Master’s thesis at the Lucerne School of Engineering and Architecture

Title	**Color-coded Components for patient-specific Applications in Medical Technologies**
Student	**Rohrer, Daniel**
Master’s degree program	**Master of Science in Engineering**
Semester	**FS20**
Lecturer	**Haack, Carsten**
External Examiner	**Mastrogiacomo, Giovanni**

Abstract German
Diese Arbeit enthält die allgemeine Prozessbeschreibung von der Erstellung und dem Lesen medizinischer Bilder (DICOM), der Segmentierung von Teilen des menschlichen Körpers auf der Grundlage dieser Bilder, der Auftragung von farbcodierten Informationen auf diese Modelle und der Verwendung der Modelle für den Mehrfarbendruck (4D) und weitere Anwendungsfälle. Die für alle diese Prozessschritte verwendeten Informationen sind patientenspezifisch oder könnten für patientenspezifische Komponenten verwendet werden.

Abstract English
This paper provides the overall process description from creating and reading of medical images (DICOM), the segmentation of parts of the human body based on those images, the application of color-coded information onto these models and the use of the models for multicolor (4D) printing and further use cases. All of these process steps and the information used are patient-specific or could be used for patient-specific components.

Place, Date Horw, 19.06.2020

© Daniel Rohrer, Lucerne School of Engineering and Architecture

---

All right reserved. The master’s thesis or parts thereof may be not reproduced in any way nor stored digitally, processed, copied or distributed without the written approval of the copyright holder.

If the thesis is published online on the website of the Lucerne University of Applied Sciences and Arts, then other conditions of use in connections with Creative Commons licenses may apply. The Creative Commons license shown on the website applies in this case.

MSE – Master Thesis

Daniel Rohrer
Itiweg 14
6072 Sachseln
+41 79 706 14 21

daniel.rohrer@stud.hslu.ch

Color-coded Components for patient-specific Applications in Medical Technologies

Advisor:
Prof. Dr. Carsten Haack
Institute for Machine- and Energy Technology IME

Expert:
Dr. Giovanni Mastrogiacomo
Head of Materials and Manufacturing Technology
Kistler Group

Industrial Partners:
Prof. Dr. Roger Abächerli
Institute for Medical Technology IMT
Lucerne University of Applied Sciences & Arts

Peacs BV, The Netherlands

Lucerne University of Applied Sciences & Arts
Master of Science in Engineering in Industrial Technologies (MSE)

Horw, 19.06.2020

Abstract

This paper provides the overall process description from creating and reading of medical images (DICOM), the segmentation of parts of the human body based on those images, the application of color-coded information onto these models and the use of the models for multicolor (4D) printing and further use cases. All of these process steps and the information used are patient-specific or could be used for patient-specific components.

Firstly, the general overview of the medical imaging methods provides a foundation of the general physics behind, its possibilities and limitations. The imaging information handling and the detailed description of its output format (DICOM) do provide further information required for the following steps, especially in regard of the segmentation process and the data consistency (meta data) throughout the overall process up to the printed models.

Secondly, various tools with different capabilities which are able to read and process the DICOM format are discussed. The capabilities range between reading and export only image information; reading, segmentation and export of 3D model; reading, segmentation, running simulations and export of result model; and reading, use information for various process steps and export results (e.g. image or 3D model). For the goal of this paper, segmentation software (e.g. 3DSlicer or GeomPeacs) and MATLAB for processing the data provide the toolset required. An advanced image recognition script able to segment the DICOM images in MATLAB would no longer require the use of the segmentation software, but this does not exist yet.

The first use case of this process is the Alvale project to enable the 4D printing of segmented hearts and applied color-codes representing timestamps of the electric signal of a PVC or VT on the heart surface measured with ECG (VIVO). Various already segmented models are converted to the output format VRML to validate the conversion tool as the process ahead has been already developed by Peacs.

The second use case to validate the process is the export of a VRML model from other patient-specific data sets, namely the simulation of the ventricular blood flow in a blood vessel and the export an 4D print of the color-coded results. Furthermore, a more generic approach with an industry standard simulation tool (ANSYS) was developed and the overall process with the export of the simulation results and subsequent printing could be carried out successfully.

Eventually, the results achieved show the validity of the process for patient-specific applications of color-coded components in medical technologies. Three use cases with different approaches of color application could be provided where the process has to be adapted, but remains the same in general. They are additionally used for a proposal of further use cases such as web-based approaches for the process. The current limitations for the automatization are outlined such as the lack of fully automated image recognition for the segmentation as well and what could be undertaken to address these issues.

Glossary

APDL	Ansys Parametric Design Language
ARVD	Arrhythmogenic Right Ventricular Dysplasia
ASCII	American Standard Code for Information Interchange
AV	Atrioventricular
CA	Contrast Agents
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CCD	Charge-coupled Device
CFD	Computational Fluid Dynamics
CHTML	Compact HTML file format
CNT	Cognition Network Technology
CRT	Cathode-Ray-Tube
CT	Computer Tomography
Coordinate Vertice	see Vertex
DDL	Digital Driving Level
DEP	Windows Dependency file format
DICOM	Digital Imaging and Communications in Medicine file format
DQE	Detective Quantum Efficiency
DSA	Digital Subtraction Angiography
ECG	Electrocardiography
EEG	Electroencephalogram
EMG	Electromyogram
FEM	Finite Element Model
GSDF	Grayscale Standard Display Function
GUI	Graphical User Interface
HTML	Hypertext Markup Language file format
HU	Hounsfield Unit
IOD	Information Object Definition
JND	Just Noticeable Difference
LUT	Look-Up-Table
MDL	MathWorks Simulink Model file format

MRE	Magnetic Resonance Elastography
MRI	Magnetic Resonance Inferometer
NM	Nuclear Medicine
OB	Other Byte
OBJ	Wavefront 3D Object file format
ODT	Open Document Text file format
PACS	Picture Archiving and Communication Systems
PVC	Premature Ventricular Contraction
RVOT	Recurrent Ventricular Arrhythmia Outflow Tract
SOP	Service-Object Pair
SPECT	Single-photon Emission Computed Tomography
STEP	Standard for the Exchange of Product Model Data file format
STL	Standard Triangulation Language file format
TRI	Triangle Mesh file format
TXT	Windows Text file format
UI	Unique Identifier
UL	Unsigned Binary Integer
Vertex	a point where higher-dimensional geometric objects meet, a point in space with additional attributes (edge point of polygon)
VOI	Value of Interest
Voxel	Volumetric Element, representing a value on a regular grid in three-dimensional space
VR	Value Representation
VRML	Virtual Reality Modeling Language (referring to VRML2.0)
VT	Ventricular Tachycardia
VTK	Visualization Toolkit Legacy file format
VTP	Visualization Toolkit Polygonal Data file format
VTU	Visualization Toolkit Unstructured Data file format
X3D	Extensible 3D file format
XML	Extensible Markup Language file format

Table of Contents

1 Introduction ... 1

1.1 Initial Situation ... 1

1.2 Objective ... 1

1.3 Problem Definition ... 1

1.4 Structure of the Paper ... 2

2 Basic Overview of the Image Creating Process of Medical Devices ... 3

2.1 Magnetic Resonance Imaging [3] ... 3

2.1.1 Basics of the Magnetic Resonance Imaging ... 3

2.1.2 Relaxation Phenomena ... 4

2.1.3 MR Imaging Technology and Applications ... 6

2.1.4 Clinical Applications of MRI ... 8

2.1.5 MRI Compatibility ... 8

2.1.6 Artifact Formation ... 9

2.2 Medical Imaging [4] ... 10

2.2.1 X-ray Imaging ... 10

2.3 Computer Tomography ... 15

2.4 Image Analysis in Medicine and Biology [5] ... 18

2.4.1 Introduction ... 18

2.4.2 Application of Cognition Network Technology for Image Analysis in Medicine and Biology ... 18

3 Electrical Phenomena of the Body and their Detection [6] ... 20

3.1 The Electrocardiogram ... 20

3.2 Application Example: Electrocardiography ... 23

4 Description of the DICOM Format ... 25

4.1 Information Object Definition ... 27

4.1.1 DICOM Information Model ... 27

4.1.2 DICOM Model of the Real World ... 28

4.2 Basic Structure ... 31

4.2.1 Value Representations ... 31

4.2.2 The Data Set ... 32

4.2.3 Unique Identifiers (UIDs) ... 34

4.3 Media Storage and File Format for Media Interchange ... 35

4.3.1 DICOM Models for Media Storage ... 35

4.3.2 DICOM File Format ... 36

4.4 Media Formats and Physical Media for Media Interchange ... 38

4.4.1 PC File System ... 38

4.5 Grayscale Standard Display Function ... 39

4.5.1 Definitions ... 39
4.5.2 Overview ... 39
4.5.3 The Grayscale Standard Display Function ... 40
4.5.4 The Process to the Presentation Look-Up-Table (Pixel Transformation Sequence) ... 42
4.6 Important Meta Data Attributes ... 43
4.7 Effects of Meta Data Attributes on Image Display ... 47
5 Reading DICOM Data Sets ... 50
5.1 MATLAB ... 50
5.1.1 Reading File Information ... 50
5.1.2 Display of the File Information ... 51
5.2 3DSlicer ... 53
5.2.1 Segmentation ... 53
5.2.2 Segmentation in 3DSlicer ... 54
5.2.3 Export of Segmentation ... 57
5.3 SimVascular ... 58
5.4 ezDICOM ... 59
5.5 GeomPeacs ... 60
6 Alvale Data Conversion Process ... 61
6.1 Geometry Generation by Peacs ... 61
6.2 ECG Data Generation by Peacs ... 61
6.3 Data Set Conversion for 4D-printable Format ... 63
6.4 Implementation of Patient Information ... 66
6.5 Conversion of other Alvale medical cases ... 69
6.6 Resolution of Data Sets ... 72
6.7 Discussion on the automation of the overall process ... 74
7 Further application for color-coded, patient-specific data ... 76
7.1 Current use in other specific use cases ... 76
7.2 Application of patient-specific color-codes to selected cases ... 82
8 Conclusion and Outlook ... 89

1 Introduction

The task of the heart is to supply the cells in the body with oxygen- and nutrient-rich blood in order to ensure their function. The heart pumps the blood through the body through regular contractions. At patients with cardiac arrhythmias, these contractions are disturbed, resulting in an uncoordinated, accelerated heart rate. Due to the resulting drop in blood pressure, dizziness and attacks of weakness can occur.

1.1 Initial Situation

The Alvale project aims to make the treatment of cardiac arrhythmias simpler and faster. At present, the localization of the origin of the cardiac arrhythmia is carried out during the treatment, which takes a lot of time. The Alvale project is developing medical diagnostics that make it possible to localize the origin of the cardiac arrhythmia in a patient-specific way even before treatment. This simplifies the planning of the treatment and you save a lot of time, which is also advantageous for the patient in terms of treatment and room costs. The data is obtained by patient-specific medical imaging & digital data processing and by a patient-specific ECG.

As part of the Alvale project, a 3D model of the heart based on patient data is to be developed, on which the origin of the cardiac arrhythmia can be seen in color. This color-coded model is to be produced using an additive manufacturing process.

1.2 Objective

The Alvale project carries out the segmentation of MRI data and supplements (MATLAB-based) information from ECG data. The further processing in common CAD systems and by means of 3D printers is missing and should be supplemented.

In a preliminary project [1] the basics and the state of the art in the field of object data generation and further processing with corresponding standards (e.g. VRML) were developed, as well as interfaces to common 3D-CAD systems and color-coded 3D printing (4D-Print). The preliminary project delivered specifications for suitable and common data formats for MATLAB programming of the interface Alvale CAD / 4D-Print. The specifications were checked exemplarily and a test case was defined with the project partners.

1.3 Problem Definition

Overall, there are three basic topics addressed in this paper:

Alvale Procedure / Data Consistency

On the basis of the abovementioned project status, the existing, current procedure is to be optimized for further data sets (times of heart rhythm), especially with regard to parameters such as Geometric resolution, discretization, material, coloring and color gradient.

DICOM Data from Segmentation

An important aspect in data generation is the step from medical MRI/CT data to geometric data. The basics of this segmentation in the Alvale project and other applications are to be investigated for the later 3D printing application (tool: 3DSlicer).

Possible Extended Application for Color-coded, Patient-specific Data

Within the framework of the Master Thesis, it will be investigated in which form data sets from other applications are available and can be adapted for 3D color printing if necessary. Possible applications are in the area of blood flow (e.g. wall shear stress) or in the area of bone/cartilage degeneration (osteoarthritis etc.). These can be fluctuations in tension, pressure, density distribution or thickness / height progression

1.4 Structure of the Paper

As an introduction to the work, important basics are first explained which serve as an underlying baseline for the understanding of the project. This comprises an overview of the different medical imaging technologies used today with its underlying physics, the machines and imaging processes. Besides that, the basics of electrocardiography are explained to give a better understanding of the medical cases which are used for this paper.

Furthermore, the output image of the medical imaging (DICOM) is closely looked at regarding standards, format and embedded information. This is the raw data used for the further processes downstream and the detailed knowledge is essential for understanding the possibilities and the restrictions present to the process.

The first step of the downstream process is the reading of the images and accessing its information, either the image data itself or the meta data. Several tools are discussed which have been used to obtain this information, including segmentation software which is used to create 3D models with the help of image recognition.

The next part is designated to the Alvale project. The overall process of segmentation, ECG data generation and its combination in one single color-coded 3D model are explained in detail. The data consistency is addressed for the output files as these files have to be able to be traced back to the initial raw data. The overall process and data consistency are then validated with several provided data sets by Peacs with different medical cases.

As the Alvale project is a very specific use case for the application of patient-specific, color-coded components, other possible use cases are discussed. This includes medical diagnostics and medical simulations, and two of the use cases are carried out to validate them including the 4D printing of the output components.

Eventually, a conclusion on the paper is given with the lessons learned and the achieved results. Moreover, future prospects are discussed with several possible use cases and the current limitations to be overcome.

2 Basic Overview of the Image Creating Process of Medical Devices

Medical devices do create images of the patients' body with several approaches. As the underlying data sets of the processed images are stored in the DICOM format, this chapter focusses in the devices which generate this data, namely Magnetic Resonance Inferometers and Computer Tomography scanners (or X-ray scanners). The chapter summarizes the information from [2] relevant to this paper.

2.1 Magnetic Resonance Imaging [3]

The Magnetic Resonance Imaging (MRI) has developed into an important imaging procedure in modern medicine. In comparison to classical X-ray examinations or computer tomography (CT), high-quality imaging of the inside of the body is possible without the use of ionizing radiation. The MRI images generated show excellent soft tissue contrast and in addition, the physician can freely select the plane to be imaged.

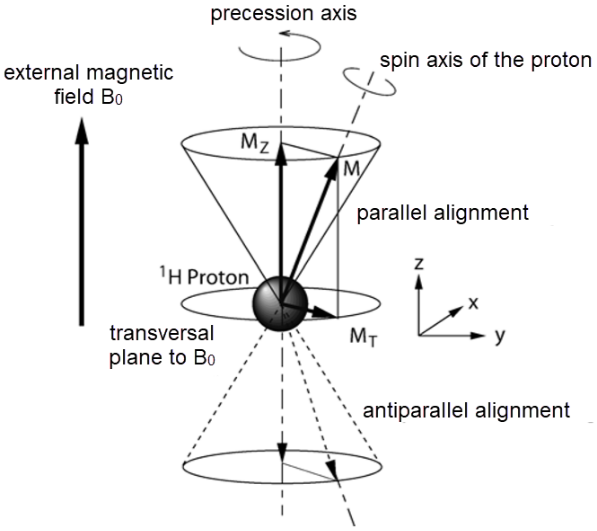


Figure 1 - Spin of a proton in a magnetic field [3]

The MRI method is based on the principle of nuclear magnetic resonance. Static magnetic fields with a strength of 0.2 to 3.0 Tesla are used for this purpose. Many nuclear magnetic resonance processes can only be described comprehensively with the aid of quantum physics. Nevertheless, classical physics is sufficient in most cases to predict system behavior.

2.1.1 Basics of the Magnetic Resonance Imaging

Atomic nuclei of odd mass number with unsaturated nuclear spins (e.g. ^{1}H, ^{13}C, ^{17}O, ^{19}F, ^{23}Na and ^{31}P) exhibit a magnetic dipole moment. For this reason, they interact with external magnetic fields. Due to their high natural abundance in human tissues and the large gyro-magnetic ratio γ hydrogen atoms (^{1}H) are best suited for MRI.

Energy Levels and Occupancy Rates

The nuclear spin of a hydrogen atom (Fig. 1) is oriented either parallel or antiparallel to a stationary magnetic field B_0 after thermal equilibrium has been reached. Since it is a quantum system at the atomic level, no other stable energy levels exist. The parallel alignment is the more energetically favorable, whereby the frequency of the two states varies depending on the strength of the external magnetic field and the temperature.

Larmor Frequency

If a magnetic field with a different orientation is applied around a charged core, which rotates like a gyroscope around its center of gravity in the spin axis, the spin axis begins to precess at the Larmor frequency. The Larmor frequency ω_0 is proportional to the field strength and is calculated according to equation (1). B_0 is the field strength of the external magnetic field in Tesla, γ the gyromagnetic ratio in MHz/T and ω_0 the Larmor frequency in MHz.

$$\omega_0 = \gamma B_0 \qquad (1)$$

2.1.2 Relaxation Phenomena

The MRI method uses the dependence of the Larmor frequency ω_0 on the external magnetic field B_0 and the splitting of the energy levels into two stable states: by excitation by means of a high-frequency electromagnetic pulse with the Larmor frequency, a transition from parallel to energetic antiparallel alignment can be forced. Only those nuclear spins are excited at any one time which exactly fulfil the resonance condition of the Larmor frequency in equation (1). The magnetization M in the fixed coordinate system (x, y, z) describes a spiral track on the surface of a sphere.

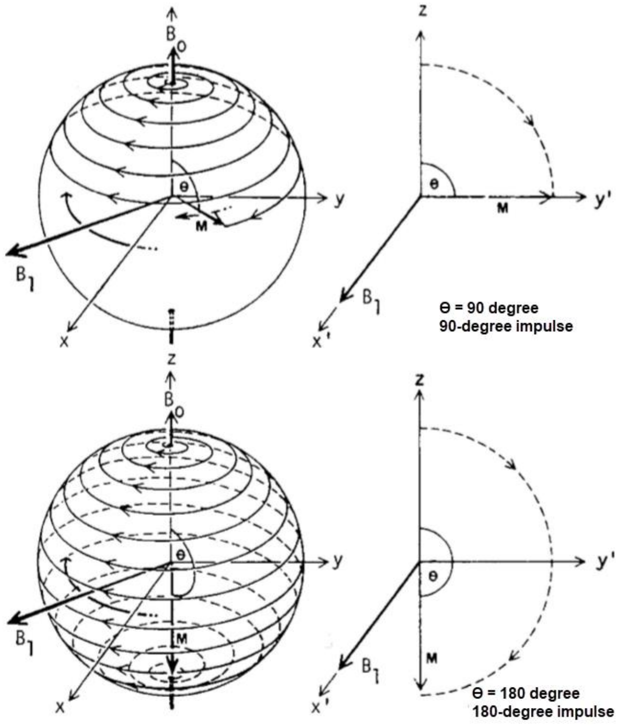


Figure 2 - Spherical Coordinate System [3]

Since the excited nuclei are in an unstable or metastable state, the absorbed energy is released in the form of electromagnetic waves at the Larmor frequency until thermal equilibrium is reached again. This process is called relaxation. Two independent effects, longitudinal (T_1) and transverse (T_2) relaxation, can be observed and described by the solutions of Bloch's equations. The relaxation times T_1 and T_2 are a measure for the speed of energy transfer: the faster the absorbed energy can be transferred to neighboring nuclei, the sooner the spin systems will reach their equilibrium state again. The longitudinal (T_1) and transverse (T_2) relaxation times describe the time, i.e., the intensity with which the energy is transferred.

T_1-Relaxation

After a disturbance of the spin system, the difference in the occupation number of the allowed energy levels and thus the magnetization M_Z in the direction of the external magnetic field is adjusted again according to the following relationship:

$$M_Z(t) = M_0 + \left(M_{Z,0} - M_0\right) * e^{\left(-\frac{t}{T_1}\right)} \qquad (2)$$

M_0 denotes the total magnetization in the thermal equilibrium state, $M_{Z,0}$ denotes the magnetization along the z-axis (longitudinal magnetization) immediately after the perturbation, t denotes the time after the perturbation, and T_1 denotes the longitudinal or Spin-Grid-Relaxation time. For 90° pulses, $M_{Z,0} = 0$, for 180° pulses, $M_{Z,0} = -M_0$.

The term longitudinal relaxation time expresses that only the component of the magnetization parallel to the external magnetic field is rebuilt with the time constant T_1 (Fig. 3). The synonymous term spin-lattice relaxation illustrates that energy is exchanged between the spins and the environment during the time T_1 in fluid media, neighboring molecules or atoms.

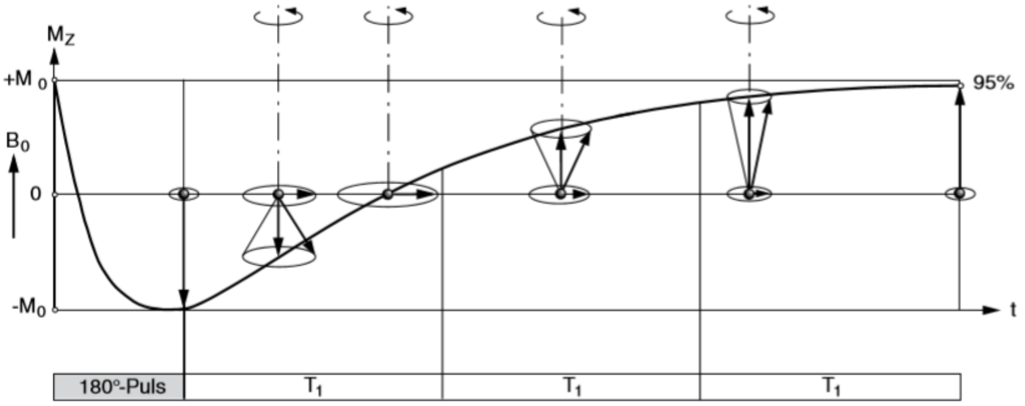


Figure 3 - Longitudinal relaxation after a 180° pulse with time constant T_1 [3]

T_2 Relaxation

For the spin signal only the transverse component M_T of the magnetization is important. By definition, it rotates in the x-y plane of the fixed coordinate system and can therefore induce an alternating voltage in a receiving coil at the Larmor frequency of the precessing spin (Fig. 4). The MRI thus registers an oscillating signal proportional to the transverse magnetization M_T and decaying with the spin-spin relaxation time T_2, the so-called Free Induction Decay Signal (FIDS). The term spin-spin relaxation time makes it clear that during the time T_2 energy is only shifted within the spin system: After a 90° pulse, all spins in the x-y plane rotate in phase, since they were rectified by the RF pulse. Because each individual atom itself generates a weak local magnetic field, each atom is in a slightly different magnetic field.

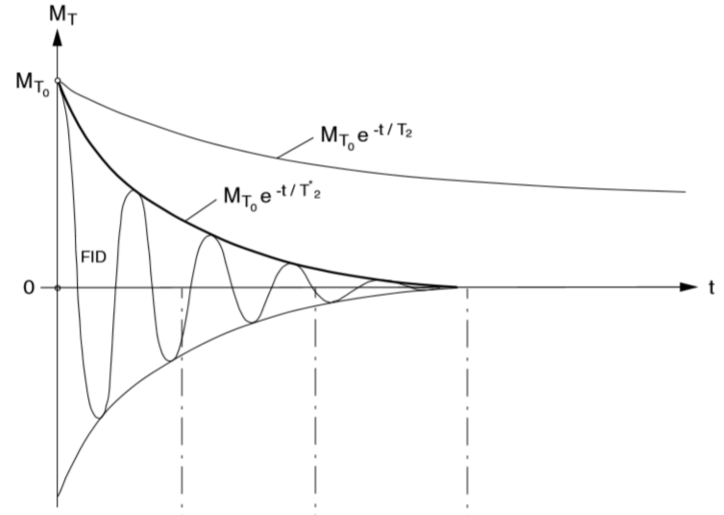


Figure 4 - T_2 Relaxation [3]

According to equation (3), the spins therefore precede with different Larmor frequencies, because B_0 is different for each atom. As a result, phase coherence is broken and the spin signals begin to cancel each other out. In a real system the spins are brought out of phase much faster than in the ideal case due to inhomogeneities of the external magnetic field. Accordingly, the signal decay is also faster. For this reason, the real time constant T_2^* is always smaller than ideal T_2.

$$M_T(t) = M_{T0} * e^{(-\frac{t}{T_2^*})} \qquad (3)$$

M_{To} denotes the transverse magnetization at time t=0. The transverse relaxation (even without inhomogeneity of the external magnetic field) is always the faster process than the reconstruction of the longitudinal magnetization: T_1 is always greater than T_2.

G_S selects a layer; G_P and G_F further breaks this down into individual voxels, which all have their own combination of Larmor frequency and phase. The MRI scans the different signals and assigns each voxel a grey level according to the received spin signal

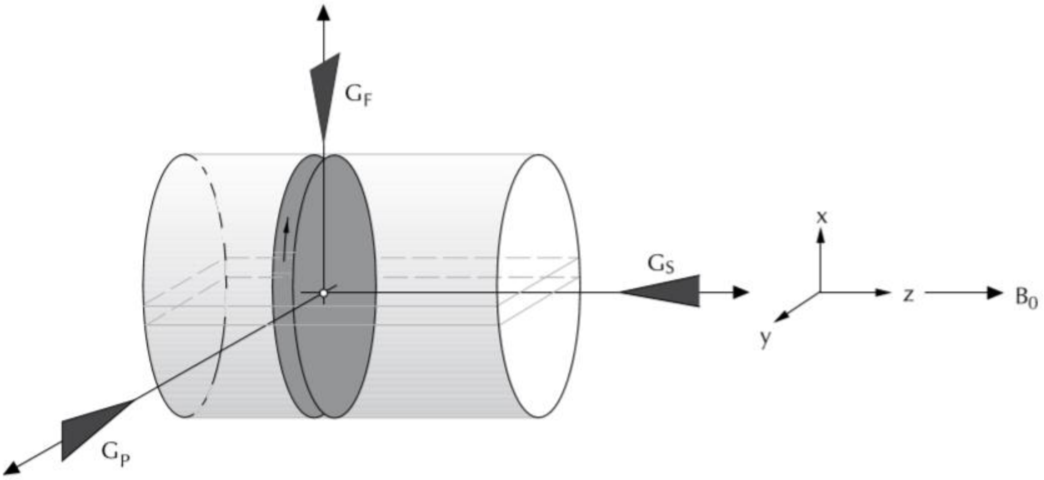


Figure 5 - Creation of Layers [3]

2.1.3 MR Imaging Technology and Applications

In MR imaging, each image element (pixel or voxel) is assigned the measured signal value (defined by the voltage induced in the receiving coils), which is determined on the one hand by tissue-specific properties such as hydrogen proton density and the T_1 and T_2 time of the tissue in the image, and on the other hand by the MR imaging sequence. Since the sensitivity of the measuring coils is not matched to a single pixel, the spin signals must also have a spatial coding. Thus, in addition to the main magnetic field B_0, three further gradient fields G_X, G_Y and G_Z are applied in the three spatial directions (Fig. 5). By applying a layer selection gradient G_S, a layer to be read out is first selected. According to equation (1), only the spins of this layer are excited when the high-frequency pulse is irradiated.

The phase coding gradient G_P divides this layer into bars, which all have a specific phase relationship to the excitation pulse. Finally, the readout gradient G_F resolves the rods into individual voxels, which in turn all have a specific combination of Larmor frequency and phase. The reading process now consists of listening to the emitted signal. To assign the received signal to a single voxel, this listening procedure is repeated with different amplitudes of the phase gradient G_P, as often as the desired number of pixels in the image requires. After sampling all phase coding steps, the image is reconstructed by means of a two-dimensional Fourier transformation, followed by a gray scale assignment for each voxel depending on the calculated spin signal in this voxel.

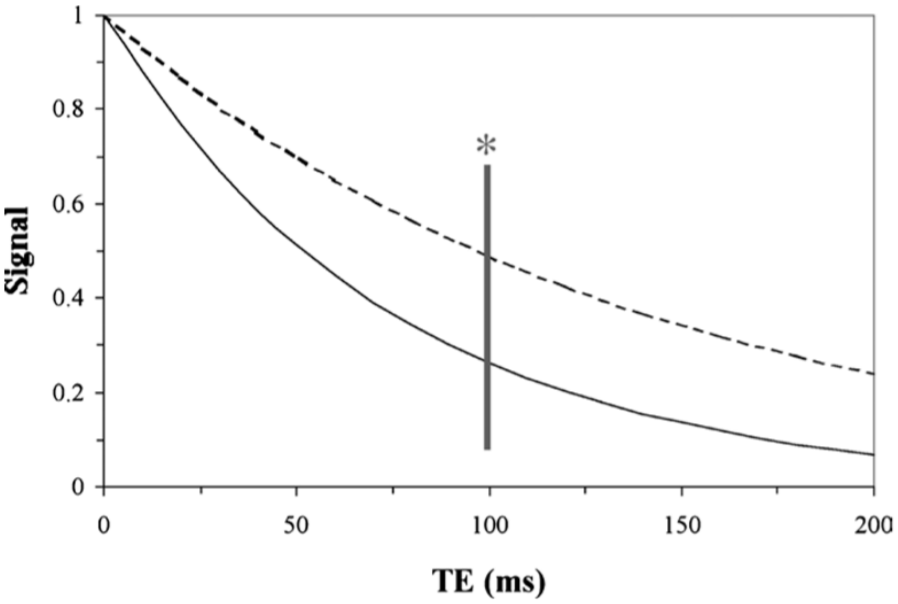


Figure 6 - T_2 Relaxation Curve of two body entities [3]

Above, the T_2 relaxation curve of muscle tissue (solid line) shows a drastically faster signal loss than the spleen (dashed line). This difference of the signal or contrast between the two tissues is the largest at high TE.

Contrast Agents

The magnetic properties of tissues and fluids can be altered in MR using contrast agents (CA); thus, in addition to the T_1 and T_2 time of tissues, the administration of CA offers a further dimension for tissue characterization. In contrast to CT or conventional X-rays, CA are not visualized directly, but can only be delimited indirectly via their effect on the tissue.

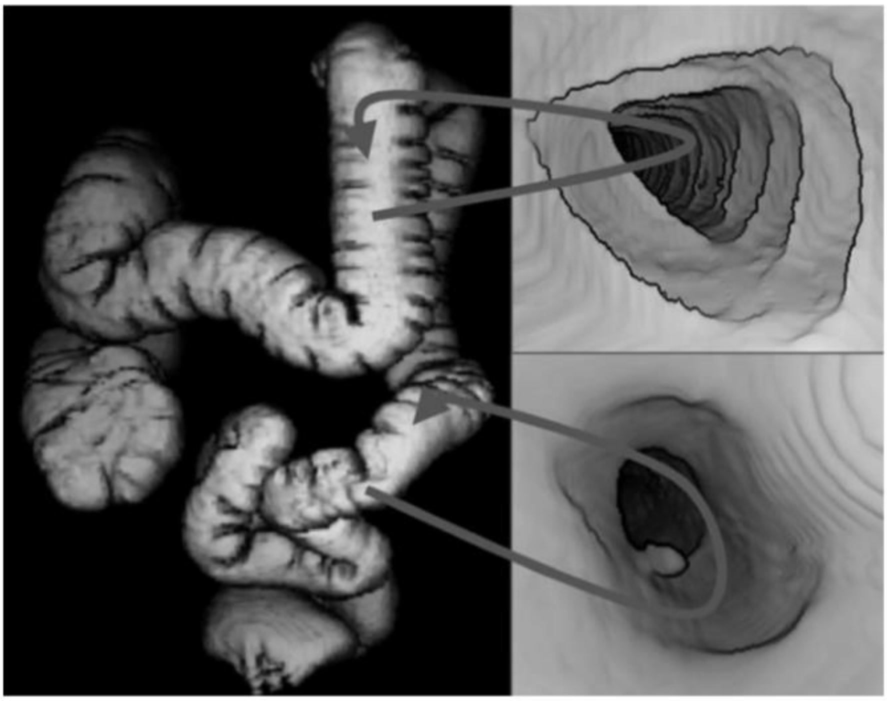


Figure 7 - Surface reconstruction of a 3D intestinal examination with CA [3]

2.1.4 Clinical Applications of MRI

The clinical applications of MRI do comprise of imaging of the heart, abdomen, joints, the musculoskeletal system, contrast-enhanced 3D MR angiography, 3D MR colonography and many more. At each of those applications other goals are pursued such as visualization of the morphology of the brain or the finding of tumors in the tissue.

2.1.5 MRI Compatibility

With the introduction of novel magnets for interventional magnetic resonance imaging (iMRI) at the beginning of the 1990s, which allow improved access to the patient, it became clear that diagnostic and material technologies can develop at different speeds. While iMRI systems are already available as serial products, adequate surgical instruments and positioning systems are still lacking today to fully exploit the potential of this new form of treatment.

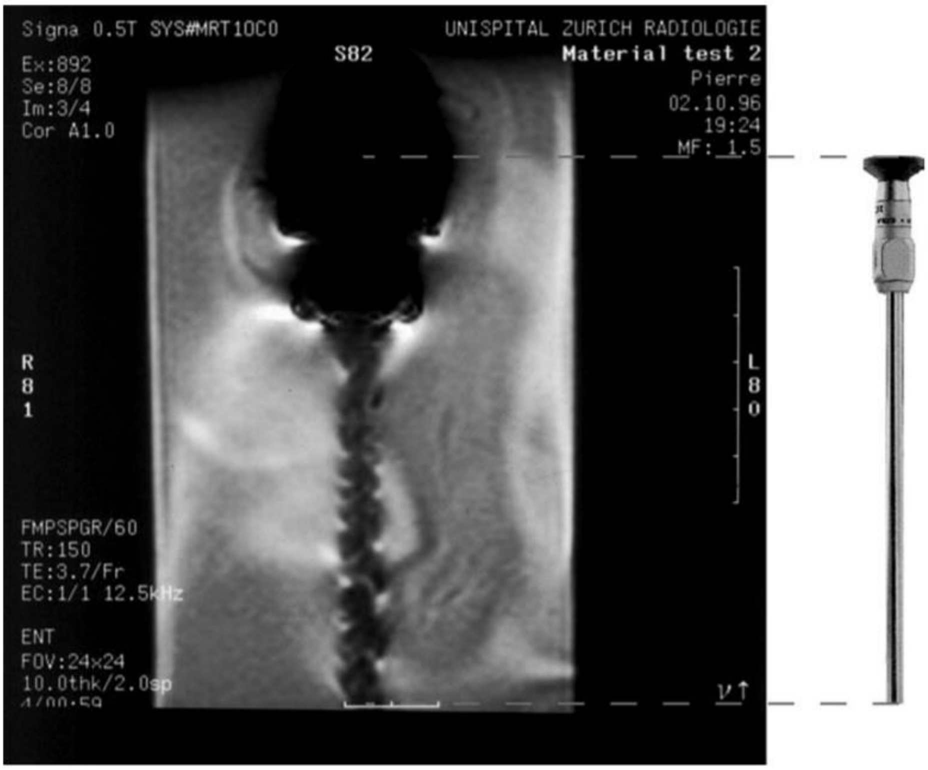


Figure 8 - MRI of an endoscope [3]

In order to be able to estimate the interference potential of existing instruments, a classification was made into three MRI compatibility classes. Magnetically incompatible instruments, such as ferromagnetic components, must be removed from the immediate vicinity of the MRI. Instruments and devices that are not used directly in MRI must meet the criteria of 1st order magnetic compatibility. Finally, instruments that are used directly in the operating field should not interfere too much with the MRI image. Such instruments are granted 2nd order magnetic compatibility. Many instruments available today, e.g. endoscopes, which are required for minimal invasive surgery, cannot be used in MRI because they are made of diamagnetic, paramagnetic and sometimes even ferromagnetic alloys. Ferromagnetic instruments must be removed from an MRI operating room because they are accelerated in the magnetic field and could potentially injure patients or personnel in free flight (magnetic incompatibility).

2.1.6 Artifact Formation

If the compatibility of the examination in patients with implants is ensured, the second question is whether the implants could possibly reduce the image quality to such an extent that the examination is useless. On the one hand, "image holes" are created at the site of the implants because there are no hydrogen protons inside the implants that are necessary for signaling. In addition, the more or less always present paramagnetic or ferromagnetic nature of the material leads to an increase or decrease of the local static magnetic field and thus to a distortion of the magnetic field in the surroundings. As a result, the hydrogen protons precess slightly slower or slightly faster than expected and are recalculated after Fourier transformation to another point in three-dimensional space or in the image plane. This creates a distortion between the actual position of spins in space and their projection in the calculated image. It results in a distortion of the image in the vicinity of the implant, which usually presents itself as a "black hole" (especially in spin echo sequences simultaneously with partially white edges).

If the implant has a particularly large para or ferromagnetic susceptibility, the "hole" is particularly large and an assessment of the surrounding structures is no longer possible. If gradient echo images, which are particularly sensitive to local magnetic field inhomogeneities (T_2^* effects), show particularly large artifacts, spin echo images can often be used to acquire still usable images. Regions of the body further away from the implant, on the other hand, can be examined without difficulty.

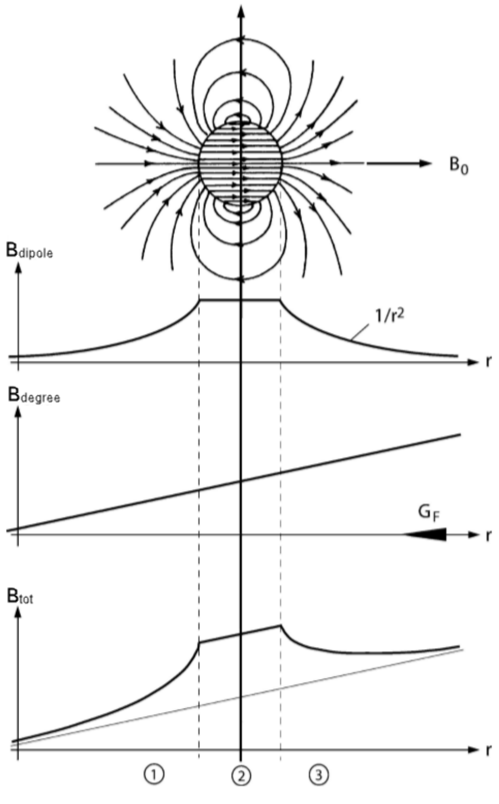


Figure 9 - Simplified illustration of artifact formation [3]

2.2 Medical Imaging [4]

Medical imaging plays a central role in the spectrum of medical technology. The various imaging techniques not only fulfil the task of answering diagnostic questions, but also form the basis for the targeted use of therapeutic procedures (e.g. radiotherapy) or image-supported interventional interactions (e.g. instrument guidance) in the human body. The goal of all imaging procedures is to get rid of the lack of direct visual insight of the physician into the human body. Accordingly, the demands on imaging systems are very high with regard to the representation and detail recognition of anatomical structures, differentiation of different tissues and their function.

None of the medical imaging procedures is able to answer diagnostic questions in their entirety with regard to their information content (anatomy, morphology, geometry, organ function, etc.). Each procedure has strengths and weaknesses. Accordingly, the diagnostic strategies with regard to their application and combination are built up according to the diagnostic question. The aim of the procedures used is therefore always to compensate for the weakness of one procedure with the strength of another.

Imaging Method	**Physical Method**	**Image Type**
X-ray image / transillumination	X-rays	projection image
Computer tomography	X-rays	sectional view
Magnetic Resonance Imaging	nuclear magnetic resonance	sectional view
Sonography	ultrasonic waves	sectional view
Scintigraphy	unstable nuclides	projection image
SPECT / PET	unstable nuclides	sectional view
Endoscopy / Microscopy	light waves	supervisory image

Table 1 – Medical Imaging Procedures [4]

2.2.1 X-ray Imaging

The classic X-ray equipment for diagnostics consists of the following basic components:

- the *emitter* (X-ray tube), which generates the X-ray radiation
- the *generator*, which supplies the emitter with energy
- the *detector*, which converts the X-ray radiation into an image signal after it has penetrated the patient
- the *device*, which assigns the tube, detector and patient to each other, the image system

Advances in X-ray technology have been made with the development of digital electronics/microprocessor technology (e.g. digital imaging systems) and the introduction of image intensifiers/TV systems, which have made it possible to separate image viewing from the place where the image was created. The digital imaging systems found their way into X-ray diagnostics with the digital subtraction angiography (DSA). Today they are a prerequisite for image acquisition, storage and image processing. Microelectronics is used to improve the quality of X-ray images, reduce the radiation dose and, last but not least, to streamline workflows.

The main advantage of X-ray imaging over all other imaging techniques is that it provides real-time overview images from the patient. This information is important for the user in order to directly follow processes or manipulations in the body under visual control. With the spread of digital imaging systems, there has been an ever-increasing shift from classic X-ray diagnostics to digital radiography. The advantages of DR include the possibility of digital image processing and access to methodologically new images.

The Digital Image

The terms "digital" and "digitized" are two much strained words that are used synonymously. In a narrower sense, a digital image is a recording in which only digital data is created at the time of creation. Examples of this are computer-generated synthetic images, count rate images in nuclear medicine, CT or MR images. On the other hand, a digitalized image is an image that initially only contains an analog value, which is then rescanned and digitized. The following figure shows the principle.

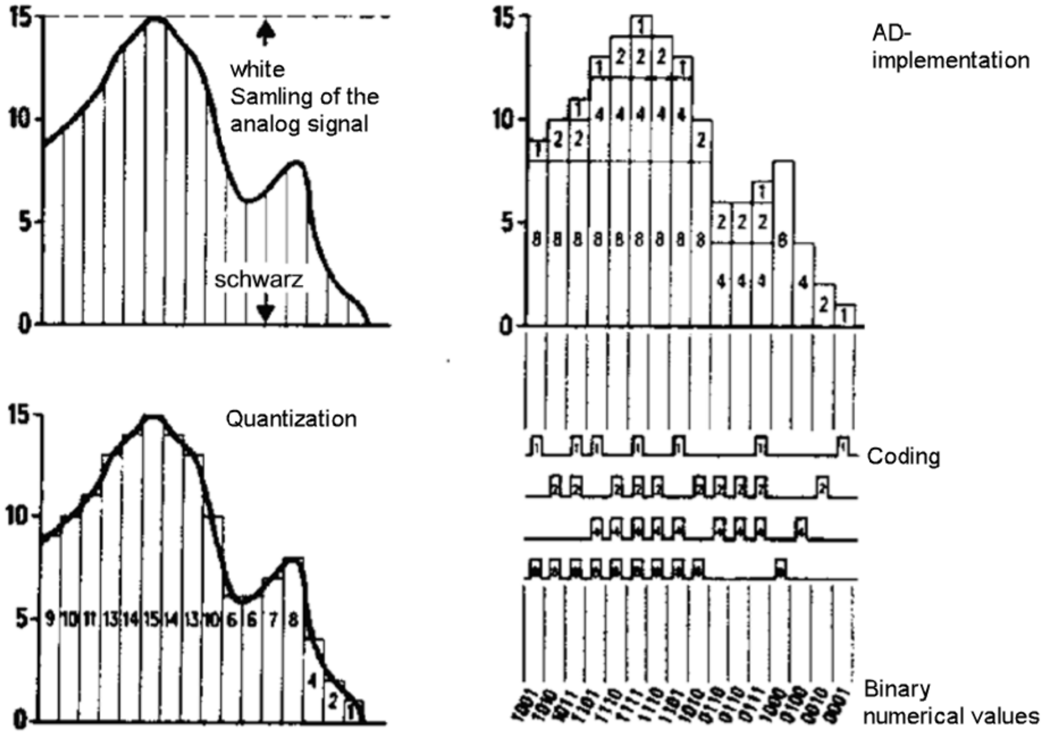

15
10
5
0
white
Samling of the analog signal
schwarz
Quantization
9 10 11 13 14 15 14 13 10 6 6 7 8 4 2 1
AD-implementation
Coding
Binary numerical values
1001 1010 1011 1101 1110 1111 1110 1101 1010 0110 0110 0111 1000 0100 0010 0001

Figure 10 - Principle of quantization [4]

In electrical engineering, digitization is the approximation of analog, i.e. continuous physical measured values by discrete measured values from a finite value set. This discretization, also called quantization, is performed for the measured value itself as well as for the time axis of the measured value determination or the spatial extension of the measured value source in the case of locally varying quantities. Figure 5 shows the principle of quantization.

Image Quality

As described above, the digital image presents itself as a finite matrix of natural numbers, the pixels. The difference between a digital and an analog image is shown schematically in the following figure. Mathematically, these are two-dimensional functions f (x,y) or g (i,j) with continuous value ranges for f, x, y or discrete values (integers) for g, i, j. The quality of a digital image depends decisively on the physical distance between two adjacent pixels (Δ i, Δ j; "pixel size"). In general, it can be said that with finer spatial rasterization, the image quality increases in terms of optical sharpness. In addition, however, one must take into account that a real imaging system has physically or technically predetermined quality limits.

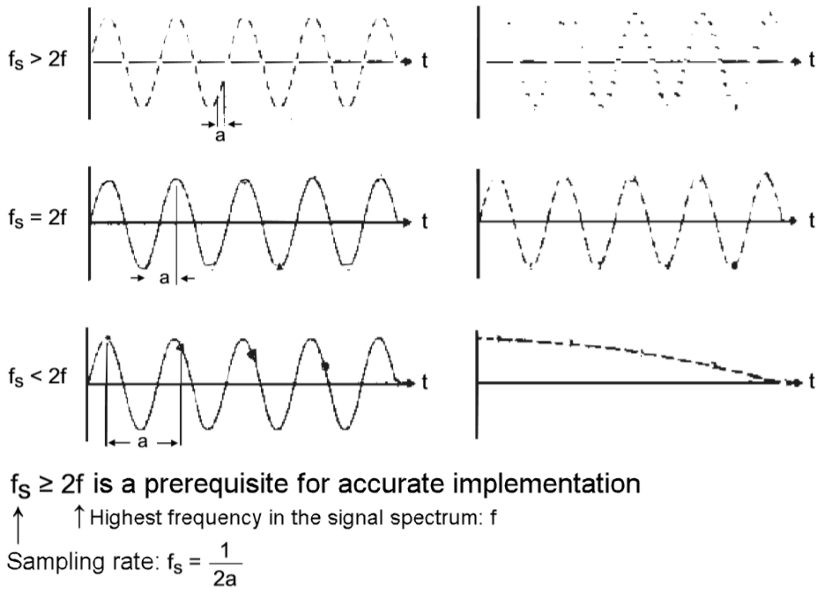

fS > 2f
t
a
fS = 2f
t
a
fS < 2f
t
a
fS ≥ 2f is a prerequisite for accurate implementation
Highest frequency in the signal spectrum: f
Sampling rate: fS = 1/2a

Figure 11 - Local resolution / local frequency [4]

The spatial resolution is the quality limit of an imaging system to reproduce spatial fluctuations of the object information to be displayed in the image. It should be noted that the spatial resolution alone is not sufficient to describe the image quality, but that the contrast resolution coupled to the spatial resolution plays a decisive role.

In X-ray technology, an upper limit for the image quality to be achieved is defined by the physics of X-rays, i.e. by quantum statistics. This means that even for an ideal imaging system there is a maximum value for the signal-to-noise ratio. The quantity DQE (detective quantum efficiency) now indicates how much of this optimum is actually achieved in a real system.

All in all, it can be stated that a multitude of influencing variables determine the quality of a digital image. Among other things, the image quality is object-dependent: "A thin patient (less scattering) has a better image than an obese patient". Equally important is the viewing environment, i.e. whether a monitor is viewed in a darkened room or in a brightly lit room, and the viewing equipment, i.e. whether the image is viewed on a high-resolution or a normal-resolution monitor.

Digital System

The principle of a digital imaging system is shown in Fig. 12 in a simplified form. The detector unit converts X-ray image information into measurement signals which are prepared by the analog-digital converter for handling in the digital system. The digital image is stored in the digital memory for the purpose of image display and image manipulation by the image processing unit. In order to display the digital image on a monitor, it must be converted into a video signal by a digital-to-analog converter. This video signal can also serve as an input signal for the documentation unit. Today, however, documentation units (laser imagers) with digital image input are common.

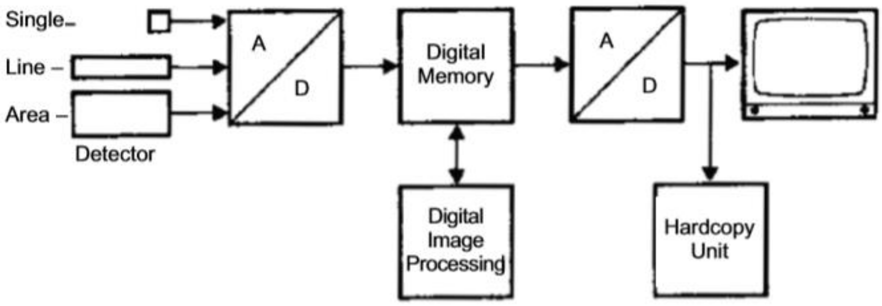

Single
Line
Area
Detector
A
D
Digital Memory
A
D
Digital Image Processing
Hardcopy Unit

Figure 12 - Principle of a digital imaging system [4]

Image Processing

For the processing of digital images can be determined in principle: Only information that is already contained in the original raw data image can be reconstructed. What is often overlooked is the fact that the image signal is already subject to processing during data acquisition (the measuring process). The analog image signal is manipulated, e.g. by logarithmization for the DSA, which is based on the exponential X-ray absorption, or by square-rooting, which is motivated by the dependence of the quantum noise on the root of the measurement signal

Simple methods of image processing can be used to display the image. The term display means the conversion of numerical values according to gray value brightness on the monitor or hardcopy. This conversion takes place with the aid of a translation rule, a so-called Look-Up-Table. Examples for manipulation in the image display are the grey value windowing or the grey value inversion. See the following figure.

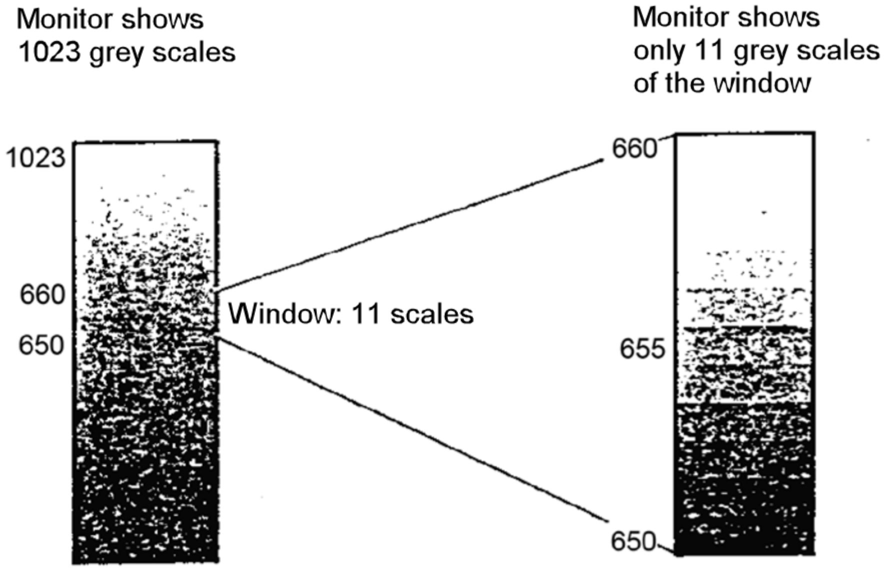


Figure 13 - Grey scale windowing for image display [4]

One speaks of actual image processing when the image itself is manipulated, i.e. the image number values are changed in their context. Example high-pass filtering: Grey-value edges (contours) present in the original image are determined by mathematical differentiation "$(d/d_i, d/d_j)g$" and the resulting pure edge image is added to the original image with the effect of emphasizing edges and fine structures.

Newer methods of image processing (artificial intelligence, neural networks, fuzzy logic) are also advancing into the automatic analysis of image content. Examples are the contour recognition of the heart, the automatic detection of stenoses in DSA images or the automatic detection of microcalcifications in mammography.

Digitization of X-ray Films

The principle here is the following: A focused laser beam is guided in a grid pattern over a developed X-ray film. Depending on the blackening of the film, the laser light is more or less absorbed. The intensity of the laser light transmitted through the film is determined by a light detector (photodiode, photomultiplier) and converted into a digital signal. The light transmission of the film is thus detected depending on the location. The principle is illustrated in figure 14.

The scanning sharpness is determined by the diameter of the laser beam. By a following image processing the grey value contrast can be increased in relation to the film image. In general, however, this does not lead to a significant improvement in image quality. The non-linear film gradation can of course not be completely compensated for in this way. Another method of film digitization can be realized by the use of CCD television cameras.

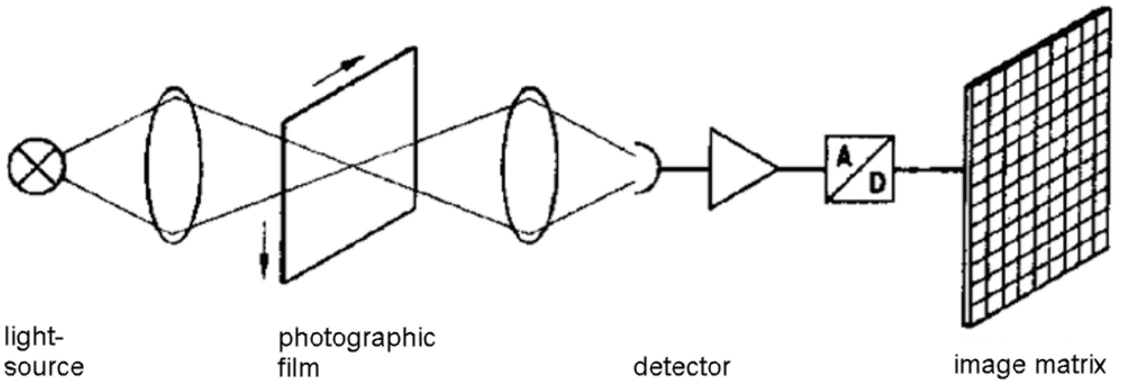


Figure 14 - Principle of X-ray digitization [4]

Since PACS (Picture Archiving and Communication Systems) are increasingly used as routine tools, the digitization of X-ray films may experience a renaissance. This method is the only viable way to make older patient images accessible to the PACS system for follow-up studies.

Picture Archiving

Today, images with matrix sizes between 512 x 512 pixels and 2048 x 2048 pixels are generated. In individual cases, e.g. for digital mammography, even images with a matrix size of 4096 x 4096 pixels or more are generated. Generally, 16 bits are stored per pixel, which is not fully exploited in terms of greyscale dynamics. 8 to 12 bits are common, but this is a concession to the computer-based binary system. This results in a data volume of 0.5 MByte for a 512^2 image, 8 MByte for a 2048^2 image and 32 MByte for a 4096^2 image. Besides the fact that the amount of data per image continues to increase, the proportion of digital images compared to analog images also increases from year to year. This ever-increasing amount of data is countered on the one hand by increasingly powerful storage media and on the other hand by methods of redundancy reduction (data compression). The general principle of a fully reversible data compression is the representation of image information in such a way that as little storage space as possible is required, but that the full information content is retained.

With the advent of PACS, fast data transfer via networks is becoming increasingly important, so compression of the data volume plays a decisive role here too.

The same applies even more so to teleradiology, which is virtually an extension of PACS via the normal telecommunications channels and thus makes it possible, for example, to transfer the digitized images from radiology to the referring physicians (telemedicine).

2.3 Computer Tomography

Computer tomography (CT) is essentially based on the principles of X-ray projection. It is a special X-ray tomography procedure that differs in its image composition fundamentally in that it provides transverse sectional images (Fig. 15). These are images of body layers which are primarily oriented perpendicular to the body axis. In contrast to the classical X-ray technique, such slice images represent the local distribution of the attenuation value μ (x, y, z). This gives it the potential to display organs of interest in principle in three-dimensional image space and to depict small differences in density of soft tissue with high contrast.

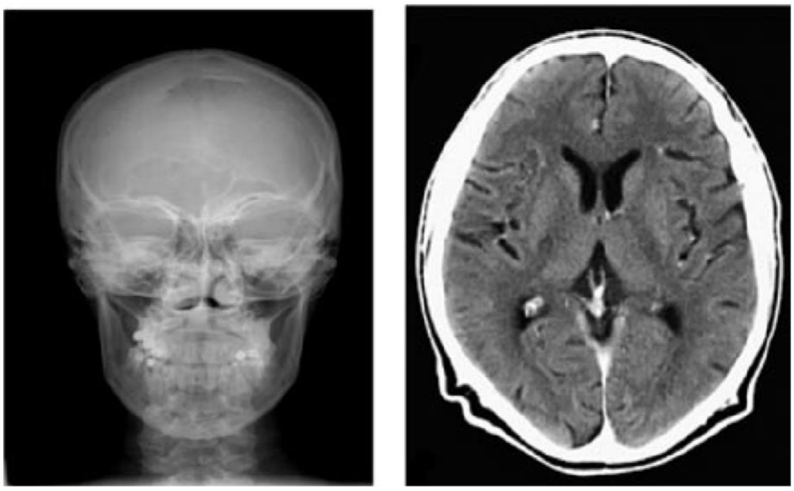


Figure 15 - From projection image to sectional view [4]

For this purpose, CT requires a large number of projections at different angles - in contrast to direct projection in classical X-ray technology. The complete three-dimensional reconstruction of μ (x, y, z) is achieved indirectly by placing and calculating layer after layer. Today, this procedure is increasingly being replaced by the so-called "Spiral-CT" technology, in which the displacement of the patient in the longitudinal direction of the body z is done continuously during the acquisition of measurement data.

Basic Principle of Image Creation

The image reconstruction method can be best explained by a measuring principle that provides a number n of parallel projections taken from m "angles". This so-called "Translations Rotations Principle" was used at the beginning of the development of CT. As shown in Figure 16, a single X-ray beam of certain dimensions is guided by a collimator from the X-ray tube through the body to the detector. Tube and detector are moved perpendicular to the beam and rotated around the center of the measuring field. The detector reacts to the incoming beams with electrical signals I whose amplitude is proportional to the intensity of these beams. The ratio of these signals to the expected signal I_0 without weakening object results in the currently present beam attenuation by the measured object.

A stationary coordinate system (x, y), in which the object function μ (x, y) is localized, is placed on the radiated layer, as well as a second system concentric to the first (ξ, η), whose ζ-axis is aligned parallel to the measuring beam, i.e. it also performs the rotation of the gantry. In this system (ξ, η) the detector registers the intensity profiles $I(\varphi, \eta)$, when the tube and the detector, with the beam moved parallel to itself, are moved by a small angle $\Delta\varphi$ rotation, then moved backwards and rotated again. This interplay continues until a total angle of rotation of 180° is reached. For each angle φ there is then a set of parallel beams whose signals are collected in $I(\varphi, \eta)$.

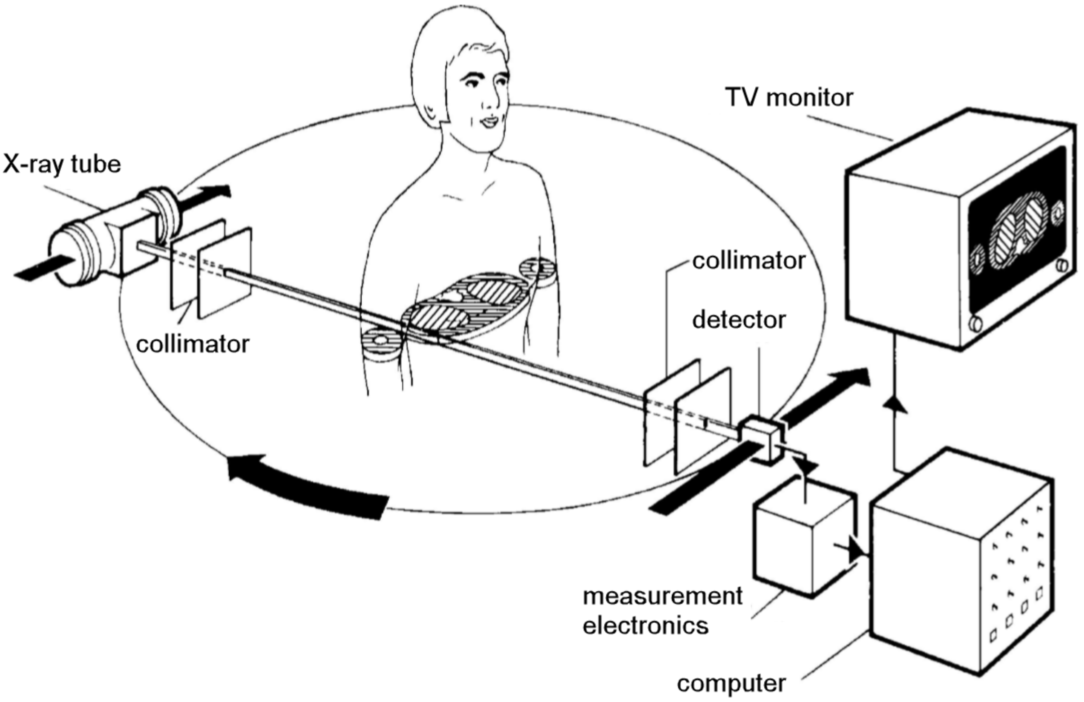


Figure 16 - Scan principle [4]

Image Display

The calculation of the attenuation value distribution μ of the irradiated layer does not yet complete the task of image representation. The distribution of the attenuation coefficients represents only an anatomical structure in the medical field of application, which must be displayed as an image. According to Hounsfield, it has become common practice to transform the linear attenuation coefficients μ (with the unit of measurement cm-1) to a dimensionless scale in which the material water is given the value 0 and air the value -1000. The conversion formula for this "CT number" is:

$$CT\ number(\mu_{rel}) = \frac{1000*(\mu_{object}*\mu_{water})}{\mu_{water}} \qquad (4)$$

The so-called CT values or "Hounsfield Units" named after the inventor of CT are related to the X-ray attenuation of water and are directly related to other biological materials via a table:

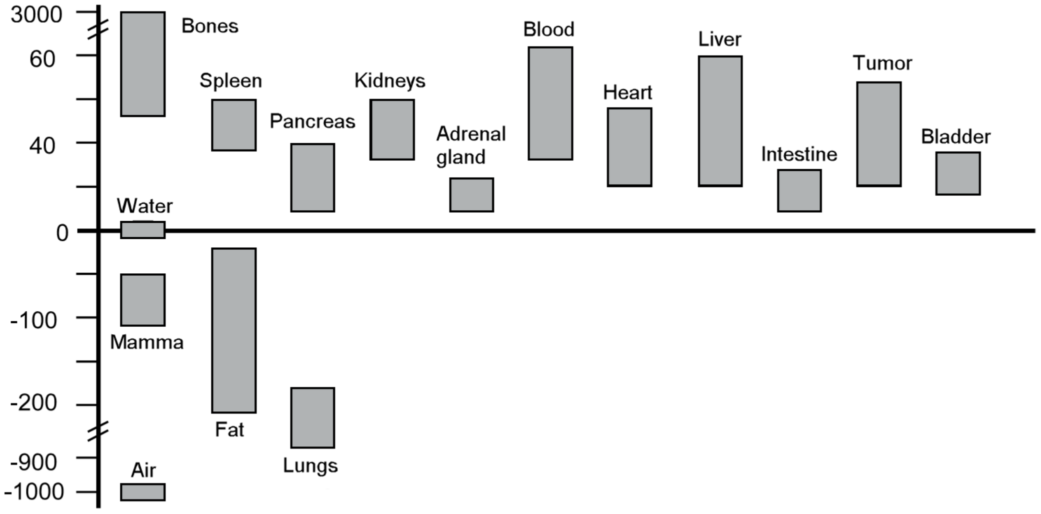


Figure 17 - Tissue types and Hounsfield Units [4]

The unit of the CT number is called 'Hounsfield Unit' (HU). This number is very suitable because the μ-values of most endogenous substances differ only slightly from the μ-value of water. The unit is used to express the deviation in per million of μ water. The Hounsfield scale starts at -1000 for air and, although in principle open at the top, ends generally at +3000 for bone. The position of the remaining tissues on the scale is shown in Figure 17.

For the morphological evaluation of a CT image, it is important to highlight the information that is essential for the respective examination. For this purpose, the attenuation values of the CT image are converted into up to 256 grey levels and displayed on a monitor. The so-called image windowing is therefore used to display small differences in attenuation. Only a part of the CT value scale is selected and spread over all brightness values between black and white. Even small differences in attenuation within the selected window become noticeable differences in grey tones, while all CT values below the window are displayed in black and all values above the window in white.

Spiral CT

In classic CT, the patient is pushed 5 mm further with the lounger and imaged layer by layer ("layered"). With the newer method, the spiral CT or helical CT, the table is advanced continuously with permanent rotation of the CT measuring system (Fig. 18). The spiral CT is not a tomography but a volume imaging method. Especially in combination with multi-line detectors (4, 8, 16 and up to 64 detectors) a considerable gain in speed can be achieved, because the table feed per revolution can be chosen even higher than the total height of the detector ensemble. From the spiral raw data, data are obtained by suitable interpolation and re-sorting, which are fed into a single layer calculation as described above.

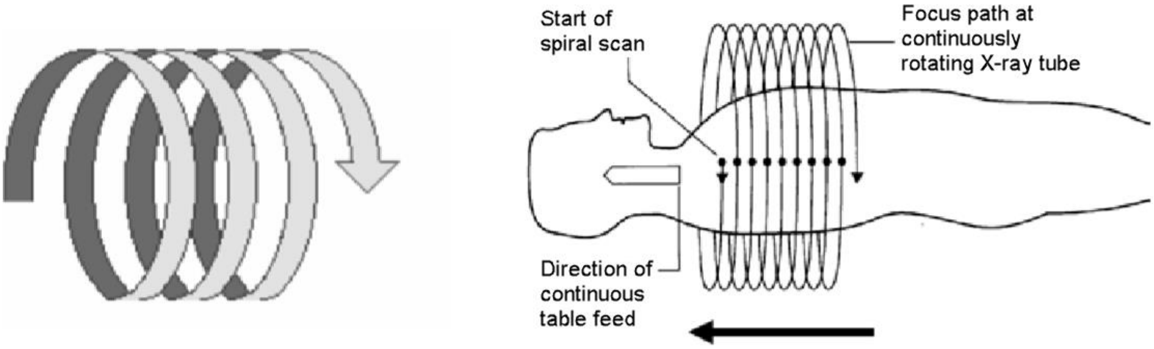


Figure 18 - Principle of spiral CT [4]

Spiral series generate "real", i.e. almost isotropic layer packages that are directly suitable for 3D representation. Typical are packages of 500 layers. With a standard data format of 512 x 512 pixels at 16 bits, this amounts to 250 MByte, i.e. a considerable data volume for reconstruction, visualization and archiving. The raw data generated can nevertheless be displayed in sections in real time during the scan (Realtime Display "RTD"), so that detailed control of the recording process is possible, during which the ionizing radiation is applied in the diagnostic area. Subsequently, on the basis of this raw data, refined reconstructions are possible over the entire raw data volume or over any sections of it.

2.4 Image Analysis in Medicine and Biology [5]

2.4.1 Introduction

Today, medical examinations without imaging techniques are unimaginable. Various methods - based on the use of ultrasound waves, X-rays, magnetic fields or light rays - are used specifically and provide extensive data material about the body and its interior. In addition, microscopic images from biopsies can be used to obtain data on the morphological properties of body tissues. By analyzing all these different types of information and consulting further clinical studies from various medical disciplines, an "overall picture" of a patient's state of health can be created, taking into account anamnesis data. Due to the flood of generated image data, image processing in general and image analysis in particular are playing an increasingly important role. Especially in the areas of diagnostic support, therapy planning and image-guided surgery, they are key technologies that are driving progress not only in these areas.
A prerequisite for the interpretation of image contents is a successful image analysis, the goal of which is the calculation of a table or pictorial description of the underlying original image. In the case of radiographs, this can be the highlighted representation and content allocation of bones, soft tissue or organs. Segmentation refers to the grouping of adjacent image points into content-related regions (so-called segments) that meet a certain homogeneity criterion. A subsequent assignment of these regions to certain classes with possible meaning in terms of content (interpretation) is the task of classification. During feature extraction, features, such as metrics for form and texture of image objects, are combined in a descriptive feature vector. During filtering, for example, image data is smoothed or edges contained therein are highlighted.

2.4.2 Application of Cognition Network Technology for Image Analysis in Medicine and Biology

In neurosciences, there are numerous application scenarios for the use of object-based procedures such as CNT (Cognition Network Technology). Figure 19 shows an example of the automatic segmentation of the lateral ventricles from magnetic resonance imaging data of the brain. A detailed evaluation of the quality of these results shows that they are comparable to those of manual segmentation.

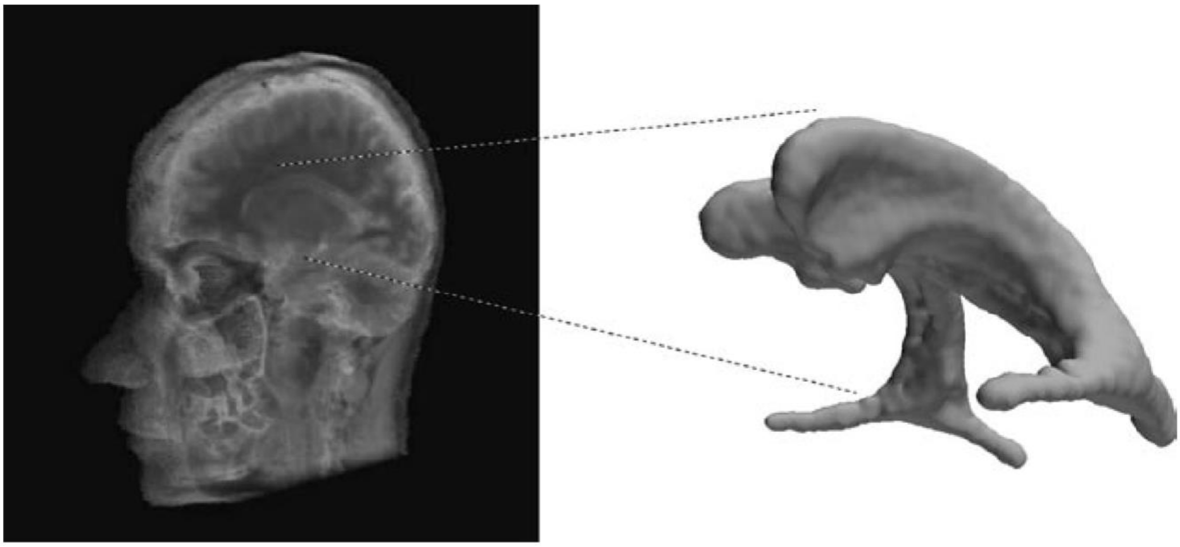


Figure 19 - Segmentation and extraction of the lateral ventricles from magnetic resonance imaging of the head [5]

Figure 20 shows the result of a first application of a CNT-based solution for computer tomographic data sets. This involves the extraction of individual organs by means of a CNT set of rules. This can segment and classify organs - such as the lung, kidney and pancreas - fully automatically and determine and quantify their morphological properties.

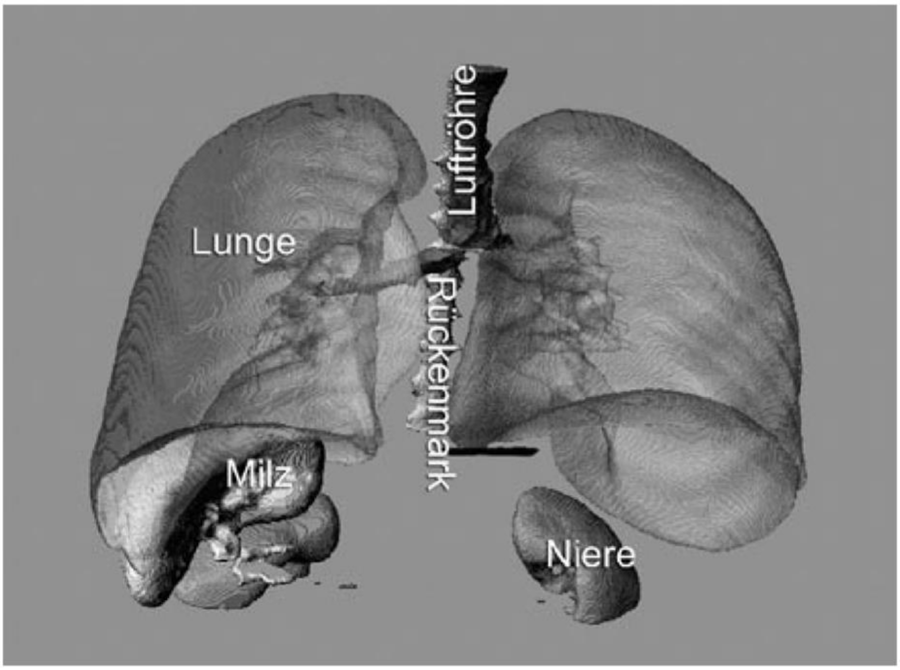


Figure 20 - Illustration of the results of CT image data analysis [5]

3 Electrical Phenomena of the Body and their Detection [6]

In the human body, both nerves and muscle cells have the property of shifting intra- and extracellular ion concentrations and thus influencing the potential distribution in their environment. By means of different synchronization processes, macroscopic cell assemblies are additionally able to change their fields in a coordinated manner and thus also generate significant electrical signals that can be measured on the body surface. This chapter first describes the electrophysiological principles of electrical signals of the human body, the synchronization mechanisms and the resulting fields, in particular the electrocardiogram (ECG), the electroencephalogram (EEG) and the electromyogram (EMG). Subsequently, the essential basics of measurement technology for recording bioelectric phenomena are explained.

3.1 The Electrocardiogram

The most well-known recording of a bioelectric signal is the electrocardiogram (ECG). The sinus node located in the atrium serves as an autonomous clock generator of the heartbeat. It generates periodic action potentials. The electrical pulses are transmitted to the different areas of the heart via excitation conduction paths. From there they pass from cell to cell with the help of gap junctions, special electrical connections of the heart muscle cells. This "all or nothing" principle of the heart leads to a similar electrical excitation of large areas, whereby the individual action potentials add up in their electrical effect, which externally leads to an increase in amplitude.

The far electric field of the heart leads to a potential distribution on the body surface (Fig. 21). A typical ECG conduction is shown in Fig. 22. It consists of positive and negative deflections, which according to Einthoven are designated as P, Q, R, S, T and U. The time course of the ECG signal and the amplitudes of the individual spikes are normally within the physiological range summarized in Table 2.

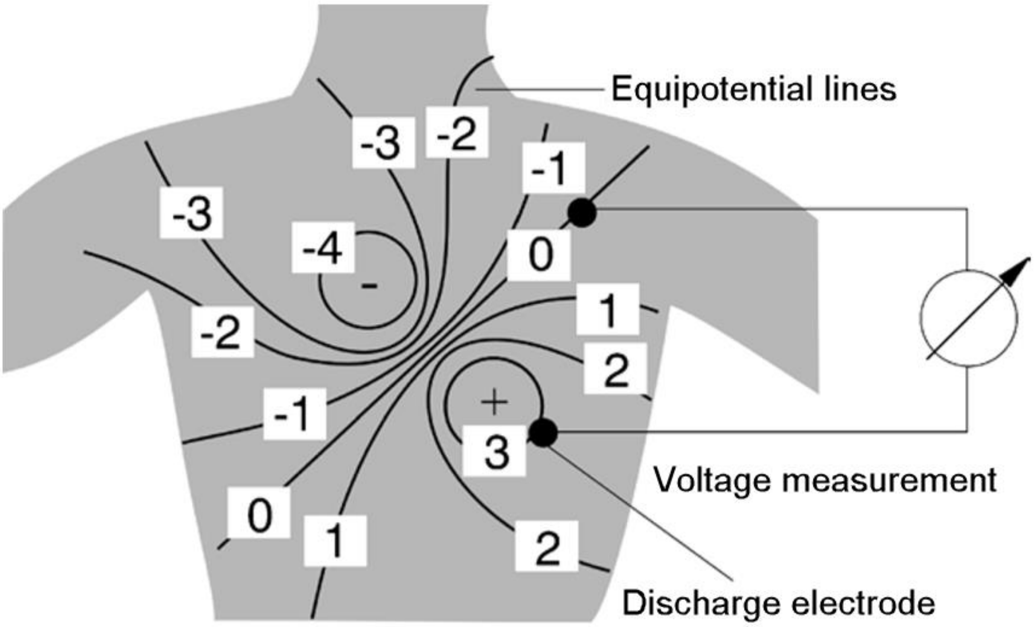


Figure 21 - Potential field of the heart at the time of the R-spike. The numbers indicate the average amplitudes in mV [6]

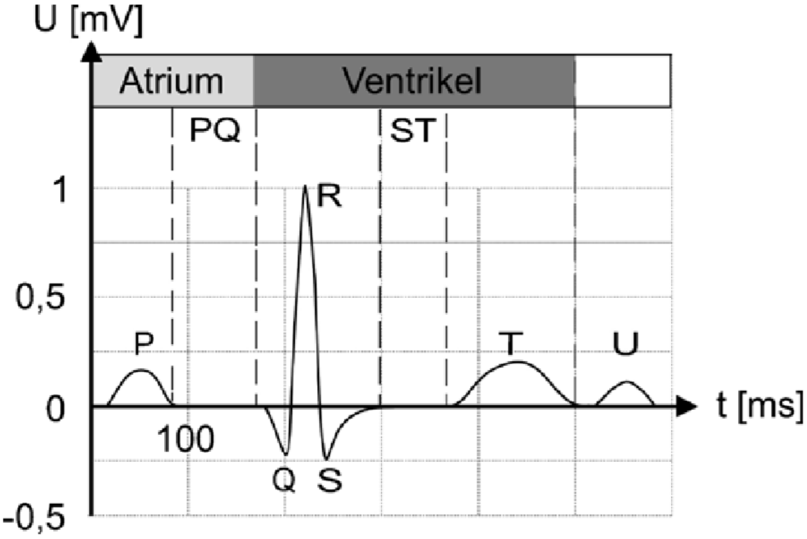


Figure 22 - Typical timing of the electrocardiogram with the designations introduced by Einthoven [6]

The individual morphological elements of the ECG can be directly related to the electrical activity of the heart. Fig. 23 shows an example of a projection of the dipole moment on the axis between the right arm and left foot. Depending on the choice of the derivation line, the projection of the cardiac dipole changes and consequently also the polarity and morphology of the ECG.

Typical Intensities of ECG waves [mV]		
Wave description	**Extremities discharge**	**Chest wall discharge**
P	< 0,25	< 0,25
PQ	0	0
Q	0,2	0,3
QRS	1	1–3
S	0,4	2
ST	0	0
T	0,4	0,5

Typical duration of ECG waves [ms]						
Desc.	P	PQ	Q	QRS	S	QT
t	90	160	< 30	80	< 40	400

Table 2 - Normal values of ECG [6]

P-Wave

The excitation of the heart begins at the sinus node and then spreads through the atria. During atrial excitation, the resulting field strength vector points to the apex of the heart, due to the course of the excitation from the upper sinus node to the basal atrioventricular (AV) node. The projection of this vector on the derivation line is small in magnitude and points in the positive direction. The P-Wave is thus an expression of the excitation propagation in both atria.

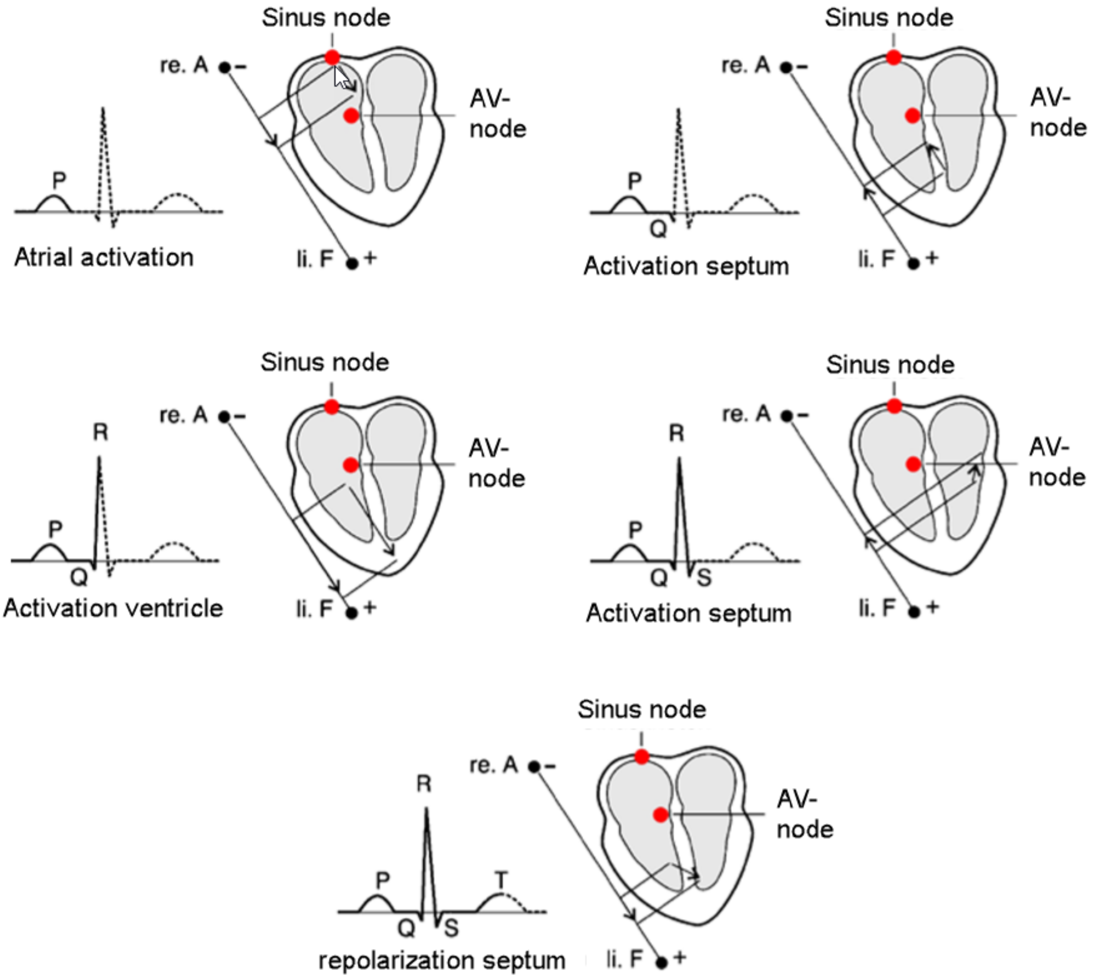

Sinus node
re. A
AV-node
P
Atrial activation
li. F
Sinus node
re. A
AV-node
P
Q
Activation septum
li. F
Sinus node
re. A
R
AV-node
P
Q
Activation ventricle
li. F
Sinus node
re. A
R
AV-node
P
Q
S
Activation septum
li. F
Sinus node
re. A
R
AV-node
P
T
Q
S
repolarization septum
li. F

Figure 23 - The propagation of excitement in the heart [6]

PQ path

After atrial excitation, the entire atrial myocardium is externally negative. There is therefore no dipole moment in the PQ time interval and no potential difference in the ECG. The PQ distance represents the transfer time from the atrium to the chamber. It extends from the end of PE to the beginning of Q and is taken as the reference line for voltage measurement (isoelectric line).

QRS complex

Ventricular excitation begins on the left side of the ventricular septum. Between this area and the unexcited tissue following the right ventricle, a vector is created that is directed to the right and to the base. The projection of this vector onto the derivation line is usually negative, so that the Q-wave associated with this excitation point is also negative. A short time later the septum including the apex of the heart is excited. The resulting propagation is now directed towards the apex of the heart. Thus, the projection of the vector onto the derivation line at this point in time and thus the R-wave has a positive polarity. The ventricular excitation propagation ends at the base of the left ventricle. The resulting field strength component now points to the right, baseline; the S-wave is therefore negative. The QRS complex thus corresponds to the propagation of excitation in both ventricles. It begins at the beginning of Q and extends to the end of the S-wave. If one of these points is missing, the corresponding zero crossing of the R point is taken as the reference point for time measurement.

ST segment
After completion of ventricular excitation, the entire heart surface is negative. Potential differences cannot be registered. This phase in the excitation process of the heart is connected to a zero line in the ECG, the ST segment (measured from the end of the S to the beginning of the T).

T-Wave
The ventricular repolarization phase of the heart begins in the subendocardial layers and progresses towards the endocardium. Thus, a field strength component is present, which points from the still excited, negative endocardial layers into the already unexcited and thus positive areas. At the time of the T-wave, which is associated with the repolarization phase, the projection of the field strength vector onto the derivation line is positive. The T-wave in the ECG is therefore positive.

3.2 Application Example: Electrocardiography

The most well-known measurement of bioelectrical signals is the electrocardiogram. The first successful ECG measurement was carried out a little more than 100 years ago by Einthoven, who placed his hands and feet in large beakers and, with the aid of a capillary electrometer, obtained a signal known today as an ECG (Fig. 24).

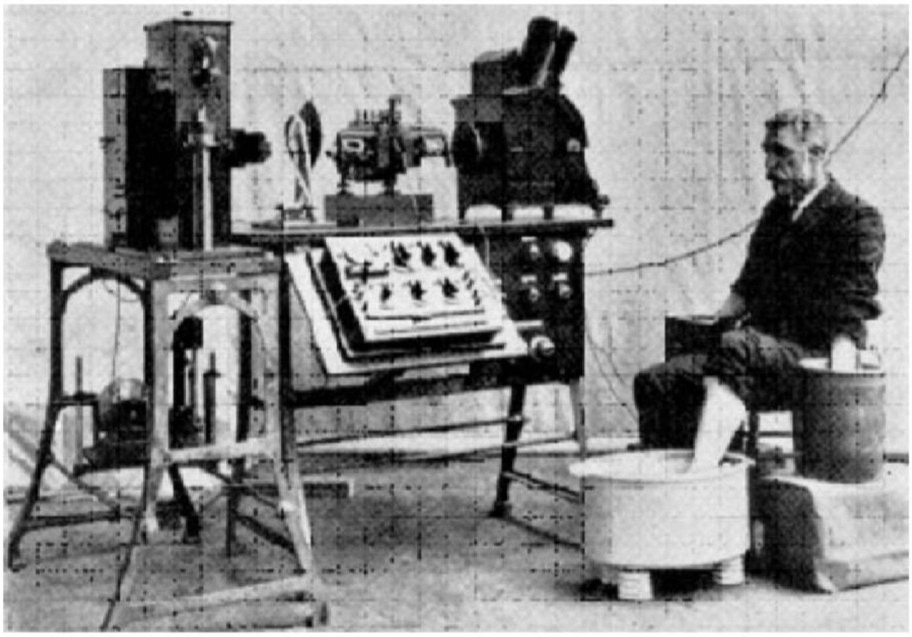


Figure 24 -First ECG device after Einthoven from the year 1911 [6]

Since then, the measurement technology has improved significantly. Self-adhesive silver-silver chloride electrodes are usually used as electrodes. Historically, three electrode arrangements have been established for ECG recording. Fig. 25 shows a patient with a modern wireless ECG device. The four electrodes arranged in the outer corners of the upper body form the basic leads according to Einthoven (bipolar lead, I, II and III) or Goldberger (unipolar lead in relation to the average value of the other electrodes, aVL, aVR, aVF). For historical reasons, the triangle "right shoulder" - "left shoulder" - "left abdomen" is used as the measurement location. The electrode glued to the right abdominal area serves as a neutral electrode to improve the signal quality. The electrodes below the nipple are called Wilson chest wall leads (V1 to V6). They are also measured unipolar with respect to the other body electrodes.

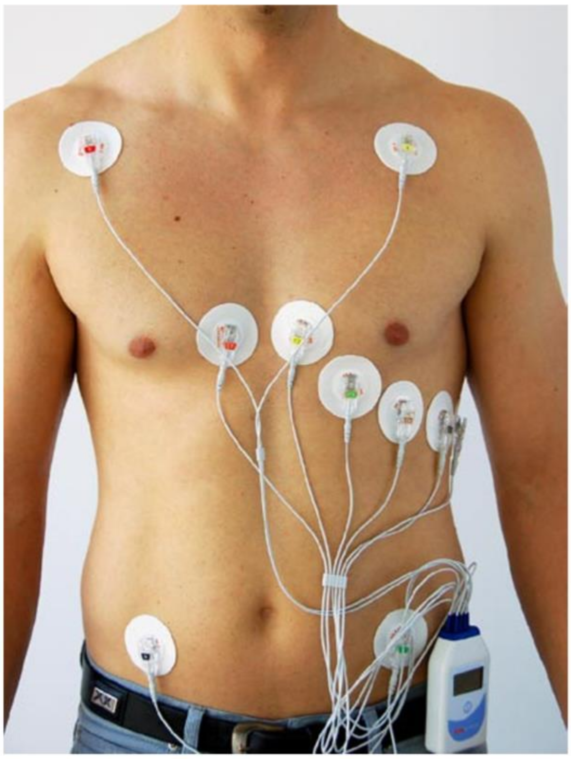


Figure 25 - Patient with a modern 12-channel ECG device [6]

The output of the ECG signals is done classically as a time signal (see Fig. 26). More and more, however, an automatic interpretation of the signals is gaining acceptance. Special algorithms analyze 10 seconds of an ECG signal, filter out artifacts, average the signals and compare them with set values. From this, diagnostic recommendations can be derived, which give the physician an indication of possible underlying cardiac diseases. Although such algorithms are accepted and widely used today, they do not replace the cardiologist, who should always examine the ECG in case of doubt.

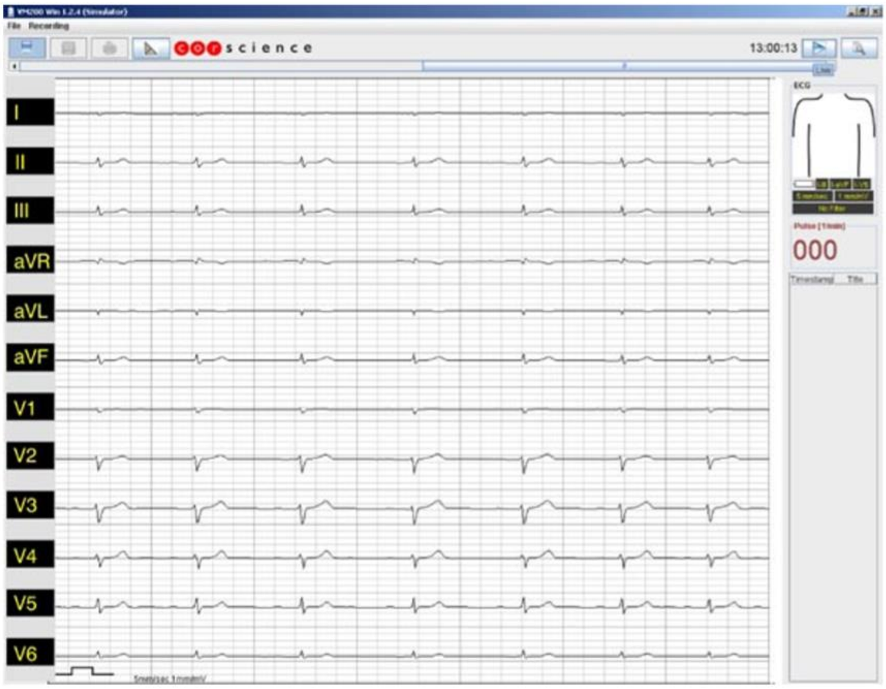


Figure 26 - Output of ECG signals [6]

4 Description of the DICOM Format

“DICOM - Digital Imaging and Communications in Medicine – is the international standard for medical images and related information. It defines the formats for medical images that can be exchanged with the data and quality necessary for clinical use.
DICOM is implemented in almost every radiology, cardiology imaging, and radiotherapy device (X-ray, CT, MRI, ultrasound, etc.), and increasingly in devices in other medical domains such as ophthalmology and dentistry. With hundreds of thousands of medical imaging devices in use, DICOM is one of the most widely deployed healthcare messaging standards in the world” [7]

DICOM is recognized by the International Organization for Standardization as the ISO 12052 standard. The patient-specific 3D models are derived from these images through the process of segmentation (see chapter 5). The understanding of the basics of the format is required for the segmentation process (potentially automatized) and the achievement of data consistency throughout the process chain.

Data Format

“DICOM groups information into data sets. That means that a file of a chest x-ray image, for example, actually contains the patient ID within the file, so that the image can never be separated from this information by mistake. This is similar to the way that image formats such as JPEG can also have embedded tags to identify and otherwise describe the image.

A DICOM data object consists of a number of attributes, including items such as name, ID, etc., and also one special attribute containing the image pixel data (i.e. logically, the main object has no “header” as such, being merely a list of attributes, including the pixel data). A single DICOM object can have only one attribute containing pixel data. For many modalities, this corresponds to a single image. However, the attribute may contain multiple “frames”, allowing storage of cine loops or other multi-frame data. Another example is NM (nuclear medicine) data, where an NM image, by definition, is a multi-dimensional multi-frame image. In these cases, three- or four-dimensional data can be encapsulated in a single DICOM object. Pixel data can be compressed using a variety of standards, including JPEG, Lossless JPEG, JPEG 2000, and Run-length encoding (RLE). LZW (zip) compression can be used for the whole data set (not just the pixel data), but this has rarely been implemented.

DICOM uses three different Data Element encoding schemes. With Explicit Value Representation (VR) Data Elements, for VRs that are not OB, OW, OF, SQ, UT, or UN, the format for each Data Element is: GROUP (2 bytes) ELEMENT (2 bytes) VR (2 bytes) LengthInByte (2 bytes) Data (variable length). For the other Explicit Data Elements or Implicit Data Elements, see section 7.1 of Part 5 of the DICOM Standard.

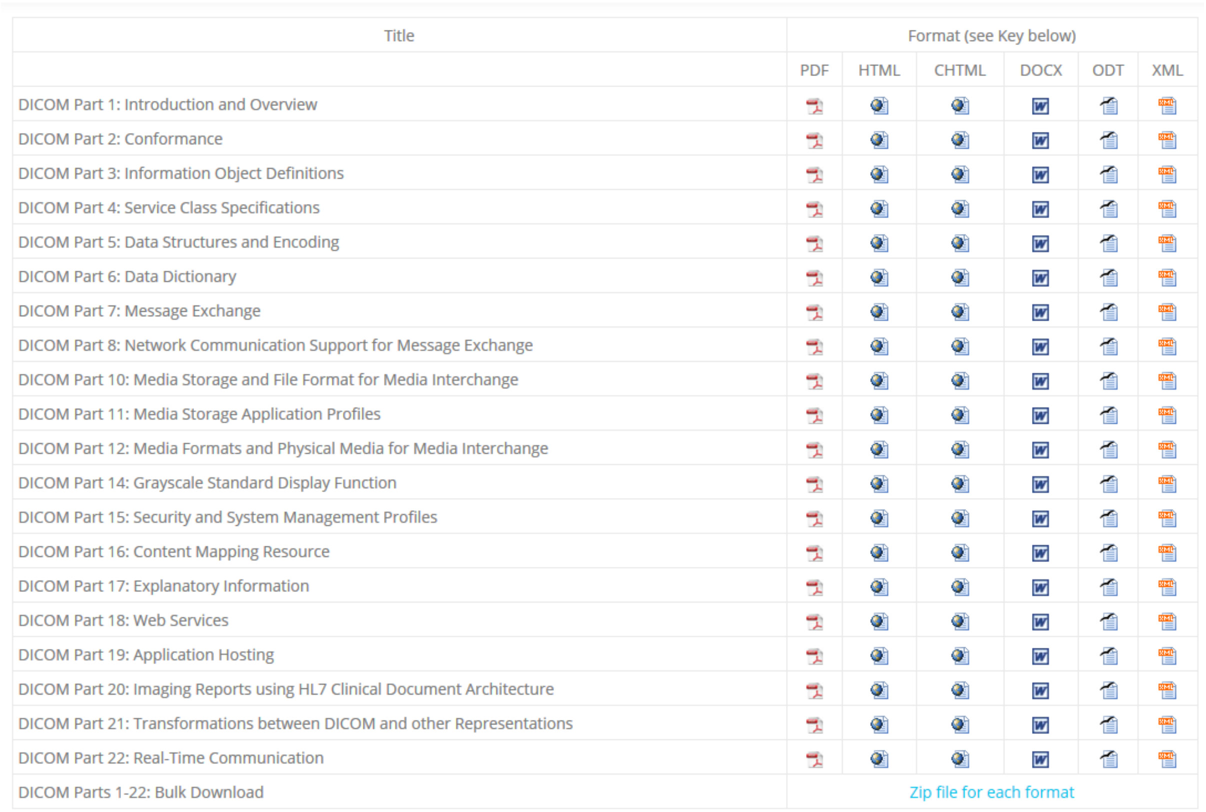
The same basic format is used for all applications, including network and file usage, but when written to a file, usually a true “header” (containing copies of a few key attributes and details of the application which wrote it) is added” [8].

Image Display

“To promote identical grayscale image display on different monitors and consistent hard-copy images from various printers, the DICOM committee developed a lookup table to display digitally assigned pixel values. To use the **DICOM grayscale standard display function (GSDF)**, images must be viewed (or printed) on devices that have this lookup curve or on devices that have been calibrated to the GSDF curve” [8].

General Overview on Current Edition

The DICOM standard is managed by the Medical Imaging & Technology Alliance [9]. The standard is updated and republished several times per year. Fig. 27 shows the current overview of the DICOM publications.



Title	Format (see Key below)					
	PDF	HTML	CHTML	DOCX	ODT	XML
DICOM Part 1: Introduction and Overview	✓	✓	✓	W	✓	✓
DICOM Part 2: Conformance	✓	✓	✓	W	✓	✓
DICOM Part 3: Information Object Definitions	✓	✓	✓	W	✓	✓
DICOM Part 4: Service Class Specifications	✓	✓	✓	W	✓	✓
DICOM Part 5: Data Structures and Encoding	✓	✓	✓	W	✓	✓
DICOM Part 6: Data Dictionary	✓	✓	✓	W	✓	✓
DICOM Part 7: Message Exchange	✓	✓	✓	W	✓	✓
DICOM Part 8: Network Communication Support for Message Exchange	✓	✓	✓	W	✓	✓
DICOM Part 10: Media Storage and File Format for Media Interchange	✓	✓	✓	W	✓	✓
DICOM Part 11: Media Storage Application Profiles	✓	✓	✓	W	✓	✓
DICOM Part 12: Media Formats and Physical Media for Media Interchange	✓	✓	✓	W	✓	✓
DICOM Part 14: Grayscale Standard Display Function	✓	✓	✓	W	✓	✓
DICOM Part 15: Security and System Management Profiles	✓	✓	✓	W	✓	✓
DICOM Part 16: Content Mapping Resource	✓	✓	✓	W	✓	✓
DICOM Part 17: Explanatory Information	✓	✓	✓	W	✓	✓
DICOM Part 18: Web Services	✓	✓	✓	W	✓	✓
DICOM Part 19: Application Hosting	✓	✓	✓	W	✓	✓
DICOM Part 20: Imaging Reports using HL7 Clinical Document Architecture	✓	✓	✓	W	✓	✓
DICOM Part 21: Transformations between DICOM and other Representations	✓	✓	✓	W	✓	✓
DICOM Part 22: Real-Time Communication	✓	✓	✓	W	✓	✓
DICOM Parts 1-22: Bulk Download	Zip file for each format					

Figure 27 - Overview of DICOM publications [10]

All the publications are available for download for free as PDF, HTML, CHTML, DOCX, ODT or XML, depending on the requirements of the user. Together, these documents describe the DICOM standard in every detail, e.g. which characters are supported (ISO-IR 100, ISO-IR144 etc.) [11, p.32]. This paper focuses on the syntax of the file format and therefore its structure and the information stored in an image file (meta data and image data). This information shall give a deeper understanding of the format and is required to read and process the DICOM images in further process steps. The nomenclature of the documents is “PS3.1” for part 1, “PS3.2” for part 2 and so on.

4.1 Information Object Definition

This chapter shall give an overview on the specification "of Information Object Definitions (IODs) that provide an abstract definition of real-world objects applicable to communication of digital medical information" [12, p.83].

4.1.1 DICOM Information Model

"The DICOM Information Model defines the structure and organization of the information related to the communication of medical images." Figure 28 "shows the relationships between the major structures of the DICOM Information Model" [12, p.107].

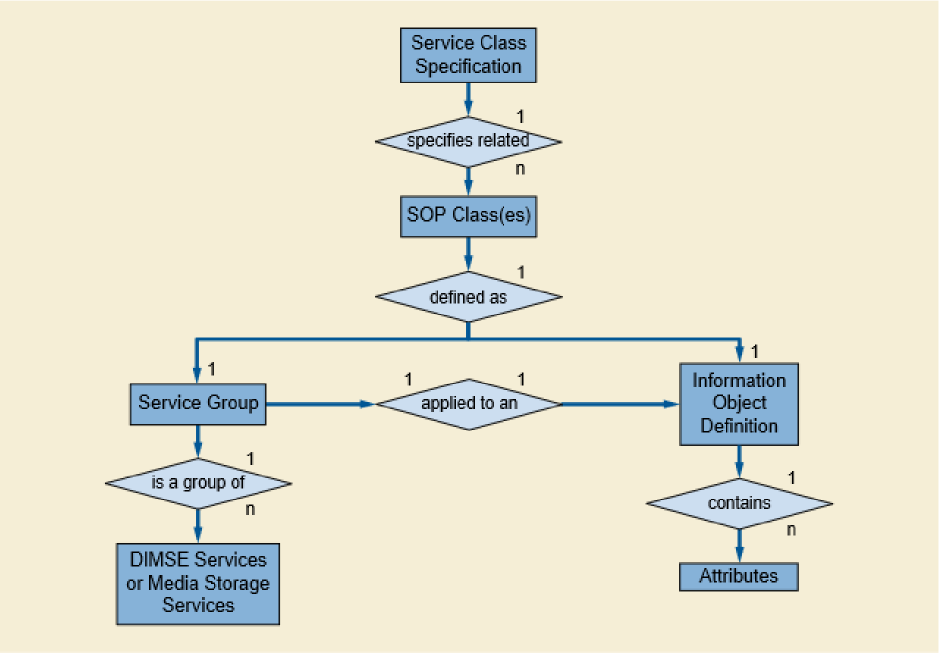

Service Class Specification
1
specifies related
n
SOP Class(es)
1
defined as
1
1
Service Group
1
applied to an
1
Information Object Definition
1
is a group of
n
1
contains
n
DIMSE Services or Media Storage Services
Attributes

Figure 28 - Major Structures of DICOM Information Model [12, p.107]

Information Object Definition

"An Information Object Definition (IOD) is an object-oriented abstract data model used to specify information about Real-World Objects. An IOD provides communicating Application Entities with a common view of the information to be exchanged" [12, p.107].

Attributes

"The Attributes of an IOD describe the properties of a Real-World Object Instance. Related Attributes are grouped into Modules that represent a higher level of semantics documented in the Module Specifications found in Annex C.

Attributes are encoded as Data Elements using the rules, the Value Representation and the Value Multiplicity concepts specified in PS3.5. For specific Data Elements, the Value Representation and Value Multiplicity are specified in the Data Dictionary in PS3.6" [12, p.108].

4.1.2 DICOM Model of the Real World

The figures 29, 30 and 31 “depict the DICOM view of the Real-World that identifies the relevant Real-World Objects and their relationships within the scope of the DICOM Standard. It provides a common framework to ensure consistency between the various Information Objects defined by the DICOM Standard” [12, p. 111].

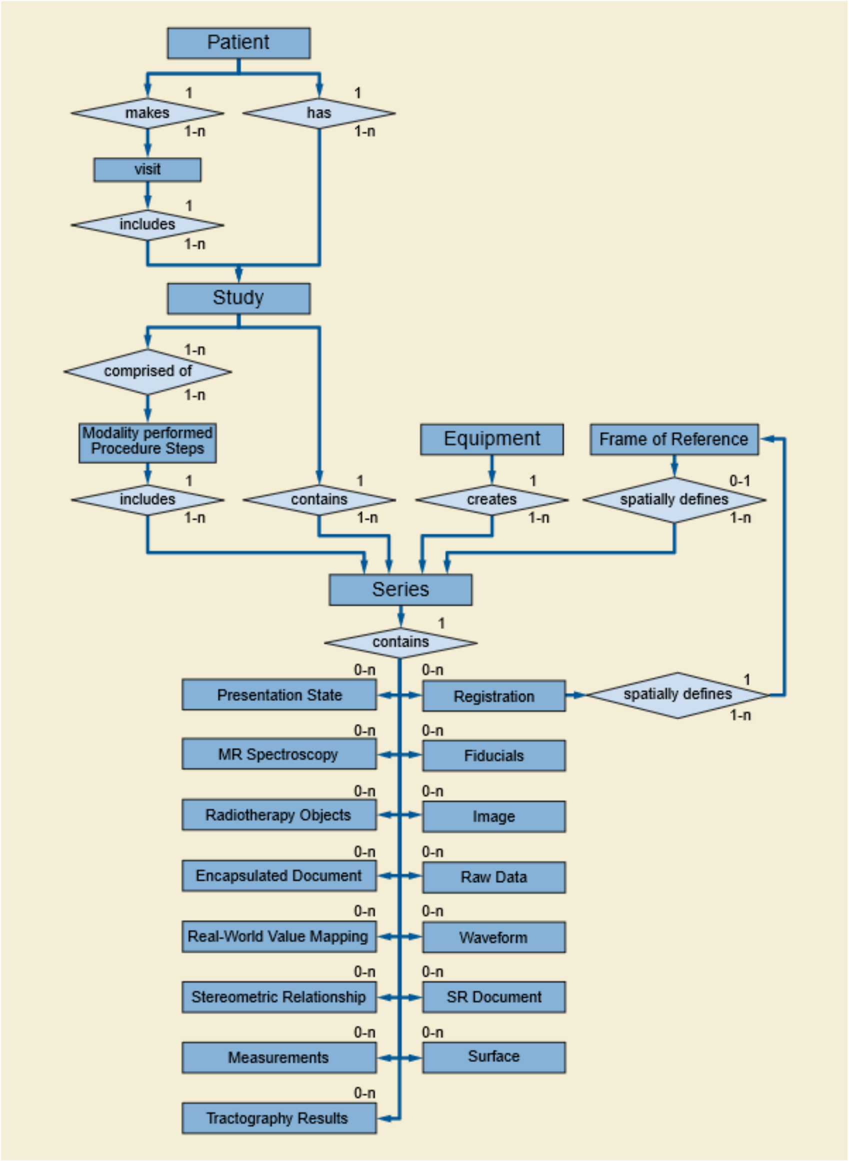

Patient
makes
1
1-n
has
1
1-n
visit
includes
1
1-n
Study
comprised of
1-n
1-n
Modality performed Procedure Steps
includes
1
1-n
contains
1
1-n
Equipment
creates
1
1-n
Frame of Reference
spatially defines
0-1
1-n
Series
contains
1
0-n
Presentation State
Registration
spatially defines
1
1-n
MR Spectroscopy
Fiducials
Radiotherapy Objects
Image
Encapsulated Document
Raw Data
Real-World Value Mapping
Waveform
Stereometric Relationship
SR Document
Measurements
Surface
Tractography Results

Figure 29 - DICOM Model of the real World [12, p. 111]

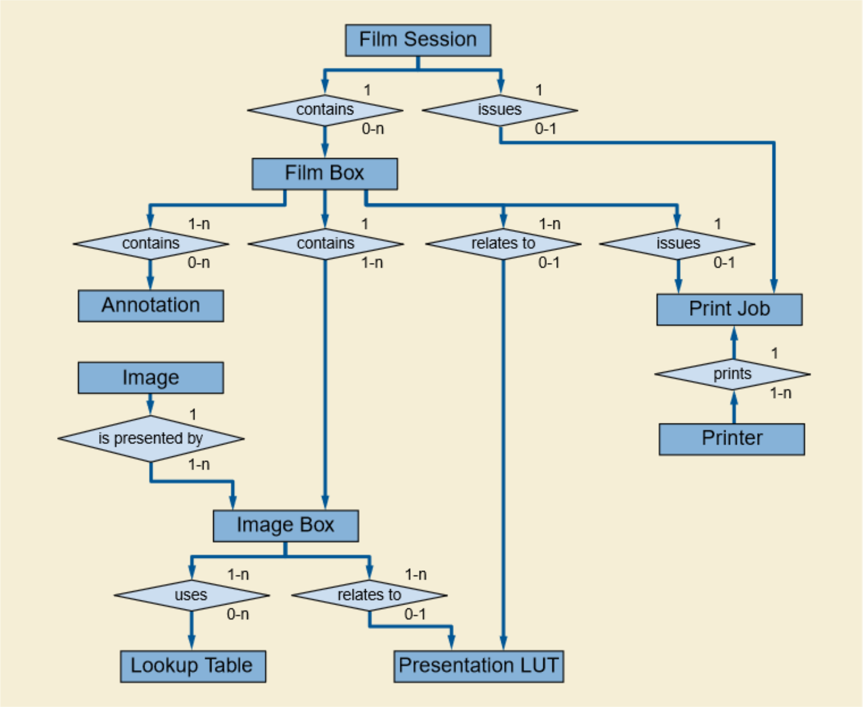

Film Session
1
contains
0-n
1
issues
0-1
Film Box
1-n
contains
0-n
1
contains
1-n
1-n
relates to
0-1
1
issues
0-1
Annotation
Print Job
1
prints
1-n
Printer
Image
1
is presented by
1-n
Image Box
1-n
uses
0-n
1-n
relates to
0-1
Lookup Table
Presentation LUT

Figure 30 - DICOM Model of the real world - PRINT [12, p.112]

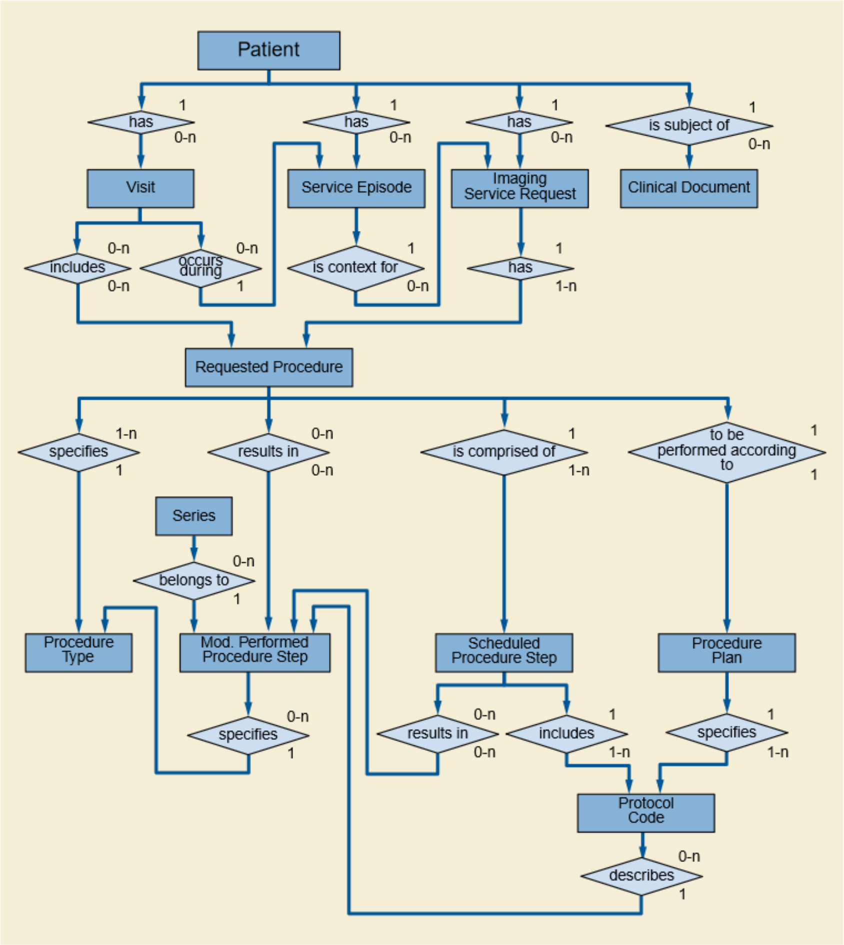

Patient
1
has
0-n
1
has
0-n
1
has
0-n
1
is subject of
0-n
Visit
Service Episode
Imaging Service Request
Clinical Document
0-n
includes
0-n
0-n
occurs during
1
1
is context for
0-n
1
has
1-n
Requested Procedure
1-n
specifies
1
0-n
results in
0-n
1
is comprised of
1-n
1
to be performed according to
1
Series
0-n
belongs to
1
Procedure Type
Mod. Performed Procedure Step
Scheduled Procedure Step
Procedure Plan
0-n
specifies
1
0-n
results in
0-n
1
includes
1-n
1
specifies
1-n
Protocol Code
0-n
describes
1

Figure 31 - Model of the Real World for the Purpose of Modality-IS Interface [12, p.114]

“The DICOM Information Model is derived from the DICOM Model of the Real World. The DICOM Information Model presented by” Figure 32 and Figure 33 “identify the various IODs specified by this Standard and their relationships. There is not always a one-to-one correspondence between DICOM Information Object Definitions and Real-World Objects. For example, a Composite IOD contains Attributes of multiple real-world objects such as Series, Equipment, Frame of Reference, Study and Patient.” [12, p.114].

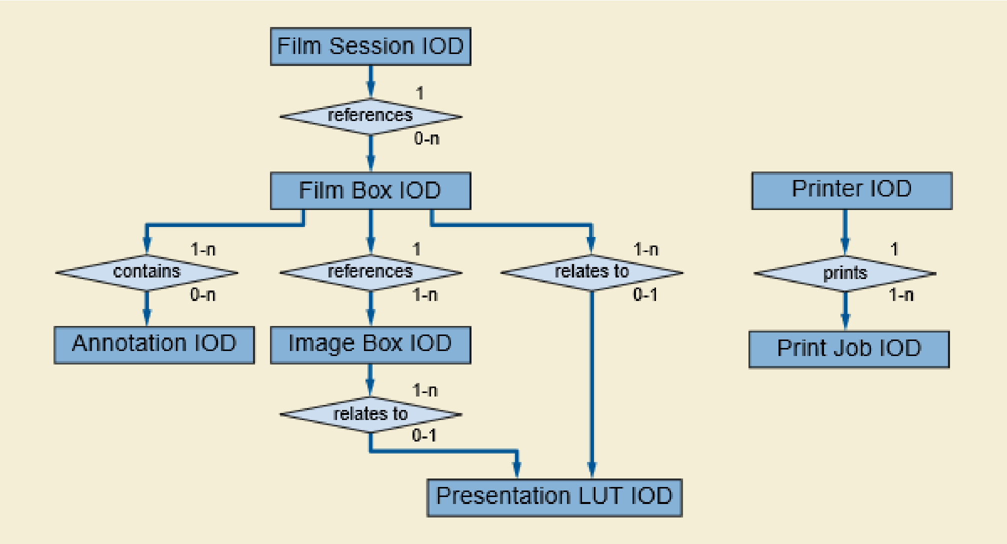

Film Session IOD
1
references
0-n
Film Box IOD
Printer IOD
1-n
contains
0-n
1
references
1-n
1-n
relates to
0-1
1
prints
1-n
Annotation IOD
Image Box IOD
Print Job IOD
1-n
relates to
0-1
Presentation LUT IOD

Figure 32 - DICOM Information Model - PRINT [12, p.112]

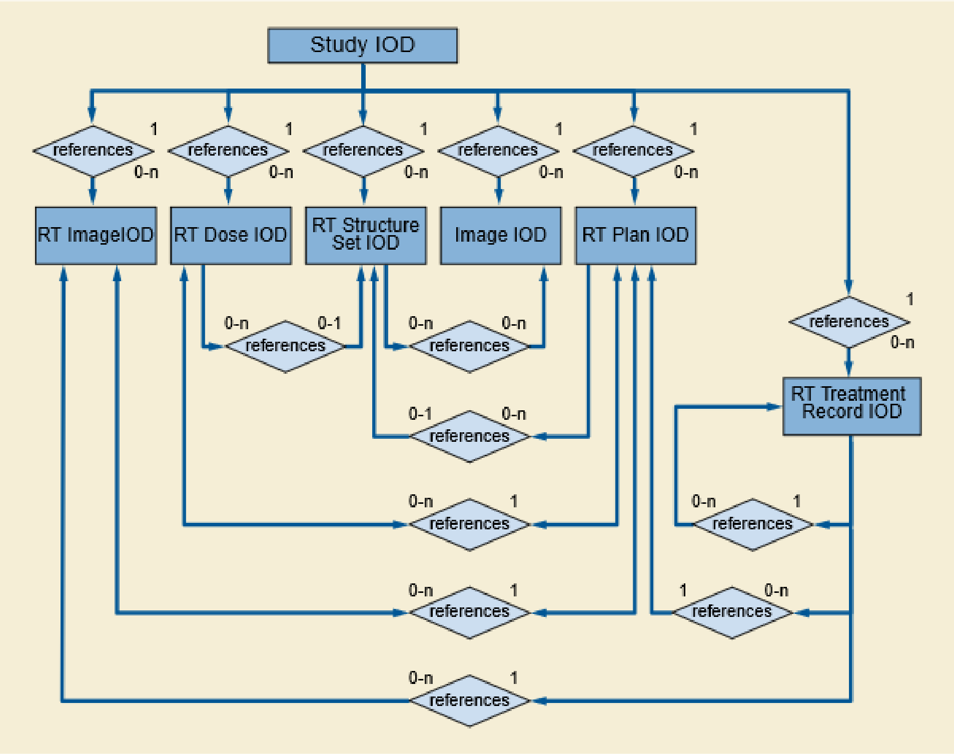

Study IOD
references
1
0-n
RT ImageIOD
RT Dose IOD
RT Structure Set IOD
Image IOD
RT Plan IOD
RT Treatment Record IOD
0-n
0-1
0-1
0-n
1

Figure 33 - DICOM Information Model - RADIOTHERAPY [12, p.113]

4.2 Basic Structure

The publication “DICOM PS3.5 2020b - Data Structures and Encoding” [11] specifies, as its name suggests, the structure and encoding of the data set. It specifies following topics:

- the encoding of Values (or Value Representations)
- the structure and usage of a Data Set
- Data Element usage and relationships to other elements
- the construction and usage of Nested Data Sets
- the construction and usage of Data Sets containing Pixel Data
- how to uniquely identify information
- the specification of the standard DICOM Transfer Syntaxes

The most important information is the structure of the Data Sets and the construction of the Data Sets containing Pixel Data.

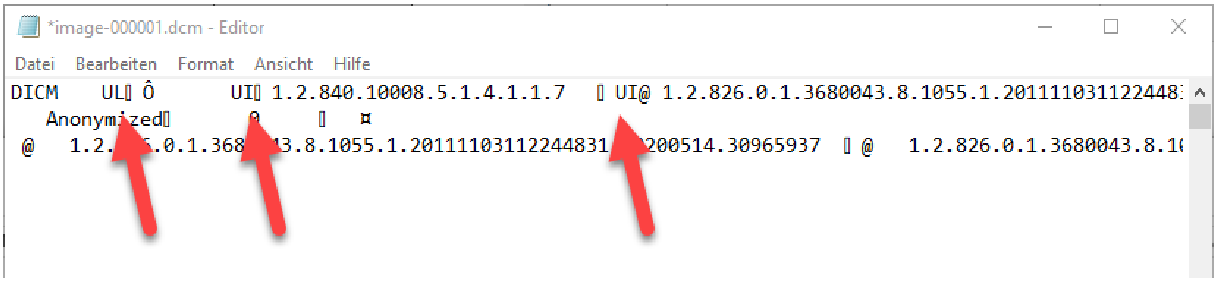
4.2.1 Value Representations

“The Value Representation of a Data Element describes the data type and format of that Data Element's Value(s). PS3.6 lists the VR of each Data Element by Data Element Tag” [11, p.37]. For PS3.6 see Figure 27. The table 6.2-1 lists all value representations of the format [11, p.38 – p.46). Below a few examples are listed.

Examples:

UL:	Unsigned binary integer 32 bits long. Represents an integer in in the range 0 <= 2^{32}
OB	Other byte, an octet-stream where the encoding of the contents is specified by the negotiated Transfer Syntax
UI:	Unique Identifier, a character string containing a unique identifier to uniquely identify a wide variety of items. The UID is a series of numeric components separated by the period "." character.

A few Value Representations within the DICOM file are shown in Fig. 34:

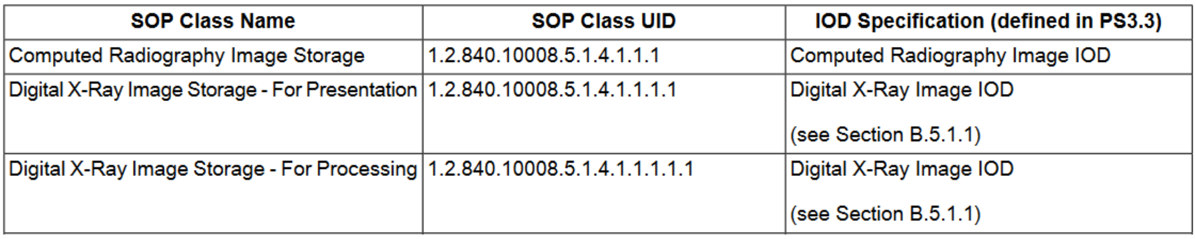


```
DICM    UL▯ Ô       UI▯ 1.2.840.10008.5.1.4.1.1.7   ▯ UI@ 1.2.826.0.1.3680043.8.1055.1.201111031122448
   Anonymi zed▯        0        ▯    ¤
 @   1.2.  5.0.1.368  3.8.1055.1.20111103112244831  200514.30965937  ▯ @   1.2.826.0.1.3680043.8.1
```

Figure 34 - Value Representations in a DICOM file [own illustration]

This illustrates that DICOM data format is stored as a mix of ASCII and binary code. Due to the binary parts it is not possible to read it like e.g. VRML (see chapters 6 & 7).

The numbers behind the VR are so called SOP Class UID (Service-Object Pair Class Unique Identifiers). “The SOP Class definition contains the rules and semantics which may restrict the use of the services in the DIMSE Service Group or the Attributes of the IOD (Information Object Definition). Examples of Service Elements are Store, Get, Find, Move, etc. Examples of Objects are CT images, MR images, but also include schedule lists, print queues, etc.” [13]. All SOP Classes are defined in PS3.4 [14, p.72 – p.76], a few of them are listed in table 3.



SOP Class Name	SOP Class UID	IOD Specification (defined in PS3.3)
Computed Radiography Image Storage	1.2.840.10008.5.1.4.1.1.1	Computed Radiography Image IOD
Digital X-Ray Image Storage - For Presentation	1.2.840.10008.5.1.4.1.1.1.1	Digital X-Ray Image IOD (see Section B.5.1.1)
Digital X-Ray Image Storage - For Processing	1.2.840.10008.5.1.4.1.1.1.1.1	Digital X-Ray Image IOD (see Section B.5.1.1)

Table 3 - Extract from Table B.5.1. Standard SOP Classes [14]

A specific Value Representation are Person Name (PN) Value Representations. A few examples given by the documentation [11, p.47] are listed below:

Examples:

- Rev. John Robert Quincy Adams, B.A. M.Div.
  "Adams^John Robert Quincy^^Rev.^B.A. M.Div."
  [One family name; three given names; no middle name; one prefix; two suffixes.]
- Susan Morrison-Jones, Ph.D., Chief Executive Officer
  "Morrison-Jones^Susan^^^Ph.D., Chief Executive Officer"
  [Two family names; one given name; no middle name; no prefix; two suffixes.]
- John Doe
  "Doe^John"
  [One family name; one given name; no middle name, prefix, or suffix. Delimiters have been omitted for the three trailing null components.]

There are further Value Representations such as Unknown (UN) or URI/URL (UR) VRs. Generally speaking, these VRs are used to store the meta data information in the file.

4.2.2 The Data Set

“A Data Set represents an instance of a real-world Information Object. A Data Set is constructed of Data Elements. Data Elements contain the encoded Values of Attributes of that object. The specific content and semantics of these Attributes are specified in Information Object Definitions” [11, p.51].

Data Elements

“A Data Element is uniquely identified by a Data Element Tag. The Data Elements in a Data Set shall be ordered by increasing Data Element Tag Number and shall occur at most once in a Data Set. A Data Element shall have one of three structures. Two of these structures contain the VR of the Data Element (Explicit VR) but differ in the way their lengths are expressed, while the other structure does not contain the VR (Implicit VR). All three structures contain the Data Element Tag, Value Length and Value for the Data Element” [11, p.51]. This structure is illustrated in Fig. 35.

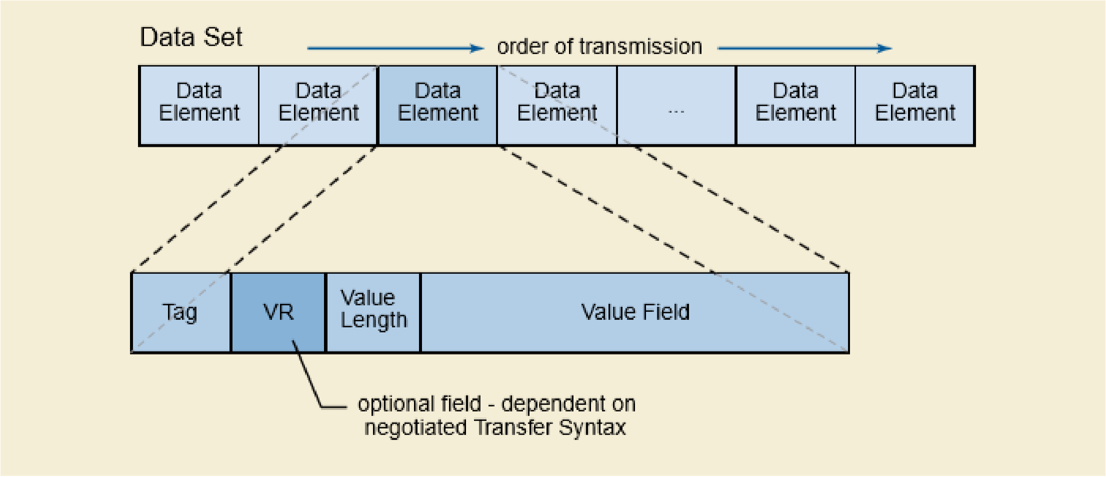


Figure 35 - DICOM Data Set and Element Structures [11, p.52]

“A Data Element is made up of fields. Three fields are common to all three Data Element structures; these are the Data Element Tag, Value Length, and Value Field. A fourth field, Value Representation (VR), is only present in the two Explicit VR Data Element structures. The definitions of the fields are:

Data Element Tag: An ordered pair of 16-bit unsigned integers representing the Group Number followed by Element Number.

Value Representation: Two single byte characters containing the VR of the Data Element. The VR for a given Data Element Tag shall be as defined by the Data Dictionary as specified in PS3.6. The two byte VR shall be encoded using only upper case letters from the DICOM default character set.

Value Length: Either:
- a 16 or 32-bit (dependent on VR and whether VR is explicit or implicit) unsigned integer containing the Explicit Length of the Value Field as the number of bytes (even) that make up the Value. It does not include the length of the Data Element Tag, Value Representation, and Value Length Fields.

- a 32-bit Length Field set to Undefined Length (FFFFFFFFH). Undefined Lengths may be used for Data Elements having the Value Representation (VR) Sequence of Items (SQ) and Unknown (UN). For Data Elements with Value Representation OW or OB Undefined Length may be used depending on the negotiated Transfer Syntax (see Section 10 and Annex A).

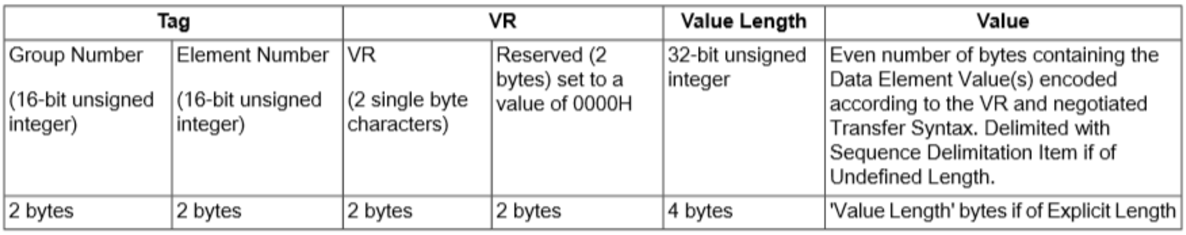
Note:
1. The decoder of a Data Set should support both Explicit and Undefined Lengths for VRs of SQ and UN and, when applicable, for VRs of OW and OB.
2. The 32-bit Value Length Field limits the maximum size of large data values such as Pixel Data sent in a Native Format (encoded in Transfer Syntaxes that use only the unencapsulated form).

Value Field: An even number of bytes containing the Value(s) of the Data Element.

The data type of Value(s) stored in this field is specified by the Data Element's VR. The VR for a given Data Element Tag can be determined using the Data Dictionary in PS3.6, or using the VR Field if it is contained explicitly within

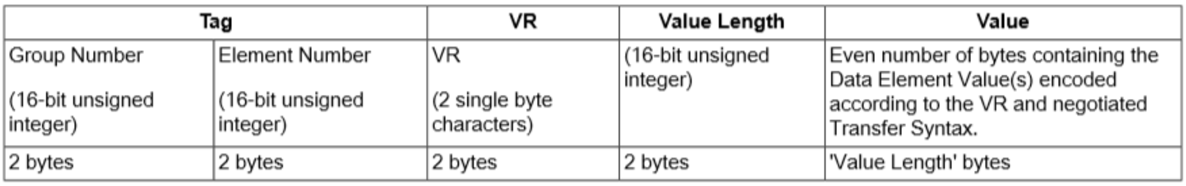
> the Data Element. The VR of Standard Data Elements shall agree with those specified in the Data Dictionary.
>
> The Value Multiplicity specifies how many Values with this VR can be placed in the Value Field. If the VM is greater than one, multiple values shall be delimited within the Value Field as defined previously in Section 6.4. The VMs of Standard Data Elements are specified in the Data Dictionary in PS3.6.
>
> Value Fields with Undefined Length are delimited through the use of Sequence Delimitation Items and Item Delimitation Data Elements, which are described further in Section 7.5" [11, p.52].

There are three different data element types, two with explicit VRs and one with implicit VR.



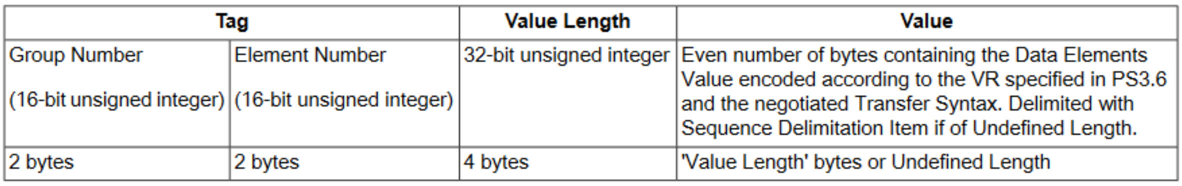
Tag		VR		Value Length	Value
Group Number (16-bit unsigned integer)	Element Number (16-bit unsigned integer)	VR (2 single byte characters)	Reserved (2 bytes) set to a value of 0000H	32-bit unsigned integer	Even number of bytes containing the Data Element Value(s) encoded according to the VR and negotiated Transfer Syntax. Delimited with Sequence Delimitation Item if of Undefined Length.
2 bytes	2 bytes	2 bytes	2 bytes	4 bytes	'Value Length' bytes if of Explicit Length

Table 4 - Data Element with Explicit VR other than in table 5 [11, p.53]



Tag		VR	Value Length	Value
Group Number (16-bit unsigned integer)	Element Number (16-bit unsigned integer)	VR (2 single byte characters)	(16-bit unsigned integer)	Even number of bytes containing the Data Element Value(s) encoded according to the VR and negotiated Transfer Syntax.
2 bytes	2 bytes	2 bytes	2 bytes	'Value Length' bytes

Table 5 - Data Element with Explicit VR of AE, AS, AT, and many more [11, p.53]



Tag		Value Length	Value
Group Number (16-bit unsigned integer)	Element Number (16-bit unsigned integer)	32-bit unsigned integer	Even number of bytes containing the Data Elements Value encoded according to the VR specified in PS3.6 and the negotiated Transfer Syntax. Delimited with Sequence Delimitation Item if of Undefined Length.
2 bytes	2 bytes	4 bytes	'Value Length' bytes or Undefined Length

Table 6 - Data Element with Implicit VR [11, p.54]

4.2.3 Unique Identifiers (UIDs)

"Unique Identifiers (UIDs) provide the capability to uniquely identify a wide variety of items. They guarantee uniqueness across multiple countries, sites, vendors and equipment. Different classes of objects, instance of objects and information entities can be distinguished from one another across the DICOM universe of discourse irrespective of any semantic context" [11, p 89].

4.3 Media Storage and File Format for Media Interchange

“This Part of the DICOM Standard specifies a general model for the storage of Medical Imaging information on removable media. The purpose of this Part is to provide a framework allowing the interchange of various types of medical images and related information on a broad range of physical storage media” [15, p. 13].

4.3.1 DICOM Models for Media Storage

First, the general communications model has to be introduced to give an overview:

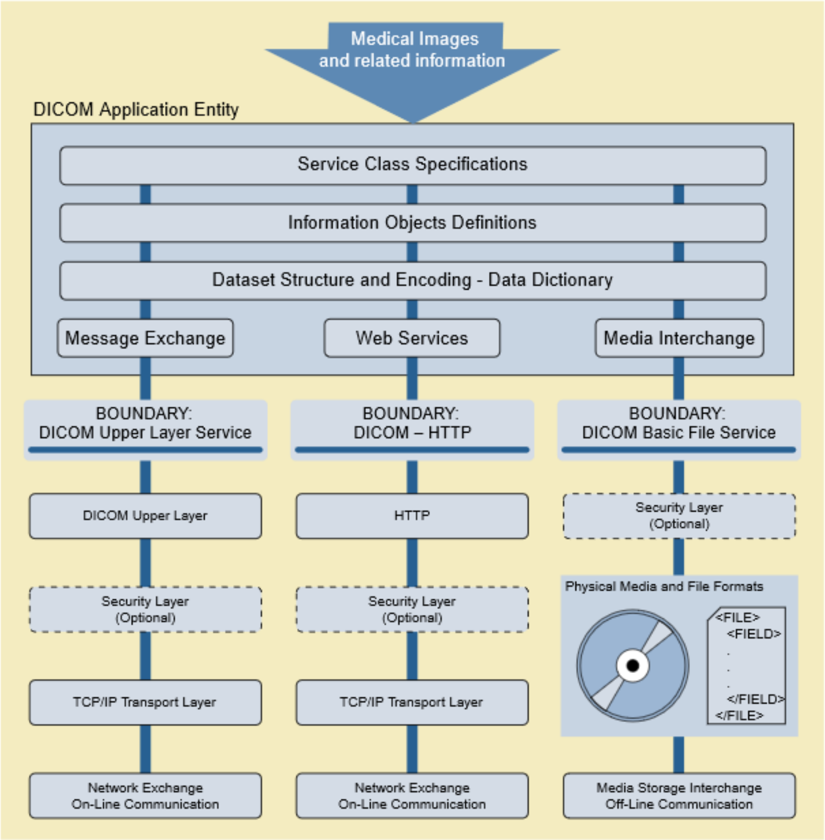

Medical Images
and related information
DICOM Application Entity
Service Class Specifications
Information Objects Definitions
Dataset Structure and Encoding - Data Dictionary
Message Exchange
Web Services
Media Interchange
BOUNDARY: DICOM Upper Layer Service
BOUNDARY: DICOM – HTTP
BOUNDARY: DICOM Basic File Service
DICOM Upper Layer
HTTP
Security Layer (Optional)
Security Layer (Optional)
Security Layer (Optional)
Physical Media and File Formats
<FILE>
<FIELD>
</FIELD>
</FILE>
TCP/IP Transport Layer
TCP/IP Transport Layer
Network Exchange On-Line Communication
Network Exchange On-Line Communication
Media Storage Interchange Off-Line Communication

Figure 36 - General Communications Model [16. p.22]

The basic file service of DICOM is the media interchange on the bottom right of Fig. 36. The DICOM Application Entity comprises of the DICOM File Format, the Basic Directory, and service / Object pairs, illustrated in Fig. 37.

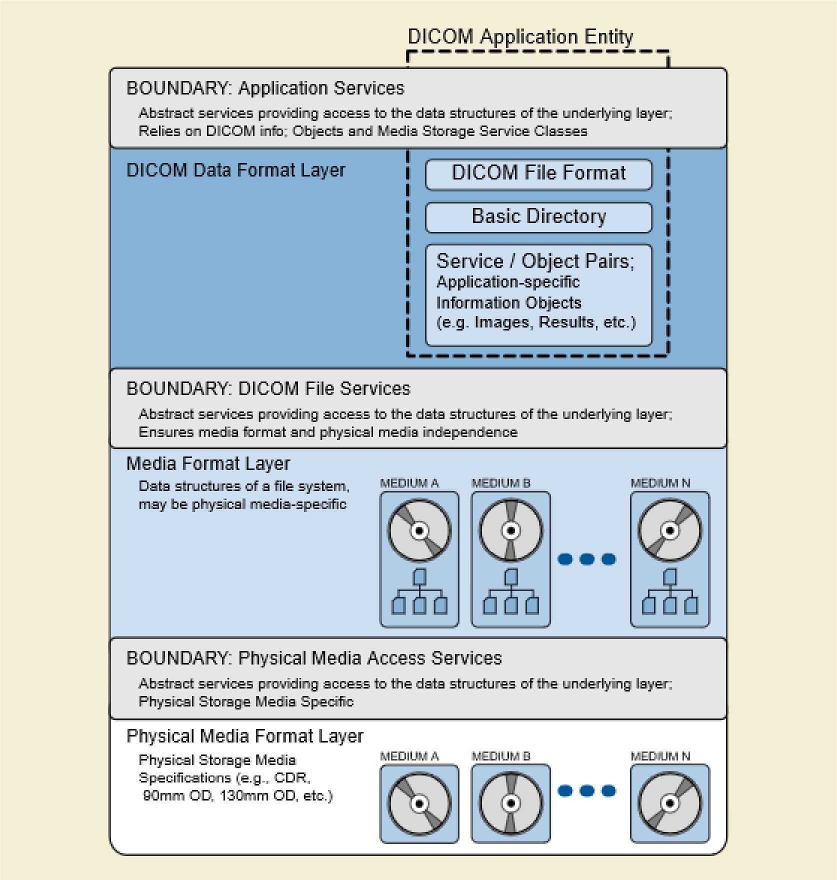

DICOM Application Entity
BOUNDARY: Application Services
Abstract services providing access to the data structures of the underlying layer;
Relies on DICOM info; Objects and Media Storage Service Classes
DICOM Data Format Layer
DICOM File Format
Basic Directory
Service / Object Pairs;
Application-specific
Information Objects
(e.g. Images, Results, etc.)
BOUNDARY: DICOM File Services
Abstract services providing access to the data structures of the underlying layer;
Ensures media format and physical media independence
Media Format Layer
Data structures of a file system,
may be physical media-specific
MEDIUM A
MEDIUM B
MEDIUM N
BOUNDARY: Physical Media Access Services
Abstract services providing access to the data structures of the underlying layer;
Physical Storage Media Specific
Physical Media Format Layer
Physical Storage Media
Specifications (e.g., CDR,
90mm OD, 130mm OD, etc.)
MEDIUM A
MEDIUM B
MEDIUM N

Figure 37 - DICOM Media Storage Model [15, p. 26]

4.3.2 DICOM File Format

“The DICOM File Format provides a means to encapsulate in a file the Data Set representing a SOP Instance related to a DICOM IOD. The byte stream of the Data Set is placed into the file after the DICOM File Meta Information. Each file contains a single SOP Instance”, see Fig. 38 [15, p.31].

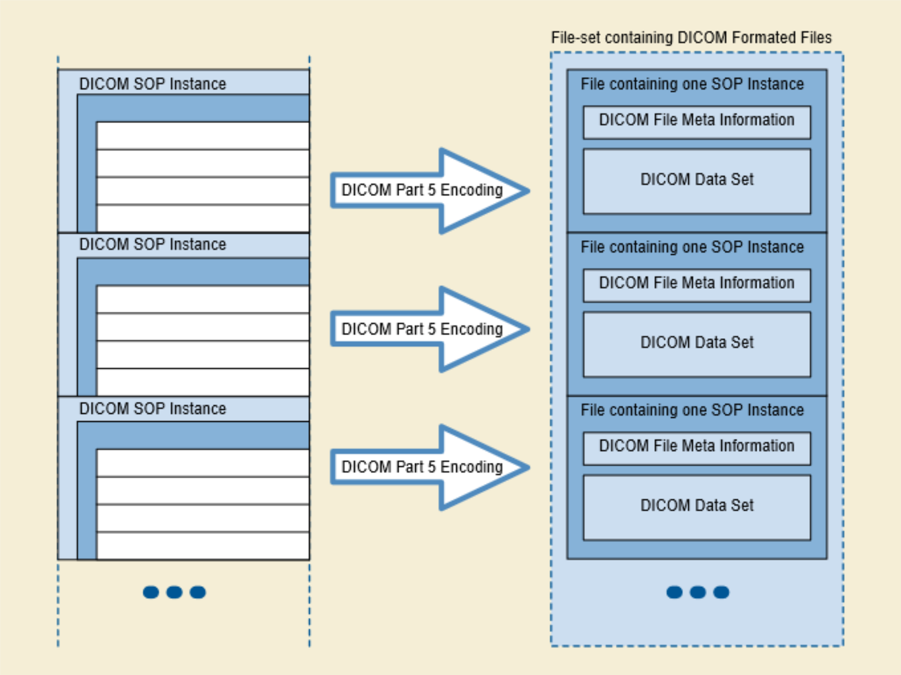
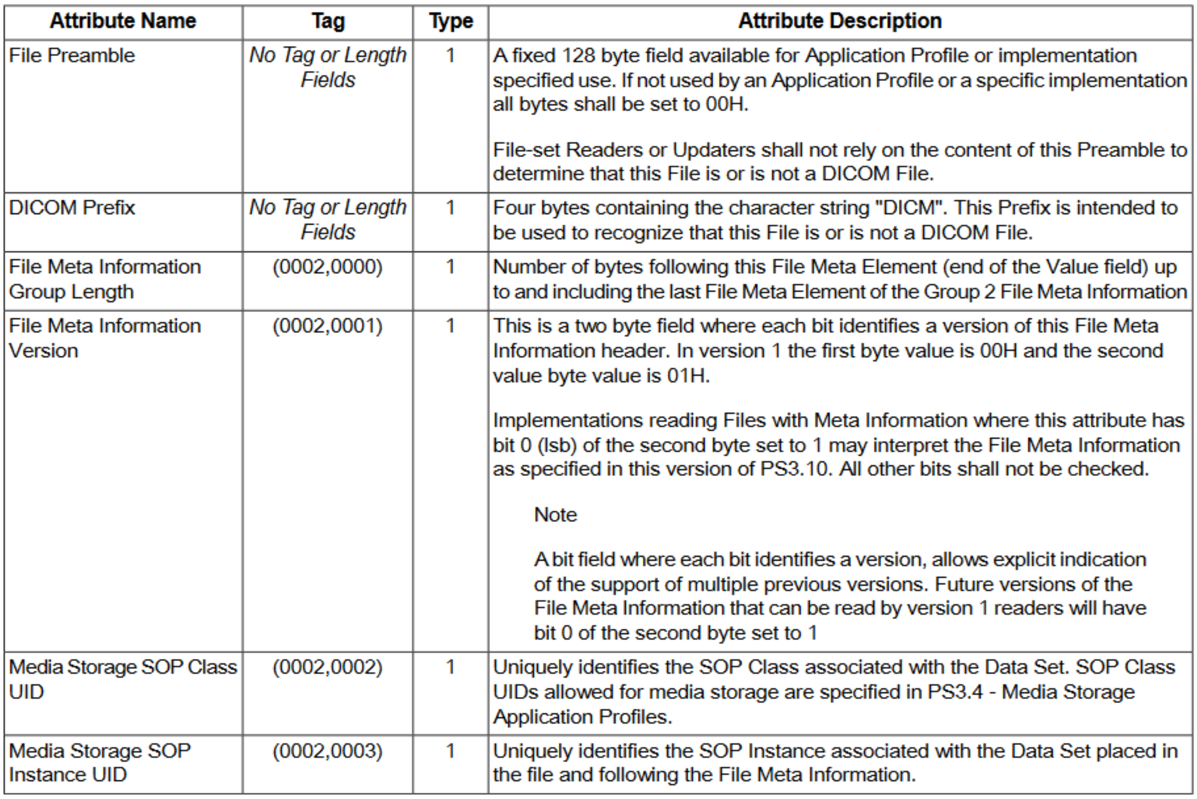

File-set containing DICOM Formated Files
DICOM SOP Instance
DICOM Part 5 Encoding
File containing one SOP Instance
DICOM File Meta Information
DICOM Data Set
DICOM SOP Instance
DICOM Part 5 Encoding
File containing one SOP Instance
DICOM File Meta Information
DICOM Data Set
DICOM SOP Instance
DICOM Part 5 Encoding
File containing one SOP Instance
DICOM File Meta Information
DICOM Data Set

Figure 38 - File-set and File Format [15, p.31]

DICOM File Meta Information

"The File Meta Information includes identifying information on the encapsulated Data Set. This header consists of a 128 byte File Preamble, followed by a 4 byte DICOM prefix, followed by the File Meta Elements shown in Table 7.1-1" [15, p.31]. An extract of this table is shown in table 7:



Attribute Name	Tag	Type	Attribute Description
File Preamble	*No Tag or Length Fields*	1	A fixed 128 byte field available for Application Profile or implementation specified use. If not used by an Application Profile or a specific implementation all bytes shall be set to 00H. File-set Readers or Updaters shall not rely on the content of this Preamble to determine that this File is or is not a DICOM File.
DICOM Prefix	*No Tag or Length Fields*	1	Four bytes containing the character string "DICM". This Prefix is intended to be used to recognize that this File is or is not a DICOM File.
File Meta Information Group Length	(0002,0000)	1	Number of bytes following this File Meta Element (end of the Value field) up to and including the last File Meta Element of the Group 2 File Meta Information
File Meta Information Version	(0002,0001)	1	This is a two byte field where each bit identifies a version of this File Meta Information header. In version 1 the first byte value is 00H and the second value byte value is 01H. Implementations reading Files with Meta Information where this attribute has bit 0 (lsb) of the second byte set to 1 may interpret the File Meta Information as specified in this version of PS3.10. All other bits shall not be checked. Note A bit field where each bit identifies a version, allows explicit indication of the support of multiple previous versions. Future versions of the File Meta Information that can be read by version 1 readers will have bit 0 of the second byte set to 1
Media Storage SOP Class UID	(0002,0002)	1	Uniquely identifies the SOP Class associated with the Data Set. SOP Class UIDs allowed for media storage are specified in PS3.4 - Media Storage Application Profiles.
Media Storage SOP Instance UID	(0002,0003)	1	Uniquely identifies the SOP Instance associated with the Data Set placed in the file and following the File Meta Information.

Table 7 - Extract of DICOM Meta File Information [15, p.32 – p.33]

This meta data can be read and displayed, for further information see chapter 5. An example of DICOMDIR file content is given in Annex A of PS3.10, again only an extract is shown below:

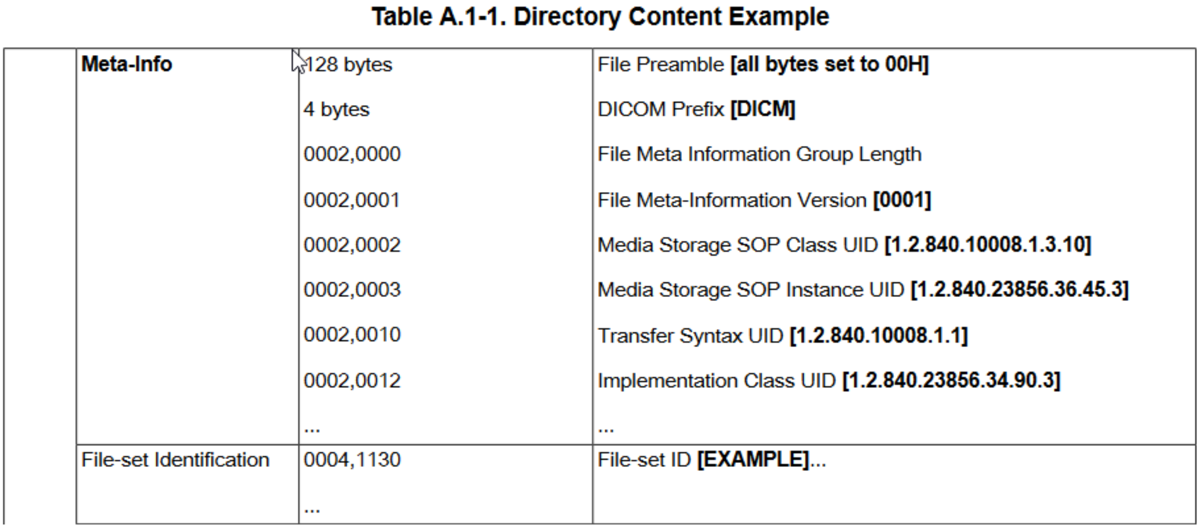


Table A.1-1. Directory Content Example

	Meta-Info	128 bytes	File Preamble **[all bytes set to 00H]**
		4 bytes	DICOM Prefix **[DICM]**
		0002,0000	File Meta Information Group Length
		0002,0001	File Meta-Information Version **[0001]**
		0002,0002	Media Storage SOP Class UID **[1.2.840.10008.1.3.10]**
		0002,0003	Media Storage SOP Instance UID **[1.2.840.23856.36.45.3]**
		0002,0010	Transfer Syntax UID **[1.2.840.10008.1.1]**
		0002,0012	Implementation Class UID **[1.2.840.23856.34.90.3]**
		...	...
	File-set Identification	0004,1130	File-set ID **[EXAMPLE]**...
		...	

Table 8 - Extract of Directory Content Example [15, p.45]

4.4 Media Formats and Physical Media for Media Interchange

The described document “facilitates the interchange of information between digital imaging computer systems in medical environments” [17, p.15]. Following figures show the structures that support a single DICOM file-set per medium partition or support multiple DICOM file-sets.

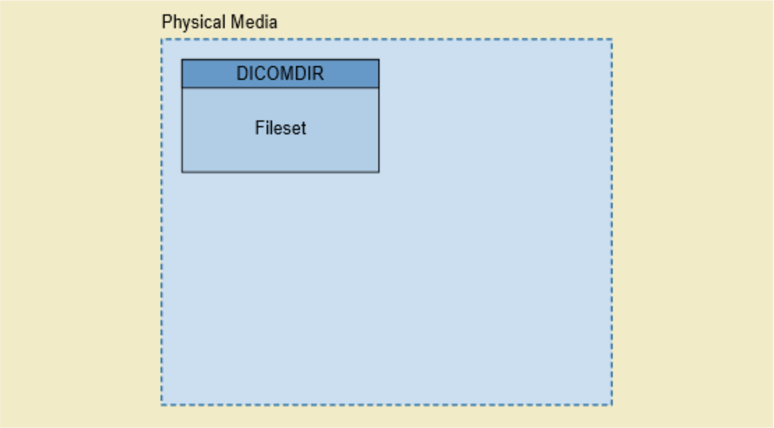


Figure 39 - Media Supporting a Single File Set [17, p.25]

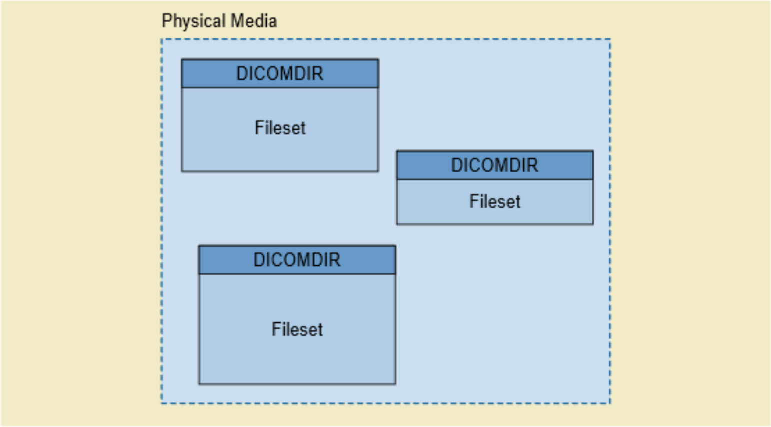


Figure 40 - Media Supporting Multiple File-sets [17, p.26]

4.4.1 PC File System

In table 9, an example of a PC file system is illustrated:

File ID	PC File system name
DICOMDIR	\DICOMDIR
FILENAME	\FILENAME
SUBDIR\FILENAME	\SUBDIR\FILENAME

Table 9 - Example File ID mappings [17, p.27]

“The PC File System provides a hierarchical structure for directories and files within directories. Each structure has a root directory that may contain references to both files and sub-directories. Sub-directories may contain references to both files and other sub-directories” [17, p.27]. There are many other files systems defined, but this paper focuses on the PC file system as this is assumed to be the most common.

4.5 Grayscale Standard Display Function

This chapter is about the specification of a standardized display function for consistent display of grayscale images [18, p.13]. This function provides methods for calibrating a particular display system for the purpose of presenting images consistently on different display media (e.g., monitors and printers). The chosen display function is based on human visual perception. Human eye contrast sensitivity is distinctly non-linear within the luminance range of display devices. This Standard uses Barten's model of the human visual system [19, 20].

4.5.1 Definitions

A few important definitions from the publication are listed below, for further definitions see [18, p.17].

Digital Driving Level (DDL)	A digital value that given as input to a Display System produces a Luminance. The set of DDLs of a Display System is all the possible discrete values that can produce Luminance values on that Display System. The mapping of DDLs to Luminance values for a Display System produces the Characteristic Curve of that Display System. The actual output for a given DDL is specific to the Display System and is not corrected for the Grayscale Standard Display Function
JND Index	The input value to the Grayscale Standard Display Function, such that one step in JND Index results in a Luminance difference that is a Just-Noticeable Difference
P-Value	A device independent value defined in a perceptually linear grayscale space. The output of the DICOM Presentation LUT (Look Up Table) is P-Values, i.e. the pixel value after all DICOM defined grayscale transformations have been applied. P-Values are the input to a Standardized Display System.

4.5.2 Overview

“Display Systems take a Digital Driving Level and produce Luminance or optical density variations that represent the image. Predictable application of image transformations, such as the modality, value-of-interest, and presentation look- up tables specified in the DICOM Standard, requires knowledge of the Characteristic Curve of the Display System. Standardizing the response function expected of the Display System simplifies the application of such image transformations across several different Display Systems such as encountered in a network environment” [18, p.23].

Figure 41 and Figure 42 “show the context for the Grayscale Standard Display Function. The Grayscale Standard Display Function is part of the image presentation. There will be a number of other modifications to the image before the Grayscale Standard Display Function is applied. The image acquisition device will adjust the image as it is formed. Other elements may perform a "window and level" to select a part of the dynamic range of the image to be presented. Yet other elements can adjust the selected dynamic range in preparation for display. The Presentation LUT outputs P-Values (presentation values). These P-Values become the Digital Driving Levels for Standardized Display Systems. The Grayscale Standard Display Function maps P-Values to the log-luminance output of the Standardized Display System. How a Standardized Display System performs this mapping is implementation dependent.

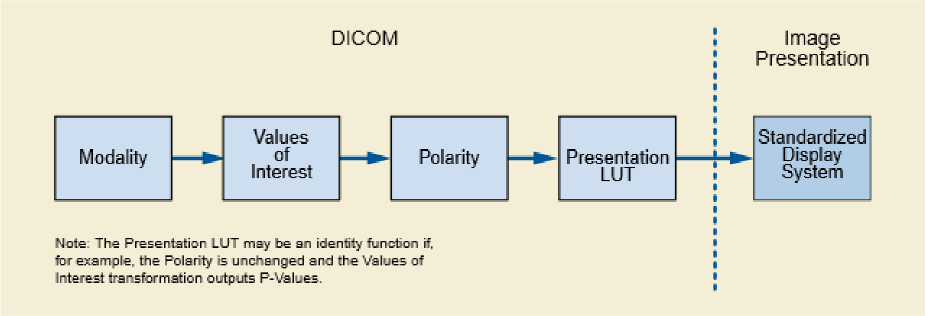

DICOM
Image Presentation
Modality
Values of Interest
Polarity
Presentation LUT
Standardized Display System
Note: The Presentation LUT may be an identity function if, for example, the Polarity is unchanged and the Values of Interest transformation outputs P-Values.

Figure 41 - The Grayscale Display Function is an element of the image presentation after several modifications to the image have been completed by other elements of the image acquisition and presentation chain [18, p.23]

The boundary between the DICOM model of the image acquisition and presentation chain, and the Standardized Display System, expressed in P-Values, is intended to be both device independent and conceptually (if not actually) perceptually linear. In other words, regardless of the capabilities of the Standardized Display System, the same range of P-Values will be presented” [18, p.23]

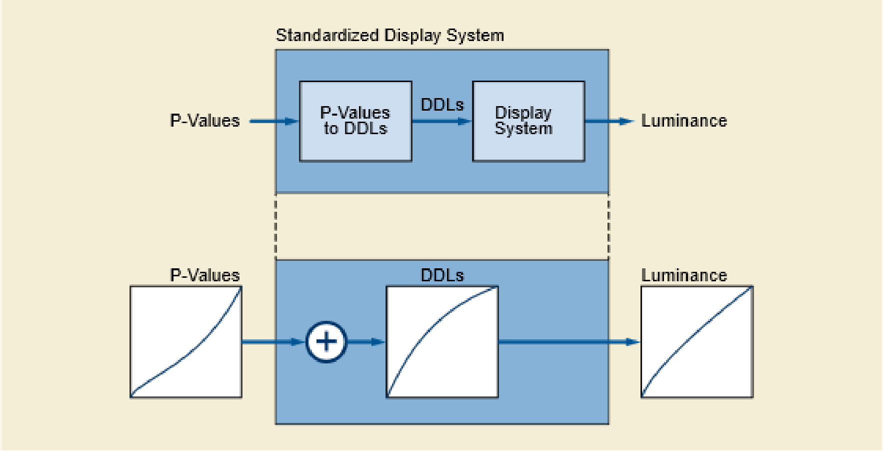

Standardized Display System
P-Values
P-Values to DDLs
DDLs
Display System
Luminance
P-Values
DDLs
Luminance

Figure 42 - The conceptual model of a Standardized Display System maps P-Values to Luminance via an intermediate transformation to Digital Driving Levels of an unstandardized Display System [18, p.24]

“The main objective is to define mathematically an appropriate Grayscale Standard Display Function for all image presentation systems. The purpose of defining this Grayscale Standard Display Function is to allow applications to know a priori how P-Values are transformed to viewed Luminance values by a Standardized Display System. In essence, defining the Grayscale Standard Display Function fixes the "units" for the P-Values output from the Presentation LUT and used as Digital Driving Levels to Standardized Display Systems.” [18, p.24]

4.5.3 The Grayscale Standard Display Function

“The Grayscale Standard Display Function is based on human Contrast Sensitivity. Human Contrast Sensitivity is distinctly non-linear within the Luminance Range of the Grayscale Standard Display Function. The human eye is relatively less sensitive in the dark areas of an image than it is in the bright areas of an image. This variation in sensitivity makes it much easier to see small relative changes in Luminance in the bright areas of the image than in the dark areas of the image. A Display Function that adjusts the brightness such that equal changes in P-Values will result in the same level of perceptibility at all driving levels is "perceptually linearized". The Grayscale Standard Display Function incorporates the notion of perceptual linearization without making it an explicit objective of PS3.14.

The Grayscale Standard Display Function is defined for the Luminance Range from 0.05 to 4000 cd/m2. The minimum Luminance corresponds to the lowest practically useful Luminance of cathode-ray-tube (CRT) monitors and the maximum exceeds the unattenuated Luminance of very bright light-boxes used for interpreting X-Ray mammography. The Grayscale Standard Display Function explicitly includes the effects of the diffused ambient Illuminance. Within the Luminance Range happen to all 1023 Just Noticeable Difference (JNDs, see Annex A)" [18, p.25].

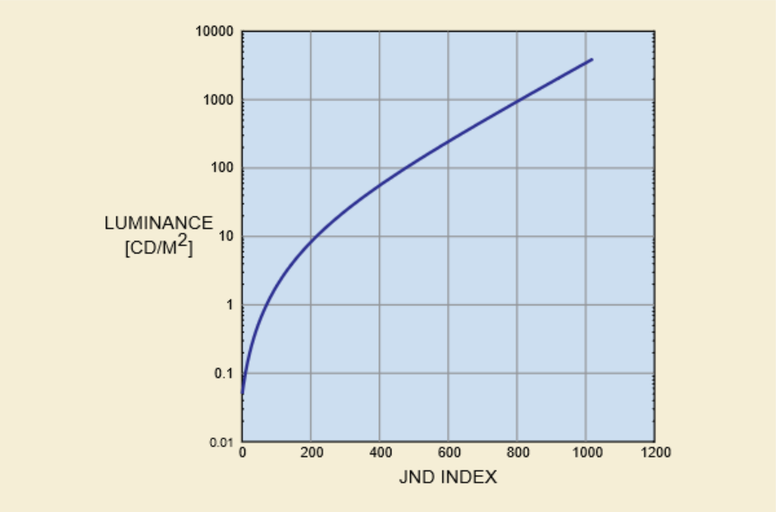


Figure 43 - The Grayscale Standard Display Function presented as logarithm-of luminance versus JND-Index [18, p.29]

Figure 43 illustrates the nonlinear distribution of the luminance of the display versus the JND Index. Table 10 shows an extract of the Grayscale Standard Display Function which shows as well that the steps of the luminance between the JNDs differ.

JND	L[cd/m 2]	JND	L[cd/m 2]	JND	L[cd/m 2]	JND	L[cd/m 2]
1	0.0500	2	0.0547	3	0.0594	4	0.0643
5	0.0696	6	0.0750	7	0.0807	8	0.0866
9	0.0927	10	0.0991	11	0.1056	12	0.1124
13	0.1194	14	0.1267	15	0.1342	16	0.1419
17	0.1498	18	0.1580	19	0.1664	20	0.1750

Table 10 – Extract of Grayscale Standard Display Function: Luminance versus JND Index [18, p.35]

4.5.4 The Process to the Presentation Look-Up-Table (Pixel Transformation Sequence)

As already mentioned in 4.5.1, the P-Values are perceptually linearly distributed, this is in respect to the perception of the human eye. These P-Values are stored in the Presentation LUT (see Fig. 41). However, the P-LUTs are generated by a previous process. The below process described is the Grayscale Transformation, further transformations are specified in [14, p.244 – p.252)

"For some SOP classes, the pixel data is guaranteed to be in P-values or to be accompanied by certain transformation parameters that produce P-values (Modality transformation, VOI (Value of Interest) Transformation, P-LUT Transformation). For other SOP classes, you have to know a priori how different vendors encode their pixel data (this is a particular problem for the old CR object). Nothing in the standard or in the image header as such will tell you" [21]. The usual SOP class is the so called "Softcopy Presentation State Storage SOP Class" specified in PS3.4 [14, p.239 – p.252].

"The Softcopy Presentation State Storage SOP Classes support a sequence of transformations that completely define the conversion of a stored image into a displayed image.
The sequence of transformations from stored pixel values into P-Values or PCS-Values is explicitly defined in a conceptual model. The actual sequence implemented may differ but must result in the same appearance" [14, p.240].

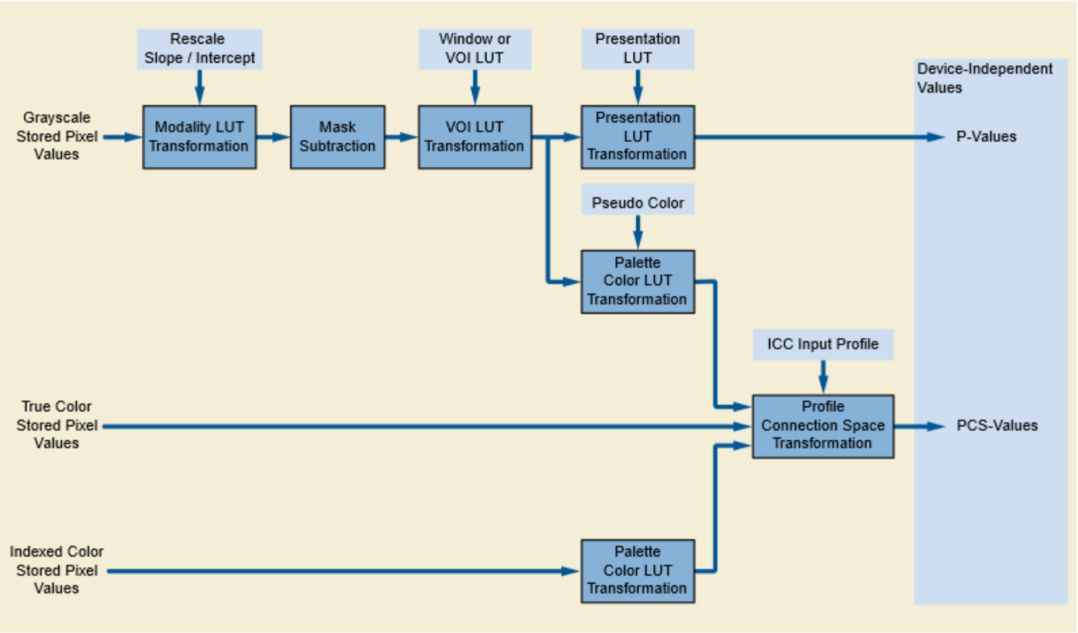

Rescale
Slope / Intercept
Window or
VOI LUT
Presentation
LUT
Device-Independent
Values
Grayscale
Stored Pixel
Values
Modality LUT
Transformation
Mask
Subtraction
VOI LUT
Transformation
Presentation
LUT
Transformation
P-Values
Pseudo Color
Palette
Color LUT
Transformation
ICC Input Profile
True Color
Stored Pixel
Values
Profile
Connection Space
Transformation
PCS-Values
Indexed Color
Stored Pixel
Values
Palette
Color LUT
Transformation

Figure 44 - Grayscale and Color Image Transformation Models [14, p.241]

"Values of MONOCHROME1 and MONOCHROME2 for Photometric Interpretation (0028,0004) in the Referenced Image SOP Instance shall be ignored, since their effect is defined by the application of the grayscale presentation state transformations" [14, p.240]. For further information, see chapter 4.6.

Modality LUT

The Modality LUT operation applies only to grayscale values. The Modality LUT transformation transforms the manufacturer dependent pixel values into pixel values that are meaningful for the modality and are manufacturer independent (e.g., Hounsfield number for CT modalities, Optical Density for film digitizers). These may represent physical units or be dimensionless. The Modality LUT in the Presentation State is modality dependent and is analogous to the same Module in an Image" [14, p. 241]

"In the case of a linear transformation, the Modality LUT is described by the Rescale Slope (0028,1053) and Rescale Intercept (0028,1052). In the case of a non-linear transformation, the Modality LUT is described by the Modality LUT Sequence. The rules for application of the Modality LUT are defined in Section C.11.1 “Modality LUT Module” in PS3.3.” [14, p.242]. For further information about Rescale, see chapter 4.6.

Mask
“The Mask operation applies only to grayscale values. The mask transformation may be applied in the case of multi-frame images for which other frames at a fixed frame position or time interval relative to the current frame may be subtracted from the current frame. Multiple mask frames may be averaged, and sub-pixel shifted before subtraction” [14, p.242]

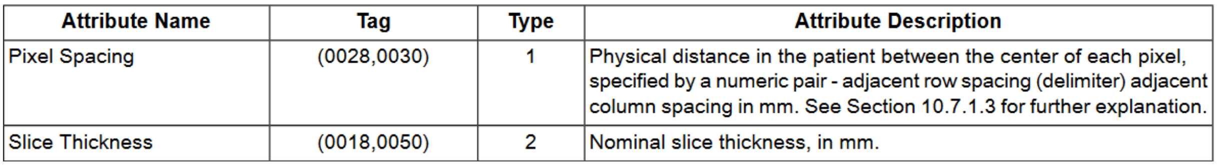
VOI LUT
“The VOI LUT operation applies only to grayscale values. The value of interest (VOI) LUT transformation transforms the modality pixel values into pixel values that are meaningful for the user or the application.
In the case of a linear transformation, the VOI LUT is described by the Window Center (0028,1050) and Window Width (0028,1051). In the case of a non-linear transformation, the VOI LUT is described by the VOI LUT Sequence” [14, p242 – p.243].
This is very relevant for the eventual P-Values, this is described more in detail in chapter 4.7.

Presentation LUT
“The Presentation LUT operation applies only to grayscale values. The Presentation LUT transformation transforms the pixel values into P-Values, a device independent perceptually linear space as defined in PS3.14 Grayscale Standard Display Function. It may be an identity function if the output of the VOI LUT transformation is in P-Values” [14, p.243].

4.6 Important Meta Data Attributes

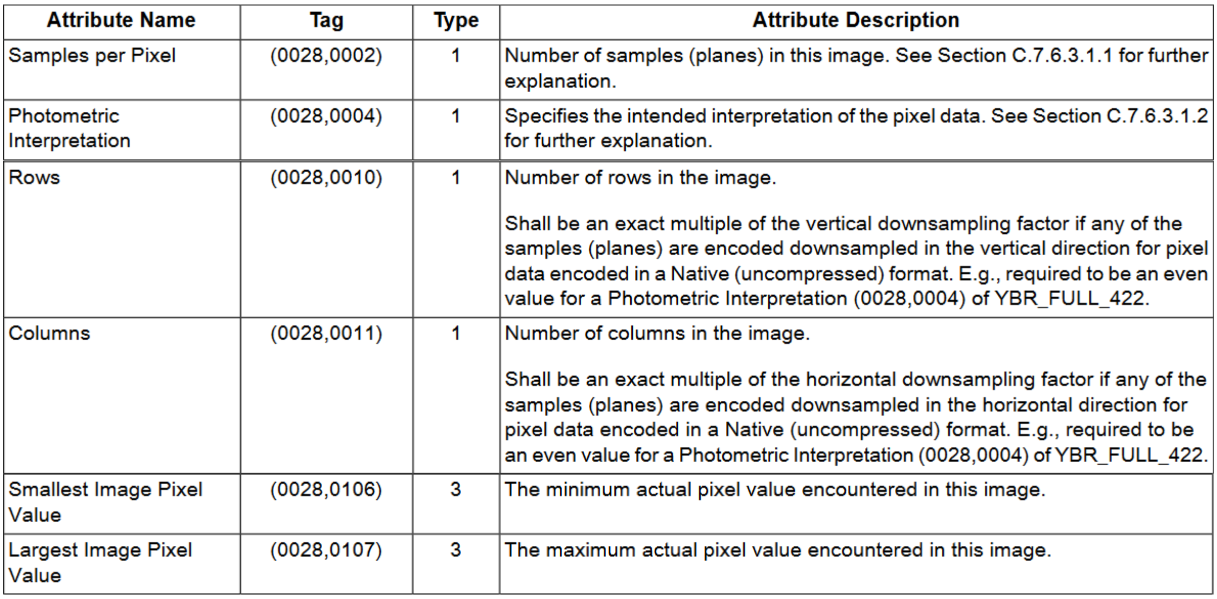
A few meta data attributes do have a major impact on the P-Values which are the basis of the grayscale image display of DICOM. Moreover, other attributes are important for the data consistency or represent the size of the image or the spacing of the CT slices. All of these attributes are specified and described in PS3.3 Annex C. The for this paper most relevant attributes are listed below.

Image Pixel Description



Attribute Name	Tag	Type	Attribute Description
Pixel Spacing	(0028,0030)	1	Physical distance in the patient between the center of each pixel, specified by a numeric pair - adjacent row spacing (delimiter) adjacent column spacing in mm. See Section 10.7.1.3 for further explanation.
Slice Thickness	(0018,0050)	2	Nominal slice thickness, in mm.

Table 11 - Excerpt of Image Plane Module Attributes [12, p.519]



Attribute Name	Tag	Type	Attribute Description
Samples per Pixel	(0028,0002)	1	Number of samples (planes) in this image. See Section C.7.6.3.1.1 for further explanation.
Photometric Interpretation	(0028,0004)	1	Specifies the intended interpretation of the pixel data. See Section C.7.6.3.1.2 for further explanation.
Rows	(0028,0010)	1	Number of rows in the image. Shall be an exact multiple of the vertical downsampling factor if any of the samples (planes) are encoded downsampled in the vertical direction for pixel data encoded in a Native (uncompressed) format. E.g., required to be an even value for a Photometric Interpretation (0028,0004) of YBR_FULL_422.
Columns	(0028,0011)	1	Number of columns in the image. Shall be an exact multiple of the horizontal downsampling factor if any of the samples (planes) are encoded downsampled in the horizontal direction for pixel data encoded in a Native (uncompressed) format. E.g., required to be an even value for a Photometric Interpretation (0028,0004) of YBR_FULL_422.
Smallest Image Pixel Value	(0028,0106)	3	The minimum actual pixel value encountered in this image.
Largest Image Pixel Value	(0028,0107)	3	The maximum actual pixel value encountered in this image.

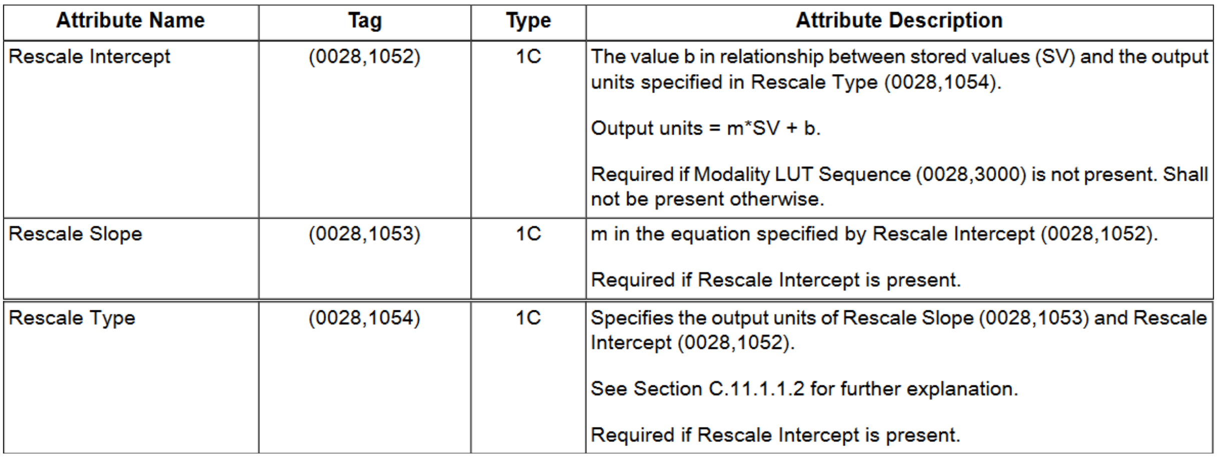
Table 12 - Excerpt of Image Pixel Description Macro Attributes [12, p.523 – p.525]

Photometric Interpretation

“The value of Photometric Interpretation (0028,0004) specifies the intended interpretation of the image pixel data” [12, p. 526]. A few interpretations are listed below, more are listed in PS3.3 p.526 – p.529.

MONOCHROME1	Pixel data represent a single monochrome image plane. The **minimum sample value** is intended to be displayed as **white** after any VOI gray scale transformations have been performed. See PS3.4. This value may be used only when Samples per Pixel (0028,0002) has a value of 1. May be used for pixel data in a Native (uncompressed) or Encapsulated (compressed) format; see Section 8.2 in PS3.5.
MONOCHROME2	Pixel data represent a single monochrome image plane. The **minimum sample value** is intended to be displayed as **black** after any VOI gray scale transformations have been performed. See PS3.4. This value may be used only when Samples per Pixel (0028,0002) has a value of 1. May be used for pixel data in a Native (uncompressed) or Encapsulated (compressed) format; see Section 8.2 in PS3.5.
PALETTE COLOR	Pixel data describe a color image with a single sample per pixel (single image plane). The pixel value is used as an index into each of the Red, Blue, and Green Palette Color Lookup Tables (0028,1101-1103&1201-1203). This value may be used only when Samples per Pixel (0028,0002) has a value of 1. May be used for pixel data in a Native (uncompressed) or Encapsulated (compressed) format; see Section 8.2 in PS3.5. When the Photometric Interpretation is Palette Color; Red, Blue, and Green Palette Color Lookup Tables shall be present.
RGB	Pixel data represent a color image described by red, green, and blue image planes. The minimum sample value for each color plane represents minimum intensity of the color. This value may be used only when Samples per Pixel (0028,0002) has a value of 3. Planar Configuration (0028,0006) may be 0 or 1. May be used for pixel data in a Native (uncompressed) or Encapsulated (compressed) format; see Section 8.2 in PS3.5.

Rescale Intercept & Rescale Slope



Attribute Name	Tag	Type	Attribute Description
Rescale Intercept	(0028,1052)	1C	The value b in relationship between stored values (SV) and the output units specified in Rescale Type (0028,1054). Output units = m*SV + b. Required if Modality LUT Sequence (0028,3000) is not present. Shall not be present otherwise.
Rescale Slope	(0028,1053)	1C	m in the equation specified by Rescale Intercept (0028,1052). Required if Rescale Intercept is present.
Rescale Type	(0028,1054)	1C	Specifies the output units of Rescale Slope (0028,1053) and Rescale Intercept (0028,1052). See Section C.11.1.1.2 for further explanation. Required if Rescale Intercept is present.

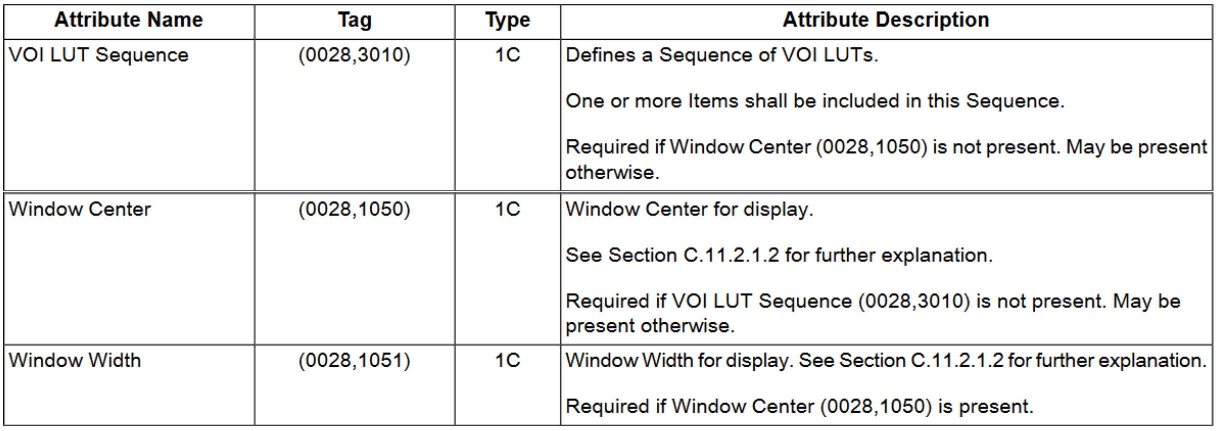
Table 13 – Excerpt of Modality LUT Macro Attributes [12, p.1293 – p.1294]

For Rescale Intercept, following statement has to be considered:
"If Image Type (0008,0008) Value 1 is ORIGINAL and Value 3 is not LOCALIZER, and Multi-energy CT Acquisition (0018,9361) is either absent or NO, output units shall be Hounsfield Units (HU)" [22].

Furthermore, following statement has to be considered for Rescale Type:
"The rescale slope and rescale intercept allow to transform the pixel values to HU or other units, as specified in the tag 0028,1054. For CT images, the unit should be HU (Hounsfield) and the default value is indeed HU when the tag 0028,1054 is not present. However, the tag may be present and may specify a different unit (OD=optical density, US=unspecified). The rescale slope and intercept are determined by the manufacturer of the hardware.
If the transformation from original pixel values to Hounsfield or Optical density is not linear, then a LUT is applied" [23].

Furthermore, if RescaleIntercept (0028,1052) and RescaleSlope (0028,1053) are not present in the image meta data information, the values are usually true Hounsfield Units. For the effect of Rescale, see chapter 4.7.

VOI LUT Module

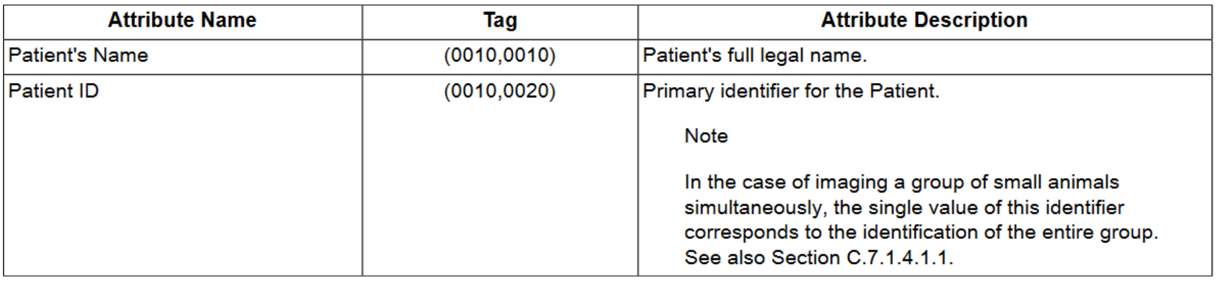


Attribute Name	Tag	Type	Attribute Description
VOI LUT Sequence	(0028,3010)	1C	Defines a Sequence of VOI LUTs. One or more Items shall be included in this Sequence. Required if Window Center (0028,1050) is not present. May be present otherwise.
Window Center	(0028,1050)	1C	Window Center for display. See Section C.11.2.1.2 for further explanation. Required if VOI LUT Sequence (0028,3010) is not present. May be present otherwise.
Window Width	(0028,1051)	1C	Window Width for display. See Section C.11.2.1.2 for further explanation. Required if Window Center (0028,1050) is present.

Table 14 - Excerpt of VOI LUT Macro Attributes [12, p.1296]

The effects of these attributes are illustrated in chapter 4.7.

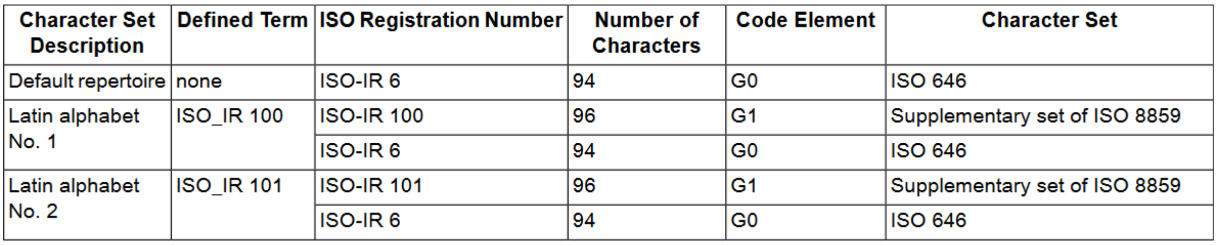
Patient Name & ID



Attribute Name	Tag	Attribute Description
Patient's Name	(0010,0010)	Patient's full legal name.
Patient ID	(0010,0020)	Primary identifier for the Patient. Note In the case of imaging a group of small animals simultaneously, the single value of this identifier corresponds to the identification of the entire group. See also Section C.7.1.4.1.1.

Table 15 - Excerpt of Performed Procedure Step Relationship Module Attribute [12, p.464]

Specific Character Set

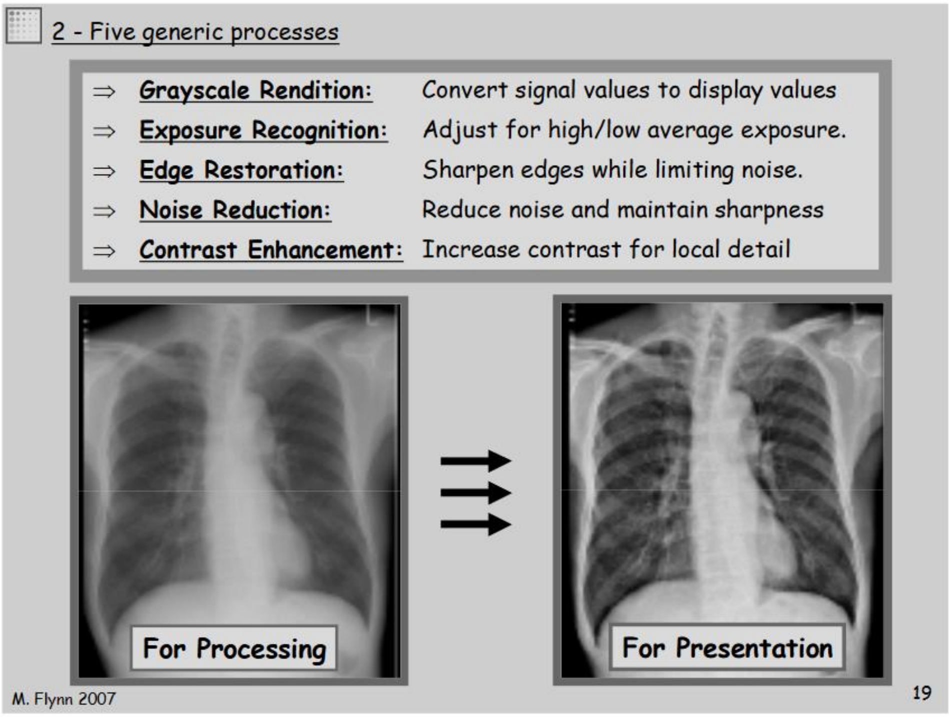


Character Set Description	Defined Term	ISO Registration Number	Number of Characters	Code Element	Character Set
Default repertoire	none	ISO-IR 6	94	G0	ISO 646
Latin alphabet No. 1	ISO_IR 100	ISO-IR 100	96	G1	Supplementary set of ISO 8859
		ISO-IR 6	94	G0	ISO 646
Latin alphabet No. 2	ISO_IR 101	ISO-IR 101	96	G1	Supplementary set of ISO 8859
		ISO-IR 6	94	G0	ISO 646

Table 16 - Excerpt of available Character Sets in DICOM [12, p.1376]

4.7 Effects of Meta Data Attributes on Image Display

The raw DICOM image data can be processed in various ways for the needs of the user. Five generic processes are listed in Fig. 45.



2 - Five generic processes

⇒ **Grayscale Rendition:**	Convert signal values to display values
⇒ **Exposure Recognition:**	Adjust for high/low average exposure.
⇒ **Edge Restoration:**	Sharpen edges while limiting noise.
⇒ **Noise Reduction:**	Reduce noise and maintain sharpness
⇒ **Contrast Enhancement:**	Increase contrast for local detail

For Processing

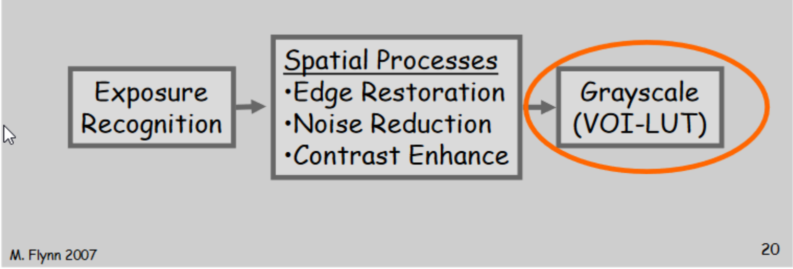
For Presentation

M. Flynn 2007

19

Figure 45 - Generic Image Processes [24]

The most relevant process is “Grayscale Rendition” which converts the signal values to display values. This process is already illustrated in Figure 44 and mentioned in chapter 4.6. Figure 46 shows the process in another form:



Exposure Recognition

Spatial Processes
•Edge Restoration
•Noise Reduction
•Contrast Enhance

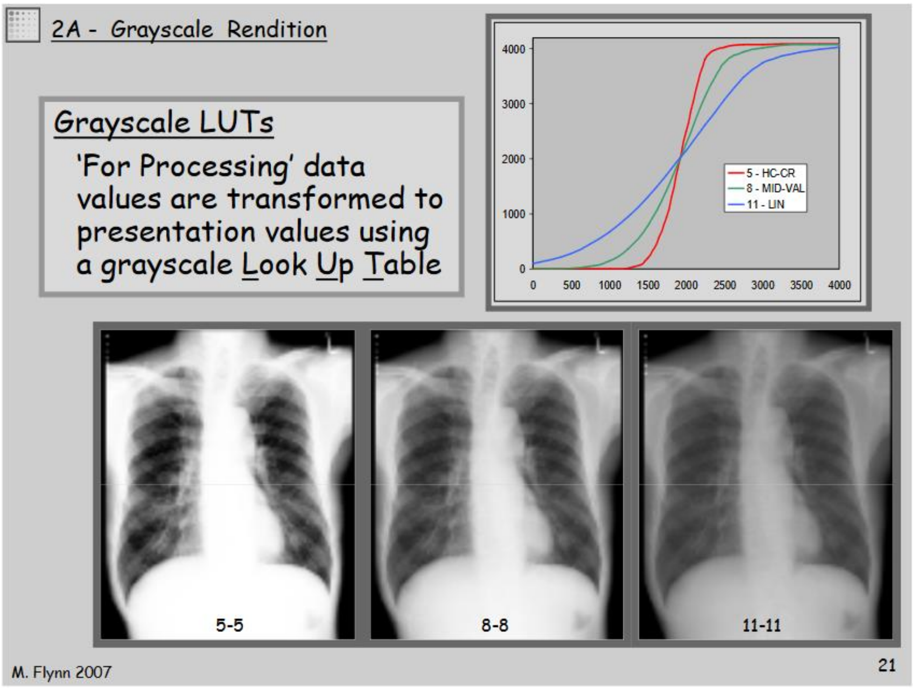
Grayscale (VOI-LUT)

M. Flynn 2007

20

Figure 46 – Grayscale Rendition with a Grayscale VOI LUT [24]

Figure 47 illustrates the effect on the displayed image on the bottom. For these images, the grayscale LUT has been adapted in way that the pixel above a certain luminescence value become white and below a certain value become black. Figure 48 shows again the process, similar to figures 42 and 43.


2A - Grayscale Rendition

Grayscale LUTs

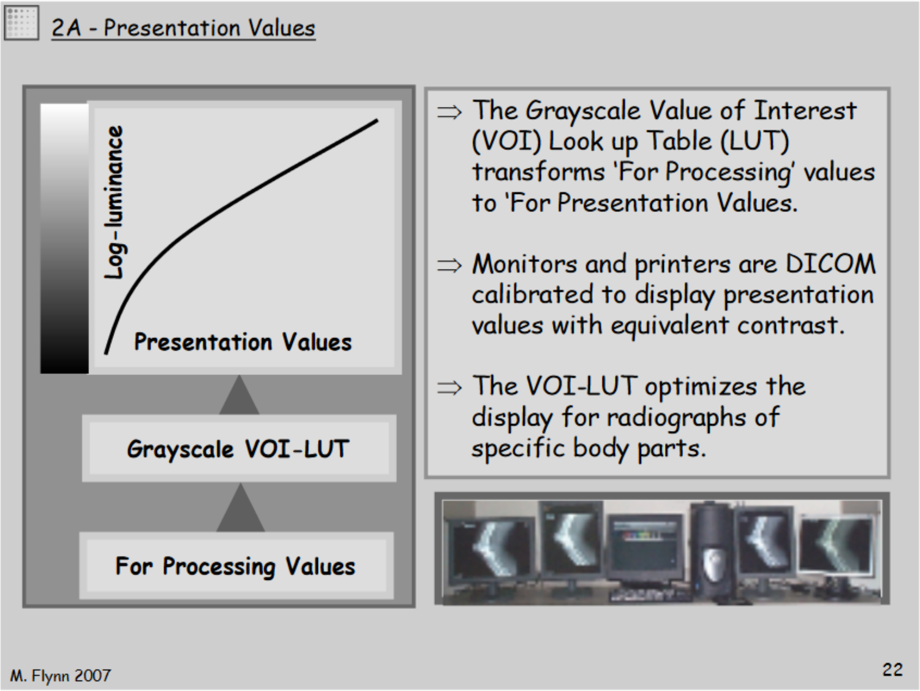
'For Processing' data values are transformed to presentation values using a grayscale Look Up Table

4000
3000
2000
1000
0
0
500
1000
1500
2000
2500
3000
3500
4000
5 - HC-CR
8 - MID-VAL
11 - LIN
5-5
8-8
11-11

M. Flynn 2007

21

Figure 47 - Change of Image Display wit LUTs [24]


2A - Presentation Values

Log-luminance
Presentation Values
Grayscale VOI-LUT
For Processing Values

⇒ The Grayscale Value of Interest (VOI) Look up Table (LUT) transforms 'For Processing' values to 'For Presentation Values.

⇒ Monitors and printers are DICOM calibrated to display presentation values with equivalent contrast.

⇒ The VOI-LUT optimizes the display for radiographs of specific body parts.

M. Flynn 2007

22

Figure 48 - Process of creating Presentation Values [24]

Digital Driving Level:

A further point to mention is the Digital Driving Level (DDL), illustrated in Fig. 42 and Fig. 48. For calibrated displays, the following circumstance applies:

"Every liquid crystal element has a slightly different response curve that must be measured and mapped to the DICOM curve. (For example, to achieve a certain Digital Drive Level (DDL) state for pixel # 2,999,999, the voltage is x, and for pixel #3,000,000 the voltage is y.) That response map is put into a "Look-Up Table" (LUT) within the display. Essentially, the LUT becomes a memory map that says, for every pixel or for the average response over the whole panel, we know which voltage will give us a certain amount of output that can be mapped to the DICOM curve" [25, p. 5]. This is true for calibrated monitors used in medical applications, how far this is applicable to non-calibrated display devices such as standard computers could not been dissolved. This is definitely a factor for the manual part of the segmentation process (see chapter 5.3).

5 Reading DICOM Data Sets

Various programs are able to read DICOM data sets, which all have a specific purpose. The most commonly used program is 3DSlicer, an open sourced software developed for the segmentation of the body with the use of DICOM files. Another widely used program is MATLAB, which enables further processing of the input data but is not used for segmentation. Besides those two applications, other capable programs are mentioned, but the list is not complete.

5.1 MATLAB

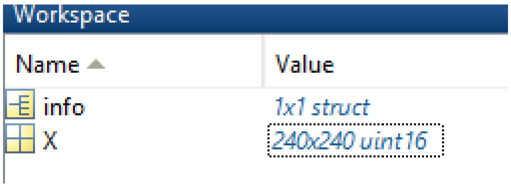
MATLAB is a powerful tool which is able to read the DICOM data into its workspace. This data then can be further processed and used for the generation of output data.

5.1.1 Reading File Information

MATLAB provides two main read commands, one for the grayscale values and one for the general file information [26]. The file has to be accessed via its filename, here "IMG00000". Of course an automatized process can be developed if the script searches for a file within a folder structure with the file extension ".dcm" (see Appendix A).

```
X = dicomread('IMG00000');       % Reads file (grayscale values)
info = dicominfo('IMG00000');    % Reads info of file
```

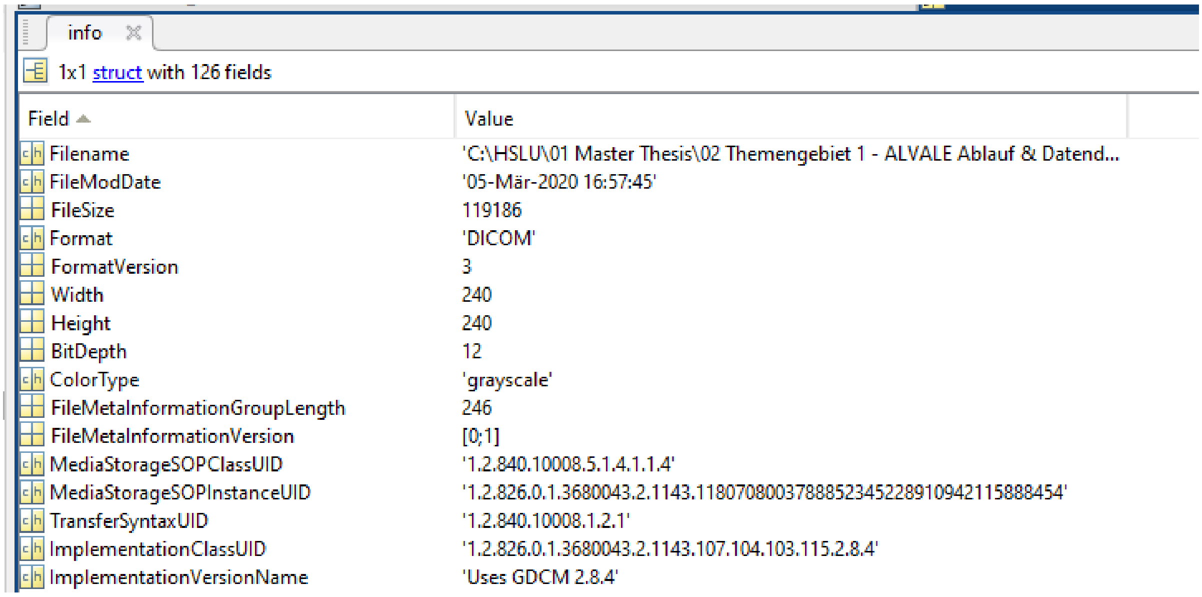
The results are illustrated below. The meta data information is stored in a structure array and various fields (Fig. 49). The grayscale image information is stored in a matrix with the pixel size of the image, so here the image has 240x240 pixels, the matrix has as well the size 240x240.



Name	Value
info	1x1 struct
X	240x240 uint16

Figure 49 - Workspace container [own illustration]

Values such as names, patient ID etc. are stored in the info structure array and can be accessed for further processing. This is relevant for the data consistency up to the final file format for 4D printing.



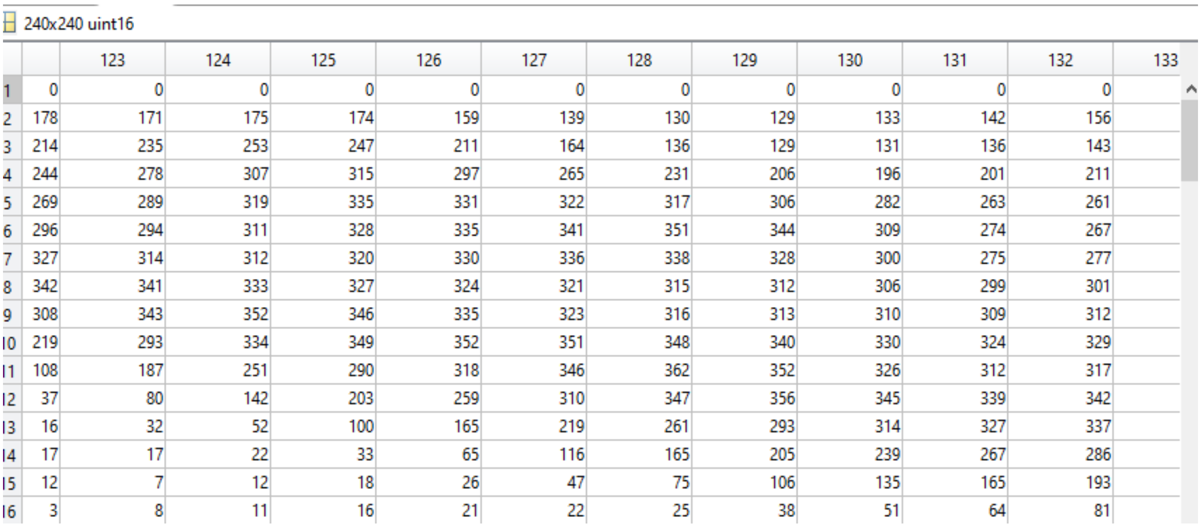
info

1x1 struct with 126 fields

Field	Value
Filename	'C:\HSLU\01 Master Thesis\02 Themengebiet 1 - ALVALE Ablauf & Datend...
FileModDate	'05-Mär-2020 16:57:45'
FileSize	119186
Format	'DICOM'
FormatVersion	3
Width	240
Height	240
BitDepth	12
ColorType	'grayscale'
FileMetaInformationGroupLength	246
FileMetaInformationVersion	[0;1]
MediaStorageSOPClassUID	'1.2.840.10008.5.1.4.1.1.4'
MediaStorageSOPInstanceUID	'1.2.826.0.1.3680043.2.1143.118070800378885234522891094211588845 4'
TransferSyntaxUID	'1.2.840.10008.1.2.1'
ImplementationClassUID	'1.2.826.0.1.3680043.2.1143.107.104.103.115.2.8.4'
ImplementationVersionName	'Uses GDCM 2.8.4'

Figure 50 - File meta data information [own illustration]

The pixel grayscale values are based on the Hounsfield Units for grayscale display. "Air absorbs X-rays almost not at all and by definition has a CT number of -1000 HU. When calibrating a CT, the attenuation of air is set to µ = 0 for simplicity. This is not correct, but in practice it would be very difficult to create a vacuum for detector calibration. The error that results from working with µ Air = 0 is always present, but it is very small and completely irrelevant for the image quality" [27]. This explains the values set as zero which is displayed as plain black on the images.



240x240 uint16

		123	124	125	126	127	128	129	130	131	132	133
1	0	0	0	0	0	0	0	0	0	0	0	
2	178	171	175	174	159	139	130	129	133	142	156	
3	214	235	253	247	211	164	136	129	131	136	143	
4	244	278	307	315	297	265	231	206	196	201	211	
5	269	289	319	335	331	322	317	306	282	263	261	
6	296	294	311	328	335	341	351	344	309	274	267	
7	327	314	312	320	330	336	338	328	300	275	277	
8	342	341	333	327	324	321	315	312	306	299	301	
9	308	343	352	346	335	323	316	313	310	309	312	
10	219	293	334	349	352	351	348	340	330	324	329	
11	108	187	251	290	318	346	362	352	326	312	317	
12	37	80	142	203	259	310	347	356	345	339	342	
13	16	32	52	100	165	219	261	293	314	327	337	
14	17	17	22	33	65	116	165	205	239	267	286	
15	12	7	12	18	26	47	75	106	135	165	193	
16	3	8	11	16	21	22	25	38	51	64	81	

Figure 51 - Section of the example 240x240 matrix [own illustration]

5.1.2 Display of the File Information

The display of the DICOM image can be done with following code:

```
imshow(X, []);                    % Display of image
```

The image is then displayed by MATLAB:

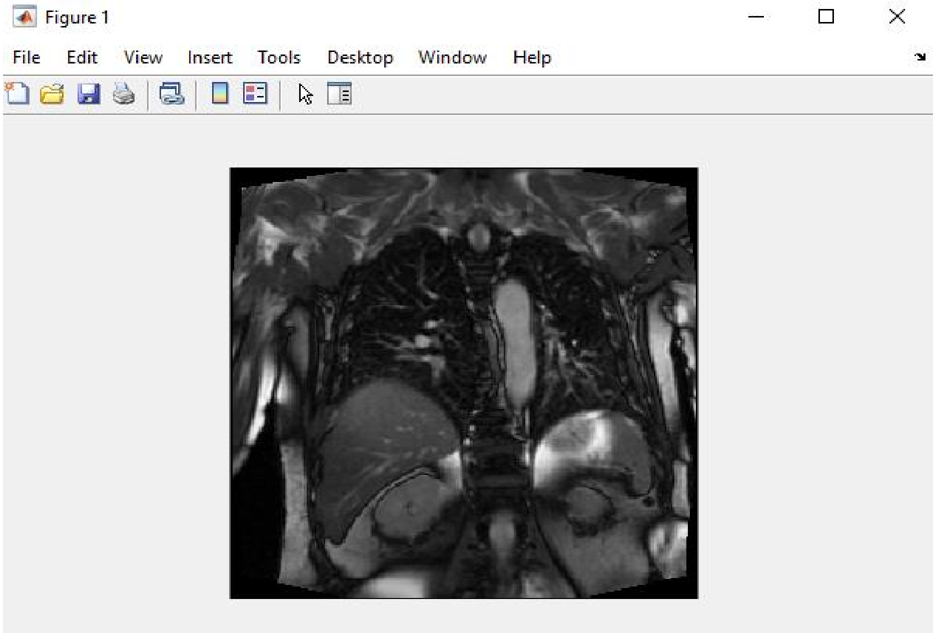


Figure 1
File Edit View Insert Tools Desktop Window Help

Figure 52 - DICOM image display with MATLAB [own illustration]

As MATLAB has stored all the grayscale values in the workspace, it is a rather simple task to only display specific values (see Appendix A). This procedure may be used in image recognition for the automatization of the segmentation process as every organ has a specific Hounsfield Unit value range (see chapter 2.3). Note the axis labeling as the image is stretched horizontally.

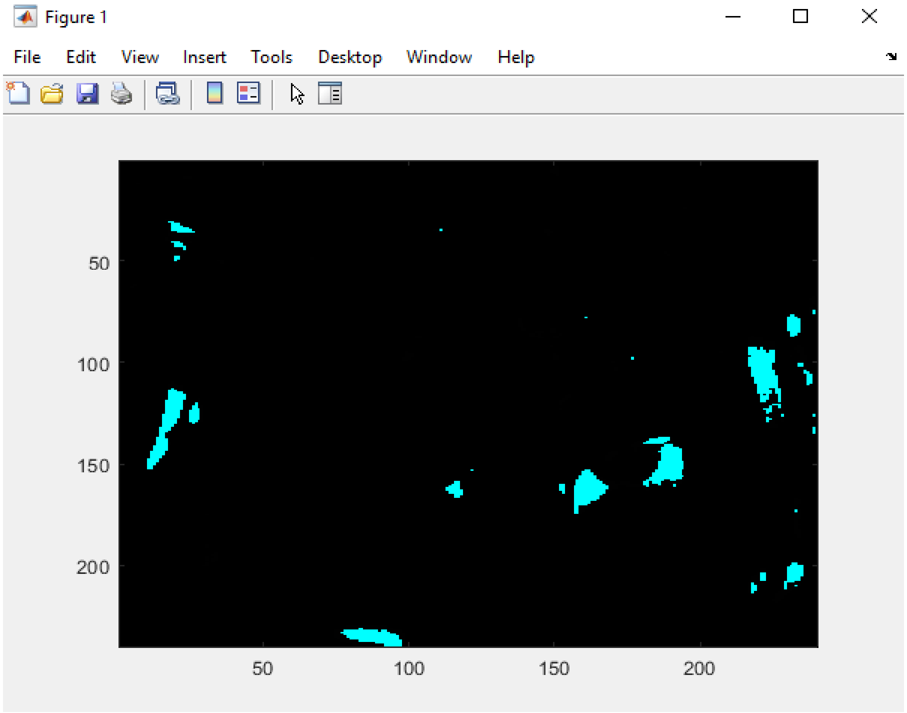


Figure 1
File Edit View Insert Tools Desktop Window Help
50
100
150
200
50
100
150
200

Figure 53 - Highlighting of Hounsfield Unit values between 300 & 500 [own illustration]

The meta data information is easily accessible in the workspace (see Appendix A), it can either be only read or used for further purposes such as implementation in the 4D printing models for data consistency.

5.2 3DSlicer

3DSlicer is a common freeware tool to read DICOM data sets and generate volumetric 3D geometries which represent the segmented body into its organs, bones etc.

5.2.1 Segmentation

This chapter shall give an overview on the segmentation as a whole, independent of 3DSlicer.

“Segmentation is a subfield of digital image processing and machine vision. The creation of content-related regions by combining adjacent pixels or voxels according to a certain homogeneity criterion is called segmentation.
In the process of machine vision, segmentation is usually the first step of image analysis and comes after image preprocessing. The process is thus:

Scene → Image acquisition → Image preprocessing → Segmentation → Feature extraction → Classification → Statement” [29]

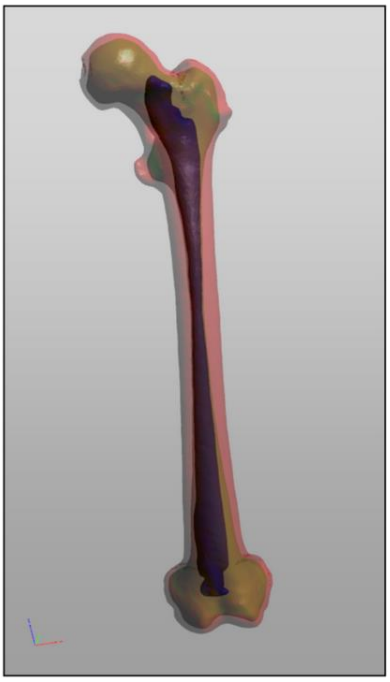


Figure 54 - Segmented and classified thighbone [28]

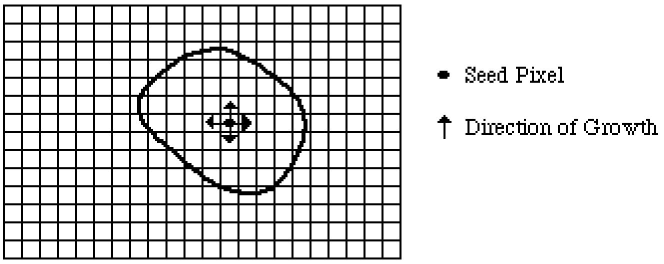
Overall, five different procedures are can be distinguished for automatic segmentation:

- Pixel-oriented
- Edge-oriented
- Region-oriented
- Model-based
- Texture-oriented

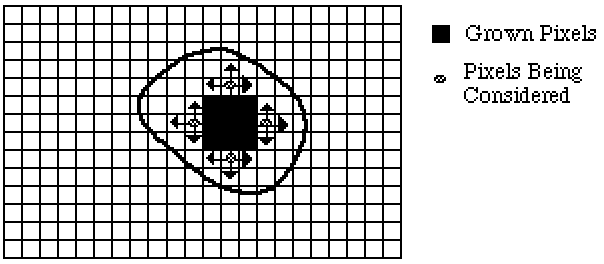
Segmentation software used to read DICOM images usually uses the region-oriented approach where it considers point sets as a whole and thus tries to find connected objects. The user marks a point (or pixel) on the image and the software connects all adjacent points with very similar grey values.

Regionally oriented methods
“The region-oriented methods consider point sets as a whole and thus try to find connected objects. Methods such as region growing, region splitting, pyramid linking and split and merge are frequently used” [29]. 3DSlicer uses the method region growing. “In this process, homogeneous image elements are merged into regions. First, the image is divided into initial cells (1×1, 2×2 or 4×4 pixels). Beginning with the selection of an initial cell as the initial region, this is then compared with the neighboring cells using a criterion (e.g. the difference in gray value between the gray value and that of the neighboring cell). If the criterion applies, the neighboring cell is added to the region. This is repeated recursively, which means that the neighboring cells of the newly added cells are also examined. If no more neighbors can be added, a region is found.
You can repeat the process for other cells that do not belong to the region until all pixels have been assigned to regions. However, this may lead to overlapping regions, depending on the selected criterion. The process is susceptible to "leakage", i.e. that regions that are actually separated by small "pixel bridges" are detected as one region” [30].

Region growing process is visualized below:


Seed Pixel
Direction of Growth

(a) Start of Growing a Region

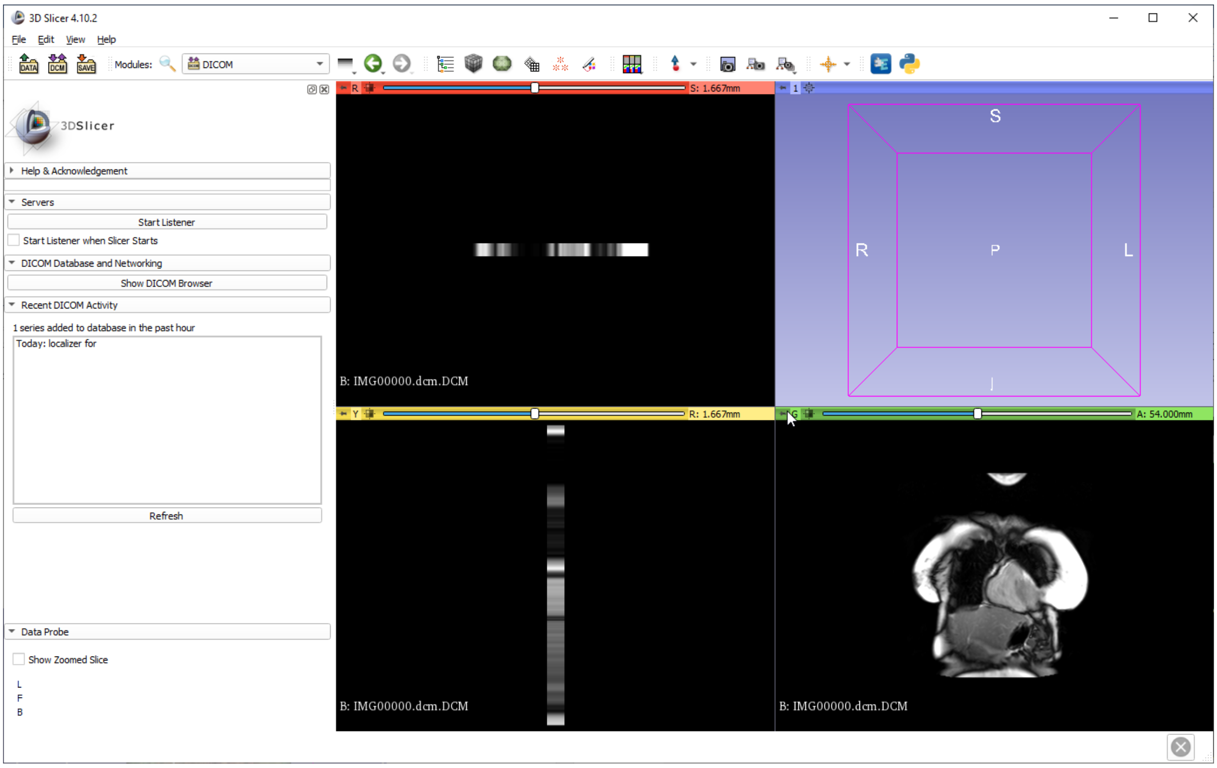

Grown Pixels
Pixels Being Considered

(b) Growing Process After a Few Iterations

Figure 55 - Region growing process based on pixels [31]

5.2.2 Segmentation in 3DSlicer

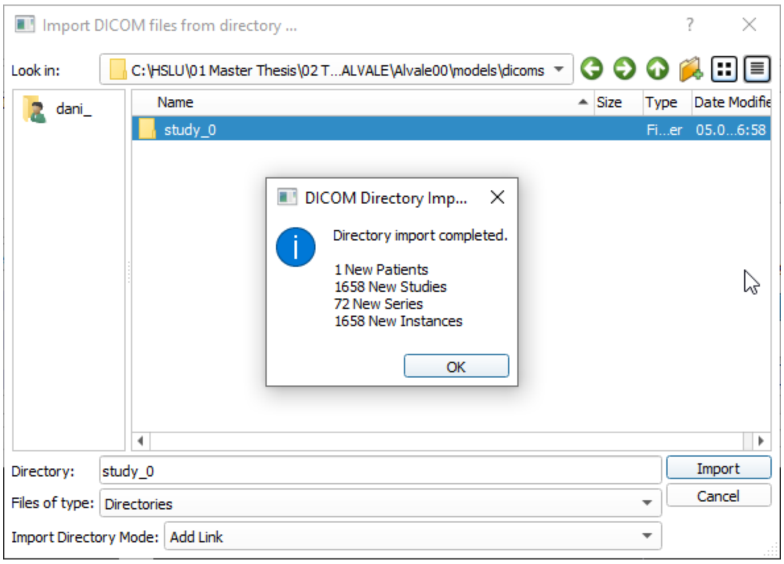
DICOM images can be imported into 3DSlicer via drag and drop (see Fig. 56). If all images from a directory shall be used, they can be loaded as a whole (see Fig. 57).


3D Slicer 4.10.2
File
Edit
View
Help
DATA
DCM
SAVE
Modules:
DICOM
3DSlicer
Help & Acknowledgement
Servers
Start Listener
Start Listener when Slicer Starts
DICOM Database and Networking
Show DICOM Browser
Recent DICOM Activity
Refresh
Data Probe
Show Zoomed Slice
L
F
B
R
S: 1.667mm
B: IMG00000.dcm.DCM
1
S
R
P
L
Y
R: 1.667mm
G
A: 54.000mm

1 series added to database in the past hour

Today: localizer for

Figure 56 - Drag and drop of one single DICOM image [own illustration]



Import DICOM files from directory ...
Look in:
C:\HSLU\01 Master Thesis\02 T...ALVALE\Alvale00\models\dicoms
dani_
Name
Size
Type
Date Modifie
study_0
Fi...er
05.0...6:58

DICOM Directory Imp...

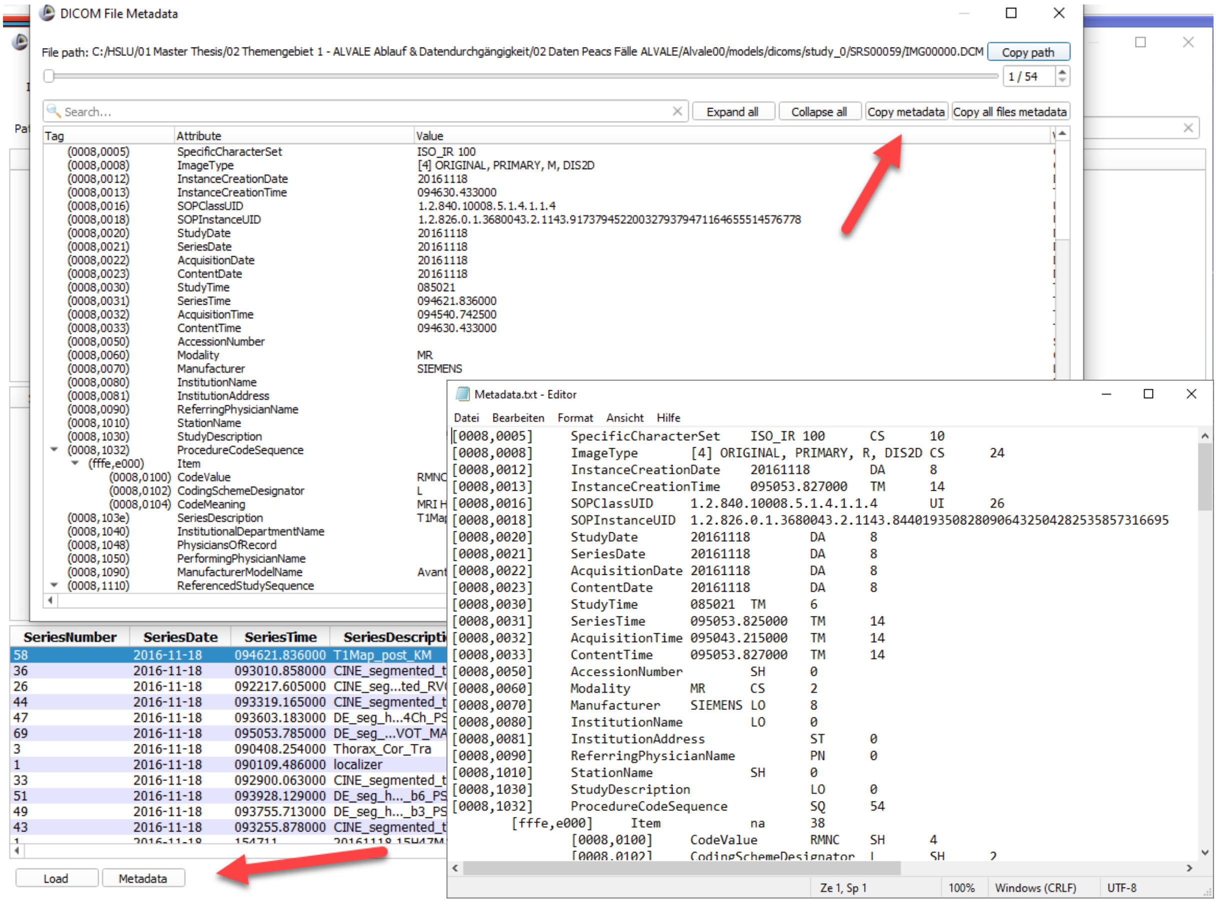
Directory import completed.

1 New Patients
1658 New Studies
72 New Series
1658 New Instances

Directory: study_0
Files of type: Directories
Import Directory Mode: Add Link

Figure 57 - Import of all DICOM images from a directory [own illustration]

These imported image sets then can be displayed as a whole to combine the image set to a complete body MRI view of the available images. All the imported image sets can be accessed anytime in the DICOM Browser of 3DSlicer. Furthermore, the meta data of the image can be accessed too. This can as well be copied and pasted into any text file or word format (see Fig. 58).



File path: C:/HSLU/01 Master Thesis/02 Themengebiet 1 - ALVALE Ablauf & Datendurchgängigkeit/02 Daten Peacs Fälle ALVALE/Alvale00/models/dicoms/study_0/SRS00059/IMG00000.DCM

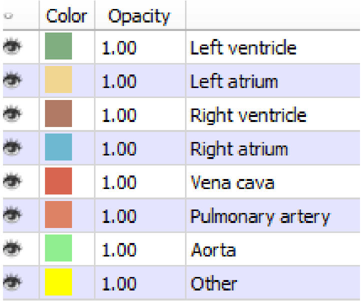
Tag	Attribute	Value
(0008,0005)	SpecificCharacterSet	ISO_IR 100
(0008,0008)	ImageType	[4] ORIGINAL, PRIMARY, M, DIS2D
(0008,0012)	InstanceCreationDate	20161118
(0008,0013)	InstanceCreationTime	094630.433000
(0008,0016)	SOPClassUID	1.2.840.10008.5.1.4.1.1.4
(0008,0018)	SOPInstanceUID	1.2.826.0.1.3680043.2.1143.9173794522003279379471164655514576778
(0008,0020)	StudyDate	20161118
(0008,0021)	SeriesDate	20161118
(0008,0022)	AcquisitionDate	20161118
(0008,0023)	ContentDate	20161118
(0008,0030)	StudyTime	085021
(0008,0031)	SeriesTime	094621.836000
(0008,0032)	AcquisitionTime	094540.742500
(0008,0033)	ContentTime	094630.433000
(0008,0050)	AccessionNumber	
(0008,0060)	Modality	MR
(0008,0070)	Manufacturer	SIEMENS
(0008,0080)	InstitutionName	
(0008,0081)	InstitutionAddress	
(0008,0090)	ReferringPhysicianName	
(0008,1010)	StationName	
(0008,1030)	StudyDescription	
(0008,1032)	ProcedureCodeSequence	
(fffe,e000)	Item	
(0008,0100)	CodeValue	RMNC
(0008,0102)	CodingSchemeDesignator	L
(0008,0104)	CodeMeaning	MRI H
(0008,103e)	SeriesDescription	T1Ma
(0008,1040)	InstitutionalDepartmentName	
(0008,1048)	PhysiciansOfRecord	
(0008,1050)	PerformingPhysicianName	
(0008,1090)	ManufacturerModelName	Avant
(0008,1110)	ReferencedStudySequence	

SeriesNumber	SeriesDate	SeriesTime	SeriesDescripti
58	2016-11-18	094621.836000	T1Map_post_KM
36	2016-11-18	093010.858000	CINE_segmented_t
26	2016-11-18	092217.605000	CINE_seg...ted_RV
44	2016-11-18	093319.165000	CINE_segmented_t
47	2016-11-18	093603.183000	DE_seg_h...4Ch_PS
69	2016-11-18	095053.785000	DE_seg_...VOT_MA
3	2016-11-18	090408.254000	Thorax_Cor_Tra
1	2016-11-18	090109.486000	localizer
33	2016-11-18	092900.063000	CINE_segmented_t
51	2016-11-18	093928.129000	DE_seg_h..._b6_PS
49	2016-11-18	093755.713000	DE_seg_h..._b3_PS
43	2016-11-18	093255.878000	CINE_segmented_t

```
[0008,0005]     SpecificCharacterSet    ISO_IR 100      CS      10
[0008,0008]     ImageType       [4] ORIGINAL, PRIMARY, R, DIS2D CS      24
[0008,0012]     InstanceCreationDate    20161118        DA      8
[0008,0013]     InstanceCreationTime    095053.827000   TM      14
[0008,0016]     SOPClassUID     1.2.840.10008.5.1.4.1.1.4       UI      26
[0008,0018]     SOPInstanceUID  1.2.826.0.1.3680043.2.1143.8440193508280906432504282535857316695
[0008,0020]     StudyDate       20161118        DA      8
[0008,0021]     SeriesDate      20161118        DA      8
[0008,0022]     AcquisitionDate 20161118        DA      8
[0008,0023]     ContentDate     20161118        DA      8
[0008,0030]     StudyTime       085021  TM      6
[0008,0031]     SeriesTime      095053.825000   TM      14
[0008,0032]     AcquisitionTime 095043.215000   TM      14
[0008,0033]     ContentTime     095053.827000   TM      14
[0008,0050]     AccessionNumber         SH      0
[0008,0060]     Modality        MR      CS      2
[0008,0070]     Manufacturer    SIEMENS LO      8
[0008,0080]     InstitutionName         LO      0
[0008,0081]     InstitutionAddress              ST      0
[0008,0090]     ReferringPhysicianName          PN      0
[0008,1010]     StationName             SH      0
[0008,1030]     StudyDescription                LO      0
[0008,1032]     ProcedureCodeSequence           SQ      54
        [fffe,e000]     Item            na      38
                [0008,0100]     CodeValue       RMNC    SH      4
                [0008,0102]     CodingSchemeDesignator  L       SH      2
```

Figure 58 - Access of meta data; copy of data and manual paste to text file [own illustration]

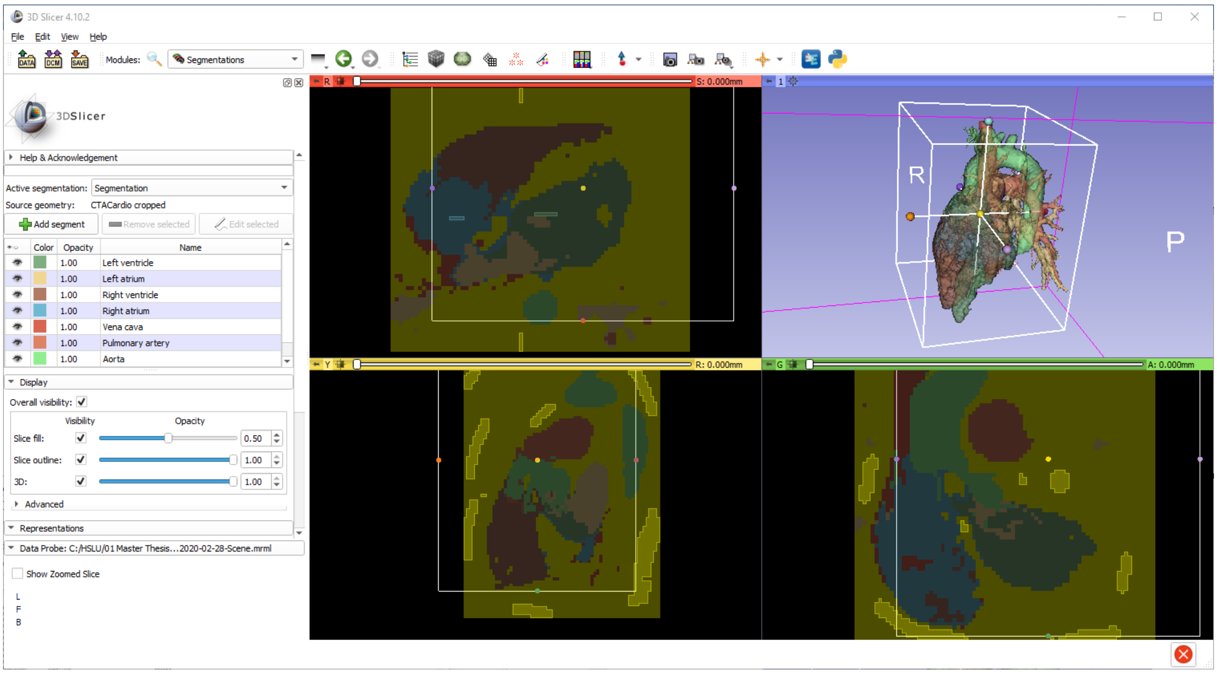
The segmentation itself starts usually with the definition of the segmented parts. All regions to be segmented can be named and a specific color can be assigned (see Fig. 59):



	Color	Opacity	
👁	🟩	1.00	Left ventricle
👁	🟨	1.00	Left atrium
👁	🟫	1.00	Right ventricle
👁	🟦	1.00	Right atrium
👁	🟥	1.00	Vena cava
👁	🟧	1.00	Pulmonary artery
👁	🟩	1.00	Aorta
👁	🟨	1.00	Other

Figure 59 - Definition of segmented objects and assigning of colors [own illustration]

After this, the to be segmented regions are then marked by the user slice by slice in all three directions, and the software assists through the region growing process (see chapter 5.2.1) to form a closed and holistic area (2D). The software further assists in the linking of the separate areas of each slice to create a 3D model of the segmentation. The below figure shows the result of an initial own segmentation:


3D Slicer 4.10.2
File
Edit
View
Help
DATA
DCM
SAVE
3DSlicer
Help & Acknowledgement
Add segment
Remove selected
Edit selected
Display
Advanced
Representations
Show Zoomed Slice
R
P
S: 0.000mm
R: 0.000mm
A: 0.000mm
L
F
B

Modules: Segmentations

Active segmentation: Segmentation

Source geometry: CTACardio cropped

	Color	Opacity	Name
👁	🟩	1.00	Left ventricle
👁	🟨	1.00	Left atrium
👁	🟫	1.00	Right ventricle
👁	🟦	1.00	Right atrium
👁	🟥	1.00	Vena cava
👁	🟧	1.00	Pulmonary artery
👁	🟩	1.00	Aorta

Overall visibility: ☑

	Visibility	Opacity
Slice fill:	☑	0.50
Slice outline:	☑	1.00
3D:	☑	1.00

Data Probe: C:/HSLU/01 Master Thesis…2020-02-28-Scene.mrml

Figure 60 - Own segmentation of heart model (standard file set provided by 3DSlicer) [own illustration]

It is visible that the detail and accuracy of the segmentation is dependent on the resolution of the images as it colors the image pixel by pixel. The resulting 3D model can be post-processed to smooth the surfaces to create an enhanced and more refined object. Figure 61 shows a better and more detailed segmentation of the same data set from a segmentation tutorial:

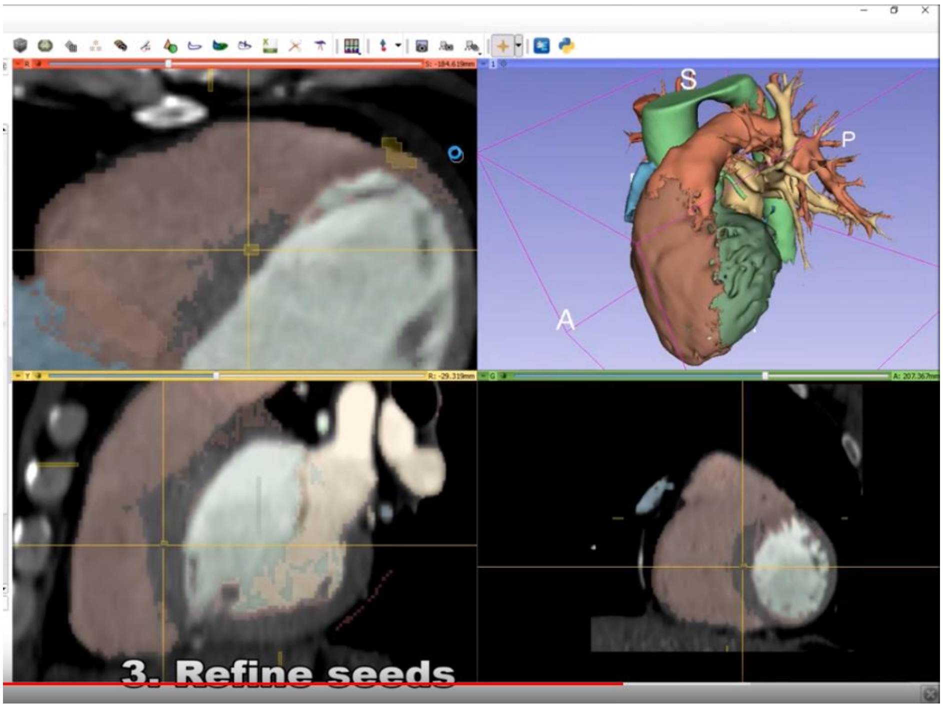


Figure 61 - Tutorial result of segmentation of heart [32]

5.2.3 Export of Segmentation

The newly created segments can be exported as STL or OBJ files. As an inherent property of the file formats, OBJ keeps the color code of the segmentation, STL only provides the geometry information. The self-segmented example of the heart could not be exported correctly, it is assumed there are unclosed or interfering geometries which leads to not being exportable. Simpler segmentations were able to be exported without any issue.

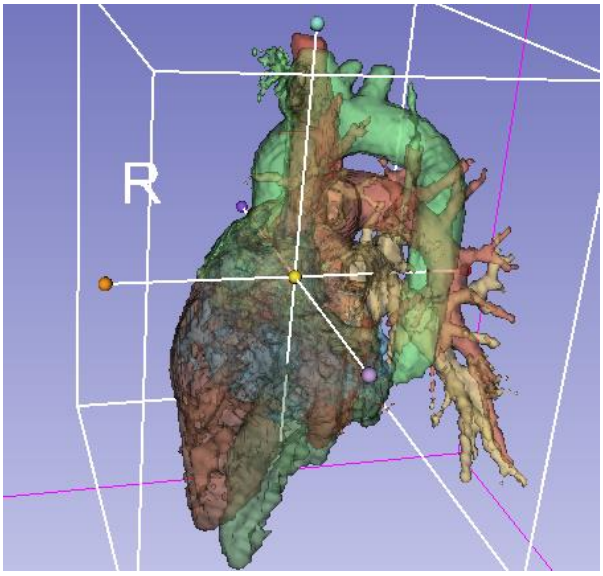
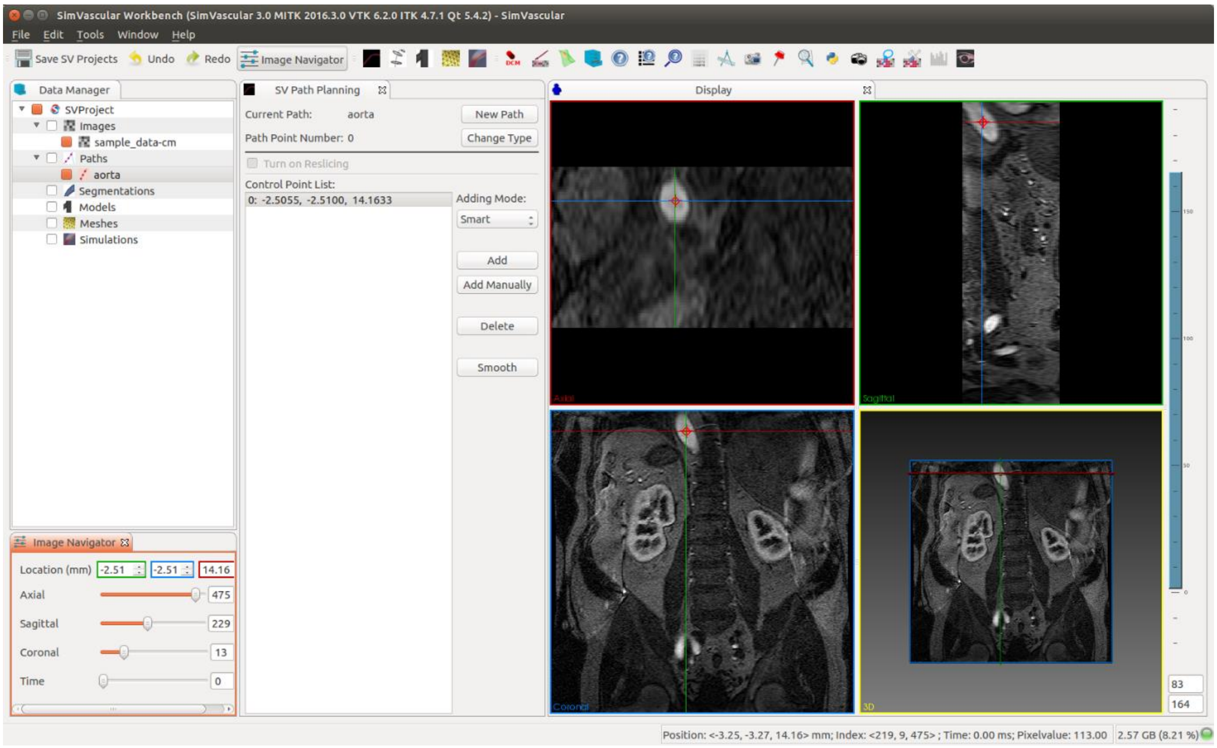


Figure 62 - Own segmentation of heart model (standard file set provided by 3DSlicer) [own illustration]

5.3 SimVascular

SimVascular [33] is an open source tool for the simulation of cardiovascular blood flow. The tool includes a segmentation software which can read DICOM images. If a DICOM image is selected to be opened, the program loads all the images from the same folder. This represents the simplest way. The advanced way is the use of the included "MITK DICOM Browser", which can scan the selected directory for DICOM images and provide further information.


SimVascular Workbench (SimVascular 3.0 MITK 2016.3.0 VTK 6.2.0 ITK 4.7.1 Qt 5.4.2) - SimVascular
File
Edit
Tools
Window
Help
Save SV Projects
Undo
Redo
Image Navigator
Data Manager
SVProject
Images
sample_data-cm
Paths
aorta
Segmentations
Models
Meshes
Simulations
SV Path Planning
New Path
Change Type
Turn on Reslicing
Add
Add Manually
Delete
Smooth
Display
Axial
Sagittal
Coronal
3D

Current Path: aorta

Path Point Number: 0

Control Point List:

0: -2.5055, -2.5100, 14.1633

Adding Mode: Smart

Location (mm) -2.51 -2.51 14.16

Axial 475

Sagittal 229

Coronal 13

Time 0

83

164

Position: <-3.25, -3.27, 14.16> mm; Index: <219, 9, 475> ; Time: 0.00 ms; Pixelvalue: 113.00 2.57 GB (8.21 %)

Figure 63 - Segmentation viewer in SimVascular [34]

This tool can only read the images and use it for the segmentation process, but it is not possible to export them in another format or use it for other purposes. The segmented model can be exported as a STL file, the simulation results (colored) can be exported as VTP or VTU file which are very specific file formats only used by very few applications (e.g. ParaView, see chapter 7.1).

5.4 ezDICOM

"ezDICOM is a medical viewer for MRI, CT and ultrasound images. It can read images from Analyze, DICOM, GE Genesis, Interfile, Siemens Magnetom, Siemens Somatom and NEMA formats. It also includes tools for converting medical images from proprietary format" [35]. This viewer is freeware and does not require any license. The DICOM images can be imported by drag and drop and are directly displayed (see Fig. 64).

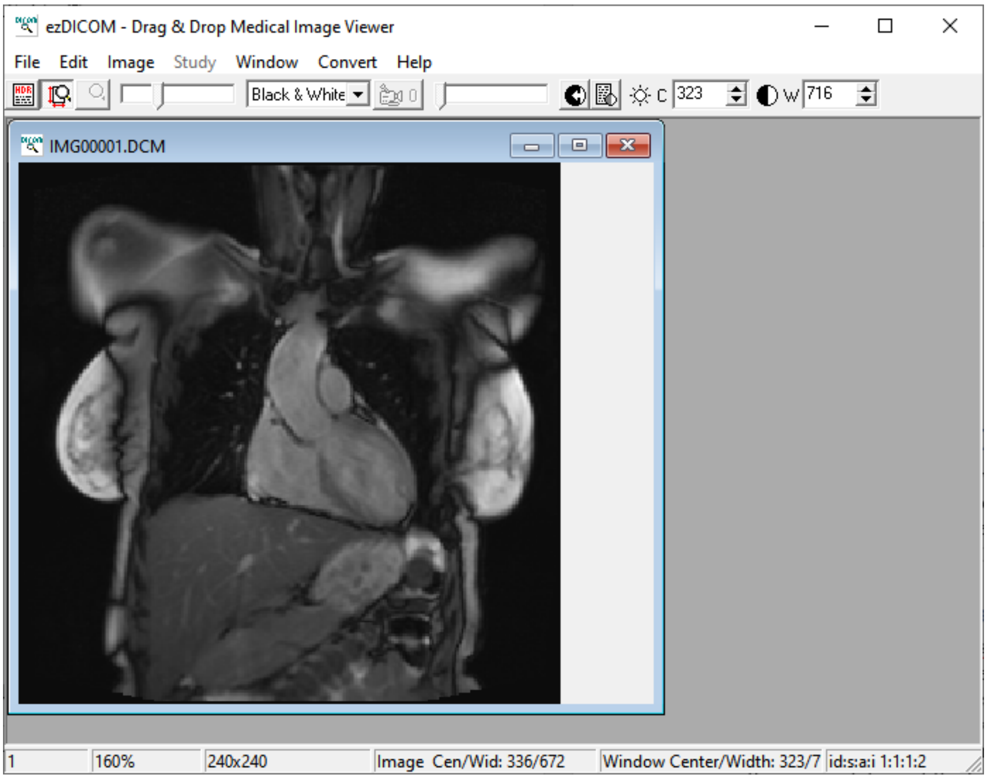
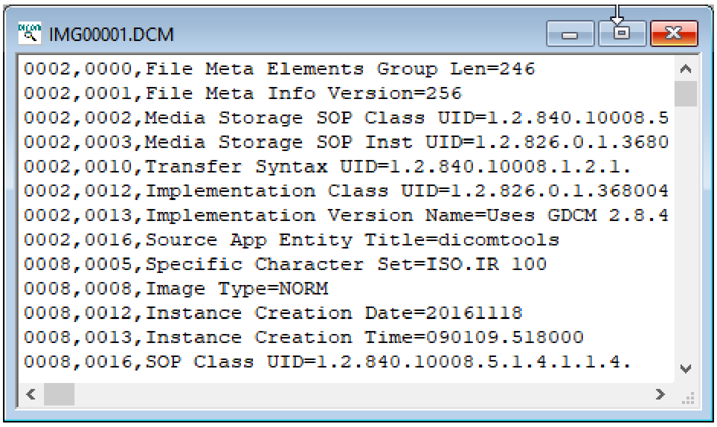

ezDICOM - Drag & Drop Medical Image Viewer
File
Edit
Image
Study
Window
Convert
Help
Black & White
C 323
W 716
IMG00001.DCM
1
160%
240x240
Image Cen/Wid: 336/672
Window Center/Width: 323/7
id:s:a:i 1:1:1:2

Figure 64 - Screenshot of ezDICOM with image imported [own illustration]

By pressing F3 or Image → View Image Information, the meta data information can be accessed (Fig. 65).


IMG00001.DCM

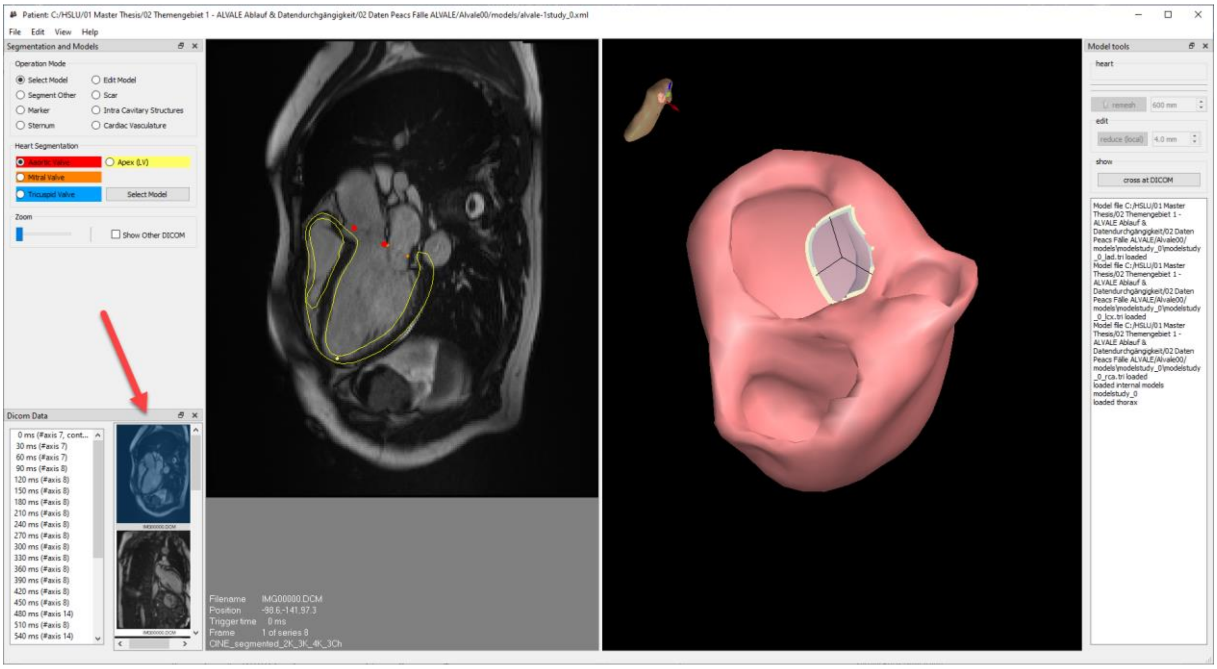
```
0002,0000,File Meta Elements Group Len=246
0002,0001,File Meta Info Version=256
0002,0002,Media Storage SOP Class UID=1.2.840.10008.5
0002,0003,Media Storage SOP Inst UID=1.2.826.0.1.3680
0002,0010,Transfer Syntax UID=1.2.840.10008.1.2.1.
0002,0012,Implementation Class UID=1.2.826.0.1.368004
0002,0013,Implementation Version Name=Uses GDCM 2.8.4
0002,0016,Source App Entity Title=dicomtools
0008,0005,Specific Character Set=ISO.IR 100
0008,0008,Image Type=NORM
0008,0012,Instance Creation Date=20161118
0008,0013,Instance Creation Time=090109.518000
0008,0016,SOP Class UID=1.2.840.10008.5.1.4.1.1.4.
```

Figure 65 - Image information display [own illustration]

The meta data cannot be exported with this program, but it can be copy-pasted after marking a complete line which pastes all lines of meta data of the image, no single line is possible to paste. The DICOM image itself can be saved as a JPG image if desired, but loses its meta data.

5.5 GeomPeacs

GeomPeacs is the proprietary segmentation tool from Peacs used in the Alvale project [36]. For more information about Alvale see chapter 6. The DICOM image is used to define the geometry (bottom left and center left), each layer of the MRI is then connected to a 3D model. Therefore, the general process is the very similar as with 3DSlicer.



Patient: C:/HSLU/01 Master Thesis/02 Themengebiet 1 - ALVALE Ablauf & Datendurchgängigkeit/02 Daten Peacs Fälle ALVALE/Alvale00/models/alvale-1study_0.xml

```
Model file C:/HSLU/01 Master
Thesis/02 Themengebiet 1 -
ALVALE Ablauf &
Datendurchgängigkeit/02 Daten
Peacs Fälle ALVALE/Alvale00/
models\modelstudy_0\modelstudy
_0_lad.tri loaded
Model file C:/HSLU/01 Master
Thesis/02 Themengebiet 1 -
ALVALE Ablauf &
Datendurchgängigkeit/02 Daten
Peacs Fälle ALVALE/Alvale00/
models\modelstudy_0\modelstudy
_0_lcx.tri loaded
Model file C:/HSLU/01 Master
Thesis/02 Themengebiet 1 -
ALVALE Ablauf &
Datendurchgängigkeit/02 Daten
Peacs Fälle ALVALE/Alvale00/
models\modelstudy_0\modelstudy
_0_rca.tri loaded
loaded internal models
modelstudy_0
loaded thorax
```

Filename IMG00000.DCM
Position -98.6,-141.97.3
Trigger time 0 ms
Frame 1 of series 8
CINE_segmented_2K_3K_4K_3Ch

File
Edit
View
Help
Segmentation and Models
Operation Mode
● Select Model
○ Edit Model
○ Segment Other
○ Scar
○ Marker
○ Intra Cavitary Structures
○ Sternum
○ Cardiac Vasculature
Heart Segmentation
Aortic Valve
Apex (LV)
Mitral Valve
Tricuspid Valve
Select Model
Zoom
☐ Show Other DICOM
Dicom Data
0 ms (#axis 7, cont...
30 ms (#axis 7)
60 ms (#axis 7)
90 ms (#axis 8)
120 ms (#axis 8)
150 ms (#axis 8)
180 ms (#axis 8)
210 ms (#axis 8)
240 ms (#axis 8)
270 ms (#axis 8)
300 ms (#axis 8)
330 ms (#axis 8)
360 ms (#axis 8)
390 ms (#axis 8)
420 ms (#axis 8)
450 ms (#axis 8)
480 ms (#axis 14)
510 ms (#axis 8)
540 ms (#axis 14)
IMG00000.DCM
IMG00000.DCM
Model tools
heart
remesh
600 mm
edit
reduce (local)
4.0 mm
show
cross at DICOM

Figure 66 - Definition of segmentation (left) and segmented 3D model (right) [own illustration]

The resulting model can be exported as an XML file or as a VTK file. The VTK file format is as well closely associated with ParaView [37]. For ParaView see chapter 7.1.
This tool needs a license from Peacs to be used and focuses on the segmentation of the heart.

6 Alvale Data Conversion Process

This chapter discusses the overall process of the Alvale project [38, 39, 40, 41] after the generation of the MRI data to the 4D printing of the color-coded heart mode. It includes the segmentation, the ECG process and data generation as well as the data conversion process of the output data from the segmentation and ECG to a 4D-printable model.

6.1 Geometry Generation by Peacs

The initial data sets for Peacs are usually MRI or CT images in the DICOM format. These images are imported to their own proprietary segmentation software "GeomPeacs" (see chapter 5.5) which works in a similar way as 3DSlicer. The user defines the area of the to be segmented organ slice by slice and the software supports the user by creating an assembled 3D model from the selected areas. The major difference of the program compared to 3DSlicer is the simplification of the user interface and its focus on the segmentation of the heart. Overall, this process is semi-automatic which requires knowledge of the user of the anatomy of the human body and the heart specifically, and its display on a DICOM image. For an experienced user, the generation of the 3D model requires roughly half an hour according to Peacs.

6.2 ECG Data Generation by Peacs

The basis of the ECG data generation is the research work with the VIVO ECGI (electrocardiographic imaging) system [38, 40]. It is a noninvasive method to locate PVC and VT (see chapter 6.5) and can replace invasive electroanatomic mapping. For the localization it uses a standard 12 lead ECG device and a 3D camera which scans the torso and registers the electrode location (see Fig. 67).

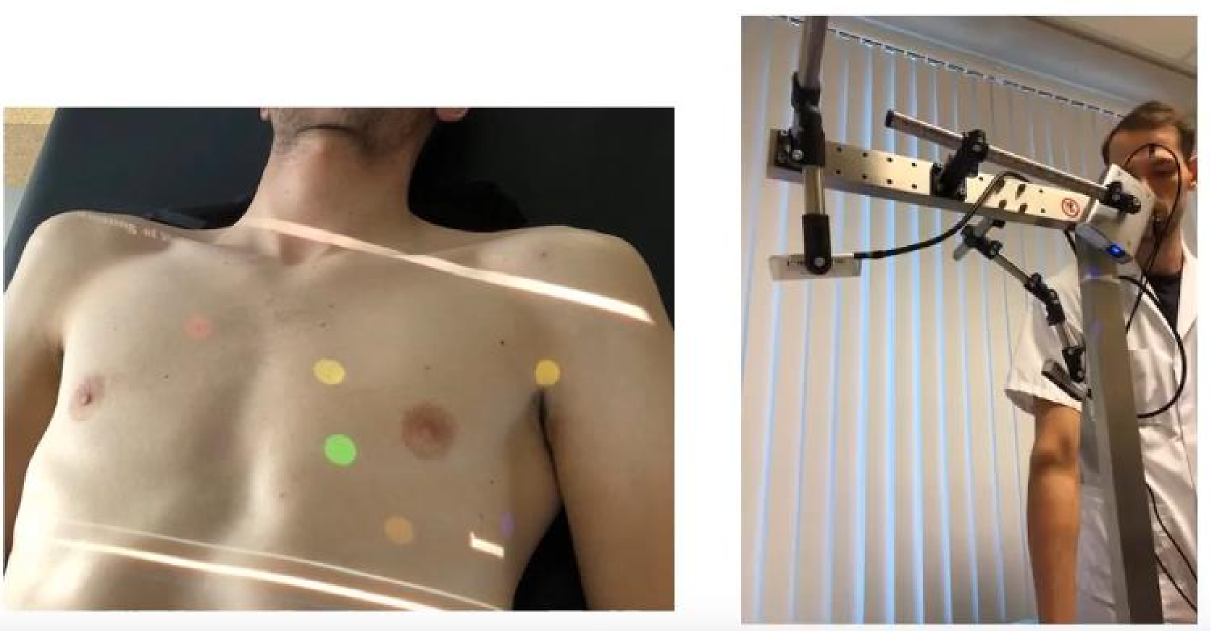
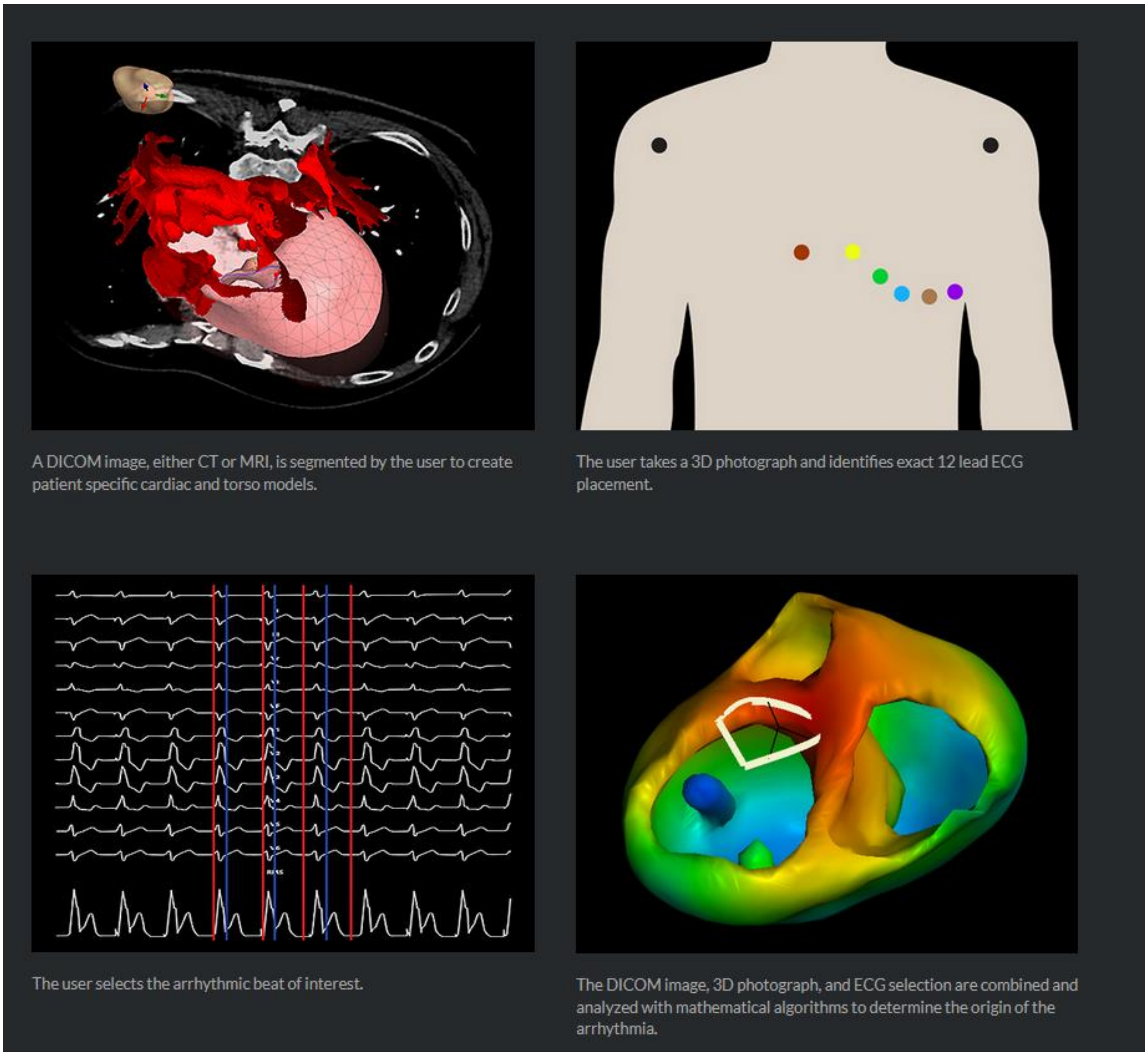


Figure 67 - Screenshot from ECG Excellence video illustrating use of torso [42]

The gained information of the ECG measurements can then be applied onto the segmented 3D model of the heart and the signals can be visualized with a color code (see Fig. 68).



A DICOM image, either CT or MRI, is segmented by the user to create patient specific cardiac and torso models.

The user takes a 3D photograph and identifies exact 12 lead ECG placement.

The user selects the arrhythmic beat of interest.

The DICOM image, 3D photograph, and ECG selection are combined and analyzed with mathematical algorithms to determine the origin of the arrhythmia.

Figure 68 - Process of VIVO ECG [43]

6.3 Data Set Conversion for 4D-printable Format

The basis of this chapter is the previous work carried out [1] to enable a conversion of Alvale data sets based on .tri and .dep to VRML data sets. The newly provided data sets have the file extension .xml and are the result files of Peacs to be used for ECGSim [44] and to display in MATLAB. The basis of the conversion are the MATLAB scripts provided by Peacs, and the VRML converter has been adapted to use the resulting workspace from the Peacs scripts. Furthermore, the colormap has been inverted in order to correct it as it has been wrong previously.

Mostly, the part with the reading of the .tri and .dep files could be replaced by simple read commands of the matrices provided into the workspace by the Peacs scripts.

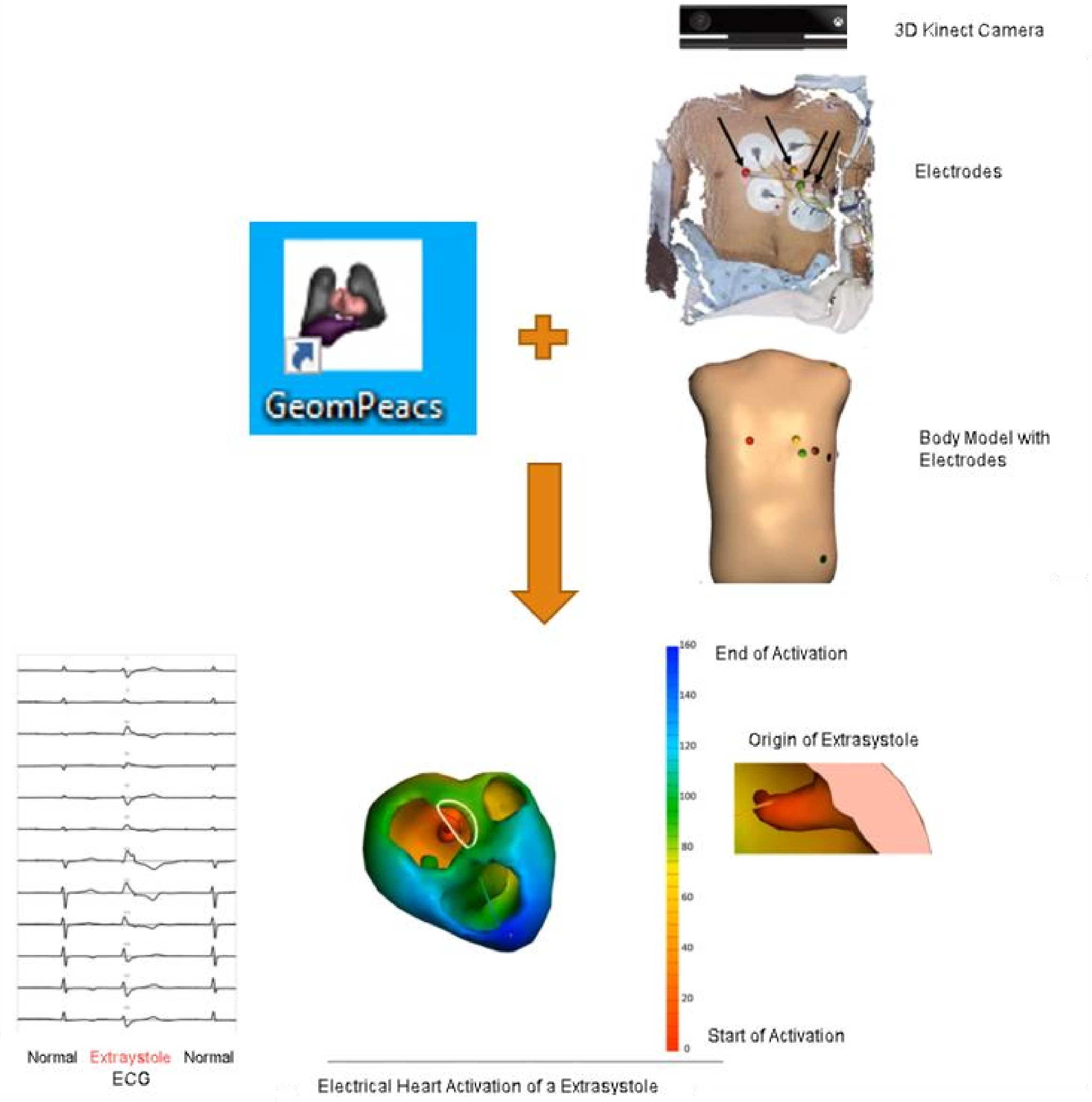

3D Kinect Camera
Electrodes
GeomPeacs
Body Model with Electrodes
End of Activation
Origin of Extrasystole
Start of Activation
160
140
120
100
80
60
40
20
0
Normal
Extraystole
Normal
ECG
Electrical Heart Activation of a Extrasystole

Figure 69 - Illustration of the data generation process without VRML conversion [own illustration, 40]

One data set (Alvale00) is the same as used at the previous project which can be used to validate the results visually regarding geometry and color application (see Fig. 70). Overall, eight cases of PVC were provided by Peacs, and all cases could be converted to VRML without any issues. Therefore, it can be stated that the conversion is successfully validated for data sets representing PVC cases.

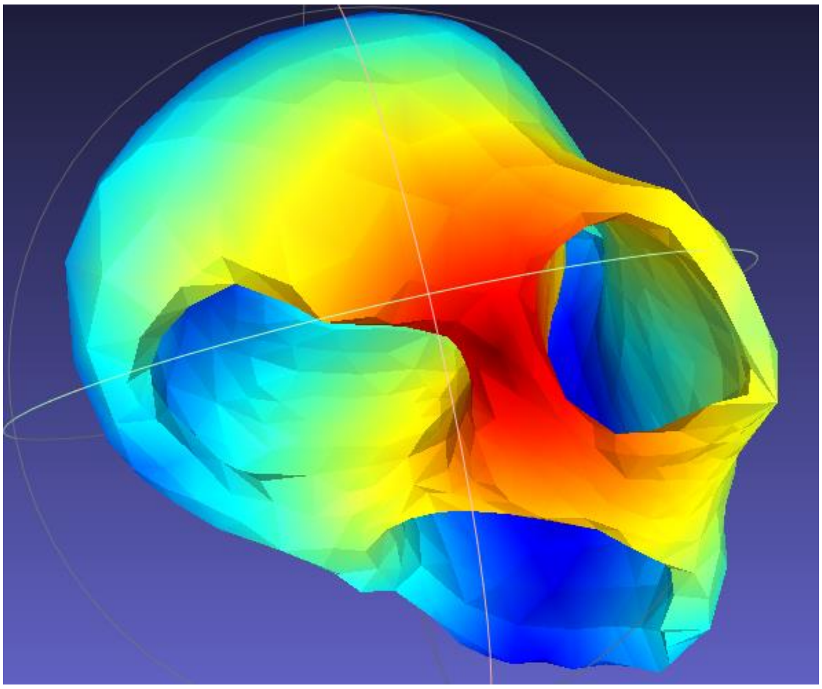


Figure 70 - Converted VRML model of Alvale00 with new converter [own illustration]

Furthermore, many other tri files are provided such as splitted (separate) parts of the heart, the lungs or the torso. As these geometries are generated by the image recognition of the 3D camera in order to locate the heart geometry and the ECG signals, they share the same coordinate system and therefore the position is displayed correctly (see Fig. 71 and Fig. 72). This circumstance would allow a merged VRML model of the data if desired, but manual adaptions would be necessary to be made such as support for the heart or a cutout in the torso to unveil the heart within the torso, and the definition of a wall thickness of the torso in the printing process to make it printed hollow.

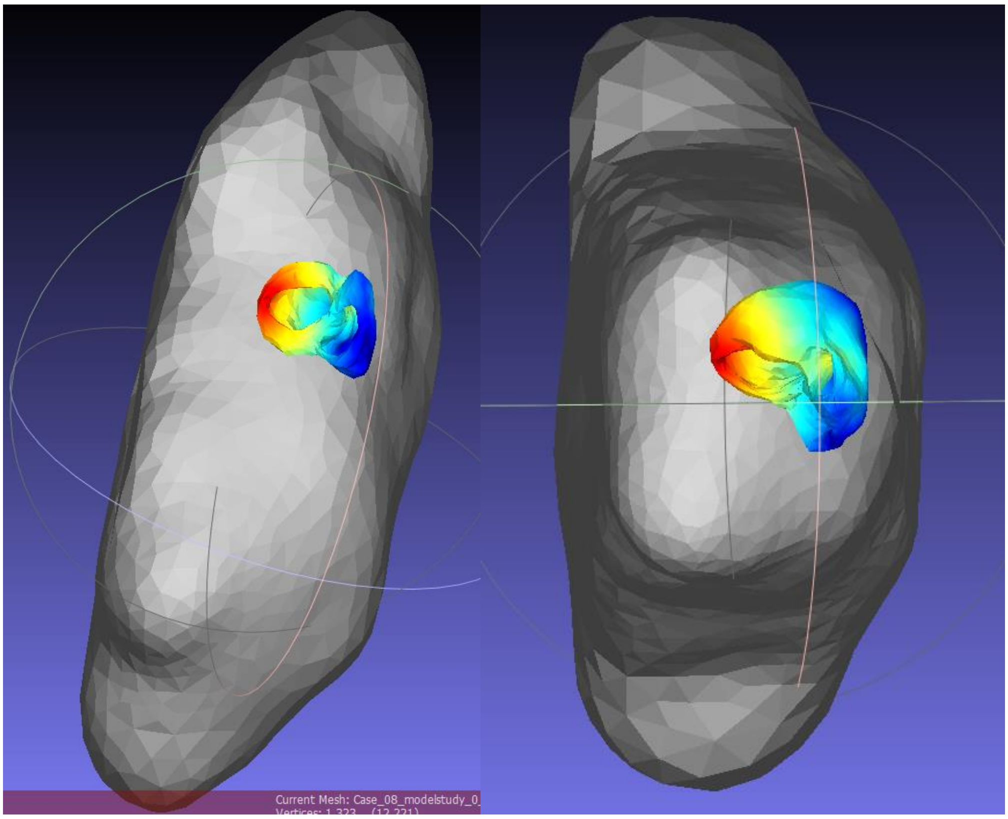

Current Mesh: Case_08_modelstudy_0
Vertices: 1 323 (12 221)

Figure 71 - Position of the heart in the torso of the patient from top and side, VRML models [own illustration]

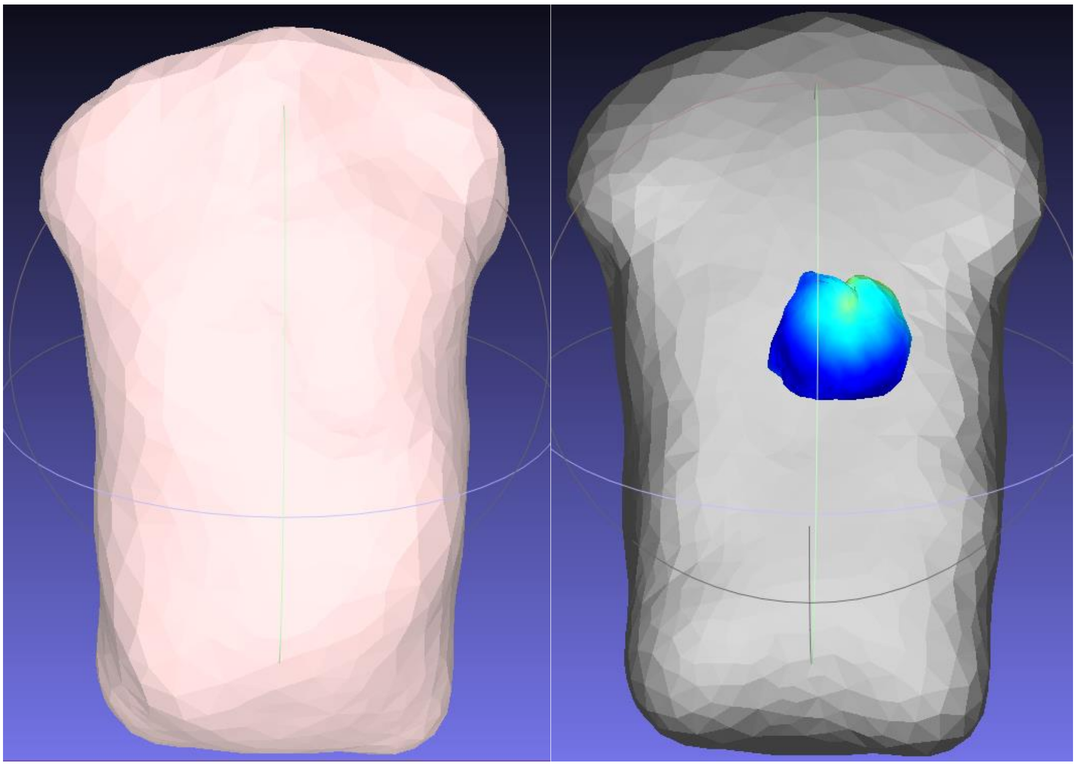


Figure 72 - Position of the heart in the torso of the patient from the front, VRML models [own illustration].

To the left: solid torso to the right: translucent torso

6.4 Implementation of Patient Information

The data continuity of the overall process is important to be able to allocate the final output data to the original file sets. Therefore, important identification meta data such as the patients' name or the patient ID need to be included in the VRML file in a way that it is accessible to any user handling this data. One important problem unsolved is the consideration of medical secrecy, as this information is not allowed to share freely without the consent of the patient. It may be necessary to use an anonymized random number for identification.

Overall, there were two ways identified how this information can be included:

Implementation into Filename

The implementation of the meta data into the filename would make the VRML file immediately recognizable to any user as the name is displayed directly in the storage folder. Furthermore, the user can search for any particular patient by searching its ID in the explorer.
The disadvantage of this process is the limitation of the length of the filename of e.g. Windows 10 of max. 256 characters [45], and the overall file path including the filename has the same limitation which reduces the possible character count of the filename. This may allow for the implementation for 8 – 12 separate meta data entries, but the readability would be very low and not user friendly. If this method is chosen, the number of meta data included would be around 1 -2 to remain viable for the user. An example of such a filename could be as following:

Alvale_PatientID#5902023_InstitutionalDepartmentName#Cedars-Sinai_Medical_Center

It is clearly visible that the contained information is very limited. Another problem of this method is the fact that the same patient may take more than one medical examination at different dates, therefore the implementation of following two data points is necessary as well for a clear identification:

- Instance Creation Date
- Instance Creation Time

If this information is included in the filename as well, it becomes increasingly difficult to read by the user and is therefore not recommended.

Implementation into the File Syntax

Another method would be the implementation of the meta data information into the file syntax. Compared to the filename implementation it allows for a virtually unlimited number of meta data points. The suggested way of this method would be the commenting out of the lines in the syntax and add the data after it. This would look as following:

```
#VRML V2.0 utf8
# Patient ID: 5902023
# Institutional Department Name: Cedars-Sinai Medical Center
# Instance Creation Date: 20161208
# Instance Creation Time: 160045.92700
```

This method is far more readable by the user. The relevant meta data is included after the first line of the VRML file and can be accessed by opening the VRML file with the Text Editor of Windows. The filename itself can be chosen and adjusted by the user independent of the meta data. The disadvantage of this method is the need for a dedicated program (or inclusion into the converter program) to look up and search for the meta data stored in the .wrl files as the explorer cannot do that.

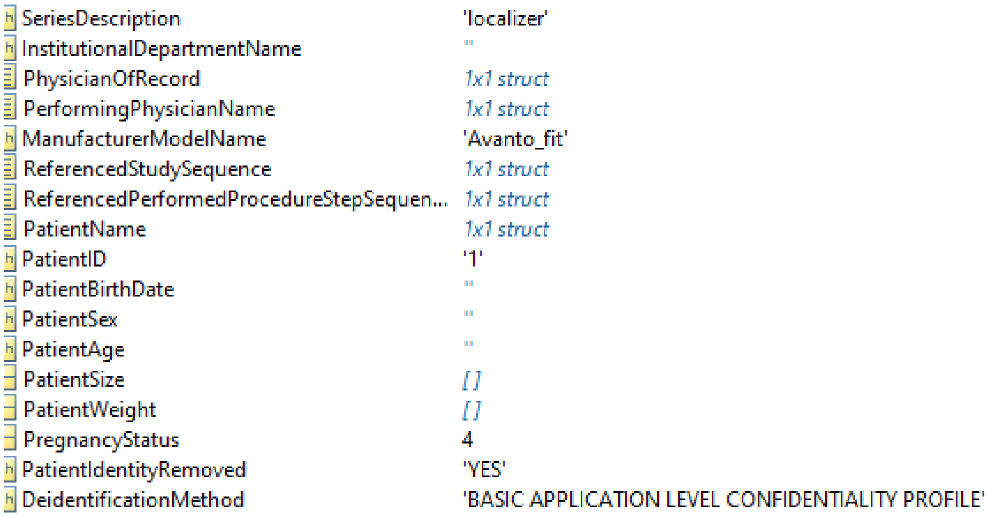
Meta Data to be Added to the Output File

As beforementioned, the use of the plain name of the patient would violate medical secrecy. Therefore, following minimal meta data is suggested to use:

- Instance Creation Date
- Instance Creation Time
- Institutional Department Name
- Physician of Record
- Performing Physician Name
- Patient ID

This information would be would be sufficient to assign the VRML file to the DICOM images.

As visible in the illustration below, the patient identity has been removed from the DICOM images (at the bottom), and the physician names are empty structures. Therefore, the suggested meta data information is not present anymore to be included to the VRML file except the Instance Creation Date. The physicians' and patients' names have to be defined by the MRI / CT user; therefore, it is important that this step is done to allow continuity of the data sets.

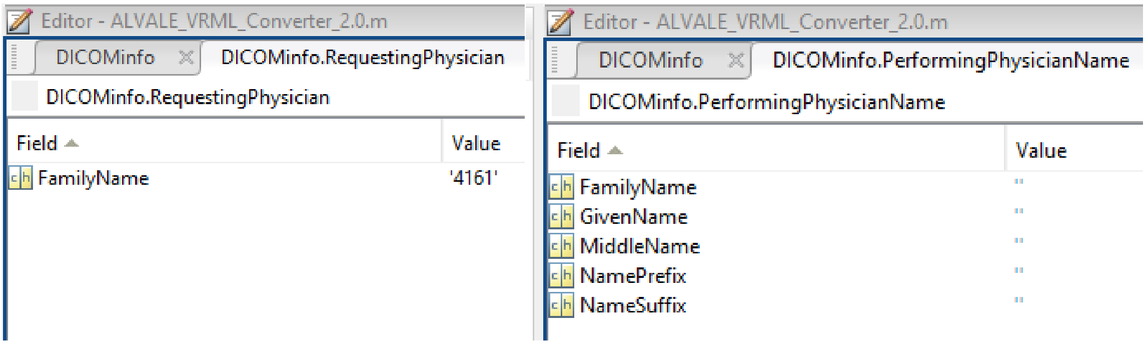


SeriesDescription	'localizer'
InstitutionalDepartmentName	''
PhysicianOfRecord	*1x1 struct*
PerformingPhysicianName	*1x1 struct*
ManufacturerModelName	'Avanto_fit'
ReferencedStudySequence	*1x1 struct*
ReferencedPerformedProcedureStepSequen...	*1x1 struct*
PatientName	*1x1 struct*
PatientID	'1'
PatientBirthDate	''
PatientSex	''
PatientAge	''
PatientSize	*[ ]*
PatientWeight	*[ ]*
PregnancyStatus	4
PatientIdentityRemoved	'YES'
DeidentificationMethod	'BASIC APPLICATION LEVEL CONFIDENTIALITY PROFILE'

Figure 73 - Excerpt of meta data (not full list) available from DICOM images [own illustration]

To show the feasibility of the process, two parameters available are chosen which share the same value structure within MATLAB (struct and character array) with the suggested meta data parameters:

- Patient ID (character array with value "1")
- Requesting Physician (struct with subfield character array with value "4161")



Editor - ALVALE_VRML_Converter_2.0.m
DICOMinfo
DICOMinfo.RequestingPhysician
Editor - ALVALE_VRML_Converter_2.0.m
DICOMinfo
DICOMinfo.PerformingPhysicianName

DICOMinfo.RequestingPhysician

Field	Value
FamilyName	'4161'

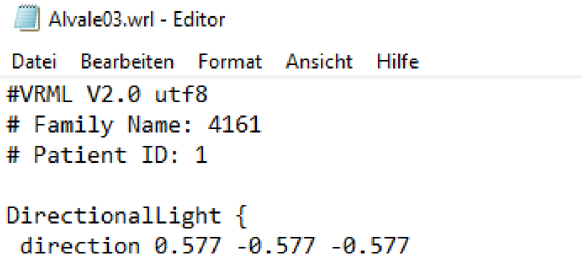
DICOMinfo.PerformingPhysicianName

Field	Value
FamilyName	''
GivenName	''
MiddleName	''
NamePrefix	''
NameSuffix	''

Figure 74 - Comparison of structures of meta data [own illustration]

This example shows as well that the physicians’ name can be anonymized too to ensure data protection for them as well.

The meta data access and inclusion to the VRML file with MATLAB can be done (see Appendix B) with following result:



```
#VRML V2.0 utf8
# Family Name: 4161
# Patient ID: 1

DirectionalLight {
 direction 0.577 -0.577 -0.577
```

Figure 75 - Reading the output file with meta data included [own illustration]

The meta data is commented out just after the header of the file which is necessary to be there to be identified as a VRML file. After the lines with the meta data the standard syntax proceeds. The reading and opening of the .wrl file with a dedicated reading program, a CAD program or import to a 3D printer is still possible without issues.

6.5 Conversion of other Alvale medical cases

At first, Peacs provided only very similar PVC (Premature Ventricular Contraction) cases, see chapter 6.3. Following further medical cases were provided:

- 1x symptomatic RVOT PVCs
- 2x recurrent ventricular arrhythmia as a consequence of dilated cardiomyopathy

“Right ventricular recurrent ventricular arrhythmia outflow tract (RVOT) tachycardia is a form of monomorphic VT originating from the outflow tract of the right ventricle or occasionally from the tricuspid annulus. It is usually seen in patients without underlying structural heart disease, although may also occur in the context of arrhythmogenic right ventricular dysplasia (ARVD)” [46]. Figure 76 shows an example of the ECG signal:

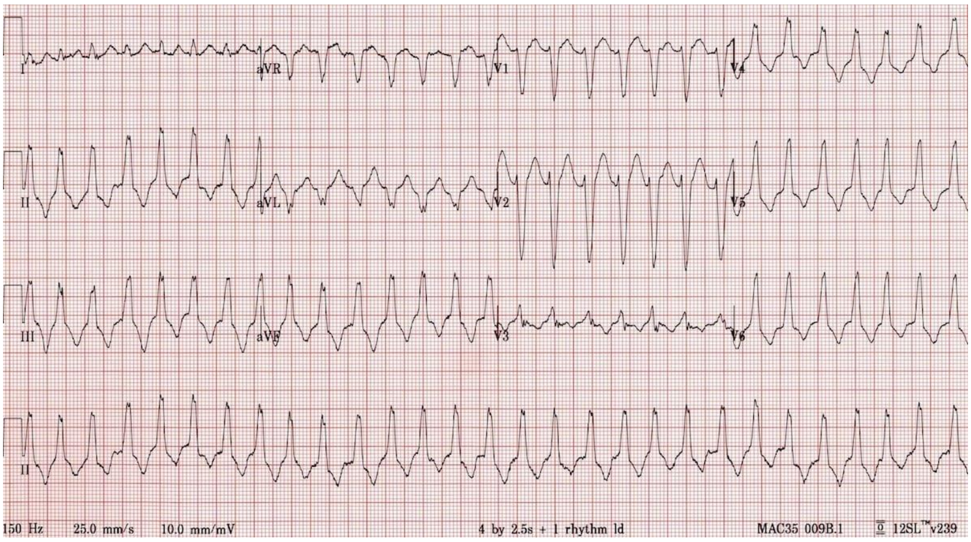

I
aVR
V1
V4
II
aVL
V2
V5
III
aVF
V3
V6
II
150 Hz
25.0 mm/s
10.0 mm/mV
4 by 2.5s + 1 rhythm ld
MAC35 009B.1
12SL™ v239

Figure 76 - ECG signals for RVOT tachycardia [46]

Figure 77 illustrates the ventricular tachycardia heartbeat compared to a normal heartbeat:

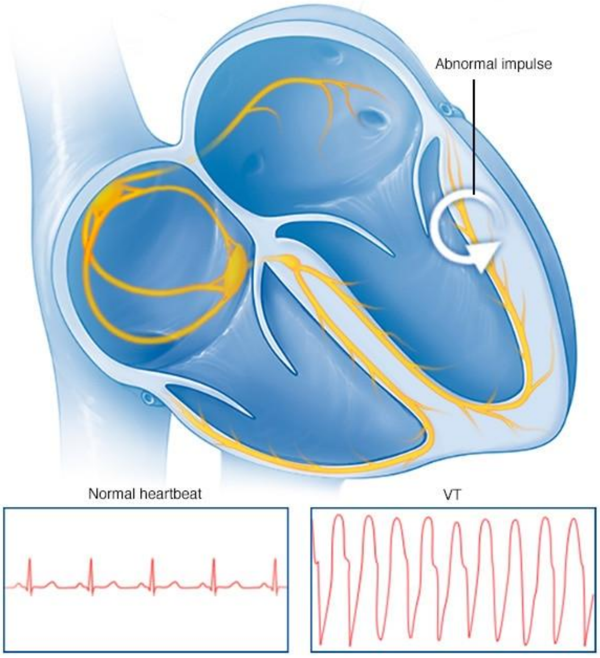

Abnormal impulse
Normal heartbeat
VT

Figure 77 - Ventricular tachycardia heartbeat [47]

Recurrent ventricular arrhythmia, as its name suggests, is the recurrence of the ventricular arrythmia. The according ECG signal is illustrated in Fig. 78.

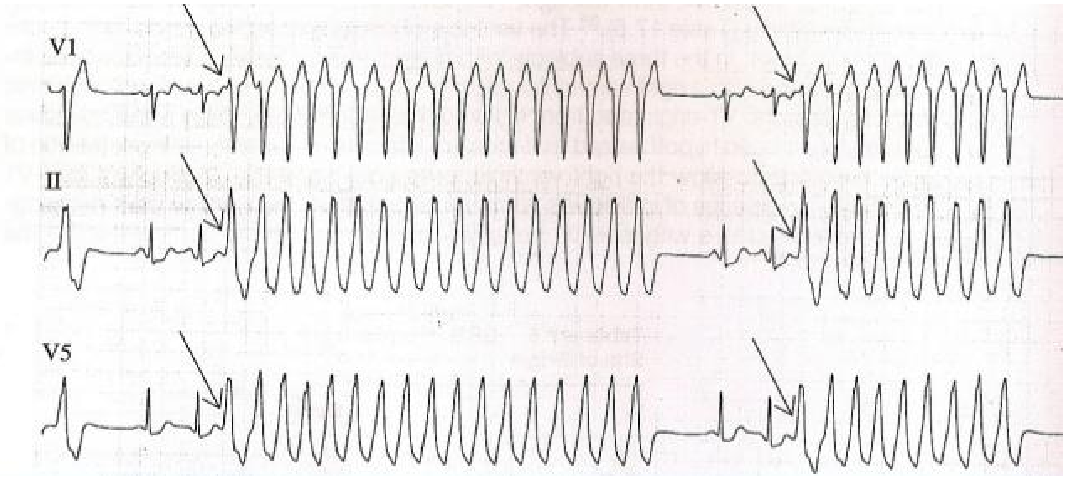


Figure 78 - ECG signal of recurrent ventricular arrhythmia [48]

The files provided were DEP and TRI files which are the same file formats as provided in the previous work [1]. The only difference was the fact that the DEP files were in the binary format instead of ASCII. This did not lead to issues at reading the files, two ways are possible: either the binary information can be translated with MATLAB standard functions or one of the provided scripts by Peacs (loadmat.m) is used which can read the timestamp information from a binary DEP file. One of the major differences to the previous files is the fact that not all of the geometry nodes have a timestamp value assigned which means no color-code can be applied. These empty spots were defined as colored in plain white as this color is not present in the colormap used and it creates a good contrast too (see Appendix D for changes of the MATLAB script). Figures 79 and 80 show the new cases:

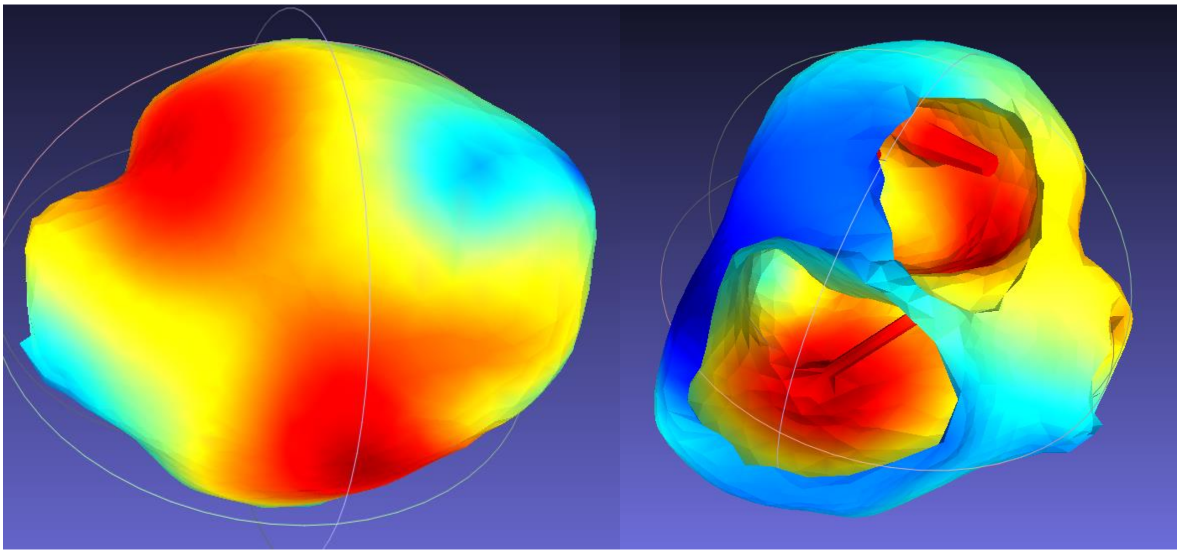


Figure 79 - Case of RVOT PVC, invasive [own illustration]

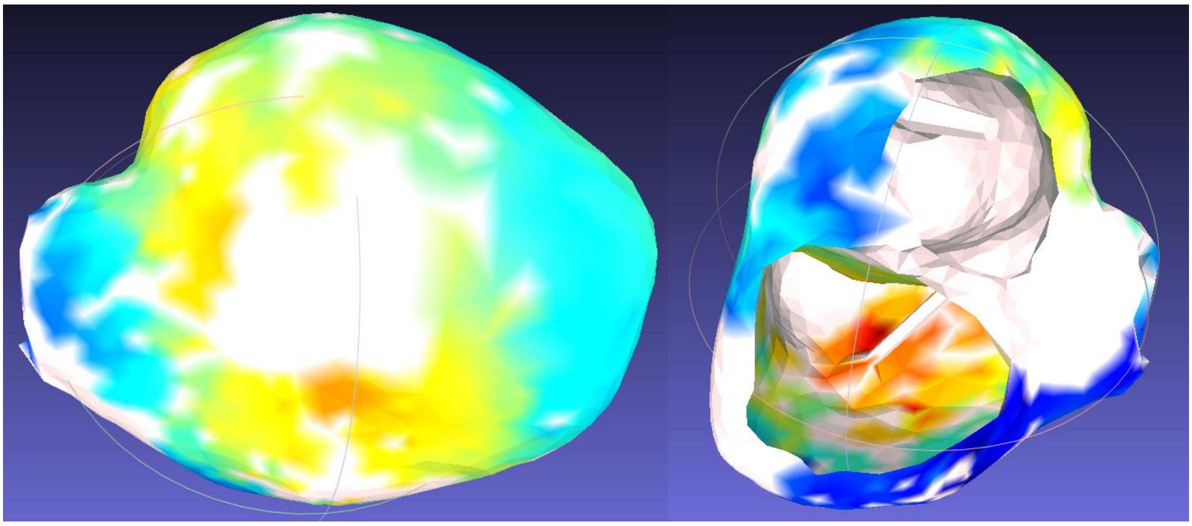


Figure 80 - Case of RVOT PVC, inverse [own illustration]

As already mentioned above, the vertices with the white spots do not have a timestamp value associated to it, here for the inverse case. For the recurrent ventricular arrhythmias, this is reversed as illustrated in Fig. 81:

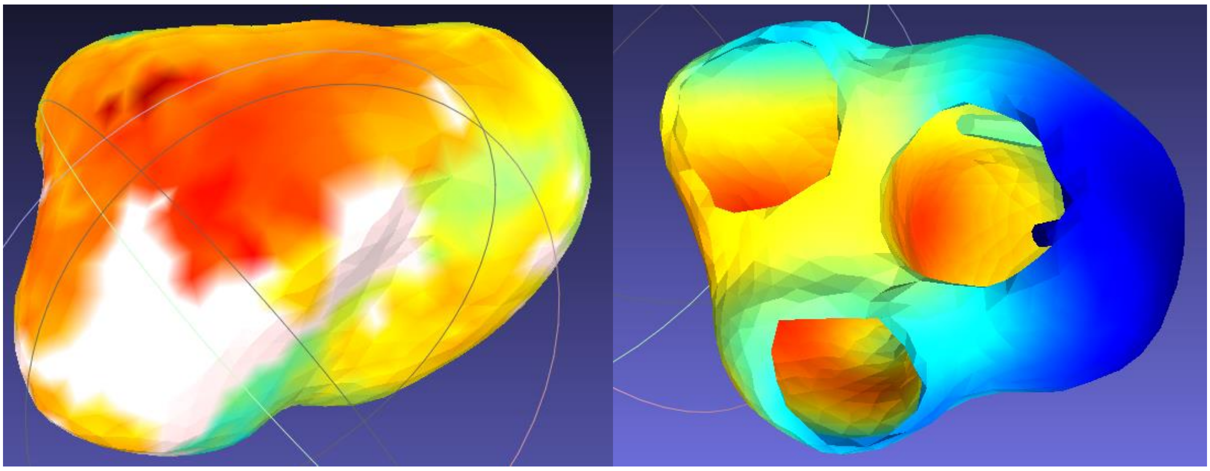


Figure 81 - Recurrent ventricular arrhythmia, invasive to the left, inverse to the right [own illustration]

In summary, all the provided cases of Alvale could correctly be converted to the VRML format for 4D printing of the models. The created MATLAB script for the conversion can be used by Peacs in the future for their purposes, either if it is 4D printing of models or further virtualization topics.

For more detailed information about the conversion process in MATLAB, see Appendix C and D.

6.6 Resolution of Data Sets

The resolution is defined with two parameters, pixel spacing and slice thickness. As each file has to be read once at a time, only a spot check has been carried out. An attempt of a MATLAB script trying to read the all the particular information out of all DICOM files at once was only partly successful.
The resolution of the provided data sets differs vastly and out of the investigated sample, the pixel spacing ranged between 1.250mm and 2.961mm, with an average of 1.706mm and a median of 1.666mm (see Appendix F). The slice thickness ranged between 5mm and 10mm, with an average of 7.848mm and a median of 8mm.

	Pixel x-axis	Pixel y-axis	Pixel Spacing [mm]	Slice Thickness [mm]
Max	290	320	2.961	10
Min	88	128	1.25	5

Median	1.706	7.848
Average	1.666	8

Table 17 - Overview on investigated values

Slice Thickness
The slice thickness can be set by the operator of the MRI and affects the image generation time, if the slice thickness is doubled, the image generation time is halved. Furthermore, it "can cause blurring in our image also known as partial voluming" [49] and therefore decrease the detail or spatial resolution of the image. If the slice thickness is increased, the amount of nuclei detected per slice increases. As each nuclei produces a signal in the measured slice, in increases the amount of signals and therefore leads to an increase in the signal-to-noise ratio (see chapter 2.2.1).

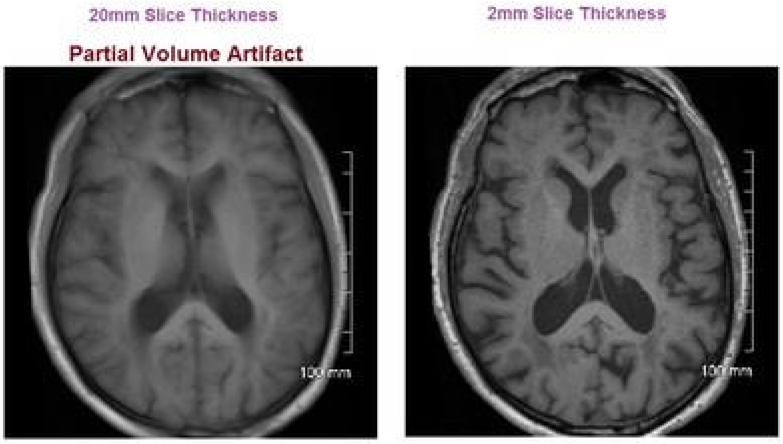


Figure 82 - Effect of slice thickness on final image [49]

In short, the thinner the selected slice thickness is, the more detail is visible on the DICOM image. The tissue voxels are smaller and therefore the image pixels are more precise. Fig. 83 illustrated the general relationship between voxels and pixels:

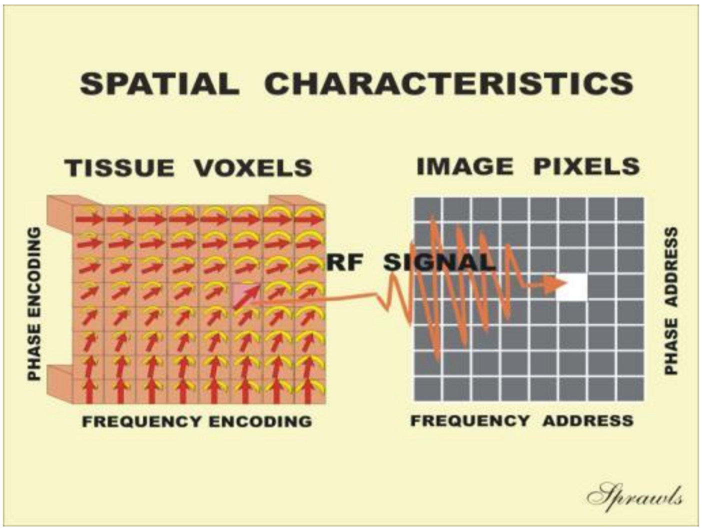

SPATIAL CHARACTERISTICS
TISSUE VOXELS
IMAGE PIXELS
PHASE ENCODING
RF SIGNAL
PHASE ADDRESS
FREQUENCY ENCODING
FREQUENCY ADDRESS
Sprawls

Figure 83 - The relationship of tissue voxels to image pixels [50]

Pixel Spacing

The pixel spacing is the physical distance between the center of each pixel, defined by the adjacent row spacing and the adjacent column spacing. This illustrated in Figure 84:

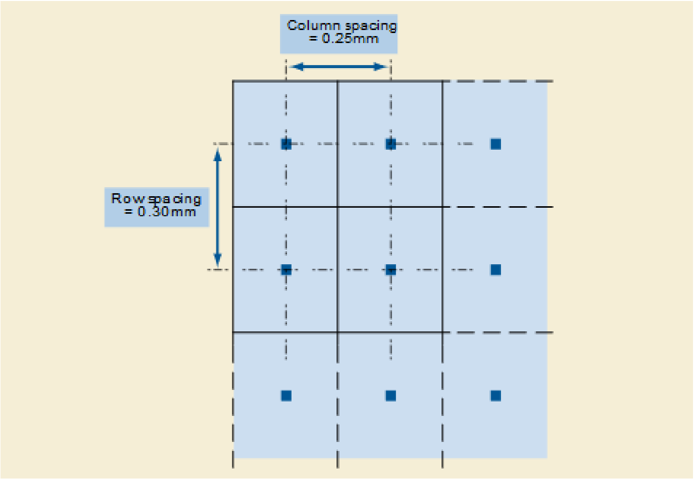

Column spacing
= 0.25mm
Row spacing
= 0.30mm

Figure 84 - Example of Pixel Spacing [51]

First, the row spacing is named, and then the column spacing. Therefore, the pixel spacing in this example would say following:

```
0028,0030,PixelSpacing=0.30   0.25
```

If the image has a size of 240x240 and the pixel spacing is 1.75mm, the image has a physical size of 420x420mm on a 1:1-ratio.

Other factors such as image acquisition time can change the output image [52] too and therefore can influence the segmentation process and can change the 3D model of the segmented body part.

6.7 Discussion on the automation of the overall process

An advantage in further medical use cases of the Alvale process would be the automatization of the overall process as this would save time and costs. The following section shall discuss the current status as well as the open topics to achieve this.

Image Data Generation (MRI or CT)

The MRI image data generation is not part of this discussion in general, but it can be presumed that this process will remain the same as today in the near future where the physician or assistant carries out the preparation of the patient and operates the equipment. One major task of the operator required to enable automatization of further steps is the definition of the important meta data information such as Patient Name or Patient ID. This is necessary to create a consistent data set where the final output data for 4D printing includes this meta data and it can be allocated to the initial MRI data set. Furthermore, the storage folder structure of the image files should be standardized to simplify the access to the image sets by the further process steps.

Overall, the storage folder structure can be predetermined as an automatization step, whereas the operating of the equipment and the meta data definition still remains partly manual labor.

Geometry Data Generation (GeomPeacs)

The geometry generation of the heart model is a semi-automatic process where the tool supports the user by connecting the selected image areas of each DICOM image layer to a complete 3D model. An experienced user can generate a 3D model in roughly half an hour and export it as an .xml-file.

If the overall process (without the patient preparations for MRI imaging and ECG measurements) of Alvale up to the 4D printing of the heart model shall be automatized, the image selection process presents the biggest obstacle to be overcome. The probably most suitable approach to fully automize this step would be the area recognition of the heart on the DICOM images using a deep learning or machine learning image analyzing tool which is implemented into GeomPeacs. Research on this field is already carried out in various projects [53, 54, 55, 56, 57]. At present, this algorithm has still to be developed and implemented into GeomPeacs.

ECG Data Generation

The ECG data generation is already mostly automatized. The main manual labor of the process is the preparation of the patient with the application of the electrodes and the set up with the 3D camera. Furthermore, the storage folder structure is currently almost the same for each data set, which can be created automatically. It can be presumed that the preparation of the patient remains manual labor, all other tasks can be automatized.

Application ECG Data onto Geometry Data

The application of the ECG data onto the 3D model of the heart is already automatized by MATLAB scripts of Peacs. These scripts can read the initial ECG and geometry data sets and combine them to one array in the MATLAB workspace. This can be displayed as well with a separate “showResults” script (see Appendix C). No further actions required.

Conversion to VRML for 4D Printing

The feasibility of the data conversion of the basic data sets (TRI for geometry and DEP for timesteps) could already be shown [1]. The difference to this particular case is the addition of the conversion script to the initial script of Peacs. This step creates a complete process from the reading of the initial ECG and geometry data sets up to the conversion to the VRML format with a color-code for the ECG information. This enables the 4D printing of the Alvale model and other use cases such as being imported into general display tools or CAD programs.

Further Automation Possibilities for other Use Cases

X3D is the successor of the VRML format and is widely used, mainly in HTML applications such as web browsers. A further possible use case would be the conversion of the VRML file or the initial data sets to X3D as it supports both XML encoding and VRML encoding [58]. This would enable a very widespread use of the output file due to the many tools supporting the format, the use of e.g. animations or video generation, or the implementation into web services.

4D Printing

The import of the VRML file to the printer software is currently a manual process and no automatic import is available today. If this can be automatized remains an open question as this would require a direct access of the Alvale process to the printer software. On the other hand, this step only takes minimal time effort and would not present a big advantage if automatized.

One important part ahead of the printing itself is the analysis of the 3D model in regard of feasibility of the printing with the used printer. The major factors are the size of the model and its minimal wall thickness. These limits are determined by the used material, printing process and the printer model, and the analysis can be done either by the printer software itself or online tools such as provided by i.materialise [59]. The analysis itself is automatic, but currently the user has to start the analysis manually. The start of the printing is done manually as well. Maybe a Python or MATLAB script could automate that as well.

Conclusions

If the preparation of the patient for the MRI imaging and the ECG procedure are excluded, it can be stated that the almost complete automatization of the Alvale process is feasible. The most significant step unsolved is the implementation of an automated image recognition into the segmentation process which selects the relevant image area to create the 3D model of the heart. All tasks up to the generation of the output VRML file are already mostly automatized. The MATLAB script has to be started manually within MATLAB at the moment (see Appendix C), a GUI which starts the process would ease the accessibility for other users. A standalone execute program or the implementation into a web service including all the functions up to the VRML export would detach the process from MATLAB and GeomPeacs and enable the use of it for a wider user base without knowledge of both programs or even the human anatomy.

The general data handling (e.g. import of the VRML model to the 3D printers' software) may remain manual, but this steps usually do not require much time investment. The only major time investment left would be the preparation of the patient for the ECG measurements and MRI imaging.

7 Further application for color-coded, patient-specific data

The Alvale project and its process is a very specific use case for a color-coded 3D model which is viable to be 4D printed. Medical technology, in particular medical diagnostics and medical simulations, is developing rapidly and in the two beforementioned areas color-coded images are often used to visualize the medical facts. This chapter shall provide another possible use case for the application of color-coded information onto the 3D geometry of either a segmented organ or a self-created 3D model.

7.1 Current use in other specific use cases

Medical Diagnostics

The Infinix-I and its software from Canon [60] uses "Parametric Imaging which displays an entire image sequence as a single composite image that is color coded in order to characterize the contrast media dynamics and to allow easier visual evaluation. Color Coded Circulation (CCC) can create movies by shifting color scale gradually so that it is easy to understand vessel flow."

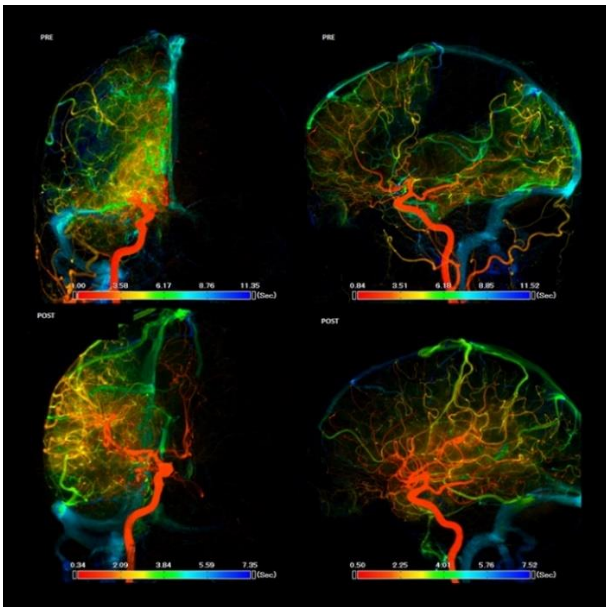

PRE
PRE
0.00 3.58 6.17 8.76 11.35 (Sec)
0.84 3.51 6.18 8.85 11.52 (Sec)
POST
POST
0.34 2.09 3.84 5.59 7.35 (Sec)
0.50 2.25 4.01 5.76 7.52 (Sec)

Figure 85 - Lateral view of DSA run of a stroke patient [60]

Another use of color-coding is the diagnosis of magnetic resonance elastography (MRE) to diagnose liver fibrosis [61]. MRE is a non-invasive method which "uses sound waves to determine tissue firmness. Physicians are able to see inside the liver without biopsies. The red indicates diseased tissue while the blue indicates healthy tissue" [62]. Combined with a segmentation of the liver this would enable a color-coded 3D model to be printed out.

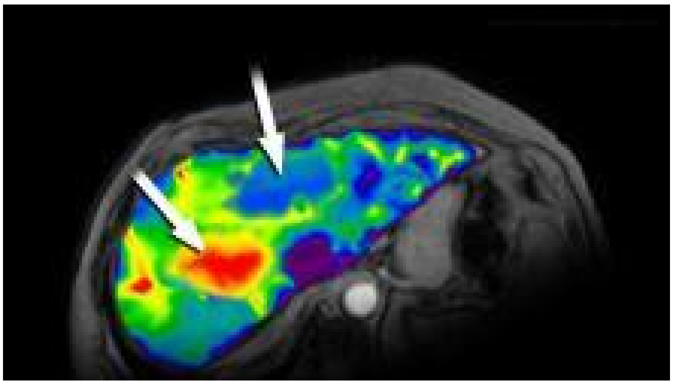


Figure 86 - MR Elastography of a liver to determine tissue firmness [62]

Medical Simulations

One paper [63] focuses on the blood flow simulation within the heart, where cardiovascular hemodynamics are computed to "provide useful insights to physicians recognize indicators leading to CVD (cardiovascular disease) and also to aid the diagnosis of CVD". The geometry is obtained by the reconstruction of the heart model with a method called image-based CFD (IB-CFD) simulation.

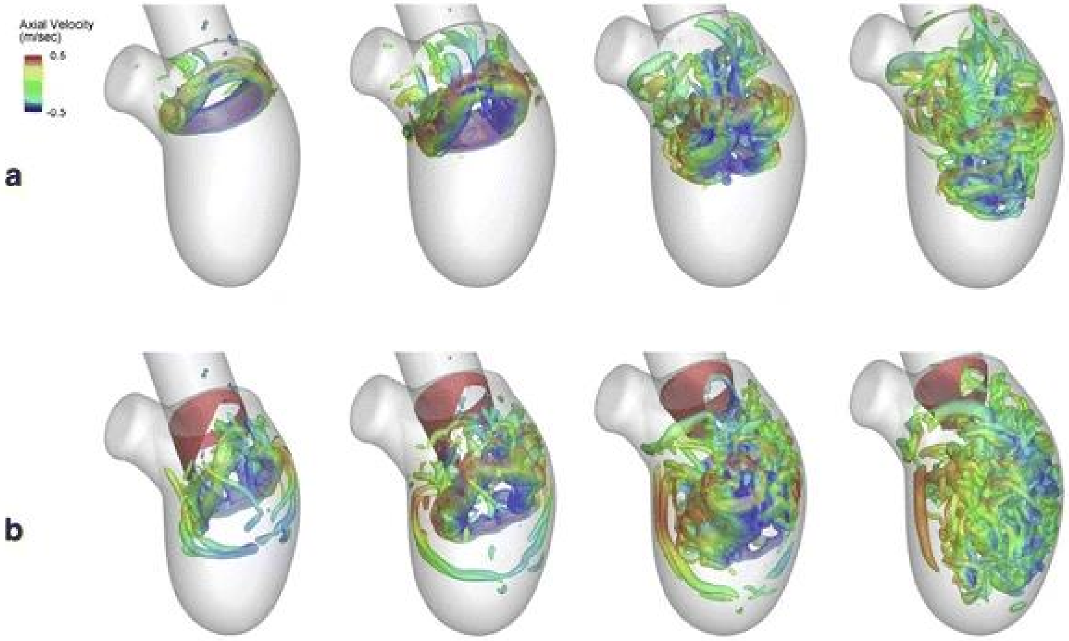


Figure 87 - "Comparison of the development of intraventricular flow with and without incorporating valve leaflets" [63]

Such simulation cases could be exported out of the simulation software and converted to VRML for 4D-printed models as a visualization object. This may not be trivial as standard CFD simulation software may not support VRML (or VRML 2.0 to be more precise, see chapter 7.2). Moreover, a conversion to X3D would open up various possibilities and use cases without the need of the simulation software.

A similar use case of modelling fluid dynamics in cardiovascular medicine aims at the minimizing of invasive instrumentation [64]. The 3D model is as well reconstructed using segmentation of DICOM images.

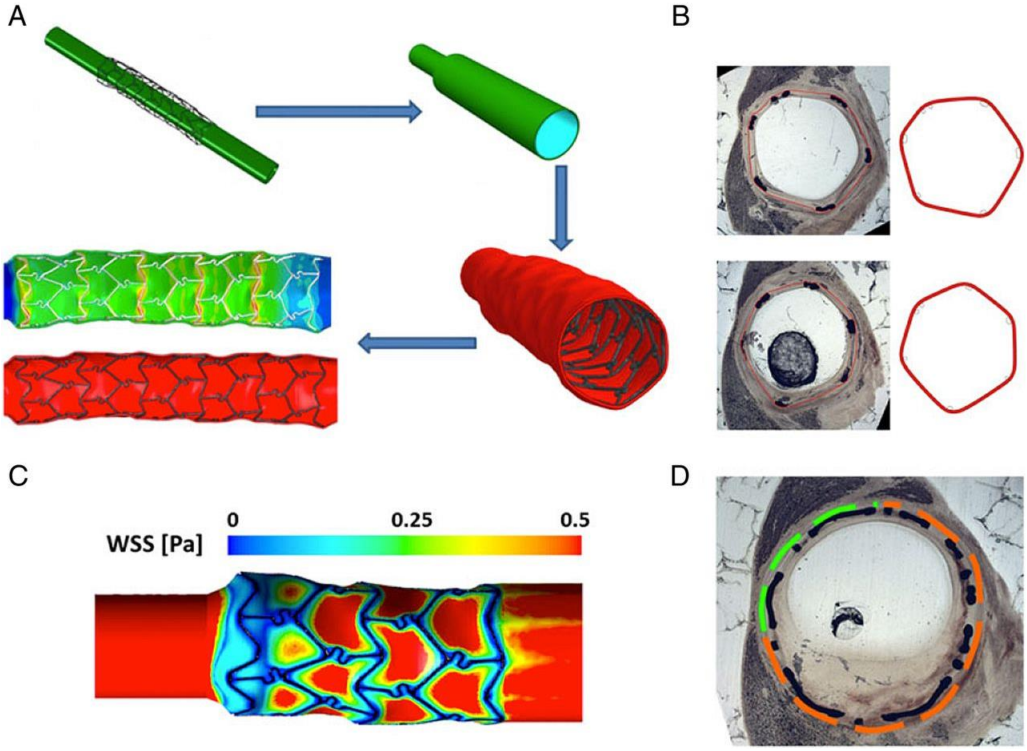

A
B
C
D
WSS [Pa]
0
0.25
0.5

Figure 88 - "A computational fluid dynamics (CFD) model demonstrating the correlation between wall shear stress (WSS) and restenosis in coronary artery disease" [64]

"CFD modelling enables investigation of pressure and flow fields at a temporal and spatial resolution unachievable by any clinical methodology. Post-processing provides additional data, generating new insights into physiology and disease processes. For example, it is difficult, and invasive, to measure arterial wall shear stress (WSS), a key factor in the development of atherosclerosis and in-stent restenosis, whereas CFD models can compute WSS and map its spatial distribution" [64]

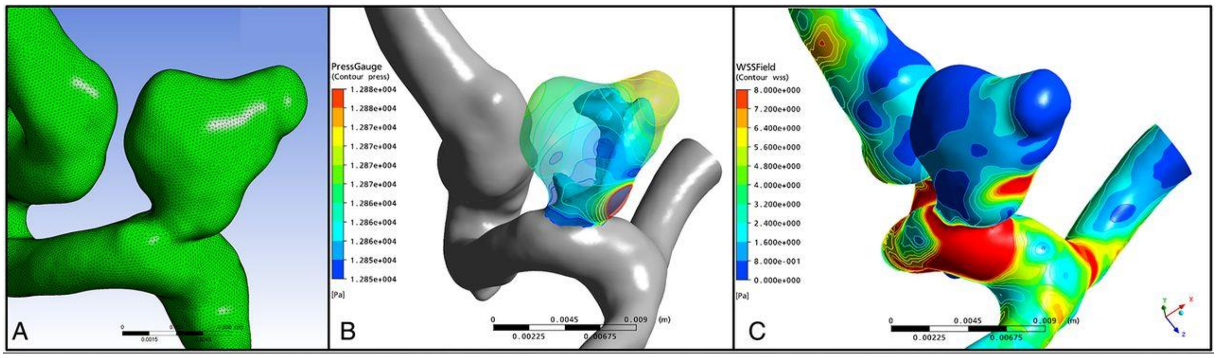

A
PressGauge
(Contour press)
1.288e+004
1.288e+004
1.287e+004
1.287e+004
1.287e+004
1.287e+004
1.286e+004
1.286e+004
1.286e+004
1.285e+004
1.285e+004
[Pa]
0
0.0045
0.009
(m)
0.00225
0.00675
B
WSSField
(Contour wss)
8.000e+000
7.200e+000
6.400e+000
5.600e+000
4.800e+000
4.000e+000
3.200e+000
2.400e+000
1.600e+000
8.000e-001
0.000e+000
[Pa]
0
0.0045
0.009
(m)
0.00225
0.00675
C

Figure 89 - "Computational fluid dynamics (CFD) model of an intracranial berry aneurysm from the @neurist project" [64]

Such models can as well be 4D-printed for visualization purposes or other use cases such as surgery preparation.

Another use of simulations [65] is the simulation of cardiovascular blood flow for the patient-specific analysis of the blood flow to improve heart surgery. The difference to the other two cases of this particular case is the use of a standalone open source simulation software called SimVascular [33].

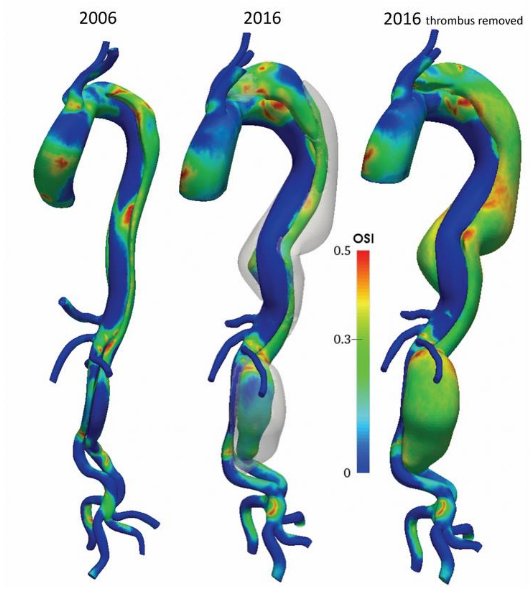
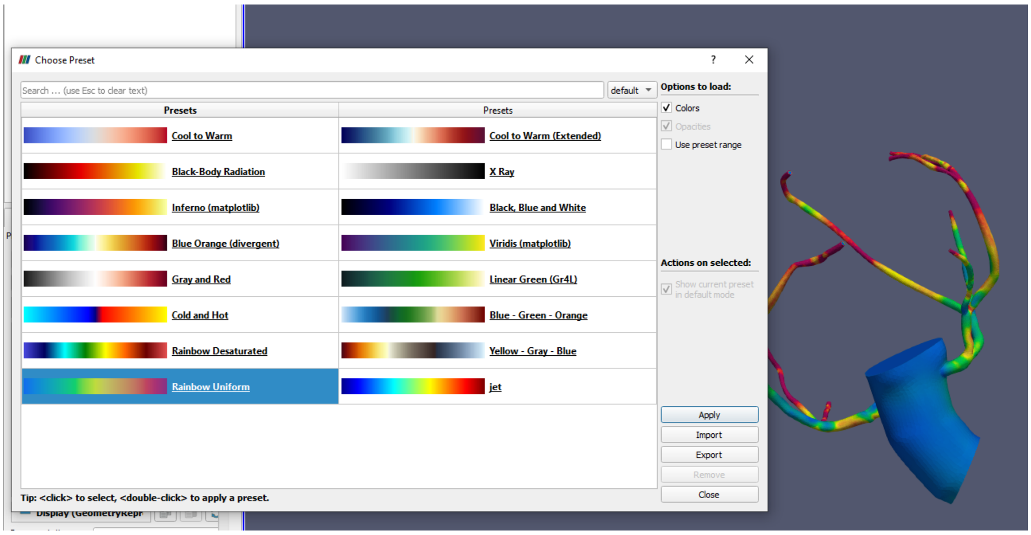

2006
2016
2016 thrombus removed
OSI
0.5
0.3
0

Figure 90 - Illustration of a simulation model [66]

This simulation software stores its results in the VTP and VTP formats primarily associated with ParaView, which is a "open-source, multi-platform data analysis and visualization application" [63]. The VTP format is a surface mesh and the VTU format is a volume mesh. Example data sets for SimVascular can be downloaded and the result files can be edited with ParaView. For example, the wall shear stress results of such a simulation can be displayed and the color-code can be edited as the user desires (see Figure 91). This edited model can be exported as a VRML file, but with the .vrml file extension. Due to this the extension, programs cannot read this file immediately, but a simple manual edit of the extension to .wrl makes it readable for those programs.


Choose Preset
?
Search ... (use Esc to clear text)
default
Presets
Cool to Warm
Black-Body Radiation
Inferno (matplotlib)
Blue Orange (divergent)
Gray and Red
Cold and Hot
Rainbow Desaturated
Rainbow Uniform
Presets
Cool to Warm (Extended)
X Ray
Black, Blue and White
Viridis (matplotlib)
Linear Green (Gr4L)
Blue - Green - Orange
Yellow - Gray - Blue
jet
Apply
Import
Export
Remove
Close
Display (GeometryRepr

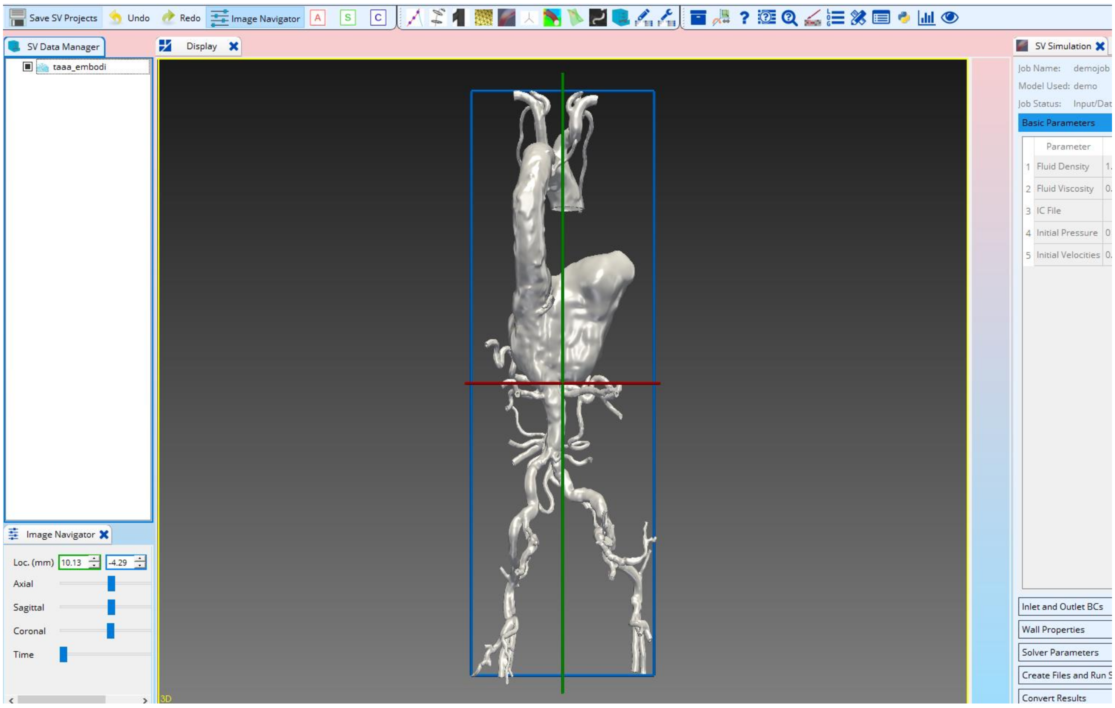
Options to load:
☑ Colors
☑ Opacities
☐ Use preset range

Actions on selected:
☑ Show current preset in default mode

Tip: <click> to select, <double-click> to apply a preset.

Figure 91 - Editing of the color code in ParaView for the wall shear stress results [own illustration]

SimVascular supports 3D model data formats such OBJ, STL or MDL (Simulink) to be imported. This enables the import of all exported segmentations of 3DSlicer as they can be exported either as STL or OBJ. These models then can be used to run the simulation of the blood flow, results stored as VTP and VTU, edited with ParaView and exported as 4D-printable VRML file.


Save SV Projects
Undo
Redo
Image Navigator
A
S
C
SV Data Manager
taaa_embodi
Display
Image Navigator
Axial
Sagittal
Coronal
Time
3D
SV Simulation
Basic Parameters
Inlet and Outlet BCs
Wall Properties
Solver Parameters
Create Files and Run S
Convert Results

Loc. (mm) 10.13 -4.29

Job Name: demojob
Model Used: demo
Job Status: Input/Dat

	Parameter	
1	Fluid Density	1.
2	Fluid Viscosity	0.
3	IC File	
4	Initial Pressure	0
5	Initial Velocities	0.

Figure 92 - Import of a STL model derived from segmentation into SimVascular [own illustration]

A further use case could be the simulation of a balloon catheter insertion into a blood vessel by using a standard FEM analysis tool such as ANSYS. Such tools are widespread in use in engineering and therefore have a substantial user base which could generate such a simulation.

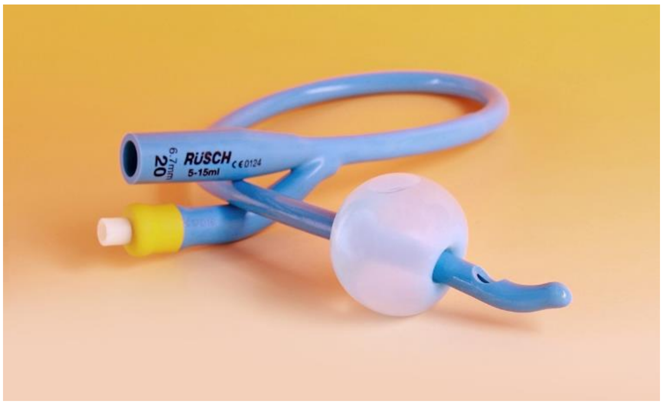


Figure 93 - Balloon catheter [68]

A balloon catheter uses a balloon filled with pressurized air or a fluid which can be expanded in a blood vessel. It is commonly used for treatment of in-stent restenosis (ISR) to recover the blood flow. This procedure introduces an inner pressure to the blood vessel which leads to stress in the vascular wall. The simulation of the introduction of the catheter could provide useful information about the size of the balloon to be chosen or potential damaging of the vessel. Balloon catheter are furthermore used in other fields as well such as urology, pain therapy, internal medicine, radiology and neurology [69].

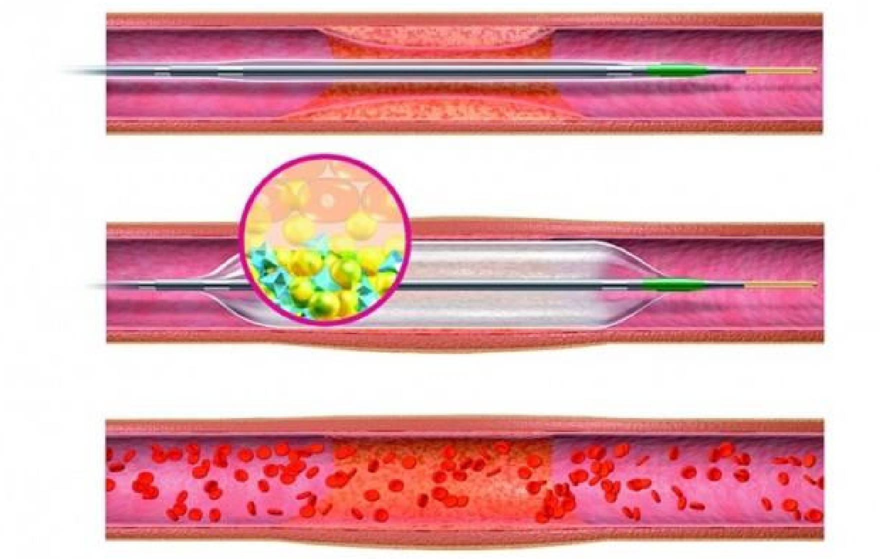


Figure 94 - Spreading of blood vessel with balloon catheter [70]

7.2 Application of patient-specific color-codes to selected cases

Two beforementioned use cases are selected, the the SimVascular process and the balloon catheter insertion simulation. The SimVascular process is already mature with the VRML conversion with ParaView. The simulation of the insertion of a balloon catheter to a patient-specific segmentation and especially the conversion to VRML has to be newly developed.

SimVascular Simulation Process and Conversion to VRML

The SimVascular simulation process has already been touched in chapters 5.3 and 7.1. The first step is either the segmentation of the particular blood vessel from DICOM images in SimVascular or the import of an already available STL model. The editing of the result files can be done in ParaView and exported as VRML. One converted simulation result was eventually 4D printed without issues.

ANSYS Simulation Results Conversion to VRML

The goal of this application is not an accurate simulation of the vascular wall stresses during balloon catheter insertion but the proof of concept of the process. This process contains the application of a local inner pressure to a hollow tube (or blood vessel), the export of the result files and the conversion into VRML for 4D printing.

Firstly, the particular use case of expanding a cardiac vessel is chosen (see Fig. 96). There a specific location is chosen where the balloon catheter should be applied (green circle). To simplify the process, a simple generic STL model is created which serves as the test bed (Fig. 95). For a patient-specific use case the STL of the segmentation can be imported.

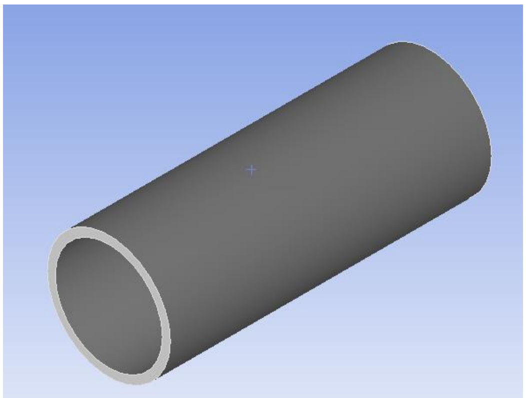


Figure 95 - Generic tube model serving as a test bed in ANSYS [own illustration]

At this model, a local inner pressure is applied to mimic the effect of the balloon catheter:

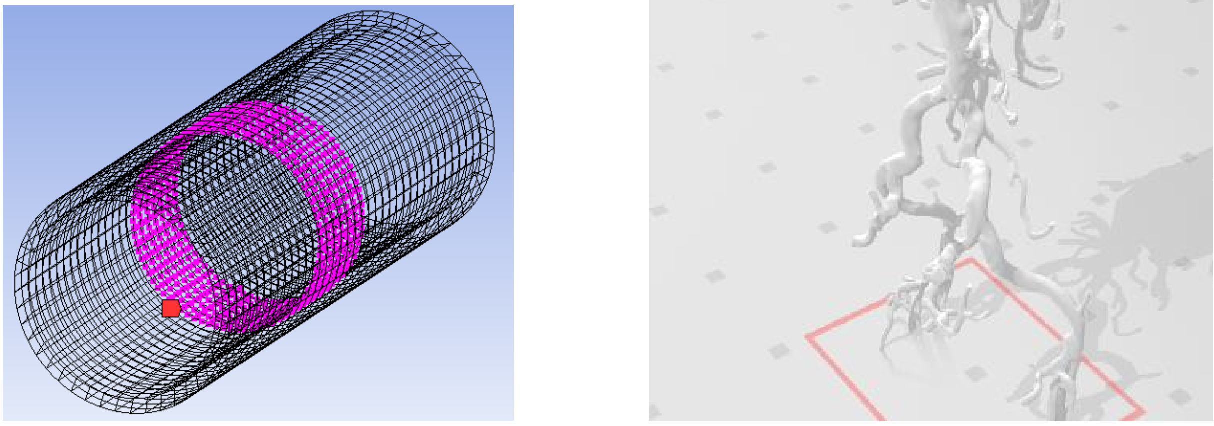


Figure 96 - Left: mesh and force visualization of tube. Right: simulated location [own illustration]

Both ends of the tube are fixated, if no boundary condition is defined the solver of ANSYS cannot run the program tells you that not enough boundary conditions are defined. Then the simulation is run and the result of this is illustrated below. The stress values and the deformation do not represent actual expectable values at a correct case, this is only to prove the concept. The fixation of both ends of the tube do not induce stress and therefore the coloring of this area is the same as for the unaffected tube sections.

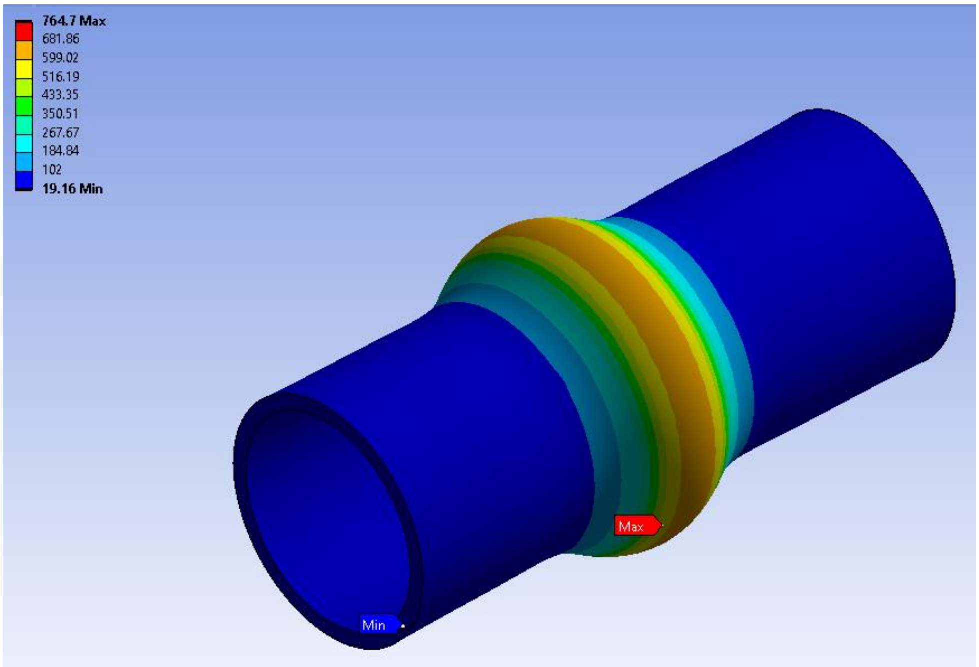
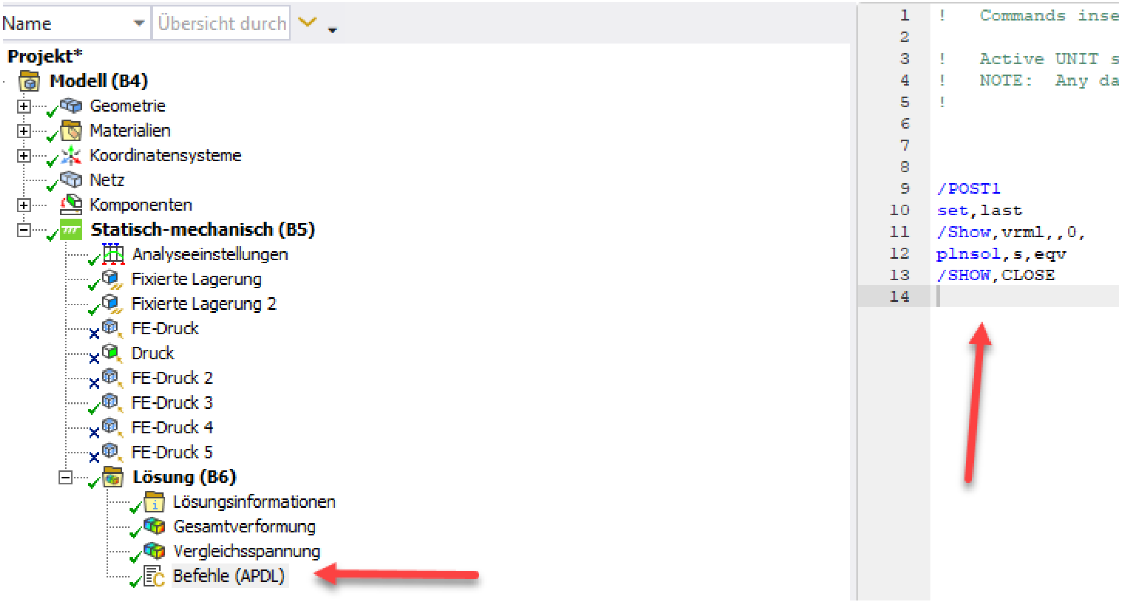

764.7 Max
681.86
599.02
516.19
433.35
350.51
267.67
184.84
102
19.16 Min
Max
Min

Figure 97 - Von Mises stress in vascular wall with exaggerated deformation [own illustration]

This result of the simulation can now be exported with the addition of an APDL (Ansys Parametric Design Language) command to the solver which automatically exports the model to the solutions folder, here VRML is chosen.


Name
Übersicht durch
Projekt*
Modell (B4)
Geometrie
Materialien
Koordinatensysteme
Netz
Komponenten
Statisch-mechanisch (B5)
Analyseeinstellungen
Fixierte Lagerung
Fixierte Lagerung 2
FE-Druck
Druck
FE-Druck 2
FE-Druck 3
FE-Druck 4
FE-Druck 5
Lösung (B6)
Lösungsinformationen
Gesamtverformung
Vergleichsspannung
Befehle (APDL)

```
!   Commands inse

!   Active UNIT s
!   NOTE:  Any da
!



/POST1
set,last
/Show,vrml,,0,
plnsol,s,eqv
/SHOW,CLOSE

```

Figure 98 - Addition of APDL command to solver [own illustration]

The exported file now can be opened using the Cortona3D Viewer:

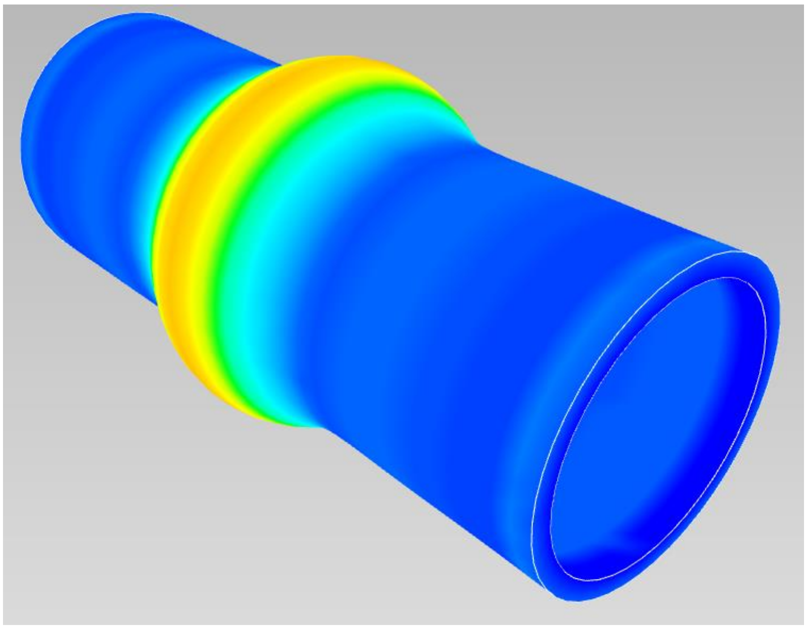


Figure 99 - Display of the exported solution model with Cortona3D Viewer [own illustration]

Unfortunately, the exported VRML file uses version 1.0, whereas most tools use VRML 2.0 which succeeded version 1.0 in the year 1997, and Cortona3D is the only tool found which displays the model correctly even though it does not officially support it [71]. This limits the further usage of the model drastically, with the most crucial fact that 4D printers do not support VRML 1.0 [72]. This creates the need to convert the export file to VRML 2.0 to prove the scenario.

Due to the obsolete status of the old version, virtually no documentation of the file syntax is available which increases the difficulty to convert it correctly. The most beneficial point is the fact that both file formats are triangulation based, therefore vertices, triangles and colors are defined similarly, and these values can be kept.

Firstly, the general syntax (plain text) of VRML 1.0 has to be replaced with the syntax of VRML 2.0. As both files are available in ASCII format, the text can be deleted and replaced with the new one. One example of the header replacement is shown below, which has to be done with every plain text.

VRML 1.0

```
#VRML V1.0 ascii
Separator {
renderCulling OFF
ShapeHints {
vertexOrdering COUNTERCLOCKWISE
faceType UNKNOWN_FACE_TYPE
}
Material {
diffuseColor [
  0.0000  0.0000  0.0000,
```

VRML 2.0

```
#VRML V2.0 utf8

Transform {
children [ Shape {
geometry IndexedFaceSet {
coord Coordinate {
point [
0.000004  0.11501  0.59970,
```

It is clearly visible that the syntax itself differs vastly from each other. Furthermore, the structure of VRML 1.0 is different to VRML 2.0 as well. As described in detail [1], the general and simplified structure of VRML 2.0 looks like following. It describes the vertices, coordination index to define triangles and the applied colors to the vertices:

```
#VRML V2.0 utf8

Transform {
 children [ Shape {
   geometry IndexedFaceSet {
      coord Coordinate {
        point [
        0.000004  0.11501  0.59970,]
        }
        coordIndex
        [
          0,1,2,-1,
        ]
      color Color {
        color [ 0 0 1 ]
      }
      }
    }
  ]
}
```

VRML 1.0 exported from ANSYS on the other hand has following structure (simplified):

Header
The standard header of the file.

```
#VRML V1.0 ascii
Separator {
renderCulling OFF
ShapeHints {
vertexOrdering COUNTERCLOCKWISE
faceType UNKNOWN_FACE_TYPE
}
```

Material
This field defines the RGB colors available for the model. `[0.000  0.000  0.000]` would be black.

```
Material {
diffuseColor [
  0.0000  0.0000  0.0000 ]
```

Normal vector
Normal vector of the triangles. Used in VRML 2.0 as well but not necessary to display the model correctly, therefore can be deleted completely.

```
Normal {
vector [
0.71308E-01  0.99730  -0.17555E-01 ]
```

Vertice definition
The vertice definition is basically the same, only the plain text syntax has to be replaced.

```
Coordinate3 {
point [
-0.97876E-06  0.11472E-01  0.58391E-01 ]
```

Coordination Index definition
Same as for the vertice definition, only the plain text has to be replaced.

```
IndexedFaceSet {
coordIndex [
   0,   1,   2,   3,-1 ]
```

Material Index definition
This is the biggest difference to VRML 2.0 where the colors are defined after the geometry definition and the values are RGB conform. In VRML 1.0 the colors are defined in a separate field (see Material, p. 85) and the colors for the geometry access the RGB values through indexing. In other words, the 23rd color defined in the Material field would apply here to the first vertice, the 19th to the second and so on.

```
materialIndex [
  23,  19,  19,  22,-1]
```

This creates the need to replace all these index values with the true RGB values for every single vertice color, and erase the `-1` which closes the loop (similar as the Coordination Index).

Normal Index
As the normal vectors are not needed, this can be erased completely as well.

```
normalIndex [
   0,
   1,
]
```

From this point on, the loop from the normal vector to the normal index repeats several times to define the full 3D model.

The overall file structure looks as following:

```
#VRML V1.0 ascii
Separator {
renderCulling OFF
ShapeHints {
vertexOrdering COUNTERCLOCKWISE
faceType UNKNOWN_FACE_TYPE
}

Material {
diffuseColor [
  0.0000  0.0000  0.0000
]
Normal {
vector [
0.71308E-01  0.99730  -0.17555E-01
]
Coordinate3 {
point [
-0.97876E-06  0.11472E-01  0.58391E-01
]
IndexedFaceSet {
coordIndex [
   0,   1,   2,   3,-1
]
materialIndex [
  23,  19,  19,  22,-1]

normalIndex [
   0,
   1,
]
Normal {
vector [
0.71308E-01  0.99730  -0.17555E-01
]
```

etc.……

The material field is only defined once at the beginning of the file, the geometry and color definition repeats several times depending on the file size and its distribution. For example, the simple generic model of the vascular wall stress simulation has more than 40 of such loops with ca. 20’000 vertices.

For the conversion of VRML 1.0 to VRML 2.0 a separate MATLAB script was created (see Appendix E) which is able to do the abovementioned necessary changes to the file syntax and structure automatically. The result of the conversion (see Fig. 100) can now be used with the standard tools and is able to be 4D printed.

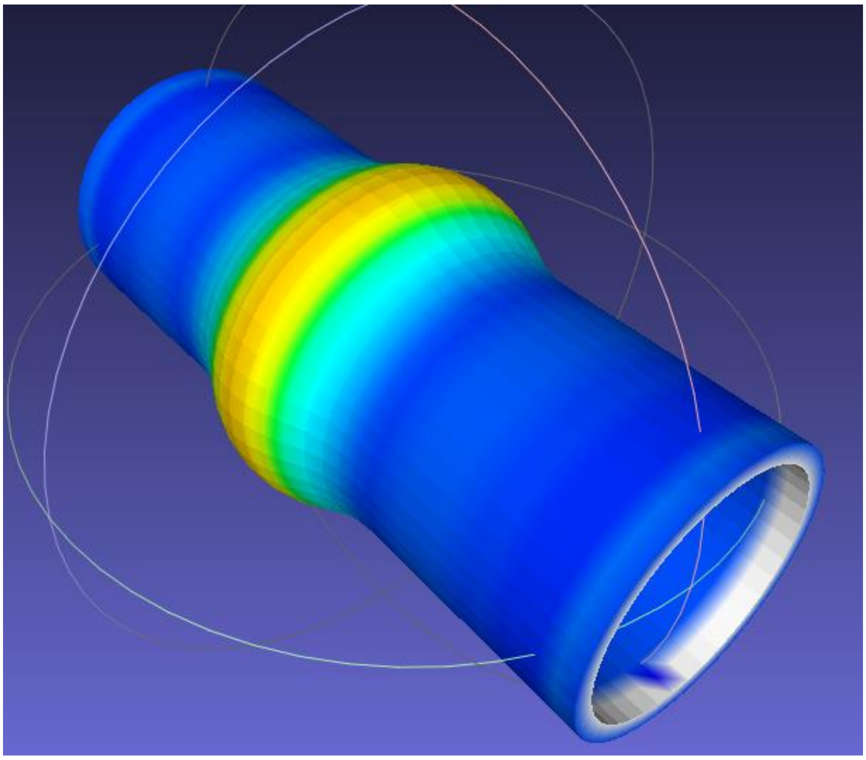


Figure 100 - VRML2.0 conversion of ANSYS simulation export [own illustration]

The geometry itself is complete and correctly converted. For the colors on the other hand, it is visible that the triangles at the inner edge have no color information. The issue could not be resolved, it is assumed that those specific color nodes were defined differently and the MATLAB script deleted this information. Nevertheless, the important section of the von Mises stresses from the insertion of the balloon catheter are displayed identical to the VRML 1.0 version. Further refinement of the script can may resolve this issue.

Conclusion

The simulation of a particular medical case with a standard FEM analysis tool, the export of the simulation results and their conversion to a 4D-printable format could be shown to be viable. Further refinement is still necessary, but the main process works as intended.

A further unrelated benefit is the possible use of the script for 4D printing of all simulation results of ANSYS Mechanical. Only two external ways to do this have been discovered as ANSYS does not provide such a solution by itself. Its usage is very limited (very few colors available) and involves much manual labor [73]. Therefore, the newly created MATLAB script could be exported as an executable and used for this application with very little to no manual labor involved.

8 Conclusion and Outlook

Three problems have been set for investigation for this paper:

1. The conversion of further data sets compared to [1] and its optimization. Furthermore, the process shall be optimized in regard of several parameters, especially a data consistency has to be provided
2. The geometry data sets are usually derived through the segmentation of DICOM images with the help of specific tools (e.g. 3DSlicer). The basics of the process from DICOM image to segmented 3D model shall be investigated for the further purpose of 3D printing
3. Other uses cases for 4D printing as Alvale shall be investigated, what data formats are available and how can they be processed for eventual printing.

The first goal of the conversion of further Alvale data sets could be achieved. Eleven different data sets with different medical conditions were converted to the VRML format with the very same script, showing the validity and consistency of the conversion. A major improvement of the process is the implementation of the conversion of the data to VRML into the MATLAB scripts by Peacs which uses the output data of their process directly. Furthermore, an example of the implementation of DICOM meta data attributes could be given for several attributes, which provides data consistency from the DICOM images to the VRML file. A proposal of the required attributes is given, but it still has to be defined which information is essential. The data privacy is still an open topic to be discussed.

The segmentation process from importing DICOM image sets, the segmentation of the images to create a 3D model and its export is explained in detail. Various different programs are presented with its differing purposes and capabilities. Especially the reading of the DICOM image data and its attributes is important for the further processing of the data. For the foundation for this process, the DICOM format is explained in very detail, from the image generation processes and the image output data generation, the properties of the format itself and the image display. Especially the image display represents a major obstacle as some attributes of the file format change the image tremendously, e.g. contrast changes for better vision of the body part to be looked at. All of this information has to be considered for the overall process, even more so if the overall process shall be automated. This automatization of the segmentation cannot be achieved due to the current lack of image analysis tools capable to replace the manual definition of the to be segmented body part.

An investigation into the field of medical diagnostics and medical simulations provided many different possible use cases which could be used for the conversion process and to be 4D printed. The two chosen simulation examples could provide the validity of two different approaches for the use case of 4D printing of patient-specific data sets. The SimVascular approach is very straightforward with tools already available to create a 4D-printable format of the simulation results. The ANSYS approach required a different conversion tool due to the fact that the simulation results only could be exported in the obsolete VRML 1.0 format and the todays' 4D printers and other software do not support this format. The conversion to VRML 2.0 could be achieved, but further refinement of the MATLAB script is required to fully automate the process.

In general, all the converted data sets are viable to be 4D printed without any further adjustments to the VRML files. Overall, six components were printed, two of them are illustrated below:

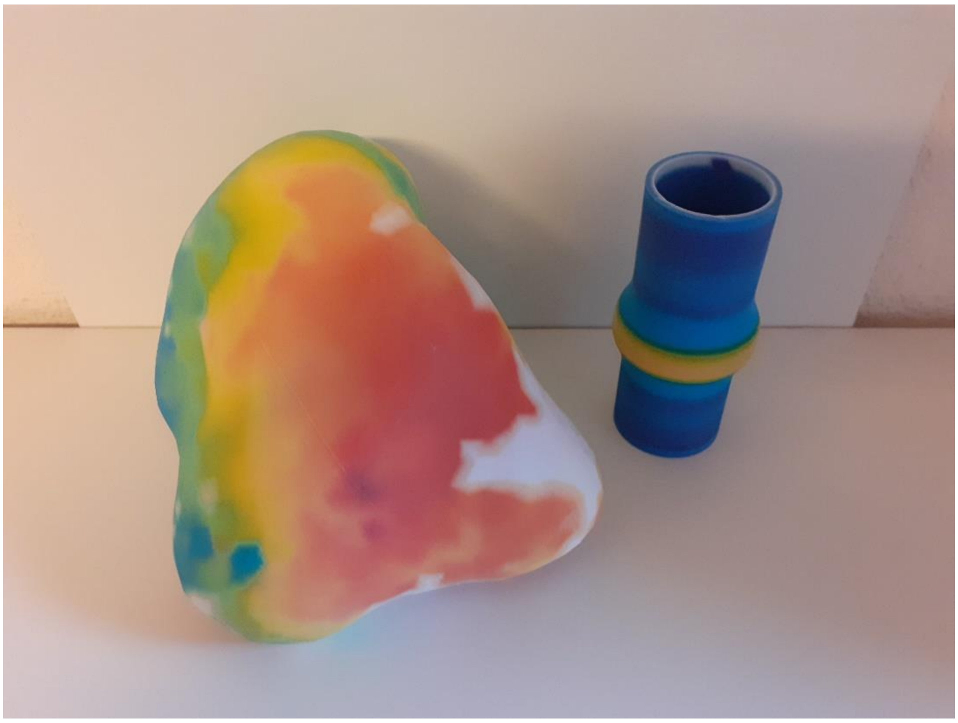


Figure 101 - 4D printed models [own illustration]

Outlook

As already mentioned in [1], the developed process of the conversion of the data sets for Alvale can be used as a foundation for further digitalization such as a web-based application where the DICOM images can be uploaded, the data is processed according the wishes of the user and 4D printable files are generated. Furthermore, the format could be changed to X3D for a fully HTML compatible solution.

A major obstacle to be overcome represents the automated image recognition for the segmentation process. Many projects are invested in this topic, but it still has to be further addressed. Furthermore, the MATLAB scripts have to be refined to be used more efficiently, especially when bigger data sets have to be processed. A possibility would be the export as a standalone executable to enable a wider user base.

As more and more medical diagnostics use color-coded models to highlight certain medical conditions or properties of body parts, and patient-specific medical simulations provide further insights and applications, the 4D printing or use in web-based application of these data sets present great chances. This can be used to improve the abovementioned medical fields and can be used for many use cases such as education, surgery preparation or data sharing across medical institutions.

Bibliography

[1] Rohrer, D. (2019). *Analysis and Definition of Data Formats and Materials for color-coded Components for patient-specific Applications in Medical Technologies*. [Unveröff. Vertiefungsarbeit], Hochschule Luzern Technik & Architektur

[2] Wintermantel, E., Ha, S.-W. (2009). *Medizintechnik. Life Science Engineering* (5. überarbeitete und erweiterte Auflage). Berlin Heidelberg: Springer-Verlag

[3] Göhde, S. C., Ladd, M. E., Papavero, L., Köver, P., Semadeni, M., Wintermantel, E. (2009). Magnetresonanztomographie. In Wintermantel, E., Ha, S.W.. *Medizintechnik. Life Science Engineering* (5. überarbeitete und erweiterte Auflage) (p.1029 - p.1070). Berlin Heidelberg: Springer-Verlag

[4] Wessels, G. (2009). Medizinische Bildgebung. In Wintermantel, E., Ha, S.W.. *Medizintechnik. Life Science Engineering* (5. überarbeitete und erweiterte Auflage) (p.1071 - p.1111). Berlin Heidelberg: Springer-Verlag

[5] Athelogou, M., Schönmeyer, R., Schmidt, G., Schäpe, A., Baatz, M., Binnig G. (2009). Bildanalyse in Medizin und Biologie. In Wintermantel, E., Ha, S.W.. *Medizintechnik. Life Science Engineering* (5. überarbeitete und erweiterte Auflage) (p.1215 - p.1237). Berlin Heidelberg: Springer-Verlag

[6] Bolz, A., Kikillus, N., Moor, C. (2009). Elektrische Phänomene des Körpers und ihre Detektion. In Wintermantel, E., Ha, S.W.. *Medizintechnik. Life Science Engineering* (5. überarbeitete und erweiterte Auflage) (p.1323 - p.1356). Berlin Heidelberg: Springer-Verlag

[7] DICOM Standard (o.D). *Overview.* Retrieved on 03.06.2020 from https://www.dicomstandard.org/about/

[8] DICOM Standard (o.D). *Key Concepts.* Retrieved on 03.06.2020 from https://www.dicomstandard.org/concepts/

[9] Medical Imaging and Technology Alliance (2020). *Standards*. Retrieved on 03.06.2020 from https://www.medicalimaging.org/standards/

[10] DICOM Standard (o.D). *Current Edition.* Retrieved on 03.06.2020 from https://www.dicomstandard.org/current/

[11] DICOM Standard (2020). *DICOM PS3.5 2020b - Data Structures and Encoding.* Retrieved on 03.06.2020 from http://dicom.nema.org/medical/dicom/current/output/pdf/part05.pdf

[12] DICOM Standard (2020). *DICOM PS3.3 2020b – Information Object Definitions.* Retrieved on 03.06.2020 from http://dicom.nema.org/medical/dicom/current/output/pdf/part03.pdf

[13] DICOM Library (2020). *SOPs.* Retrieved on 03.06.2020 from https://www.dicomlibrary.com/dicom/sop/

[14] DICOM Standard (2020). *DICOM PS3.4 2020b – Service Class Specifications.* Retrieved on 03.06.2020 from http://dicom.nema.org/medical/dicom/current/output/pdf/part04.pdf

[15] DICOM Standard (2020). *DICOM PS3.10 2020b - Media Storage and File Format for Media Interchange.* Retrieved on 03.06.2020 from http://dicom.nema.org/medical/dicom/current/output/pdf/part10.pdf

[16] DICOM Standard (2020). *DICOM PS3.1 2020b - Introduction and Overview.* Retrieved on 03.06.2020 from http://dicom.nema.org/medical/dicom/current/output/pdf/part1.pdf

[17] DICOM Standard (2020). *DICOM PS3.12 2020b - Media Formats and Physical Media for Media Interchange.* Retrieved on 03.06.2020 from http://dicom.nema.org/medical/dicom/current/output/pdf/part12.pdf

[18] DICOM Standard (2020). *DICOM PS3.14 2020b – Grayscale Standard Display Function.* Retrieved on 03.06.2020 from http://dicom.nema.org/medical/dicom/current/output/pdf/part14.pdf

[19] Barten, P. G. J. (1992). *Physical model for the Contrast Sensitivity of the human eye.* Proc. SPIE 1666, 57-72

[20] Barten, P.G.J. (1993). *Spatio-temporal model for the Contrast Sensitivity of the human eye and its temporal aspects.* Proc. SPIE 1913-01

[21] Marco Eichelberg (05.04.2005). *What is p-values???.* Retrieved on 03.06.2020 from https://groups.google.com/forum/#!topic/comp.protocols.dicom/p_hk0e7-iwo

[22] DICOM Standard Browser (2020). *Rescale Intercept.* Retrieved on 03.06.2020 from https://dicom.innolitics.com/ciods/ct-image/ct-image/00281052

[23] Brandoli., P. (17.04.2012). *rescale slope and rescale intercept*. Retrieved on 03.06.2020 from https://stackoverflow.com/questions/10193971/rescale-slope-and-rescale-intercept

[24] Flynn, M. (2007). *Digital Image Processing in Radiography*. Retrieved on 03.06.2020 from https://www.aapm.org/meetings/amos2/pdf/29-7999-58461-92.pdf

[25] Abileah, A. (2005). *DICOM Calibration for Medical Displays. A Comparison of Two Methods.* Retrieved on 04.06.2020 from https://medicaldisplaysforless.com/Dome/WP/DICOM_Calibration_for_Medical_Displays.pdf

[26] MathWorks Help Center (2020). *dicomread.* Retrieved on 04.06.2020 from https://ch.mathworks.com/help/images/ref/dicomread.html

[27] Wikipedia (2019). *Hounsfield-Skala. Beispiele*. Retrieved on 04.06.2020 from https://de.wikipedia.org/wiki/Hounsfield-Skala.

[28] Newe, A., Ganslandt, T. (2013). *Simplified Generation of Biomedical 3D Surface Model Data for Embedding into 3D Portable Document Format (PDF) Files for Publication and Education.* Retrieved on 06.06-2020 from https://doi.org/10.1371/journal.pone.0079004

[29] Wikipedia (2020). *Segmentation (Bildverarbeitung).* Retrieved on 06.06.2020 from https://de.wikipedia.org/wiki/Segmentierung_(Bildverarbeitung)

[30] Wikipedia (2019). *Region Growing* Retrieved on 06.06.2020 from https://de.wikipedia.org/wiki/Region_Growing

[31] Marshall, D. (1997). *Region Growing*. Retrieved on 06.06.2020 from https://users.cs.cf.ac.uk/Dave.Marshall/Vision_lecture/node35.html

[32] PerkLab Research (2017). *Whole heart segmentation from cardiac CT in 10 minutes*. Retrieved on 06.06.2020 from https://www.youtube.com/watch?v=BJoIexIvtGo

[33] SimVascular (2017). *SimVascular. The only fully opensource software package providing a complete pipeline from medical image data segmentation to patient specific blood flow simulation and analysis.* Retrieved on 06.06.2020 from https://simvascular.github.io/index.html

[34] SimVascular (2017). *SimVascular. Quick Guide.* Retrieved on 06.06.2020 from https://simvascular.github.io/docsQuickGuide.html

[35] Sourceforge (2013). *ezDICOM*. Retrieved on 06.06.2020 from https://sourceforge.net/projects/ezdicom/

[36] Peacs (2020). *Alvale*. Retrieved on 06.06.2020 from https://peacs.nl/?p=317

[37] Paraview (2018). *ParaView/Data formats. VTK(Visualization ToolKit) files.* Retrieved on 06.06.2020 https://www.paraview.org/Wiki/ParaView/Data_formats#VTK.28Visualization_ToolKit.29_files

[38] van Dam, P.M., Boyle N.G., Laks M.M., Tung R. (2016). *Localization of premature ventricular contractions from the papillary muscles using the standard 12-lead electrocardiogram: a feasibility study using a novel cardiac isochrone positioning system.* Retrieved on 06.06.2020 from https://doi.org/10.1093/europace/euw347

[39] Perez-Alday, E.A., Thomas, J.A., Kabir, M., Sedaghat, G., Rogovoy, N., van Dam, E., van Dam, P.M., Woodward, W., Fuss, C., Ferencik, M., Tereshchenko, L.G. (2018). *Torso geometry reconstruction and body surface electrode localization using three-dimensional photography. Journal of Electrocardiology.* Retrieved on 06.06.2020 from https://doi.org/10.1016/j.jelectrocard.2017.08.035

[40] Misra, S., van Dam, P.M. Chrispin, J., Assis, F., Keramati, A., Kolandaivelu, A., (2018). *Initial validation of a novel ECGI system for localization of premature ventricular contractions and ventricular tachycardia in structurally normal and abnormal hearts. Journal of Electrocardiology.* Retrieved on 06.06.2020 from https://doi.org/10.1016/j.jelectrocard.2018.05.018

[41] Cluitmans, M., Brooks, D.H., MacLeod, R., Dössel, O., Guillem, M.S., van Dam, P.M, Svehlikova, J., He, B., Sapp, J., Wang, L., Bear, L. (2018). *Validation and Opportunities of Electrocardiographic Imaging: From Technical Achievements to Clinical Applications. Frontiers in Physiology.* Retrieved on 06.06.2020 from https://www.sci.utah.edu/publications/Clu2018b/fphys-09-01305.pdf

[42] ECG-Excellence (o.D.). *Electrocardiagram recording*. Retrieved on 06.06.2020 from https://www.ecg-excellence.com/

[43] VIVO – Catheter Precision (2018). *VIVO Overview*. Retrieved on 06.06.2020 from https://www.catheterprecision.com/vivo/

[44] ECGSIM (o.D.). *Introduction*. Retrieved on 06.06.2020 from https://www.ecgsim.org/introduction.php

[45] Vanosdol, Z.. (2018). *Microsoft Community. 256 Character Limit still exists on Windows 10?*. Retrieved on 06.06.2020 from https://answers.microsoft.com/en-us/windows/forum/all/256-character-limit-still-exists-on-windows-10/88c58c92-4835-474b-83c1-79277d55317a

[46] Burns, E. (2019). *Life in the Fastlane. Right Ventricular Outflow Tract (RVOT) Tachycardia.* Retrieved on 06.06.2020 from https://litfl.com/right-ventricular-outflow-tract-rvot-tachycardia/

[47] Mayo Clinic (o.D). *Ventricular tachycardia. Overview.* Retrieved on 06.06.2020 from https://www.mayoclinic.org/diseases-conditions/ventricular-tachycardia/symptoms-causes/syc-20355138

[48] Slideshare (2011). *Recurrent ventricular arrhythmia after cardiac surgery*. Retrieved on 06.06.2020 from https://www.slideshare.net/salah_atta/recurrent-ventricular-arrhythmia-after-cardiac-surgery

[49] MRI Shark (2019). *Slice Thickness*. Retrieved on 06.06.2020 from http://www.mrishark.com/slice-thickness.html

[50] Sprawls, P. (o.D). *Spatial Characteristics of the Magnetic Resonance Image*. Retrieved on 06.06.2020 from http://www.sprawls.org/mripmt/MRI09/index.html

[51] DICOM Standard Browser (2020). *Pixel Spacing Attribute*. Retrieved on 06.06.2020 from https://dicom.innolitics.com/ciods/ct-image/image-plane/00280030

[52] mrimaster.com (o.D). *Bandwidth and Image Quality.* Retrieved on 06.06.2020 from https://mrimaster.com/technique%20bandwidth.html

[53] Lakhani, P., Gray, D.L., Pett, C.R. et al (2018). *Hello World Deep Learning in Medical Imaging*. J Digit Imaging 31, p.283–289. Retrieved on 10.06.2020 from https://doi.org/10.1007/s10278-018-0079-6

[54] Dutta, S. (2020). *A 2020 Guide to Deep Learning for Medical Imaging and the Healthcare Industry*. Retrieved on 10.06.2020 from https://nanonets.com/blog/deep-learning-for-medical-imaging/

[55] Rajchl, M., Ira Ktena. S., Pawlowski, N. (2018*). An Introduction to Biomedical Image Analysis with TensorFlow and DLTK*. Retrieved on 10.06.2020 from https://blog.tensorflow.org/2018/07/an-introduction-to-biomedical-image-analysis-tensorflow-dltk.html

[56] Dutta-Roy, T. (2017). *Medical Image Analysis with Deep Learning — I*. Retrieved on 10.06.2020 from https://medium.com/@taposhdr/medical-image-analysis-with-deep-learning-i-23d518abf531

[57] Patil, S. (2019). *Medical Image Analysis with Deep Learning.* Retrieved on 10.06.2020 from https://towardsdatascience.com/medical-image-analysis-with-deep-learning-9557cad44944

[58] web3D Consortium (2020). *What is X3D?.* Retrieved on 10.06.2020 from https://www.web3d.org/x3d/what-x3d

[59] i.materialise (2020). *Uploaded Models. Analysis*. Retrieved on 10.06.2020 from https://i.materialise.com/en/

[60] Canon Medical Sysems (o.D). *Clinical Gallery. Infinix-i.* Retrieved on 10.06.2020 from https://global.medical.canon/products/angiography/angio_clinicalgallery_parametric_imaging

[61] Singh S., Venkatesh S.K., Wang Z., et al (2015). *Diagnostic performance of magnetic resonance elastography in staging liver fibrosis: a systematic review and meta-analysis of individual participant data*. Clin Gastroenterol Hepatol. doi:10.1016/j.cgh.2014.09.046 Retrieved on 10.06.2020 from https://pubmed.ncbi.nlm.nih.gov/25305349/?from_term=ehman+rl+AND+mayo&from_pos=2

[62] National Institute of Biomedical Imaging and Bioengineering (o.D). *Image Library*. Retrieved on 10.06.2020 from https://www.nibib.nih.gov/news-events/multimedia/image-gallery

[63] Doost, S.N., Ghista, D., Su, B. *et al* (2016). *Heart blood flow simulation: a perspective review.* BioMed Eng OnLine 15**,** 101. Retrieved on 10.06.2020 from https://doi.org/10.1186/s12938-016-0224-8

[64] Morris P.D., Narracott A., von Tengg-Kobligk H., et al (2016). *Computational fluid dynamics modelling in cardiovascular medicine*. Heart ;102:18-28. Retrieved on 10.06.2020 from https://heart.bmj.com/content/102/1/18

[65] Cardiovascular Biomechanics Computation Lab (o.D). *Marsden Lab*. Retrieved on 10.06.2020 from https://cbcl.stanford.edu/

[66] Cardiovascular Biomechanics Computation Lab (o.D). *Mission*. Retrieved on 10.06.2020 https://cbcl.stanford.edu/about/mission

[67] ParaView (o.D). *Welcome to ParaView*. Retrieved on 10.06.2020 from https://www.paraview.org/

[68] Katheterladen (o.D). *Rüsch Silasil Ballonkatheter*. Retrieved on 10.06.2020 from https://www.katheterladen.de/katheter/lange-liegedauer/222/ruesch-silasil-ballonkatheter

[69] MedLexi.de (2019). *Ballonkatheter. Formen, Arten & Typen*. Retrieved on 10.06.2020 from https://medlexi.de/Ballonkatheter

[70] dicardiology.com (2017). *Drug-eluting Balloon is Noninferior to Drug-eluting Stent in Patients With In-stent Restenosis.* Retrieved on 10.06.2020 from https://www.dicardiology.com/article/drug-eluting-balloon-noninferior-drug-eluting-stent-patients-stent-restenosis

[71] Cortona3D.com (o.D). *Cortona3D Viewer Supports VRML 2.0*. Retrieved on 10.06.2020 from https://www.cortona3d.com/en/kb/articles/cortona3d-viewer-supports-vrml-20

[72] Stratasys.com (o.D). *Objet260 Connex User Guide*. p.29. Retrieved on 10.06.2020 from http://articles.stratasys.com/en/articles/2042470-objet260-connex-user-guide

[73] Miller, E. (2014). *Color 3D Printing ANSYS ANSYS Mechanical and Mechanical APDL Results.* Retrieved on 10.06.2020 from http://www.padtinc.com/blog/color-3d-printing-ansys-ansys-mechanical-and-mechanical-apdl-results/

List of Illustrations

Figure 1 - Spin of a proton in a magnetic field [3] 3
Figure 2 - Spherical Coordinate System [3] 4
Figure 3 - Longitudinal relaxation after a 180° pulse with time constant T_1 [3] 5
Figure 4 - T_2 Relaxation [3] 5
Figure 5 - Creation of Layers [3] 6
Figure 6 - T_2 Relaxation Curve of two body entities [3] 7
Figure 7 - Surface reconstruction of a 3D intestinal examination with CA [3] 7
Figure 8 - MRI of an endoscope [3] 8
Figure 9 - Simplified illustration of artifact formation [3] 9
Figure 10 - Principle of quantization [4] 11
Figure 11 - Local resolution / local frequency [4] 12
Figure 12 - Principle of a digital imaging system [4] 12
Figure 13 - Grey scale windowing for image display [4] 13
Figure 14 - Principle of X-ray digitization [4] 14
Figure 15 - From projection image to sectional view [4] 15
Figure 16 - Scan principle [4] 16
Figure 17 - Tissue types and Hounsfield Units [4] 16
Figure 18 - Principle of spiral CT [4] 17
Figure 19 - Segmentation and extraction of the lateral ventricles from magnetic resonance imaging of the head [5] 18
Figure 20 - Illustration of the results of CT image data analysis [5] 19
Figure 21 - Potential field of the heart at the time of the R-spike. The numbers indicate the average amplitudes in mV [6] 20
Figure 22 - Typical timing of the electrocardiogram with the designations introduced by Einthoven [6] 21
Figure 23 - The propagation of excitement in the heart [6] 22
Figure 24 -First ECG device after Einthoven from the year 1911 [6] 23
Figure 25 - Patient with a modern 12-channel ECG device [6] 24
Figure 26 - Output of ECG signals [6] 24
Figure 27 - Overview of DICOM publications [10] 26
Figure 28 - Major Structures of DICOM Information Model [12, p.107] 27
Figure 29 - DICOM Model of the real World [12, p. 111] 28
Figure 30 - DICOM Model of the real world - PRINT [12, p.112] 29
Figure 31 - Model of the Real World for the Purpose of Modality-IS Interface [12, p.114] 29
Figure 32 - DICOM Information Model - PRINT [12, p.112] 30
Figure 33 - DICOM Information Model - RADIOTHERAPY [12, p.113] 30
Figure 34 - Value Representations in a DICOM file [own illustration] 31
Figure 35 - DICOM Data Set and Element Structures [11, p.52] 33
Figure 36 - General Communications Model [16. p.22] 35
Figure 37 - DICOM Media Storage Model [15, p. 26] 36
Figure 38 - File-set and File Format [15, p.31] 36
Figure 39 - Media Supporting a Single File Set [17, p.25] 38
Figure 40 - Media Supporting Multiple File-sets [17, p.26] 38
Figure 41 - The Grayscale Display Function is an element of the image presentation after several modifications to the image have been completed by other elements of the image acquisition and presentation chain [18, p.23) 40
Figure 42 - The conceptual model of a Standardized Display System maps P-Values to Luminance via an intermediate transformation to Digital Driving Levels of an unstandardized Display System [18, p.24] 40

Figure 43 - The Grayscale Standard Display Function presented as logarithm-of luminance versus JND-Index [18, p.29]............ 41
Figure 44 - Grayscale and Color Image Transformation Models [14, p.241] 42
Figure 45 - Generic Image Processes [24] 47
Figure 46 – Grayscale Rendition with a Grayscale VOI LUT [24]............ 47
Figure 47 - Change of Image Display wit LUTs [24] 48
Figure 48 - Process of creating Presentation Values [24] 48
Figure 49 - Workspace container [own illustration]............ 50
Figure 50 - File meta data information [own illustration] 51
Figure 51 - Section of the example 240x240 matrix [own illustration] 51
Figure 52 - DICOM image display with MATLAB [own illustration]............ 52
Figure 53 - Highlighting of Hounsfield Unit values between 300 & 500 [own illustration] 52
Figure 54 - Segmented and classified thighbone [28]............ 53
Figure 55 - Region growing process based on pixels [31] 54
Figure 56 - Drag and drop of one single DICOM image [own illustration]............ 54
Figure 57 - Import of all DICOM images from a directory [own illustration]............ 55
Figure 58 - Access of meta data; copy of data and manual paste to text file [own illustration]............ 55
Figure 59 - Definition of segmented objects and assigning of colors [own illustration] 56
Figure 60 - Own segmentation of heart model (standard file set provided by 3DSlicer) [own illustration] 56
Figure 61 - Tutorial result of segmentation of heart [32] 57
Figure 62 - Own segmentation of heart model (standard file set provided by 3DSlicer) [own illustration] 57
Figure 63 - Segmentation viewer in SimVascular [34] 58
Figure 64 - Screenshot of ezDICOM with image imported [own illustration]............ 59
Figure 65 - Image information display [own illustration] 59
Figure 66 - Definition of segmentation (left) and segmented 3D model (right) [own illustration]....... 60
Figure 67 - Screenshot from ECG Excellence video illustrating use of torso [42] 61
Figure 68 - Process of VIVO ECG [43] 62
Figure 69 - Illustration of the data generation process without VRML conversion [own illustration, 40] 63
Figure 70 - Converted VRML model of Alvale00 with new converter [own illustration]............ 64
Figure 71 - Position of the heart in the torso of the patient from top and side, VRML models [own illustration)] 65
Figure 72 - Position of the heart in the torso of the patient from the front, VRML models [own illustration]. 65
Figure 73 - Excerpt of meta data (not full list) available from DICOM images [own illustration] 67
Figure 74 - Comparison of structures of meta data [own illustration] 67
Figure 75 - Reading the output file with meta data included [own illustration]............ 68
Figure 76 - ECG signals for RVOT tachycardia [46]............ 69
Figure 77 - Ventricular tachycardia heartbeat [47] 69
Figure 78 - ECG signal of recurrent ventricular arrhythmia [48]............ 70
Figure 79 - Case of RVOT PVC, invasive [own illustration] 70
Figure 80 - Case of RVOT PVC, inverse [own illustration] 71
Figure 81 - Recurrent ventricular arrhythmia, invasive to the left, inverse to the right [own illustration] 71
Figure 82 - Effect of slice thickness on final image [49] 72
Figure 83 - The relationship of tissue voxels to image pixels [50] 73
Figure 84 - Example of Pixel Spacing [51] 73
Figure 85 - Lateral view of DSA run of a stroke patient [60] 76

Figure 86 - MR Elastography of a liver to determine tissue firmness [62] 76
Figure 87 - “Comparison of the development of intraventricular flow with and without incorporating valve leaflets” [63] 77
Figure 88 - “A computational fluid dynamics (CFD) model demonstrating the correlation between wall shear stress (WSS) and restenosis in coronary artery disease” [64] 78
Figure 89 - “Computational fluid dynamics (CFD) model of an intracranial berry aneurysm from the @neurist project” [64] 78
Figure 90 - Illustration of a simulation model [66] 79
Figure 91 - Editing of the color code in ParaView for the wall shear stress results [own illustration] . 79
Figure 92 - Import of a STL model derived from segmentation into SimVascular [own illustration].. 80
Figure 93 - Balloon catheter [68] 81
Figure 94 - Spreading of blood vessel with balloon catheter [70]... 81
Figure 95 - Generic tube model serving as a test bed in ANSYS [own illustration]............................ 82
Figure 96 - Left: mesh and force visualization of tube. Right: simulated location [own illustration] .. 82
Figure 97 - Von Mises stress in vascular wall with exaggerated deformation [own illustration] 83
Figure 98 - Addition of APDL command to solver [own illustration] 83
Figure 99 - Display of the exported solution model with Cortona3D Viewer [own illustration].......... 84
Figure 100 - VRML2.0 conversion of ANSYS simulation export [own illustration] 88
Figure 101 - 4D printed models [own illustration].. 90

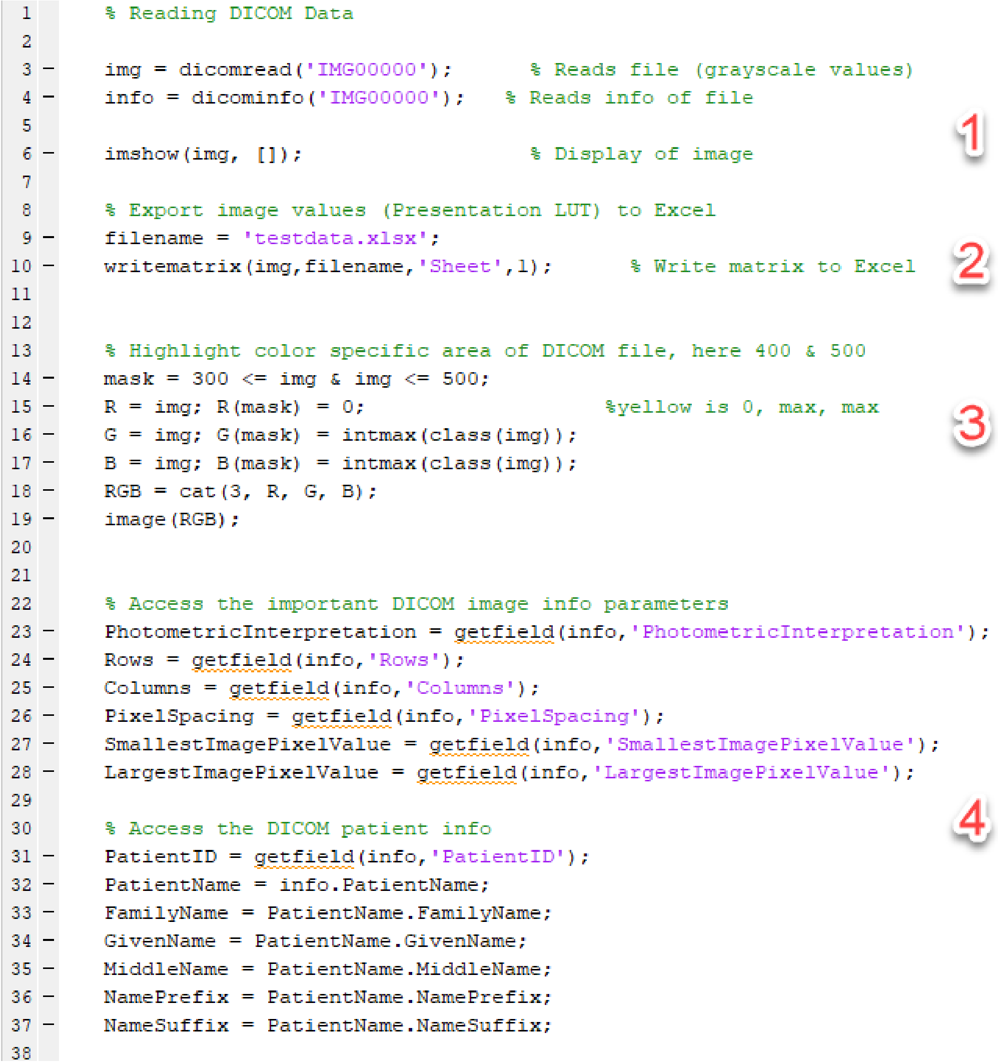
Appendix A

Reading DICOM with MATLAB

Name of scripts: DICOM_Reader.m
ALVALE_VRML_Converter_2.0.m

1. Reading of DICOM image information and meta attributes and store in Workspace
2. Possibility: Export of Presentation LUT to an Excel sheet
3. Possibility: Highlighting of specific grayscale values in image display with MATLAB
4. Accessing of specific DICOM meta attributes to store in Workspace for further processing
5. Access from DICOM image data from a subfolder

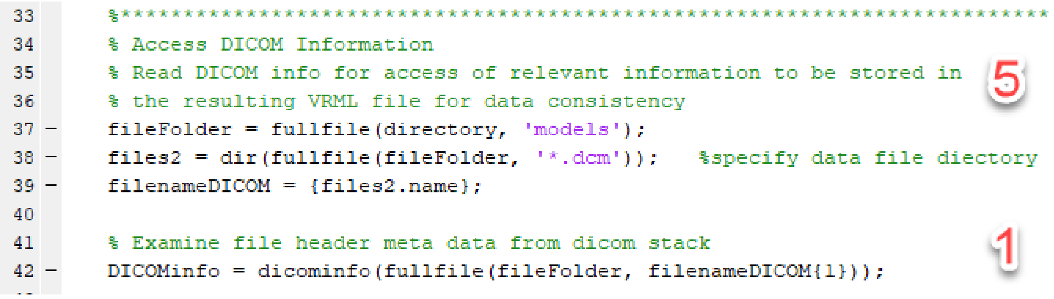
DICOM_Reader.m:



```
% Reading DICOM Data

img = dicomread('IMG00000');       % Reads file (grayscale values)
info = dicominfo('IMG00000');   % Reads info of file

imshow(img, []);                   % Display of image

% Export image values (Presentation LUT) to Excel
filename = 'testdata.xlsx';
writematrix(img,filename,'Sheet',1);      % Write matrix to Excel


% Highlight color specific area of DICOM file, here 400 & 500
mask = 300 <= img & img <= 500;
R = img; R(mask) = 0;                  %yellow is 0, max, max
G = img; G(mask) = intmax(class(img));
B = img; B(mask) = intmax(class(img));
RGB = cat(3, R, G, B);
image(RGB);


% Access the important DICOM image info parameters
PhotometricInterpretation = getfield(info,'PhotometricInterpretation');
Rows = getfield(info,'Rows');
Columns = getfield(info,'Columns');
PixelSpacing = getfield(info,'PixelSpacing');
SmallestImagePixelValue = getfield(info,'SmallestImagePixelValue');
LargestImagePixelValue = getfield(info,'LargestImagePixelValue');

% Access the DICOM patient info
PatientID = getfield(info,'PatientID');
PatientName = info.PatientName;
FamilyName = PatientName.FamilyName;
GivenName = PatientName.GivenName;
MiddleName = PatientName.MiddleName;
NamePrefix = PatientName.NamePrefix;
NameSuffix = PatientName.NameSuffix;

```

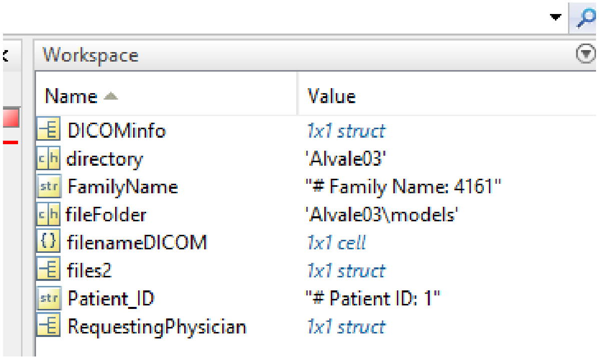
This script shall only show the possibilities of MATLAB to read DICOM images, store the meta attributes and the image information into the Workspace for further use. This can be displaying the image, export the image information or the meta attributes (e.g. in a text file), or implementing into the VRML file format (see Appendix B).

Excerpt from ALVALE_VRML_Converter_2.0.m:



```
%************************************************************************
% Access DICOM Information
% Read DICOM info for access of relevant information to be stored in
% the resulting VRML file for data consistency
fileFolder = fullfile(directory, 'models');
files2 = dir(fullfile(fileFolder, '*.dcm'));   %specify data file diectory
filenameDICOM = {files2.name};

% Examine file header meta data from dicom stack
DICOMinfo = dicominfo(fullfile(fileFolder, filenameDICOM{1}));
```

Workspace with meta attributes read with ALVALE_VRML_Converter_2.0.m:



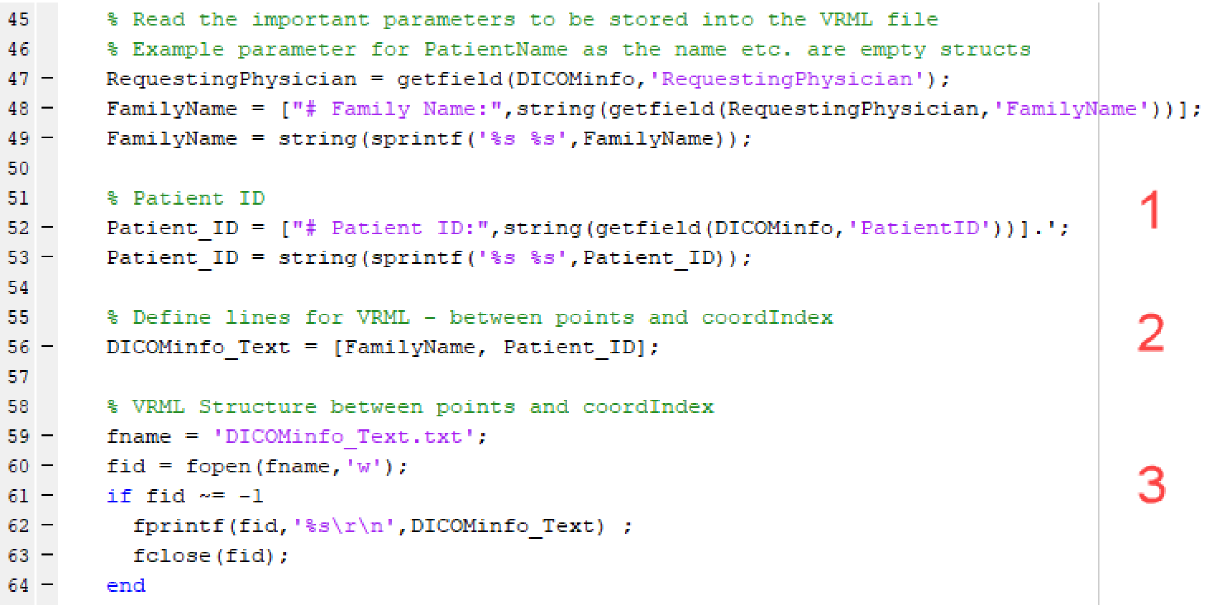
Name	Value
DICOMinfo	1x1 struct
directory	'Alvale03'
FamilyName	"# Family Name: 4161"
fileFolder	'Alvale03\models'
filenameDICOM	1x1 cell
files2	1x1 struct
Patient_ID	"# Patient ID: 1"
RequestingPhysician	1x1 struct

Appendix B

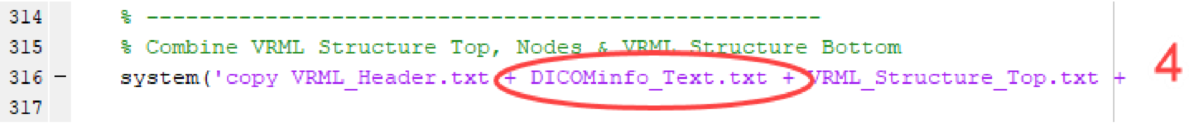
Adding of DICOM Attributes into VRML with MATLAB

Name of Script: ALVALE_VRML_Converter_2.0.m

1. Reading of important attributes from the DICOMinfo structure in the Workspace (see Appendix A), with adding "#" to comment out the lines for VRML syntax
2. Combine those attributes in one Workspace container (here DICOMinfo_Text)
3. Export this container to a text file
4. The VRML file is created as a text file at first while combining all the separate parts of the overall syntax of the file. The DICOMinfo_Text is added right below the VRML header



```
% Read the important parameters to be stored into the VRML file
% Example parameter for PatientName as the name etc. are empty structs
RequestingPhysician = getfield(DICOMinfo,'RequestingPhysician');
FamilyName = ["# Family Name:",string(getfield(RequestingPhysician,'FamilyName'))];
FamilyName = string(sprintf('%s %s',FamilyName));

% Patient ID
Patient_ID = ["# Patient ID:",string(getfield(DICOMinfo,'PatientID'))].';
Patient_ID = string(sprintf('%s %s',Patient_ID));

% Define lines for VRML - between points and coordIndex
DICOMinfo_Text = [FamilyName, Patient_ID];

% VRML Structure between points and coordIndex
fname = 'DICOMinfo_Text.txt';
fid = fopen(fname,'w');
if fid ~= -1
  fprintf(fid,'%s\r\n',DICOMinfo_Text) ;
  fclose(fid);
end
```



```
% ---------------------------------------------------
% Combine VRML Structure Top, Nodes & VRML Structure Bottom
system('copy VRML_Header.txt + DICOMinfo_Text.txt + VRML_Structure_Top.txt +
```

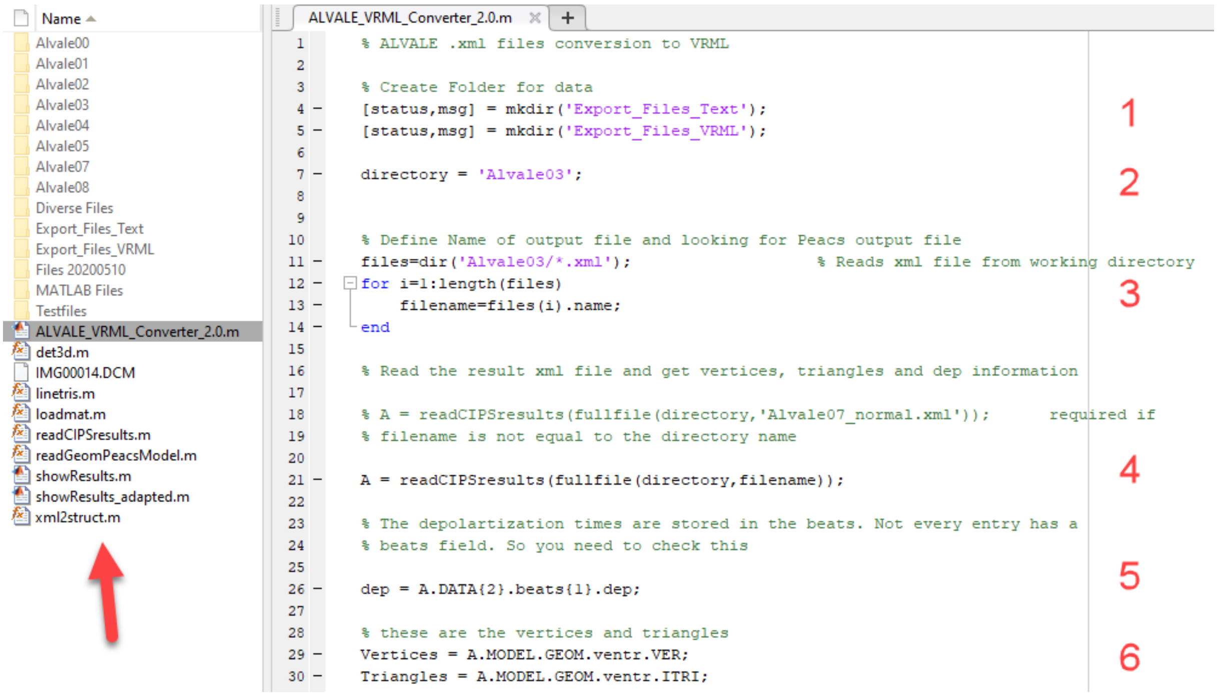
Appendix C

Conversion of Alvale Data Sets to VRML (Part 1)

Name of Script: ALVALE_VRML_Converter_2.0.m

1. Creating directories where the export files are eventually stored if not already created
2. Define the working directory to be accessed with the files
   Note: This has to be adjusted manually for every run if the directory shall change. For a enhanced script without manual adjustments, the overall process from this point to the end could be looped
3. Looking for the Peacs output file in the XML format and define filename
4. Accessing Peacs function "*readCIPSresults*" which reads the data from the XML file
5. Read the timestamp information from the Workspace container "A"
6. Read the vertices and triangles form the Workspace container "A"

General Note: All function files are provided by Peacs and are necessary to read the XML file (see arrow below)


Name
Alvale00
Alvale01
Alvale02
Alvale03
Alvale04
Alvale05
Alvale07
Alvale08
Diverse Files
Export_Files_Text
Export_Files_VRML
Files 20200510
MATLAB Files
Testfiles
ALVALE_VRML_Converter_2.0.m
det3d.m
IMG00014.DCM
linetris.m
loadmat.m
readCIPSresults.m
readGeomPeacsModel.m
showResults.m
showResults_adapted.m
xml2struct.m
ALVALE_VRML_Converter_2.0.m
1
2
3
4
5
6

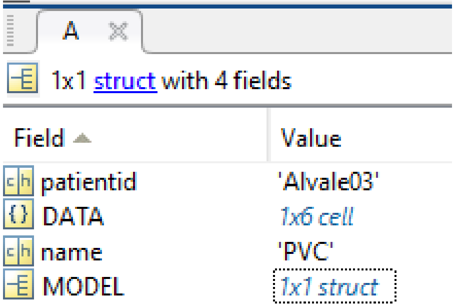
```
% ALVALE .xml files conversion to VRML

% Create Folder for data
[status,msg] = mkdir('Export_Files_Text');
[status,msg] = mkdir('Export_Files_VRML');

directory = 'Alvale03';


% Define Name of output file and looking for Peacs output file
files=dir('Alvale03/*.xml');                    % Reads xml file from working directory
for i=1:length(files)
    filename=files(i).name;
end

% Read the result xml file and get vertices, triangles and dep information

% A = readCIPSresults(fullfile(directory,'Alvale07_normal.xml'));      required if
% filename is not equal to the directory name

A = readCIPSresults(fullfile(directory,filename));

% The depolartization times are stored in the beats. Not every entry has a
% beats field. So you need to check this

dep = A.DATA{2}.beats{1}.dep;

% these are the vertices and triangles
Vertices = A.MODEL.GEOM.ventr.VER;
Triangles = A.MODEL.GEOM.ventr.ITRI;
```

Data loaded from XML file

Name of function: readCIPSresults.m

Only the most relevant data for the conversion to VRML is showed here

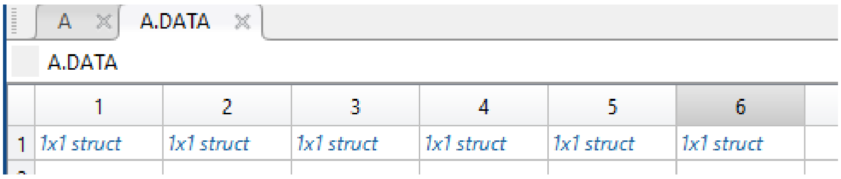
Overall:



1x1 struct with 4 fields

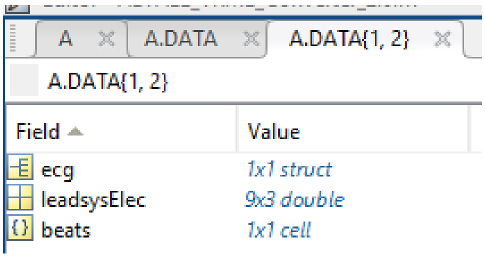
Field	Value
patientid	'Alvale03'
DATA	1x6 cell
name	'PVC'
MODEL	1x1 struct

DATA:



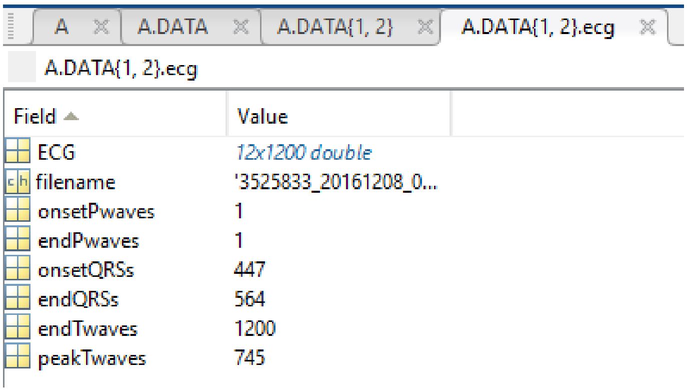
A.DATA

	1	2	3	4	5	6
1	1x1 struct	1x1 struct	1x1 struct	1x1 struct	1x1 struct	1x1 struct


A
A.DATA
A.DATA{1, 2}

A.DATA{1, 2}

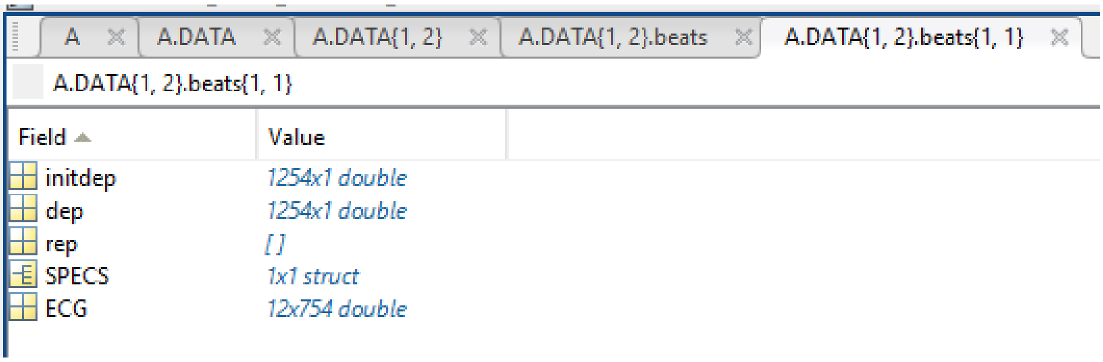
Field	Value
ecg	1x1 struct
leadsysElec	9x3 double
beats	1x1 cell


A
A.DATA
A.DATA{1, 2}
A.DATA{1, 2}.ecg

A.DATA{1, 2}.ecg

Field	Value
ECG	12x1200 double
filename	'3525833_20161208_0...
onsetPwaves	1
endPwaves	1
onsetQRSs	447
endQRSs	564
endTwaves	1200
peakTwaves	745

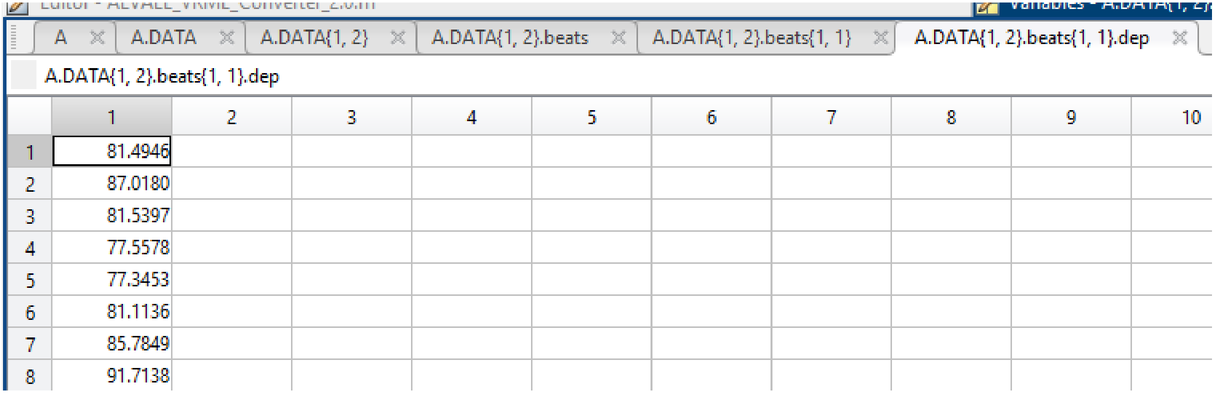
Beats:


A
A.DATA
A.DATA{1, 2}
A.DATA{1, 2}.beats
A.DATA{1, 2}.beats{1, 1}

A.DATA{1, 2}.beats{1, 1}

Field	Value
initdep	1254x1 double
dep	1254x1 double
rep	[ ]
SPECS	1x1 struct
ECG	12x754 double

Timestamp values:

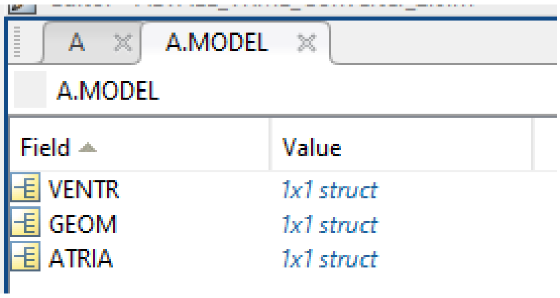

A
A.DATA
A.DATA{1, 2}
A.DATA{1, 2}.beats
A.DATA{1, 2}.beats{1, 1}
A.DATA{1, 2}.beats{1, 1}.dep

A.DATA{1, 2}.beats{1, 1}.dep

	1	2	3	4	5	6	7	8	9	10
1	81.4946									
2	87.0180									
3	81.5397									
4	77.5578									
5	77.3453									
6	81.1136									
7	85.7849									
8	91.7138									

These values are accessed in step 5.

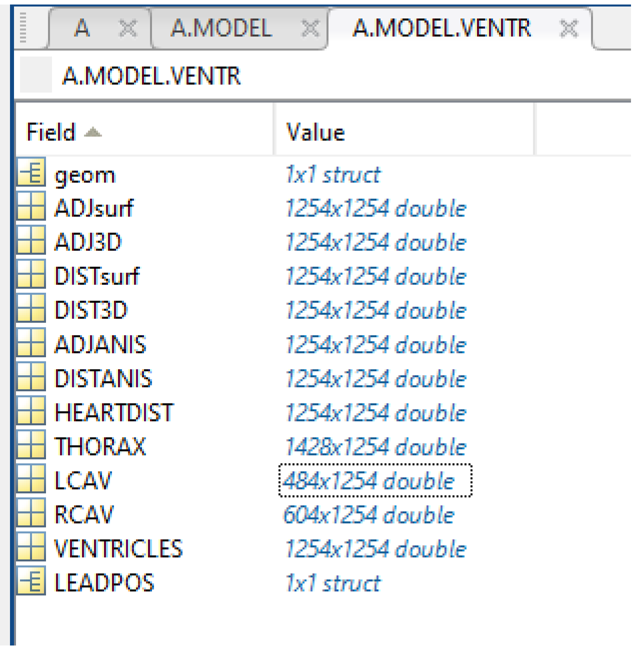
MODEL



A.MODEL

Field	Value
VENTR	1x1 struct
GEOM	1x1 struct
ATRIA	1x1 struct

Ventricles:

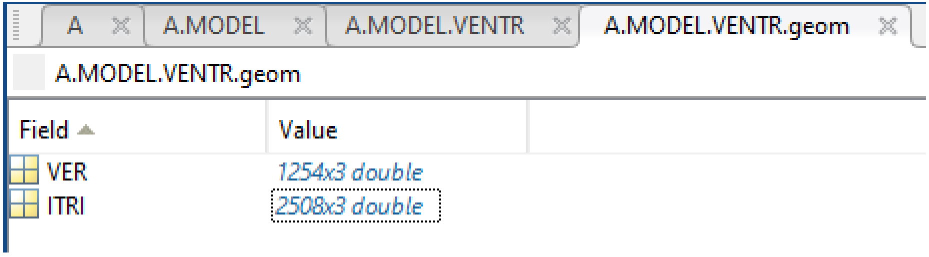


A.MODEL.VENTR

Field	Value
geom	*1x1 struct*
ADJsurf	*1254x1254 double*
ADJ3D	*1254x1254 double*
DISTsurf	*1254x1254 double*
DIST3D	*1254x1254 double*
ADJANIS	*1254x1254 double*
DISTANIS	*1254x1254 double*
HEARTDIST	*1254x1254 double*
THORAX	*1428x1254 double*
LCAV	*484x1254 double*
RCAV	*604x1254 double*
VENTRICLES	*1254x1254 double*
LEADPOS	*1x1 struct*

All geometry information (incl. other geometries such as the thorax scan from the 3D camera) is stored in this field.

geom:



A.MODEL.VENTR.geom

Field	Value
VER	*1254x3 double*
ITRI	*2508x3 double*

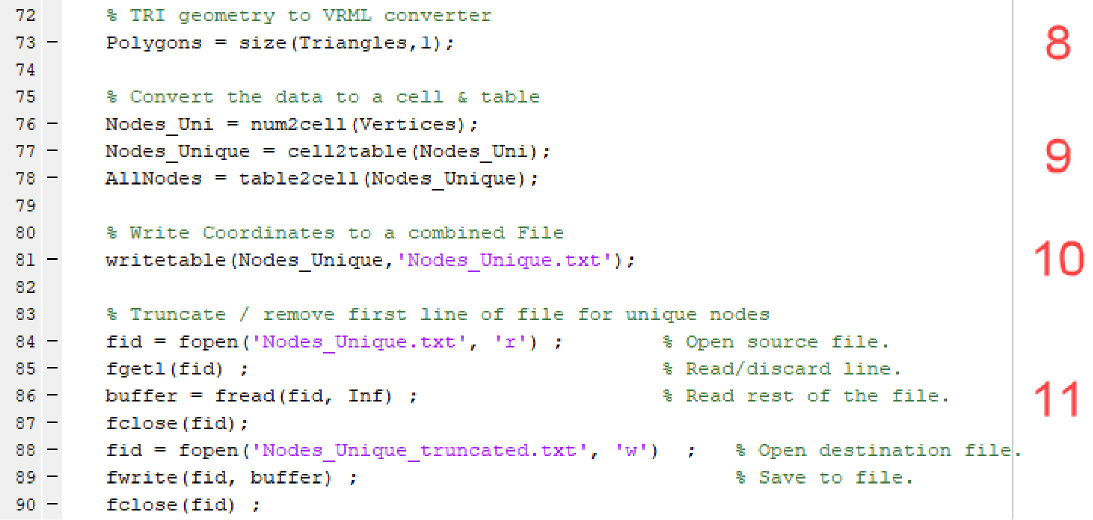
The geometry for the heart and the coordination indexes for the triangles are stored in the geom field. The VER and ITRI fields are accessed in step 6.

Conversion of Alvale Data Sets to VRML (Part 2)

Name of Script: ALVALE_VRML_Converter_2.0.m

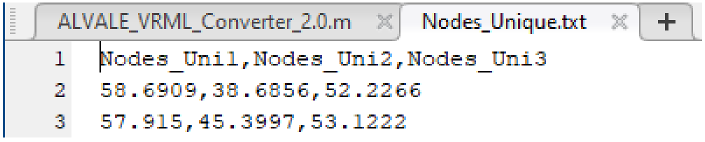
7. Accessing DICOM information (see Appendix B)
8. Get number of polygons from heart geometry
9. Convert array of vertices to exportable array for the TXT format
10. Export vertice array to TXT file
11. Edit TXT file, delete first line as it is not part of a VRML file, and export again
12. Adding placeholder for comma (required for VRML syntax
13. Add comma at placeholder position and export final VRML node text file

General Note: These exports and imports could be avoided with an enhanced script, all of these steps could be done within MATLAB and only the final VRML node structure would be required to export

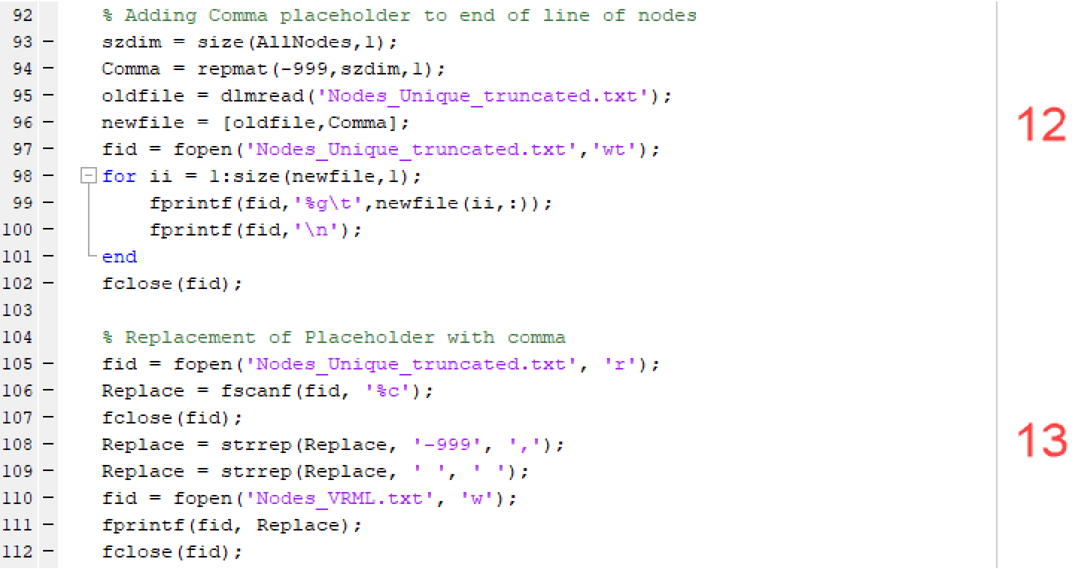


```
% TRI geometry to VRML converter
Polygons = size(Triangles,1);

% Convert the data to a cell & table
Nodes_Uni = num2cell(Vertices);
Nodes_Unique = cell2table(Nodes_Uni);
AllNodes = table2cell(Nodes_Unique);

% Write Coordinates to a combined File
writetable(Nodes_Unique,'Nodes_Unique.txt');

% Truncate / remove first line of file for unique nodes
fid = fopen('Nodes_Unique.txt', 'r') ;        % Open source file.
fgetl(fid) ;                                  % Read/discard line.
buffer = fread(fid, Inf) ;                    % Read rest of the file.
fclose(fid);
fid = fopen('Nodes_Unique_truncated.txt', 'w')  ;  % Open destination file.
fwrite(fid, buffer) ;                              % Save to file.
fclose(fid) ;
```

First line of the first export is not part of a VRML file:



```
Nodes_Uni1,Nodes_Uni2,Nodes_Uni3
58.6909,38.6856,52.2266
57.915,45.3997,53.1222
```

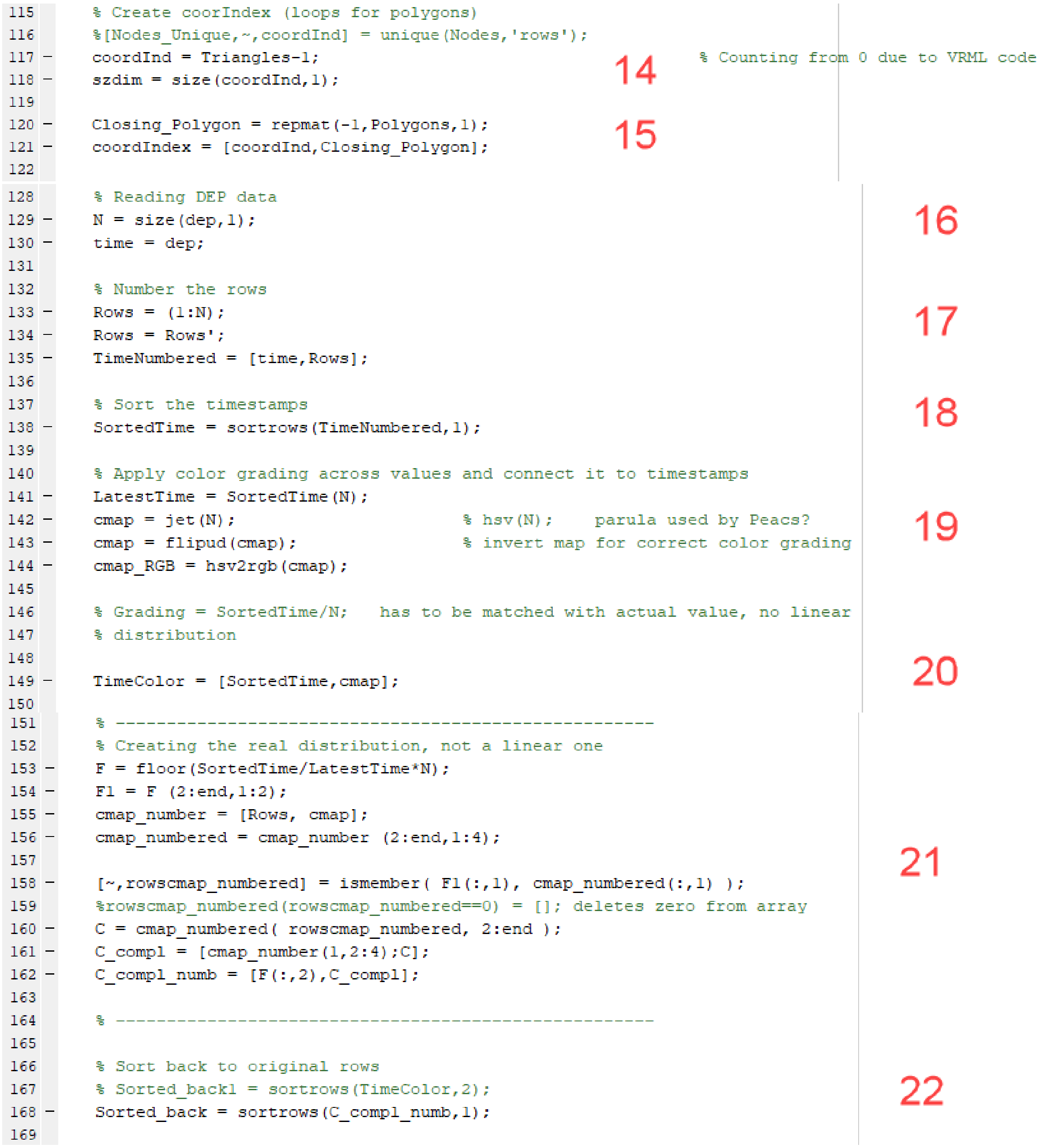


```
% Adding Comma placeholder to end of line of nodes
szdim = size(AllNodes,1);
Comma = repmat(-999,szdim,1);
oldfile = dlmread('Nodes_Unique_truncated.txt');
newfile = [oldfile,Comma];
fid = fopen('Nodes_Unique_truncated.txt','wt');
for ii = 1:size(newfile,1);
    fprintf(fid,'%g\t',newfile(ii,:));
    fprintf(fid,'\n');
end
fclose(fid);

% Replacement of Placeholder with comma
fid = fopen('Nodes_Unique_truncated.txt', 'r');
Replace = fscanf(fid, '%c');
fclose(fid);
Replace = strrep(Replace, '-999', ',');
Replace = strrep(Replace, ' ', ' ');
fid = fopen('Nodes_VRML.txt', 'w');
fprintf(fid, Replace);
fclose(fid);
```

Conversion of Alvale Data Sets to VRML (Part 3)

Name of Script: ALVALE_VRML_Converter_2.0.m

14. Creating the coordination indexes of the VRML format, counting from 0 due to VRML syntax
15. Adding “-1” to the coordination indexes to close the polygon due to VRML syntax
16. Getting the number of timestamps. The dep to time conversion is not necessary if the code below uses the dep variable
17. Indexing of the timestamps
18. Sorting the timestamps according their value (from low to high)
19. Create colormap with the same size as the number of timestamps. Inverting of the colormap to achieve correct linking of values (low = red, high = blue in this case)
20. Combine timestamps and colormap in one array
21. Create true distribution of colormap according the values of the timestamps. The initial colormap is a linear distribution, but the timestamps are not linearly distributed
22. Sort back the color values to the position of the allocated timestamps

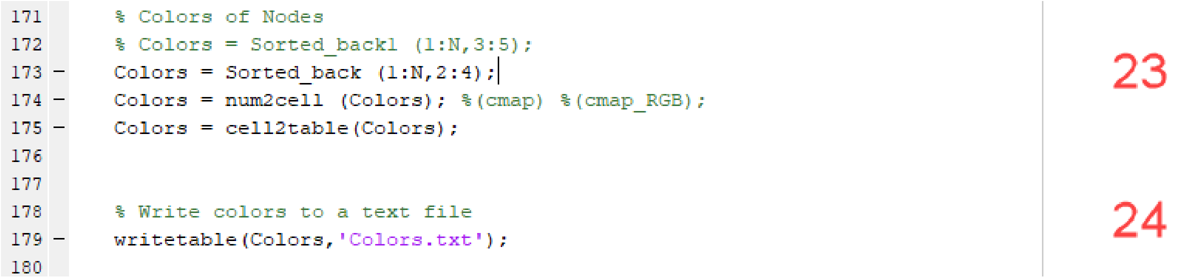


```matlab
% Create coorIndex (loops for polygons)
%[Nodes_Unique,~,coordInd] = unique(Nodes,'rows');
coordInd = Triangles-1;                                    % Counting from 0 due to VRML code
szdim = size(coordInd,1);

Closing_Polygon = repmat(-1,Polygons,1);
coordIndex = [coordInd,Closing_Polygon];

% Reading DEP data
N = size(dep,1);
time = dep;

% Number the rows
Rows = (1:N);
Rows = Rows';
TimeNumbered = [time,Rows];

% Sort the timestamps
SortedTime = sortrows(TimeNumbered,1);

% Apply color grading across values and connect it to timestamps
LatestTime = SortedTime(N);
cmap = jet(N);                      % hsv(N);    parula used by Peacs?
cmap = flipud(cmap);                % invert map for correct color grading
cmap_RGB = hsv2rgb(cmap);

% Grading = SortedTime/N;   has to be matched with actual value, no linear
% distribution

TimeColor = [SortedTime,cmap];

% -----------------------------------------------------
% Creating the real distribution, not a linear one
F = floor(SortedTime/LatestTime*N);
F1 = F (2:end,1:2);
cmap_number = [Rows, cmap];
cmap_numbered = cmap_number (2:end,1:4);

[~,rowscmap_numbered] = ismember( F1(:,1), cmap_numbered(:,1) );
%rowscmap_numbered(rowscmap_numbered==0) = []; deletes zero from array
C = cmap_numbered( rowscmap_numbered, 2:end );
C_compl = [cmap_number(1,2:4);C];
C_compl_numb = [F(:,2),C_compl];

% -----------------------------------------------------

% Sort back to original rows
% Sorted_back1 = sortrows(TimeColor,2);
Sorted_back = sortrows(C_compl_numb,1);
```

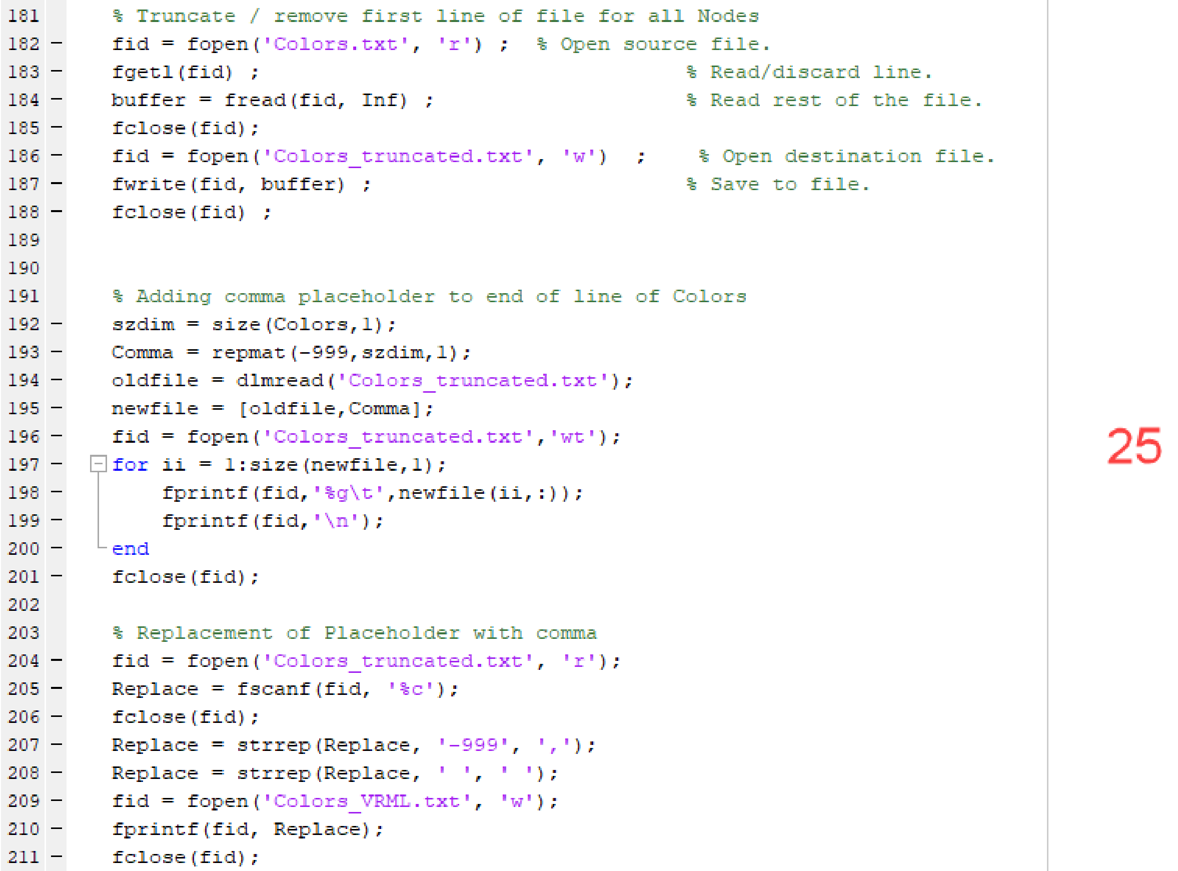
Conversion of Alvale Data Sets to VRML (Part 4)

Name of Script: ALVALE_VRML_Converter_2.0.m

23. Create array only with the relevant colormap values and prepare for export
24. Export array as TXT file
25. Use the same strategy as used between steps 11-13 for creating VRML syntax for color nodes



```
% Colors of Nodes
% Colors = Sorted_back1 (1:N,3:5);
Colors = Sorted_back (1:N,2:4);
Colors = num2cell (Colors); %(cmap) %(cmap_RGB);
Colors = cell2table(Colors);


% Write colors to a text file
writetable(Colors,'Colors.txt');

```

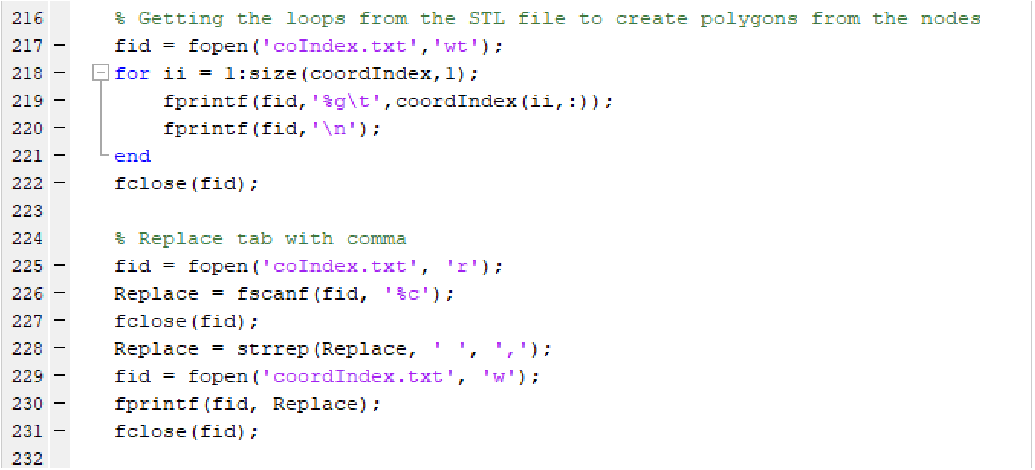


```
% Truncate / remove first line of file for all Nodes
fid = fopen('Colors.txt', 'r') ;  % Open source file.
fgetl(fid) ;                                  % Read/discard line.
buffer = fread(fid, Inf) ;                    % Read rest of the file.
fclose(fid);
fid = fopen('Colors_truncated.txt', 'w')  ;    % Open destination file.
fwrite(fid, buffer) ;                         % Save to file.
fclose(fid) ;


% Adding comma placeholder to end of line of Colors
szdim = size(Colors,1);
Comma = repmat(-999,szdim,1);
oldfile = dlmread('Colors_truncated.txt');
newfile = [oldfile,Comma];
fid = fopen('Colors_truncated.txt','wt');
for ii = 1:size(newfile,1);
    fprintf(fid,'%g\t',newfile(ii,:));
    fprintf(fid,'\n');
end
fclose(fid);

% Replacement of Placeholder with comma
fid = fopen('Colors_truncated.txt', 'r');
Replace = fscanf(fid, '%c');
fclose(fid);
Replace = strrep(Replace, '-999', ',');
Replace = strrep(Replace, ' ', ' ');
fid = fopen('Colors_VRML.txt', 'w');
fprintf(fid, Replace);
fclose(fid);
```

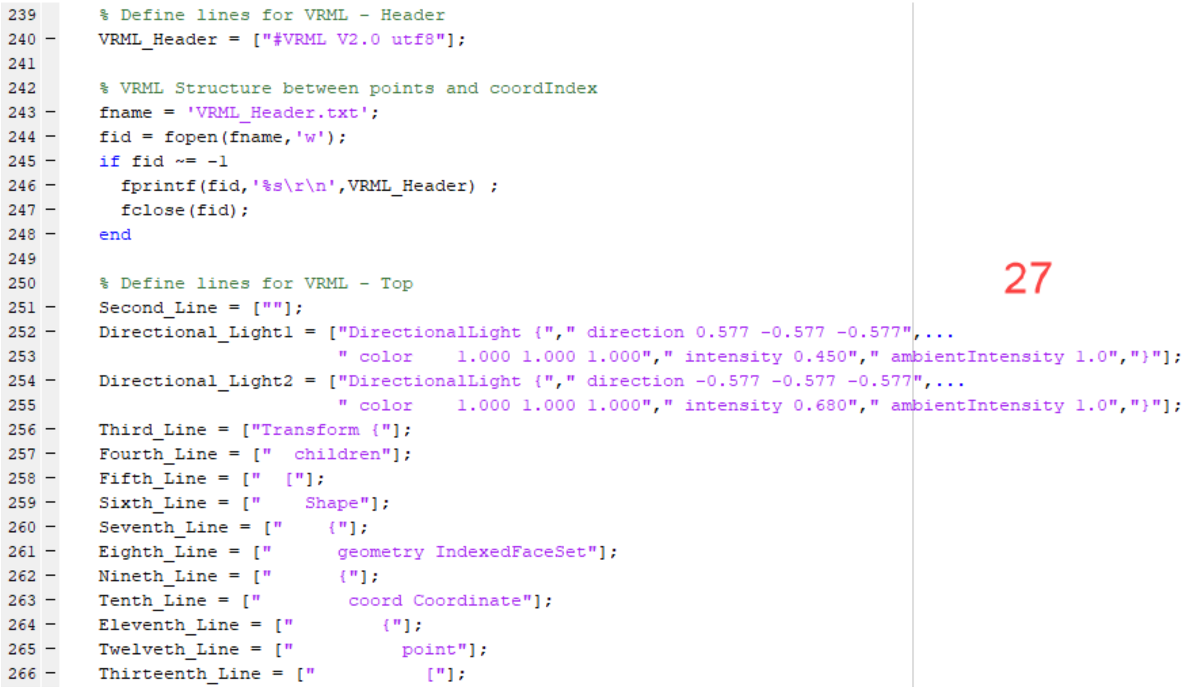
Conversion of Alvale Data Sets to VRML (Part 5)

Name of Script: ALVALE_VRML_Converter_2.0.m

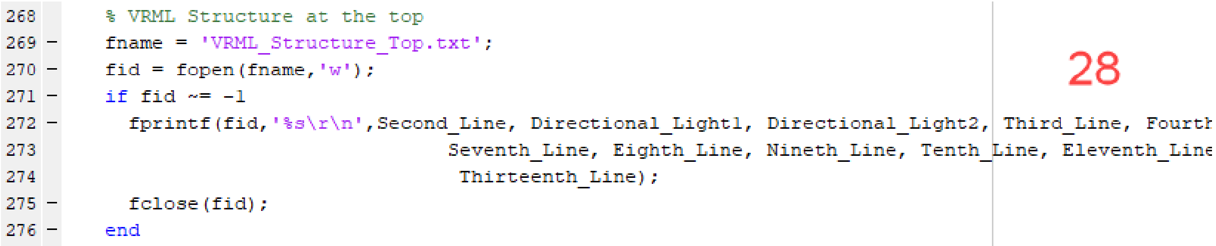
26. Export coordination indexes with correct VRML syntax as TXT file
27. Create VRML syntax at the beginning of the VRML file. Could be done in fewer lines
28. Export this syntax to a TXT file



```
% Getting the loops from the STL file to create polygons from the nodes
fid = fopen('coIndex.txt','wt');
for ii = 1:size(coordIndex,1);
    fprintf(fid,'%g\t',coordIndex(ii,:));
    fprintf(fid,'\n');
end
fclose(fid);

% Replace tab with comma
fid = fopen('coIndex.txt', 'r');
Replace = fscanf(fid, '%c');
fclose(fid);
Replace = strrep(Replace, '	', ',');
fid = fopen('coordIndex.txt', 'w');
fprintf(fid, Replace);
fclose(fid);

```



```
% Define lines for VRML - Header
VRML_Header = ["#VRML V2.0 utf8"];

% VRML Structure between points and coordIndex
fname = 'VRML_Header.txt';
fid = fopen(fname,'w');
if fid ~= -1
  fprintf(fid,'%s\r\n',VRML_Header) ;
  fclose(fid);
end

% Define lines for VRML - Top
Second_Line = [""];
Directional_Light1 = ["DirectionalLight {"," direction 0.577 -0.577 -0.577",...
                      " color    1.000 1.000 1.000"," intensity 0.450"," ambientIntensity 1.0","}"];
Directional_Light2 = ["DirectionalLight {"," direction -0.577 -0.577 -0.577",...
                      " color    1.000 1.000 1.000"," intensity 0.680"," ambientIntensity 1.0","}"];
Third_Line = ["Transform {"];
Fourth_Line = ["  children"];
Fifth_Line = ["  ["];
Sixth_Line = ["    Shape"];
Seventh_Line = ["    {"];
Eighth_Line = ["      geometry IndexedFaceSet"];
Nineth_Line = ["      {"];
Tenth_Line = ["        coord Coordinate"];
Eleventh_Line = ["        {"];
Twelveth_Line = ["          point"];
Thirteenth_Line = ["          ["];
```

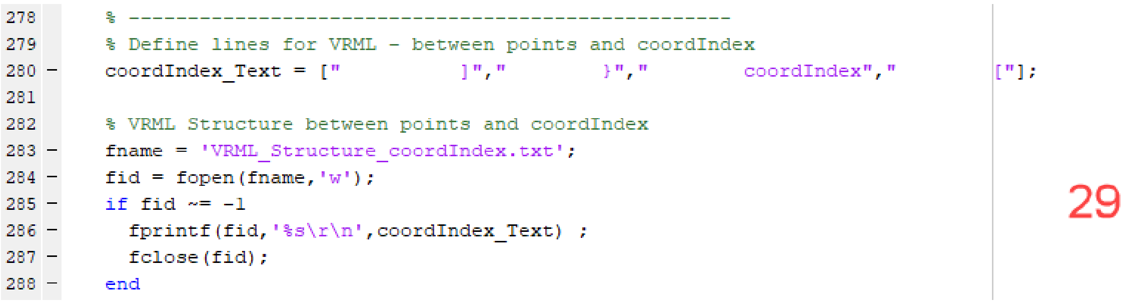


```
% VRML Structure at the top
fname = 'VRML_Structure_Top.txt';
fid = fopen(fname,'w');
if fid ~= -1
  fprintf(fid,'%s\r\n',Second_Line, Directional_Light1, Directional_Light2, Third_Line, Fourth
                              Seventh_Line, Eighth_Line, Nineth_Line, Tenth_Line, Eleventh_Line
                               Thirteenth_Line);
  fclose(fid);
end
```

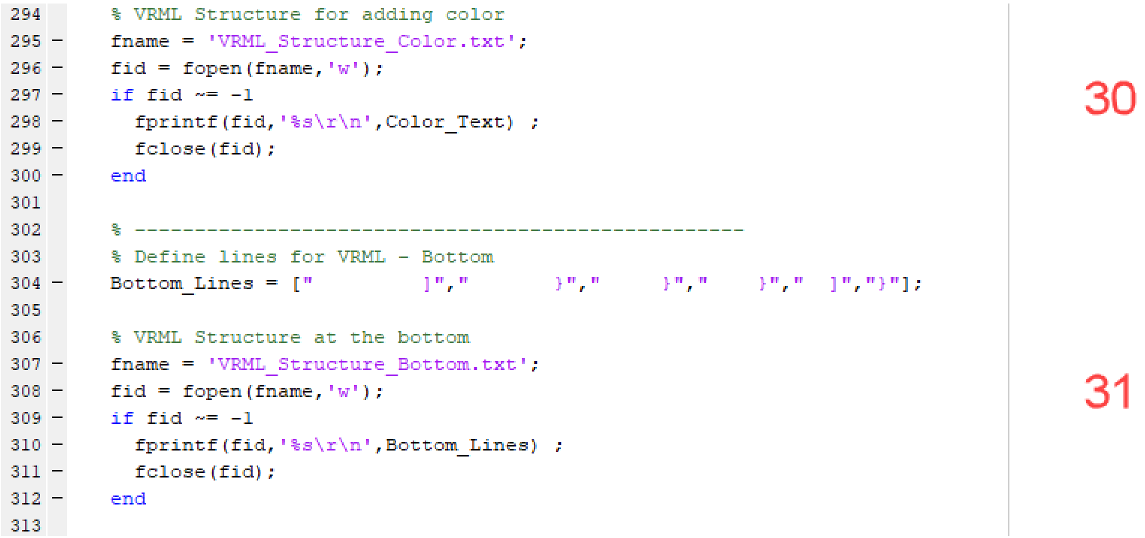
Conversion of Alvale Data Sets to VRML (Part 6)

Name of Script: ALVALE_VRML_Converter_2.0.m

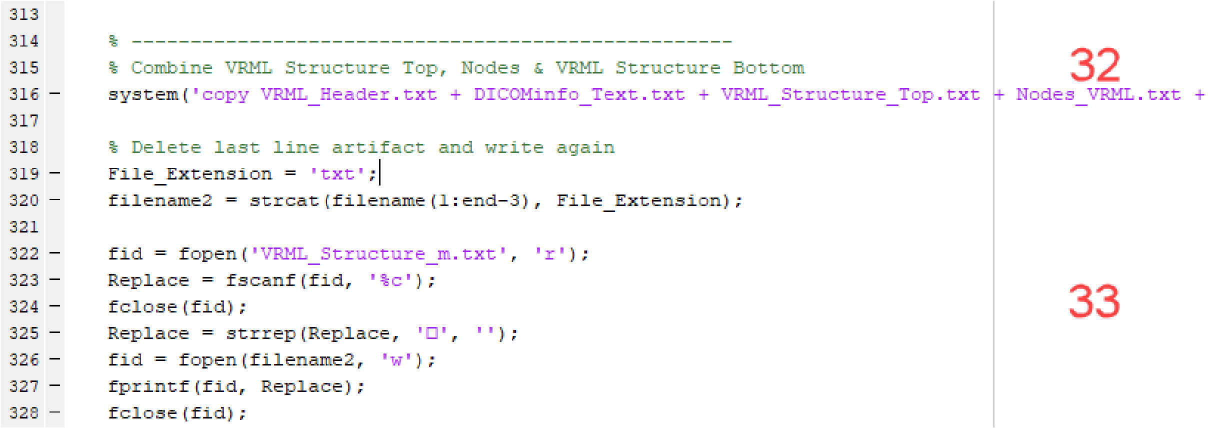
29. Create VRML syntax for coordination index
30. Create VRML syntax for colors
31. Create VRML syntax at the bottom of the VRML file
32. Combine all exported text files to one text file comprising all the data required for the VRML format
33. Delete unwanted artifacts which occur due to the combination of the text files



```
% ---------------------------------------------------
% Define lines for VRML - between points and coordIndex
coordIndex_Text = ["          ]","        }","        coordIndex","        ["];

% VRML Structure between points and coordIndex
fname = 'VRML_Structure_coordIndex.txt';
fid = fopen(fname,'w');
if fid ~= -1
  fprintf(fid,'%s\r\n',coordIndex_Text) ;
  fclose(fid);
end
```



```
% VRML Structure for adding color
fname = 'VRML_Structure_Color.txt';
fid = fopen(fname,'w');
if fid ~= -1
  fprintf(fid,'%s\r\n',Color_Text) ;
  fclose(fid);
end

% ---------------------------------------------------
% Define lines for VRML - Bottom
Bottom_Lines = ["          ]","        }","      }","    }","  ]","}"];

% VRML Structure at the bottom
fname = 'VRML_Structure_Bottom.txt';
fid = fopen(fname,'w');
if fid ~= -1
  fprintf(fid,'%s\r\n',Bottom_Lines) ;
  fclose(fid);
end
```

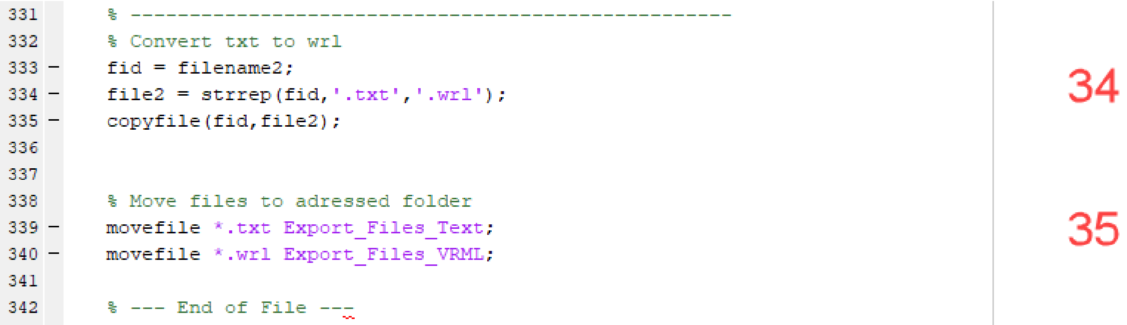


```

% ---------------------------------------------------
% Combine VRML Structure Top, Nodes & VRML Structure Bottom
system('copy VRML_Header.txt + DICOMinfo_Text.txt + VRML_Structure_Top.txt + Nodes_VRML.txt +

% Delete last line artifact and write again
File_Extension = 'txt';
filename2 = strcat(filename(1:end-3), File_Extension);

fid = fopen('VRML_Structure_m.txt', 'r');
Replace = fscanf(fid, '%c');
fclose(fid);
Replace = strrep(Replace, '□', '');
fid = fopen(filename2, 'w');
fprintf(fid, Replace);
fclose(fid);
```

Conversion of Alvale Data Sets to VRML (Part 7)

Name of Script: ALVALE_VRML_Converter_2.0.m

34. Copy text file and exchange the file extension of TXT with WRL
35. Move files to the created folders. The text files are not necessary anymore, could be deleted as well. The VRML file has the same name as the initial XML file from Peacs

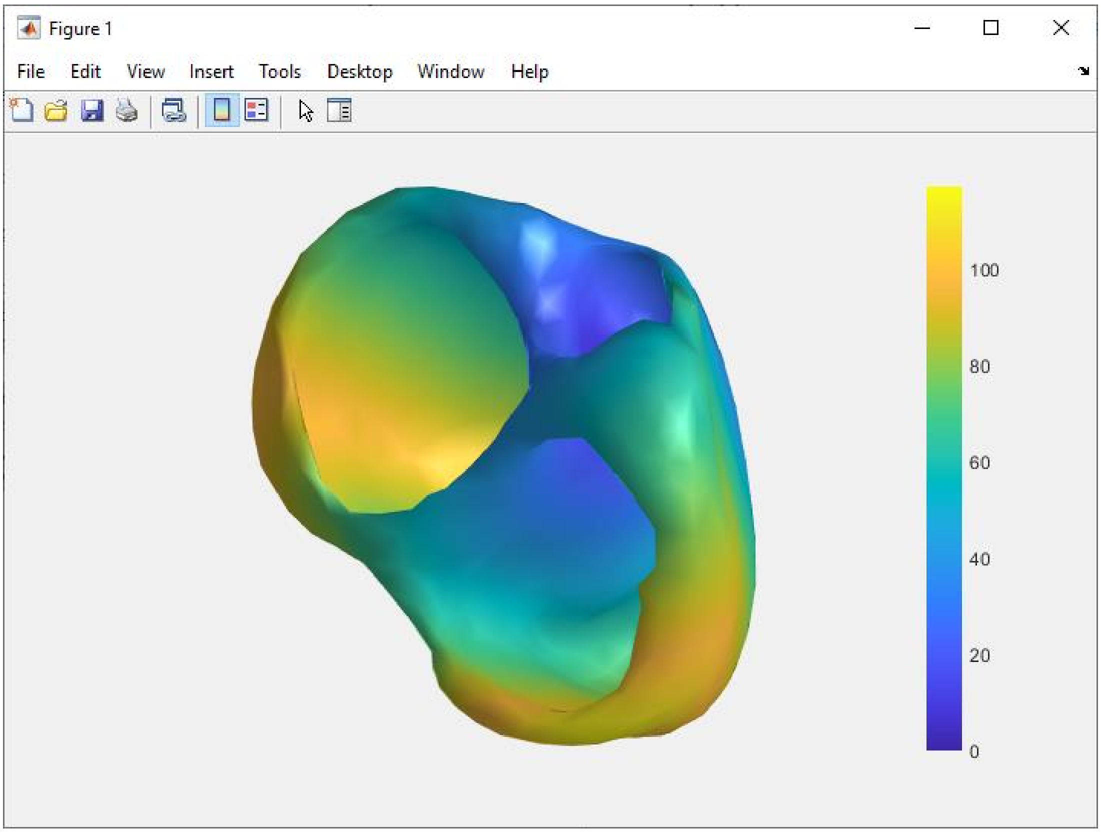


```
% ----------------------------------------------------
% Convert txt to wrl
fid = filename2;
file2 = strrep(fid,'.txt','.wrl');
copyfile(fid,file2);


% Move files to adressed folder
movefile *.txt Export_Files_Text;
movefile *.wrl Export_Files_VRML;

% --- End of File ---
```

Display XML file with MATLAB

Name of script: showResults.m

This script displays the XML file provided by Peacs. Below only the MATLAB display is shown. The colormap is predefined by MATLAB and can be freely adjusted via Edit → Colormap. It can also be used for the validation of the conversion models.



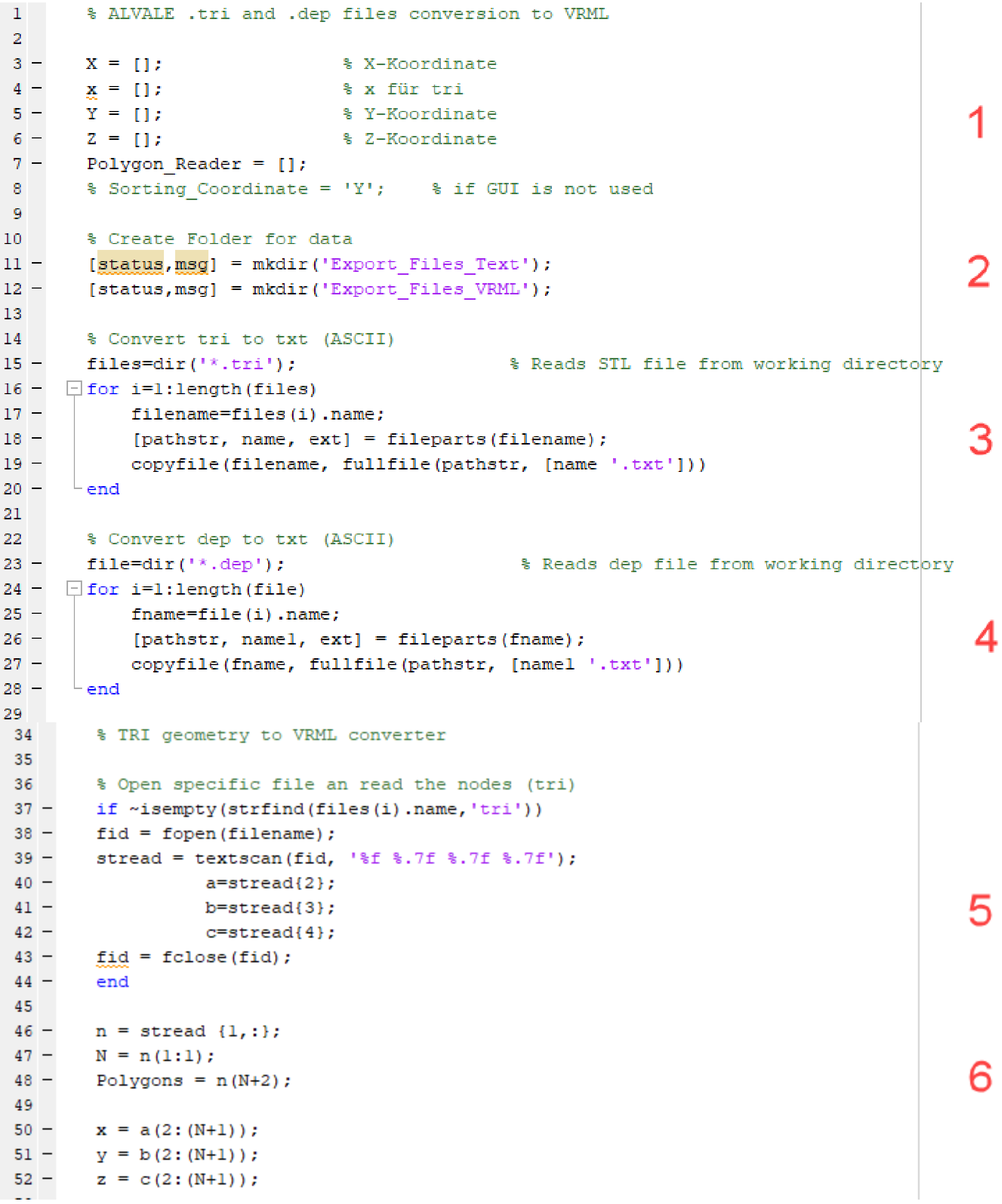
Appendix D

Changes to Original Alvale Script for New Cases (Part 1)

Name of scripts: ALVALE_VRML_Converter_BinaryDep_Invasive.m
ALVALE_VRML_Converter_BinaryDep_Inverse.m

As the initial data sets are not XML but DEP and TRI, the accessing of the data to the workspace had to be changed. Additionally, as the invasive cases do not contain color information for each vertice, these vertices had to be colored by a definition, here white.

1. Creating placeholder variables for geometry readin
2. Create directories for storing the final results and intermediate results
3. Replace .tri file extension with .txt file extension
4. Replace .dep file extension with .txt file extension
5. Read the vertices from the tri file
6. Create coordinates x, y and z with the vertices

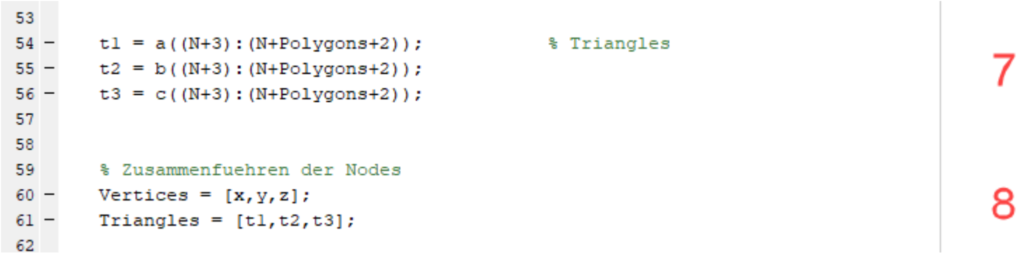


```
% ALVALE .tri and .dep files conversion to VRML

X = [];                % X-Koordinate
x = [];                % x für tri
Y = [];                % Y-Koordinate
Z = [];                % Z-Koordinate
Polygon_Reader = [];
% Sorting_Coordinate = 'Y';    % if GUI is not used

% Create Folder for data
[status,msg] = mkdir('Export_Files_Text');
[status,msg] = mkdir('Export_Files_VRML');

% Convert tri to txt (ASCII)
files=dir('*.tri');                   % Reads STL file from working directory
for i=1:length(files)
    filename=files(i).name;
    [pathstr, name, ext] = fileparts(filename);
    copyfile(filename, fullfile(pathstr, [name '.txt']))
end

% Convert dep to txt (ASCII)
file=dir('*.dep');                     % Reads dep file from working directory
for i=1:length(file)
    fname=file(i).name;
    [pathstr, name1, ext] = fileparts(fname);
    copyfile(fname, fullfile(pathstr, [name1 '.txt']))
end

% TRI geometry to VRML converter

% Open specific file an read the nodes (tri)
if ~isempty(strfind(files(i).name,'tri'))
fid = fopen(filename);
stread = textscan(fid, '%f %.7f %.7f %.7f');
          a=stread{2};
          b=stread{3};
          c=stread{4};
fid = fclose(fid);
end

n = stread {1,:};
N = n(1:1);
Polygons = n(N+2);

x = a(2:(N+1));
y = b(2:(N+1));
z = c(2:(N+1));
```

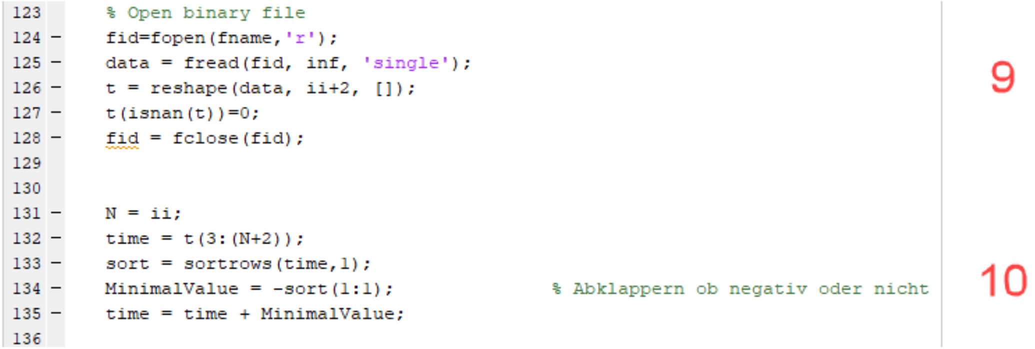
Changes to Original Alvale Script for New Cases (Part 2)

Name of scripts: ALVALE_VRML_Converter_BinaryDep_Invasive.m
ALVALE_VRML_Converter_BinaryDep_Inverse.m

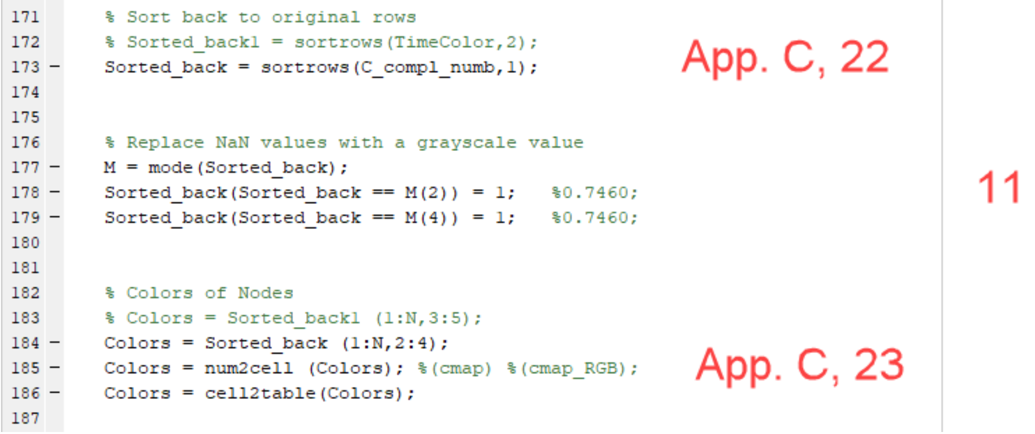
7. Create triangles
8. Combine all coordinates and triangles in one Workspace container each. From here, the same steps 9-15 from Appendix C are applied
9. Open binary file and read it
10. Create timestamps. After that, the inverse file is processed according the steps 17-35 (final step). The invasive file is processed according steps 17-22.
11. Determine empty color nodes (vertices without defined color) as white. From that point on the same process from steps 23-35 (end) from Appendix C are applied



```
t1 = a((N+3):(N+Polygons+2));           % Triangles
t2 = b((N+3):(N+Polygons+2));
t3 = c((N+3):(N+Polygons+2));


% Zusammenfuehren der Nodes
Vertices = [x,y,z];
Triangles = [t1,t2,t3];

```



```
% Open binary file
fid=fopen(fname,'r');
data = fread(fid, inf, 'single');
t = reshape(data, ii+2, []);
t(isnan(t))=0;
fid = fclose(fid);


N = ii;
time = t(3:(N+2));
sort = sortrows(time,1);
MinimalValue = -sort(1:1);              % Abklappern ob negativ oder nicht
time = time + MinimalValue;

```



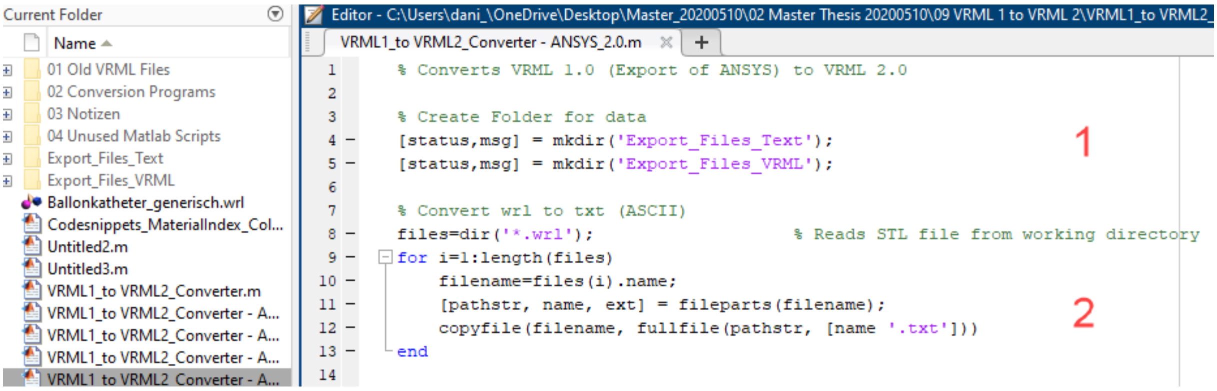
```
% Sort back to original rows
% Sorted_back1 = sortrows(TimeColor,2);
Sorted_back = sortrows(C_compl_numb,1);


% Replace NaN values with a grayscale value
M = mode(Sorted_back);
Sorted_back(Sorted_back == M(2)) = 1;   %0.7460;
Sorted_back(Sorted_back == M(4)) = 1;   %0.7460;


% Colors of Nodes
% Colors = Sorted_back1 (1:N,3:5);
Colors = Sorted_back (1:N,2:4);
Colors = num2cell (Colors); %(cmap) %(cmap_RGB);
Colors = cell2table(Colors);

```

Appendix E

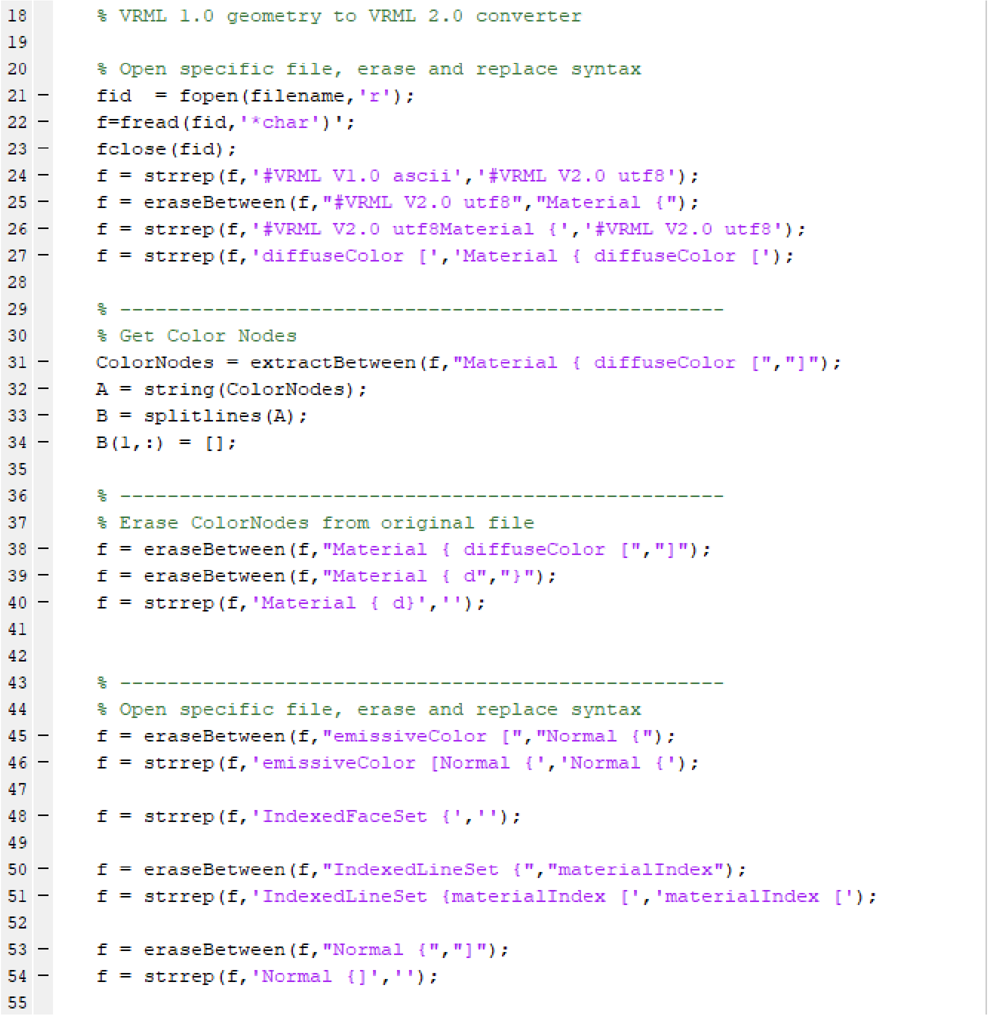
Conversion of VRML 1.0 Export from ANSYS to VRML 2.0 (Part 1)

Name of Script: VRML1_to_VRML2_Converter - ANSYS_2.0.m

1. Creating directories where the export files are eventually stored if not already created
2. Copy VRML file and exchange file extension with TXT. Here the to be converted file has to be in the current folder of MATLAB
3. Erase the unnecessary parts of the VRML1.0 file and replace the syntax with VRML2.0
4. Store all colors from the Material Index (or available colors for the model) in the Workspace


Current Folder
Name
01 Old VRML Files
02 Conversion Programs
03 Notizen
04 Unused Matlab Scripts
Export_Files_Text
Export_Files_VRML
Ballonkatheter_generisch.wrl
Codesnippets_MaterialIndex_Col...
Untitled2.m
Untitled3.m
VRML1_to VRML2_Converter.m
VRML1_to VRML2_Converter - A...
VRML1_to VRML2_Converter - A...
VRML1_to VRML2_Converter - A...
VRML1_to VRML2_Converter - A...
Editor - C:\Users\dani_\OneDrive\Desktop\Master_20200510\02 Master Thesis 20200510\09 VRML 1 to VRML 2\VRML1_to VRML2_
VRML1_to VRML2_Converter - ANSYS_2.0.m
1
2

```
% Converts VRML 1.0 (Export of ANSYS) to VRML 2.0

% Create Folder for data
[status,msg] = mkdir('Export_Files_Text');
[status,msg] = mkdir('Export_Files_VRML');

% Convert wrl to txt (ASCII)
files=dir('*.wrl');                          % Reads STL file from working directory
for i=1:length(files)
    filename=files(i).name;
    [pathstr, name, ext] = fileparts(filename);
    copyfile(filename, fullfile(pathstr, [name '.txt']))
end

```

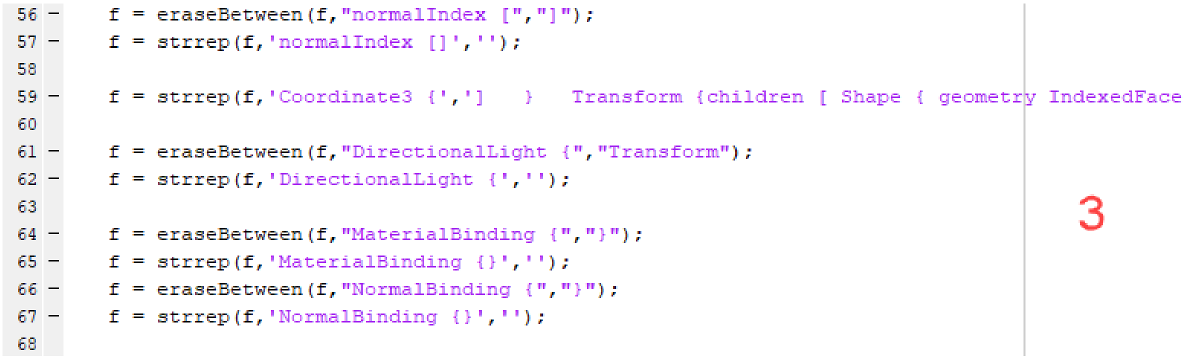


```
% VRML 1.0 geometry to VRML 2.0 converter

% Open specific file, erase and replace syntax
fid  = fopen(filename,'r');
f=fread(fid,'*char')';
fclose(fid);
f = strrep(f,'#VRML V1.0 ascii','#VRML V2.0 utf8');
f = eraseBetween(f,"#VRML V2.0 utf8","Material {");
f = strrep(f,'#VRML V2.0 utf8Material {','#VRML V2.0 utf8');
f = strrep(f,'diffuseColor [','Material { diffuseColor [');

% ---------------------------------------------------
% Get Color Nodes
ColorNodes = extractBetween(f,"Material { diffuseColor [","]");
A = string(ColorNodes);
B = splitlines(A);
B(1,:) = [];

% ---------------------------------------------------
% Erase ColorNodes from original file
f = eraseBetween(f,"Material { diffuseColor [","]");
f = eraseBetween(f,"Material { d","}");
f = strrep(f,'Material { d}','');


% ---------------------------------------------------
% Open specific file, erase and replace syntax
f = eraseBetween(f,"emissiveColor [","Normal {");
f = strrep(f,'emissiveColor [Normal {','Normal {');

f = strrep(f,'IndexedFaceSet {','');

f = eraseBetween(f,"IndexedLineSet {","materialIndex");
f = strrep(f,'IndexedLineSet {materialIndex [','materialIndex [');

f = eraseBetween(f,"Normal {","]");
f = strrep(f,'Normal {]','');

```

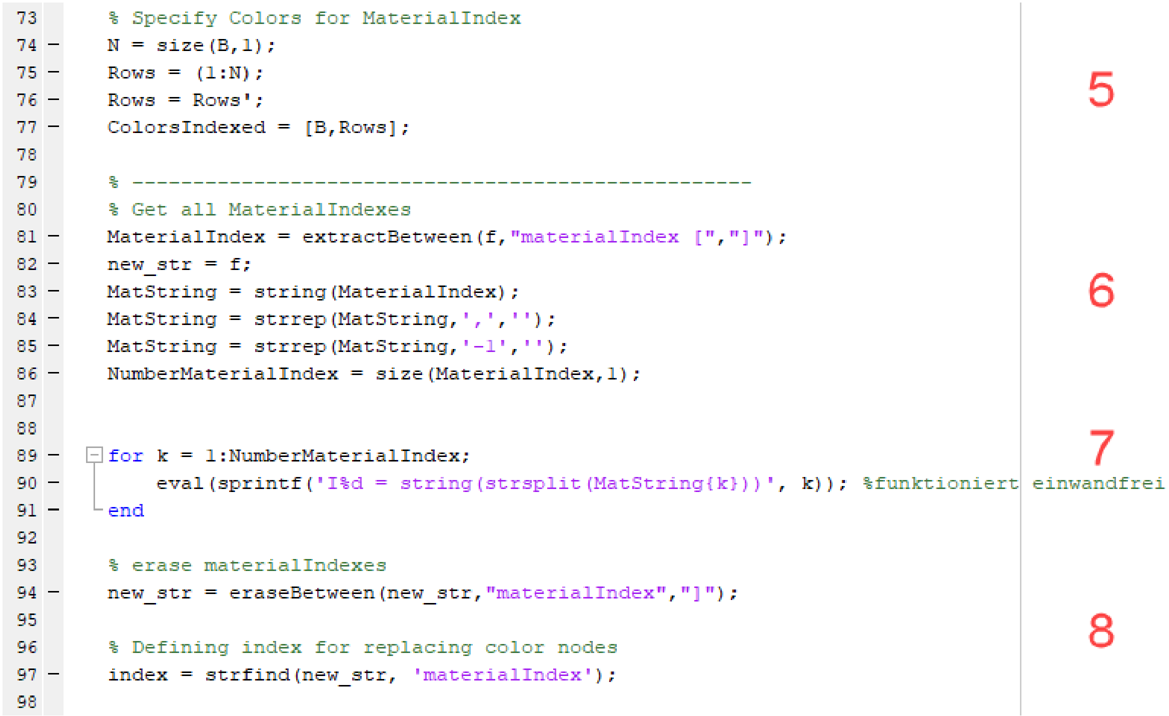
Conversion of VRML 1.0 Export from ANSYS to VRML 2.0 (Part 2)

Name of Script: VRML1_to_VRML2_Converter - ANSYS_2.0.m

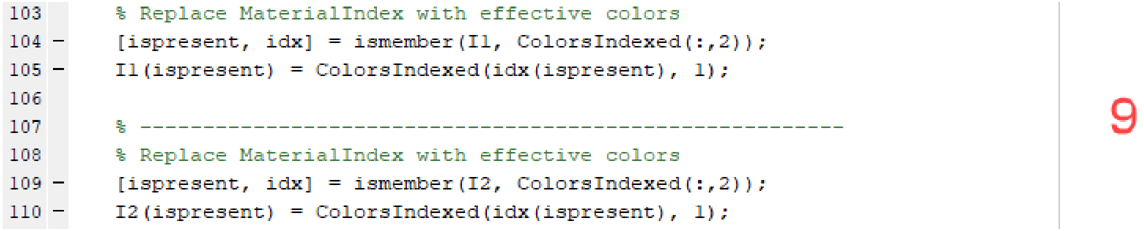
5. Index colors from Material Index
6. Get all the indexed color values assigned to the geometry from the file
7. Separate all indexed color values in separate arrays to access later
8. Search for all the lines in the file where all color assignments start (e.g. line 229768) and store in the Workspace
9. Replace the indexed color values (e.g. 19 for the 1st vertice) with RGB values defined in the Material Index (e.g. 19 defines the RGB value 0 1 0.5). Here every replacement of each color array (here 43) are exchanged separately, which means this script only works if 43 color arrays exist. For more or less color arrays, this has to be adapted manually. A better script could loop this with far less code required.



```
f = eraseBetween(f,"normalIndex [","]");
f = strrep(f,'normalIndex []','');

f = strrep(f,'Coordinate3 {','] 	}   Transform {children [ Shape { geometry IndexedFace

f = eraseBetween(f,"DirectionalLight {","Transform");
f = strrep(f,'DirectionalLight {','');

f = eraseBetween(f,"MaterialBinding {","}");
f = strrep(f,'MaterialBinding {}','');
f = eraseBetween(f,"NormalBinding {","}");
f = strrep(f,'NormalBinding {}','');
```



```
% Specify Colors for MaterialIndex
N = size(B,1);
Rows = (1:N);
Rows = Rows';
ColorsIndexed = [B,Rows];

% -----------------------------------------------------
% Get all MaterialIndexes
MaterialIndex = extractBetween(f,"materialIndex [","]");
new_str = f;
MatString = string(MaterialIndex);
MatString = strrep(MatString,',','');
MatString = strrep(MatString,'-1','');
NumberMaterialIndex = size(MaterialIndex,1);


for k = 1:NumberMaterialIndex;
    eval(sprintf('I%d = string(strsplit(MatString{k}))', k)); %funktioniert einwandfrei
end

% erase materialIndexes
new_str = eraseBetween(new_str,"materialIndex","]");

% Defining index for replacing color nodes
index = strfind(new_str, 'materialIndex');
```

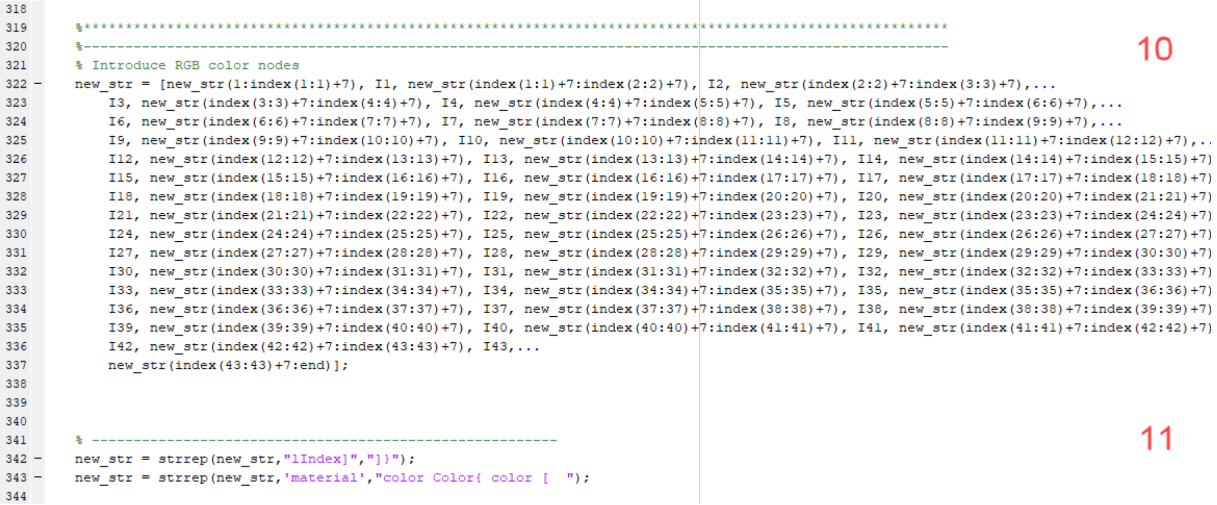


```
% Replace MaterialIndex with effective colors
[ispresent, idx] = ismember(I1, ColorsIndexed(:,2));
I1(ispresent) = ColorsIndexed(idx(ispresent), 1);

% -----------------------------------------------------------
% Replace MaterialIndex with effective colors
[ispresent, idx] = ismember(I2, ColorsIndexed(:,2));
I2(ispresent) = ColorsIndexed(idx(ispresent), 1);
```

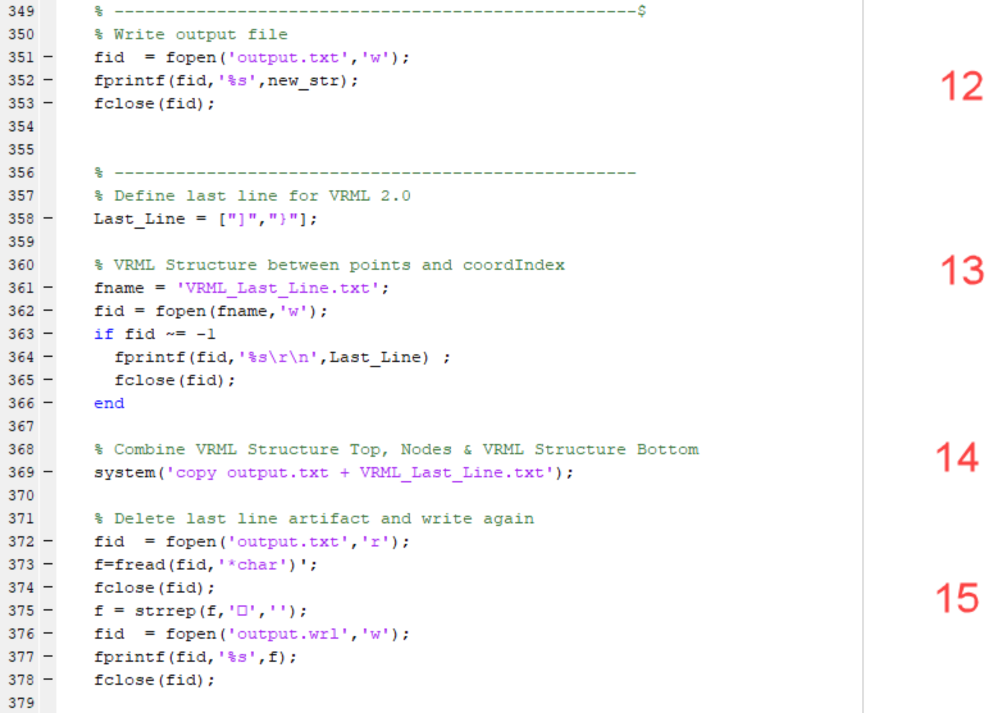
Conversion of VRML 1.0 Export from ANSYS to VRML 2.0 (Part 3)

Name of Script: VRML1_to_VRML2_Converter - ANSYS_2.0.m

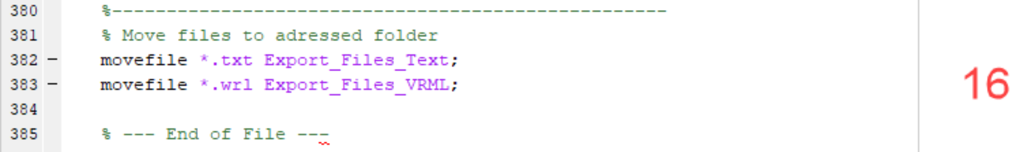
10. The RGB values defined in step 9 are now inserted to the file at the indexed lines defined at step 8. The same issue as with step 9 applies here too. An enhanced script could loop this
11. Update syntax with VRML2.0 syntax
12. Export the adapted file to a text file
13. Create additional required VRML2.0 syntax
14. Combine syntax and output file
15. Delete artifacts due to the combination of the files
16. Move result files to the created folders



```

%************************************************************************************************
%------------------------------------------------------------------------------------------------
% Introduce RGB color nodes
new_str = [new_str(1:index(1:1)+7), I1, new_str(index(1:1)+7:index(2:2)+7), I2, new_str(index(2:2)+7:index(3:3)+7),...
    I3, new_str(index(3:3)+7:index(4:4)+7), I4, new_str(index(4:4)+7:index(5:5)+7), I5, new_str(index(5:5)+7:index(6:6)+7),...
    I6, new_str(index(6:6)+7:index(7:7)+7), I7, new_str(index(7:7)+7:index(8:8)+7), I8, new_str(index(8:8)+7:index(9:9)+7),...
    I9, new_str(index(9:9)+7:index(10:10)+7), I10, new_str(index(10:10)+7:index(11:11)+7), I11, new_str(index(11:11)+7:index(12:12)+7),..
    I12, new_str(index(12:12)+7:index(13:13)+7), I13, new_str(index(13:13)+7:index(14:14)+7), I14, new_str(index(14:14)+7:index(15:15)+7)
    I15, new_str(index(15:15)+7:index(16:16)+7), I16, new_str(index(16:16)+7:index(17:17)+7), I17, new_str(index(17:17)+7:index(18:18)+7)
    I18, new_str(index(18:18)+7:index(19:19)+7), I19, new_str(index(19:19)+7:index(20:20)+7), I20, new_str(index(20:20)+7:index(21:21)+7)
    I21, new_str(index(21:21)+7:index(22:22)+7), I22, new_str(index(22:22)+7:index(23:23)+7), I23, new_str(index(23:23)+7:index(24:24)+7)
    I24, new_str(index(24:24)+7:index(25:25)+7), I25, new_str(index(25:25)+7:index(26:26)+7), I26, new_str(index(26:26)+7:index(27:27)+7)
    I27, new_str(index(27:27)+7:index(28:28)+7), I28, new_str(index(28:28)+7:index(29:29)+7), I29, new_str(index(29:29)+7:index(30:30)+7)
    I30, new_str(index(30:30)+7:index(31:31)+7), I31, new_str(index(31:31)+7:index(32:32)+7), I32, new_str(index(32:32)+7:index(33:33)+7)
    I33, new_str(index(33:33)+7:index(34:34)+7), I34, new_str(index(34:34)+7:index(35:35)+7), I35, new_str(index(35:35)+7:index(36:36)+7)
    I36, new_str(index(36:36)+7:index(37:37)+7), I37, new_str(index(37:37)+7:index(38:38)+7), I38, new_str(index(38:38)+7:index(39:39)+7)
    I39, new_str(index(39:39)+7:index(40:40)+7), I40, new_str(index(40:40)+7:index(41:41)+7), I41, new_str(index(41:41)+7:index(42:42)+7)
    I42, new_str(index(42:42)+7:index(43:43)+7), I43,...
    new_str(index(43:43)+7:end)];



% --------------------------------------------------------
new_str = strrep(new_str,"lIndex]","]}");
new_str = strrep(new_str,'material',"color Color{ color [  ");

```



```
% ---------------------------------------------------$
% Write output file
fid  = fopen('output.txt','w');
fprintf(fid,'%s',new_str);
fclose(fid);


% ---------------------------------------------------
% Define last line for VRML 2.0
Last_Line = ["]","}"];

% VRML Structure between points and coordIndex
fname = 'VRML_Last_Line.txt';
fid = fopen(fname,'w');
if fid ~= -1
  fprintf(fid,'%s\r\n',Last_Line) ;
  fclose(fid);
end

% Combine VRML Structure Top, Nodes & VRML Structure Bottom
system('copy output.txt + VRML_Last_Line.txt');

% Delete last line artifact and write again
fid  = fopen('output.txt','r');
f=fread(fid,'*char')';
fclose(fid);
f = strrep(f,'□','');
fid  = fopen('output.wrl','w');
fprintf(fid,'%s',f);
fclose(fid);

```



```
%---------------------------------------------------
% Move files to adressed folder
movefile *.txt Export_Files_Text;
movefile *.wrl Export_Files_VRML;

% --- End of File ---
```

Appendix F

Investigation of Pixel Spacing and Slice Thickness in Alvale DICOM images

Important Parameters

(0028,0030) PixelSpacing
(0018,0050) SliceThickness

Data Set	No. of data	Folder	Pixel x-axis	Pixel y-axis	Pixel Spacing [mm]	Slice Thickness [mm]
Alvale00	1658	SRS00000	240	240	1.666	8
		SRS00001	240	240	1.666	8
		SRS00002	240	240	1.666	8
		SRS00003	240	240	1.666	8
		SRS00004	240	240	1.666	8
		SRS00005	256	208	1.406	6
		SRS00006	290	320	1.25	5
		SRS00007	256	256	1.563	10
		SRS00008	240	240	2.083	8
		SRS00009	192	224	1.518	6
		SRS00010	192	224	1.518	6
		SRS00020	156	192	1.771	8
		SRS00030	256	208	1.406	8
		SRS00040	192	156	1.771	8
		SRS00050	176	256	1.328	8
		SRS00060	208	256	1.406	8
		SRS00071	256	224	1.328	8
Alvale01	1246	SE00000	240	240	1.666	8
		SE00072	256	224	1.328	8
Alvale02	2183	AAHEART_SCOUT_0006	256	256	1.563	10
		CINE_REALTIME_RVOT_0034	88	128	2.813	8
		CINE_REALTIME_SAX_INLINEVF_0028	128	128	2.961	8
		TI-SCOUT_10_MIN_P_I_SAX_B5_0051	144	192	2.042	8
Alvale03	1519	SRS00000	240	240	1.666	8
		SRS00061	156	192	1.875	8
Alvale04	943	SRS00000	240	240	1.666	8
		SRS00054	192	192	1.875	8
Alvale05	1127	SRS00000	240	240	1.666	8
		SRS00035	192	156	1.771	8
Alvale07	1619	SRS00000	240	240	1.666	8
		SRS00065	192	192	1.875	8
Alvale08	2120	SRS00014	240	240	1.666	8
		SRS02133	256	256	1.523	8

Average:	1.706060606	7.848484848
Median:	1.666	8