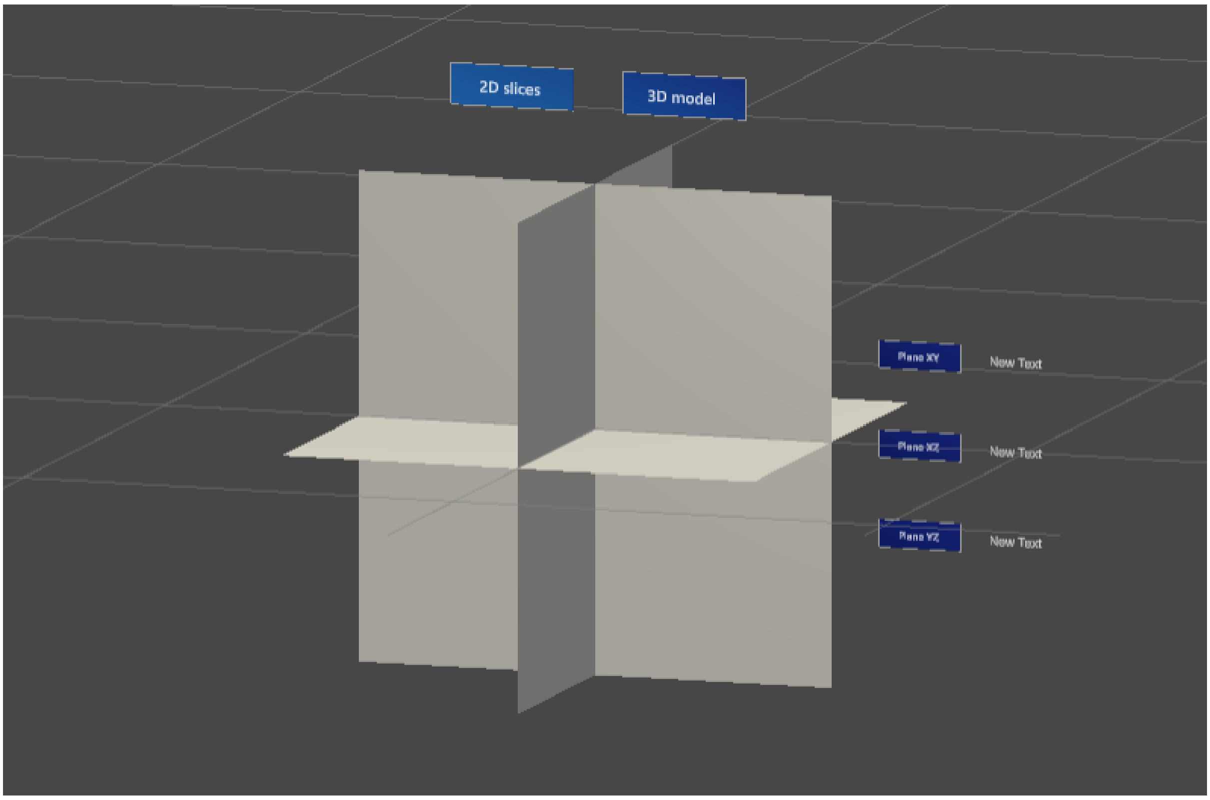

Figure 27. Making of 2D mode in *Unity*

The "2D slices" button activates a game object which contains three planes, another three buttons and an UI canvas with three TextMeshPro elements. The 2D slices mode development started from adding three perpendicular planes to the scene, which represent accordingly the slice XY, slice XZ and slice YZ. The functionality of this mode is based on the fact that each of these plane moves in an appropriate direction, for example the plane representing the slice XY has to move only along the Z axis. To make it possible, three scripts taken from the

MRTK package are added to each plane – *NearInteractionGrabbable*, *ObjectManipulator* and *ConstraintManager*. Due to the *NearInteractionGrabbable* and *ObjectManipulator* scripts it is possible to move, rotate and scale an object that these scripts are added to. The *ConstraintManager* script, as the name of it says, can add some extra constraints on movement, rotation and scaling of an object. In this instance it disables the rotation and scaling of every plane and additionally limits movement direction only to one axis. Apart from the aforementioned scripts every plane object also contains an additional script for controlling the image loaded onto this plane and a coordinate number value that is shown next to this plane's button. The additional scripts names are *sliceXYLattice*, *sliceXZLattice* and *sliceYZLattice* for the lattice structure and *sliceXYLamp*, *sliceXZLamp* and *sliceYZLamp* for the lamp – each for appropriate plane object. As far as these scripts are concerned, the coding gets a little bit more expanded. The codes *sliceXY_*, *sliceXZ_* and *sliceYZ_* are analogous and differs from each other when it comes to a folder path for loading pictures and a position of slices in space, thus six different scripts are required. The working principle of these scripts is to save the current coordinate position of a slice every time it is moved. According to that position an appropriate piece from the image sequence is loaded into the moved plane object and an appropriate coordinate position number value is loaded into the *TextMeshPro* object. For that to be feasible the UI canvas with three *TextMeshPro* elements is inserted into the scene – again in this case each element is assigned to an appropriate plane object. When it comes to the buttons that appear in the "2D slices" mode these are used for switching on and off a respective plane along with the *TextMeshPro* object proper to it. The working principle for these buttons is the same as for the buttons characterized before. The script *showHide2* belonging to these buttons differ only slightly from before mentioned *hideShow2* script. The first difference between these scripts consists in the fact the script *showHide2* is switching on or off two game objects instead of one as it was earlier. The second difference is that in this case the game objects assigned to the script are shown in the scene from the start. In the script *hideShow2* the game object assigned is initially hidden. Another thing worth mentioning when it comes to developing the 2D slices mode is that the special shader was used for plane objects – double sided shaders from *Ciconia Studio* (42). If it was not for this asset, it would not be possible to look at the 2D slices from both sides. This treatment despite its simple implementation significantly improves the usability of the 2D slices mode.

The "3D model" button from the app starting view activates a game object that contains the 3D model of the lattice, the section plane, the hand lamp and two buttons. The lattice mesh object contains the same *MRTK* scripts as the slice planes from the 2D slices mode – *NearInteractionGrabbable*, *ObjectManipulator* and *ConstraintManager*. Their role was described in detail while talking about the 2D slices so there is no need to duplicate the characterization of them.

The section plane appearing in the scene is a simple plane object. It has the *MRTK* scripts *NearInteractionGrabbable*, *ObjectManipulator* and *ConstraintManager* added to it. Thanks to them it is possible to grab it and move around. However, to make it possible to create the cross section of the 3D model with it a dedicated *CrossSection* shader from Abdullah Aldandarawy is added to it (43). Moreover, the exact same shader has to be also selected for the lattice mesh in order to use the section plane. The working principle of this functionality is that every part of the mesh above the intersection of the section plane and the mesh disappears. In the early stage of the development the *MRTK* shaders were used for this functionality. However, they turned out not to be sufficient for the application since they were not able to perform capping of the internal areas of the cross section. Thanks to the free shaders uploaded by Abdullah Aldandarawy this asset could be introduced to the application.

The hand lamp occurring in the scene is a assembly of two cylinders and a directional light. The cylinder were chosen to be black and yellow in appropriate sizes in order to create a kind of metaphor to a torch. The hand lamp resembling an everyday object that has a very similar function in real world is making a favor to the end user who thanks to this can without any additional instruction figure out how to use this object. When it comes to the light source of the hand lamp, first idea was for it to be a spot light, which has a specified location and range over which the light falls off with constrained angle, resulting in a cone-shaped region of illumination and light fading away proportionally with the distance from the light source (44). It could perfectly reflect the type of light that comes out of a torch in real world environment. Unfortunately, *Unity* does not support rendering spotlights in scenes in a real time. It is only possible to see this type of light affecting a *Unity* scene if a scene is previously "baked", which means that a scene has to be rendered prior to running the application. It would not suit for the application as in the app it is essential that the light coming from the hand lamp affects the scene in a real time. What is more, practically only light that suit the requirement is the directional light. The directional light works similar to a sunlight that can be encounter in the real world environment. It does not have any specific source point in the scene, it only has a direction. Moreover, it does not diminish so the position of an object in the scene does not affect how much it is illuminated by the directional light. What is more, there can only be one active directional light at a time in a *Unity* scene. For the application it means that when the hand lamp is being switched on, the directional light that previously illuminated everything in the scene is temporarily disabled. Although this type of light does not resemble the light coming from a torch, it works fairly sufficient for the application and does not prevent the end user from examining some particular areas of the 3D model with the help of the light that they can control.

The two buttons of the 3D model mode are dedicated to switching of and off the section plane and the hand lamp. For the section plane the script *showHide* is used and for the hand lamp the script named *hideShow*. Both scripts are presented in the last pages of the dissertation.

4.4 3D Slicer

The computed tomography dataset contains slices cut only in one direction. In the application 2D slices mode the slices planes in the Unity scene are positioned in three different directions XY, XZ and YZ. To generate 3 sets of slices, every in different direction the software called 3D Slicer was used.

3D Slicer is a free and open-source software for image analysis and scientific visualization with a focus on clinical and biomedical applications. It has a fairly big community of knowledgeable users and developers working together to improve medical computing (45). Although the software targets at the medial user, it can be used for the industrial and the product development purposes just as well.

The *Screen Capture* module that was used is a feature that enables the user to collect the image sequence of slices of the examined object in any direction. This feature generated three image sequences that were later used in the application for displaying the CT slices on the plane objects in the 2D slices mode. The software was set to generate 131 slices in the XY direction, 501 in the XZ direction and 155 in the YZ direction for the lattice structure dataset and 327 slices in the XY direction, 43 in the XZ direction and 181 in the YZ direction for the lamp dataset. The image sequences generated are .PNG, 1000 x 1000 pixels files. The difference in the number of slices for each direction results from the facts that the objects are not cubic, having different dimensions in each direction, and it was set to keep the same distance between each slice for every direction. The chosen quantity of the slices provides a smooth translation from one slice to another and at the same time does not overload the application size.

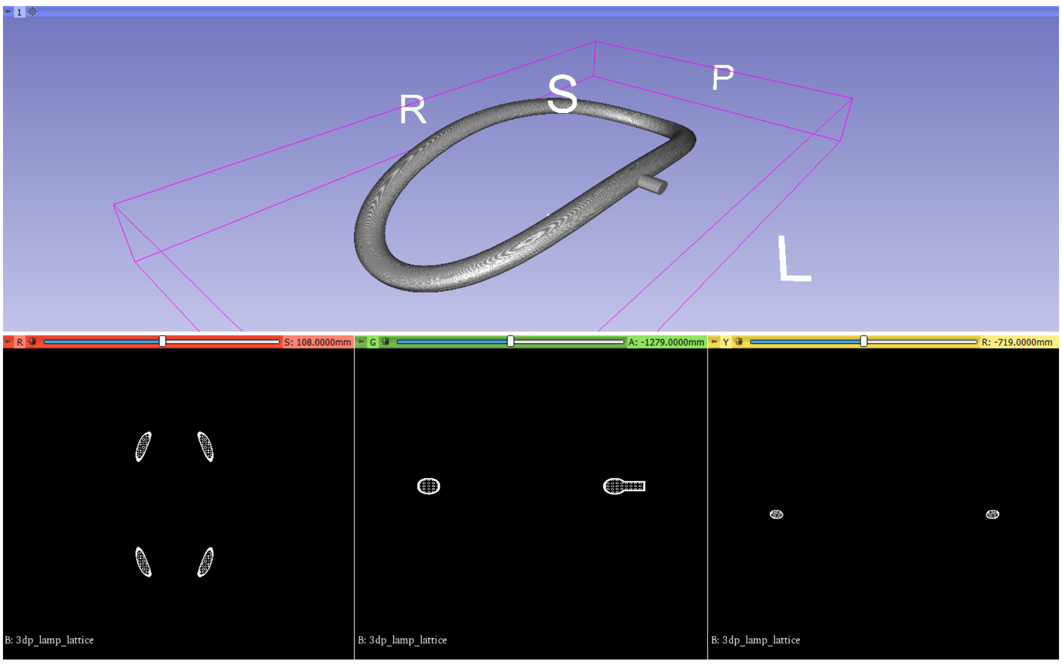

Figure 28. *3D Slicer* interface

5 Conclusions

The industrial CT scan technology is constantly growing. There is a strong need for the accurate quantitative inspection methods as much as for the accurate qualitative methods. The VR and AR technologies are growing even more rapidly, being used by the industries as digital entertainment, fashion, medical, just to name a few. The prices of head-mounted displays are decreasing and it is becoming more and more common for people to acquire this piece of technology for their households . Despite all these factors, not many scientific research or any has been done concerning the use of VR or AR technologies for visualizing industrial CT scans, which leaves an impression that this technology crossover is a total novelty. Nevertheless, there seems to be distinct assets that this can introduce to qualitative inspection of industrial CT datasets for the product development process and engineering. The fact that this technology merge is still in its infant phase means that a lot development is to be done there. The possible directions of the growth were shown in this dissertation with the emphasis on how the VR visualization of the CT data can be successfully adapted specifically to the process of the product design development. It was accented how visualizing the data in a VR environment is helpful for checking internal defects or possible deviation from the desired geometry in the case of the lattice structure specimen, and how providing a designer with an intuitive tool for checking internal structures without having to become an expert on reading CT scans can bring extra advantages in the case of the lamp.

The application visualizing the computed tomography data in the virtual reality that could be potentially used in various industries for qualitative inspections was introduced. The state of art was featured concerning VR, AR, CT scans, and all of them combined as a technology for the medical, design and engineering use cases. In the next part of the dissertation, it was presented how using the virtual reality technology as a visualization method through a digital VR application could bring added value to the process of viewing industrial CT scans datasets for the product development process. In the end, the whole process of developing the application within multiple pieces of software was presented in detail.

Although the application as it is now could be a valuable asset itself for the designers and engineering, there are some aspects that could be introduced in addition to elevate the value of the application even more. Firstly, the application could profit from having also an augmented reality mode. The user could examine the part using the model targeting thus seeing the internal structure of the part in its natural environment. Moreover, the 3D model could be generated as a 3D volume texture instead of the .OBJ mesh model as it is now. Nevertheless, the mesh approach is often more useful for engineering data as it suits more for calculating distances between the manufactured part and the CAD model. Additional features in the application could be a 3D paint brush for marking specific areas of an object and a multi-user mode to examine an object at the same time in a one virtual room by a few people from different locations.

References

1. The Nobel Prize in Physiology or Medicine 1979. *The Nobel Prize.* [Online] [Cited: October 1, 2021.] https://www.nobelprize.org/prizes/medicine/1979/summary/.

2. Rogowska, Jadwiga. Overview and Fundamentals of Medical Image Segmentation. [book auth.] Isaac Bankman. *Handbook of Medical Image Processing and Analysis.* 2008.

3. *The basic principles of computed tomographyand magnetic resonance imaging.* Karen S. Caldemeyer, Kenneth A. Buckwalter. 5, s.l. : Journal of the American Academy of Dermatology, 1999, Vol. 41.

4. Prokop, M., and M. Galanski. *Spiral and Multislice Computed Tomography of the Body.* 2003.

5. *What are the basic concepts of temporal, contrast, and spatial resolution in cardiac CT?* Eugene Lin, Adam Alessio. 6, s.l. : J Cardiovasc Comput Tomogr, 2009, Vol. 3.

6. *Industrial applications of computed tomography.* L. De Chiffre, S. Carmignato, J.-P. Kruth, R. Schmitt, A. Weckenmann. 63, s.l. : CIRP Annals - Manufacturing Technology, 2014.

7. *New X-ray computed tomography methods for research and industry.* Johann Kastner, Christoph Heinzl, Bernhard Plank, Dietmar Salaberger, Christian Gusenbauer, Sascha Senck. s.l. : 7th Conference on Industrial Computed Tomography, Leuven, Belgium (iCT 2017), 2017.

8. *Format DICOM – komunikacja pomiędzy systemami diagnostyki obrazowej i planowania leczenia radioterapii.* Osewski, W. ,Dolla, Ł. ,Hawrylewicz, L. ,Garbaciok, S. ,Ślosarek, K. 5, s.l. : Inżynier i Fizyk Medyczny, 2014, Vol. 3.

9. Sergei Vardomatski. Augmented And Virtual Reality After Covid-19. *Forbes.* [Online] September 14, 2021. [Cited: September 26, 2021.] https://www.forbes.com/sites/forbestechcouncil/2021/09/14/augmented-and-virtual-reality-after-covid-19/?sh=7f56cf052d97.

10. Maghan McDowell. Why AR clothing try-on is nearly here. *VOGUE BUSINESS.* [Online] July 26, 2021. [Cited: September 26, 2021.] https://www.voguebusiness.com/technology/why-ar-clothing-try-on-is-nearly-here.

11. Barrett Cordero. Could Augmented And Virtual Reality Be The Next Big Things In The Meetings And Events Industry? *Forbes.* [Online] July 27, 2021. [Cited: September 26, 2021.] https://www.forbes.com/sites/theyec/2021/07/27/could-augmented-and-virtual-reality-be-the-next-big-things-in-the-meetings-and-events-industry/?sh=296b3c1525ad.

12. Erickson, Mandy. Virtual reality system helps surgeons, reassures patients. *Stanford MEDICINE.* [Online] July 11, 2017. [Cited: September 26, 2021.] https://med.stanford.edu/news/all-news/2017/07/virtual-reality-system-helps-surgeons-reassures-patients.html.

13. *Current and future applications of virtual reality for medicine.* RICHARD M. SATAVA, SHAUN B. JONES. 3, s.l. : Proceedings of the IEEE, 1998, Vol. 86.

14. *Force and moment generating capacity of lower limb muscles before and after tendon.* S. L. Delp, F. R. Zajac. s.l. : Clinical Orthopaedics, 1992, Vol. 284.

15. *Virtual reality surgical simulator: The first steps.* R. M. Satava. s.l. : Surgical Endoscopy, 1993, Vol. 7.

16. *Electronic imaging of the human body. Data storage and interchange format standards.* Whitlock, V. M. Spitzer and D. G. s.l. : Proc. Electronic Imaging of the Human Body Working Group, 1992.

17. *The exploration of cross-sectional data with a virtual endoscope.* W. E. Lorensen, F. A. Jolesz, and R. Kikinis,. s.l. : Interactive Technology and the New Paradigm for Healthcare, 1995.

18. *Virtual Reality (VR) in Healthcare Market Size, Share & COVID-19 Impact Analysis, By Component, By Application, and Regional Forecast, 2021-2028.* s.l. : FORTUNE BUSINESS INSIGHTS, 2021.

19. FDA to host workshop on medical applications of AR, VR. [Online] January 23, 2020. [Cited: September 27, 2021.] https://www.auntminnie.com/index.aspx?sec=log&URL=https%3a%2f%2fwww.auntminnie.com%2findex.aspx%3fsec%3dsup%26sub%3dadv%26pag%3ddis%26ItemID%3d127955.

20. *Immersive 3D virtual reality imaging in planning minimally invasive and complex adult cardiac surgery.* Amir H. Sadeghi, Wouter Bakhuis, Frank Van Schaagen, Frans B.S. Oei, Jos A. Bekkers, Alexander P.W.M. Maat, Edris A.F. Mahtab, Ad J.J.C. Bogers, Yannick J.H.J. Taverne.

21. *Mixed reality computed tomography-based surgical planning for partial nephrectomy.* Soichiro Yoshida M.D., Ph.D., Maki Sugimoto M.D., Ph.D., Shohei Fukuda M.D., Naoji Taniguchi M.S., Kazutaka Saito M.D., Ph.D., Yasuhisa Fujii M.D., Ph.D.

22. *VR Multi-user Conference Room for Surgery Planning.* Oleksii Bashkanov, Patrick Saalfeld, Hariharasudhan Gunasekaran, Mathews Jabaraj, Bernhard Preim, Tobias Huber, Florentine H¨uttl, Werner Kneist, Christian Hansen. s.l. : 18. Jahrestagung der Deutschen Gesellschaft für Computer- und Roboterassistierte Chirurgie e.V., 2019.

23. JOHNS HOPKINS MEDICINE. *Johns Hopkins Performs Its First Augmented Reality Surgeries in Patients.* [Online] [Cited: 03 13, 2022.] https://www.hopkinsmedicine.org/news/articles/johns-hopkins-performs-its-first-augmented-reality-surgeries-in-patients.

24. *SlicerVR for Medical Intervention Training and Planning in Immersive Virtual Reality.* Csaba Pinter, Andras Lasso, Saleh Choueib, Mark Asselin, Jean-Christophe Fillion-Robin, Jean-Baptiste Vimort, Ken Martin, Matthew A. Jolley.

25. *Volumetric Medical Data Visualization for Collaborative VR Environments.* Roland Fischer, Kai-Ching Chang, René Weller, Gabriel Zachmann. s.l. : International Conference on Virtual Reality and Augmented Reality: EuroVR 2020: Virtual Reality and Augmented Reality, 2020.

26. *Accuracy of industrial computed tomography measurements: Experimental results from an international comparison.* Carmignato, Simone. 61, s.l. : CIRP Annals - Manufacturing Technology, 2012.

27. W, Sun, B, Brown S and K, Leach R. *An overview of industrial X-ray computed tomography.* Teddington : National Physical Laboratory, United Kingdom, 2011.

28. *Industrial Computed Tomography Market Report.* s.l. : GRAND VIEW RESEARCH, 2021.

29. *Special issue on the 6th conference on industrial computed tomography 2016 (iCT2016).* Kastner, Johann. s.l. : Case Studies in Nondestructive Testing and Evaluation, 2016.

30. *Visualizing Industrial CT Volume Data for Nondestructive Testing Applications.* Runzhen Huang, Kwan-Liu Ma, Patrick McCormick, William Ward. s.l. : IEEE Visualization Conference, 2003.

31. *Interior Surface Unwrapping System of Volume CT Data.* Kan Wang, Junye Luo, William G. Wee, Patrick Howard, Andrew Ferro. s.l. : International Conference on Computational Science and Computational Intelligence, 2015.

32. Research and Markets. *Industrial Computed Tomography Market - Growth, Trends, COVID-19 Impact, and Forecasts (2021 - 2026).* 2021.

33. *Applications of Virtual Reality in Engineering and Product Design: Why, What, How, When and Where.* Aurora Berni, Yuri Borgianni. s.l. : Electronics, 2020.

34. *Virtual Reality for Product Development in Manufacturing Industries.* Laura Schina, Mariangela Lazoi, Roberto Lombardo, Angelo Corallo. Lecce, Italy : Augmented Reality, Virtual Reality, and Computer Graphics - Third International Conference, AVR 2016.

35. *Ford's Virtual Reality Prototyping Reinvents Vehicle Design.* Baron, Elizabeth. Highlands Ranch, Colorado, USA : Product Design & Development, 2016.

36. *Action Research on Virtual-Reality-Assisted Product and Process Design.* Angelo Corallo, Mariangela Lazoi, Gabriele Papadia, Claudio Pascarelli. s.l. : IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT, 2020.

37. Lavik, Matias. mlavik1/UnityVolumeRendering. *GitHub.* [Online] [Cited: 03 29, 2022.] https://github.com/mlavik1/UnityVolumeRendering.

38. [Online] [Cited: 03 01, 2022.] https://store.unity.com/products/unity-personal.

39. [Online] [Cited: 03 01, 2022.] https://www.statista.com/statistics/1249850/oculus-quest-2-units-sold-by-quarter/.

40. What is the Mixed Reality Toolkit. [Online] [Cited: 03 05, 2022.] https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/?view=mrtkunity-2021-05.

41. *Oculus Integration.* [Online] [Cited: 03 05, 2022.] https://assetstore.unity.com/packages/tools/integration/oculus-integration-82022#description.

42. Free Double Sided Shaders. *Unity Asset Store.* [Online] Ciconia Studio. [Cited: 03 08, 2022.] https://assetstore.unity.com/packages/vfx/shaders/free-double-sided-shaders-23087.

43. Cross Section. *Unity Asset Store.* [Online] [Cited: 03 09, 2022.] https://assetstore.unity.com/packages/vfx/shaders/cross-section-66300.

44. Types of light. *Unity Documentation.* [Online] [Cited: 03 09, 2022.] https://docs.unity3d.com/Manual/Lighting.html.

45. 3D Slicer image computing platform. *What is 3D Slicer?* [Online] [Cited: 03 12, 2022.] https://www.slicer.org/.

List of abbreviations

CT – Computed Tomography

VR – Virtual Reality

AR – Augmented Reality

HMT – Head-Mounted Display

CAGR - Compound Annual Growth Rate

NDT – Nondestructive Testing

Table of figures

C# scripts

reset

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class reset : MonoBehaviour
{
    public void resetScene()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
    }

    void Start()
    {

    }

    void Update()
    {

    }
}
```

hideShow

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class hideShow : MonoBehaviour
{
    public GameObject obj;

    void Start()
    {
        obj.SetActive(false);
    }

    void Update()
    {

    }

    public void whenButtonClicked()
    {
        if (obj.activeInHierarchy == true)
            obj.SetActive(false);
        else
            obj.SetActive(true);
    }
}
```

hideShow2

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class hideShow2 : MonoBehaviour
{
    public GameObject obj1;
    public GameObject obj2;

    void Start()
    {
        obj1.SetActive(false);
        obj2.SetActive(false);
    }

    void Update()
    {

    }

    public void whenButtonClicked()
    {
        if (obj1.activeInHierarchy == true)
            obj1.SetActive(false);
        else
            obj1.SetActive(true);

        obj2.SetActive(false);
    }
}
```

showHide

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class showHide : MonoBehaviour
{
    public GameObject obj1;
    public GameObject obj2;
    public GameObject plane;

    void Start()
    {
        obj1.SetActive(true);
        obj2.SetActive(false);
        plane.SetActive(false);
    }

    void Update()
    {

    }

    public void whenButtonClicked()
    {
        if (obj1.activeInHierarchy == true)
        {
            obj1.SetActive(false);
            obj2.SetActive(true);
            plane.SetActive(true);
        }
        else
        {
            obj1.SetActive(true);
            obj2.SetActive(false);
            plane.SetActive(false);
        }
    }
}
```

showHide2

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class showHide2 : MonoBehaviour
{
    public GameObject obj1;
    public GameObject obj2;

    void Start()
    {
        obj1.SetActive(true);
        obj2.SetActive(true);
    }

    void Update()
    {

    }

    public void whenButtonClicked()
    {
        if (obj1.activeInHierarchy == true)
            obj1.SetActive(false);
        else
            obj1.SetActive(true);

        if (obj2.activeInHierarchy == true)
            obj2.SetActive(false);
        else
            obj2.SetActive(true);
    }
}
```

sliceXYLattice

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System;
using TMPro;

public class sliceXY : MonoBehaviour
{
    Vector3 tempPos;

    Texture2D myTexture;
    float sliceNumber;

    public TextMeshProUGUI slicePositionText;

    void Start()
    {

    }

    void Update()
    {
        tempPos = transform.position;

        if (tempPos.z > 0.1443f)
        {
            tempPos.z = 0.1443f;
            sliceNumber = 132f;
        }
        else if (tempPos.z < -0.1443f)
        {
            tempPos.z = -0.1443f;
            sliceNumber = 0f;
        }
        else
        {
            sliceNumber = Convert.ToSingle(Math.Round(tempPos.z / 0.0022041977443609f +
67f));
        }
        transform.position = tempPos;

        if (sliceNumber < 10)
        {
            myTexture = Resources.Load("XY/image_0000" + sliceNumber) as Texture2D;
        }
        else if (sliceNumber < 100)
        {
            myTexture = Resources.Load("XY/image_000" + sliceNumber) as Texture2D;
        }
        else
        {
            myTexture = Resources.Load("XY/image_00" + sliceNumber) as Texture2D;
        }

        GetComponent<Renderer>().material.mainTexture = myTexture;

        slicePositionText.text = Math.Round(tempPos.z*1000,2).ToString() + " mm";
    }
}
```

sliceXZLattice

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System;
using TMPro;

public class sliceXZ : MonoBehaviour
{
    Vector3 tempPos;

    Texture2D myTexture;
    float sliceNumber;

    public TextMeshProUGUI slicePositionText;

    void Start()
    {

    }

    void Update()
    {
        tempPos = transform.position;

        if (tempPos.y > 0.5463f)
        {
            tempPos.y = 0.5463f;
            sliceNumber = 500f;
        }
        else if (tempPos.y < -0.5463f)
        {
            tempPos.y = -0.5463f;
            sliceNumber = 0f;
        }
        else
        {
            sliceNumber = Convert.ToSingle(Math.Round(tempPos.y / 0.00218962075848303f +
251f));
        }
        transform.position = tempPos;

        if (sliceNumber < 10)
        {
            myTexture = Resources.Load("XZ/image_0000" + sliceNumber) as Texture2D;
        }
        else if (sliceNumber < 100)
        {
            myTexture = Resources.Load("XZ/image_000" + sliceNumber) as Texture2D;
        }
        else
        {
            myTexture = Resources.Load("XZ/image_00" + sliceNumber) as Texture2D;
        }

        GetComponent<Renderer>().material.mainTexture = myTexture;

        slicePositionText.text = Math.Round(tempPos.y*1000, 2).ToString() + " mm";
    }
}
```

sliceYZLattice

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System;
using TMPro;

public class sliceYZ : MonoBehaviour
{
    Vector3 tempPos;

    Texture2D myTexture;
    float sliceNumber;

    public TextMeshProUGUI slicePositionText;

    void Start()
    {

    }

    void Update()
    {
        tempPos = transform.position;

        if (tempPos.x > 0.1664f)
        {
            tempPos.x = 0.1664f;
            sliceNumber = 154f;
        }
        else if (tempPos.x < -0.1664f)
        {
            tempPos.x = -0.1664f;
            sliceNumber = 0f;
        }
        else
        {
            sliceNumber = Convert.ToSingle(Math.Round(tempPos.x / 0.00217641161290323f +
78f));
        }
        transform.position = tempPos;

        if (sliceNumber < 10)
        {
            myTexture = Resources.Load("YZ/image_0000" + sliceNumber) as Texture2D;
        }
        else if (sliceNumber < 100)
        {
            myTexture = Resources.Load("YZ/image_000" + sliceNumber) as Texture2D;
        }
        else
        {
            myTexture = Resources.Load("YZ/image_00" + sliceNumber) as Texture2D;
        }

        GetComponent<Renderer>().material.mainTexture = myTexture;

        slicePositionText.text = Math.Round(tempPos.x*1000, 2).ToString() + " mm";
    }
}
```

sliceXYLamp

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System;
using TMPro;

public class sliceXYLamp : MonoBehaviour
{
    Vector3 tempPos;

    Texture2D myTexture;
    float sliceNumber;

    public TextMeshProUGUI slicePositionText;

    void Start()
    {

    }

    void Update()
    {
        tempPos = transform.position;

        if (tempPos.z > 0.314f)
        {
            tempPos.z = 0.314f;
            sliceNumber = 326f;
        }
        else if (tempPos.z < -0.3144f)
        {
            tempPos.z = -0.3144f;
            sliceNumber = 0f;
        }
        else
        {
            sliceNumber = Convert.ToSingle(Math.Round(tempPos.z / 0.00192293577981651f +
163f));
        }
        transform.position = tempPos;

        if (sliceNumber < 10)
        {
            myTexture = Resources.Load("LAMP/XY/image_0000" + sliceNumber) as Texture2D;
        }
        else if (sliceNumber < 100)
        {
            myTexture = Resources.Load("LAMP/XY/image_000" + sliceNumber) as Texture2D;
        }
        else
        {
            myTexture = Resources.Load("LAMP/XY/image_00" + sliceNumber) as Texture2D;
        }

        GetComponent<Renderer>().material.mainTexture = myTexture;

        slicePositionText.text = Math.Round(tempPos.z * 1000, 2).ToString() + " mm";
    }
}
```

sliceXZLamp

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System;
using TMPro;

public class sliceXZLamp : MonoBehaviour
{
    Vector3 tempPos;

    Texture2D myTexture;
    float sliceNumber;

    public TextMeshProUGUI slicePositionText;

    void Start()
    {

    }

    void Update()
    {
        tempPos = transform.position;

        if (tempPos.y > 0.0415f)
        {
            tempPos.y = 0.0415f;
            sliceNumber = 42f;
        }
        else if (tempPos.y < -0.042f)
        {
            tempPos.y = -0.042f;
            sliceNumber = 0f;
        }
        else
        {
            sliceNumber = Convert.ToSingle(Math.Round(tempPos.y / 0.00195348837209302f +
21f));
        }
        transform.position = tempPos;

        if (sliceNumber < 10)
        {
            myTexture = Resources.Load("LAMP/XZ/image_0000" + sliceNumber) as Texture2D;
        }
        else if (sliceNumber < 100)
        {
            myTexture = Resources.Load("LAMP/XZ/image_000" + sliceNumber) as Texture2D;
        }
        else
        {
            myTexture = Resources.Load("LAMP/XZ/image_00" + sliceNumber) as Texture2D;
        }

        GetComponent<Renderer>().material.mainTexture = myTexture;

        slicePositionText.text = Math.Round(tempPos.y * 1000, 2).ToString() + " mm";
    }
}
```

sliceYZLamp

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System;
using TMPro;

public class sliceYZLamp : MonoBehaviour
{
    Vector3 tempPos;

    Texture2D myTexture;
    float sliceNumber;

    public TextMeshProUGUI slicePositionText;

    void Start()
    {

    }

    void Update()
    {
        tempPos = transform.position;

        if (tempPos.x > 0.177f)
        {
            tempPos.x = 0.177f;
            sliceNumber = 180f;
        }
        else if (tempPos.x < -0.1775f)
        {
            tempPos.x = -0.1775f;
            sliceNumber = 0f;
        }
        else
        {
            sliceNumber = Convert.ToSingle(Math.Round(tempPos.x / 0.00196132596685083f +
90f));
        }
        transform.position = tempPos;

        if (sliceNumber < 10)
        {
            myTexture = Resources.Load("LAMP/YZ/image_0000" + sliceNumber) as Texture2D;
        }
        else if (sliceNumber < 100)
        {
            myTexture = Resources.Load("LAMP/YZ/image_000" + sliceNumber) as Texture2D;
        }
        else
        {
            myTexture = Resources.Load("LAMP/YZ/image_00" + sliceNumber) as Texture2D;
        }

        GetComponent<Renderer>().material.mainTexture = myTexture;

        slicePositionText.text = Math.Round(tempPos.x * 1000, 2).ToString() + " mm";
    }
}
```