



MEDIZINISCHE UNIVERSITÄT  
INNSBRUCK

**Leopold Franzens Universität  
Innsbruck**

Institut für Informatik

**Medizinische Universität  
Innsbruck**

Universitätsklinik für HNO

# BlueEye

## RINGEL

3D Darstellung der Trachea  
mit dem Marching Cubes Algorithmus

Bakkalaureatsarbeit

Verfasser: Abdullatif Yesil  
0118704

Betreuer: Prof. Dr. Wolfgang Freysinger  
Universitätsklinik für Hals-, Nasen- und Ohrenheilkunde  
Medizinische Universität Innsbruck

Innsbruck, 8. Juli 2009  
Diese Dokumentation ist verfügbar unter:  
<http://blueeye.yesil.at>

## Inhaltsverzeichnis

1.	Einleitung .....	5
1.1.	Projektbeschreibung .....	5
2.	Grundlagen.....	6
2.1.	CT Bilder (Computertomographie) .....	6
2.2.	DICOM Format .....	7
2.3.	Hounsfield Skala .....	7
3.	Material und Methoden.....	8
3.1.	C++ .....	8
3.2.	CMAKE.....	8
3.3.	FLTK.....	9
3.4.	VTK .....	11
3.5.	VTK Methoden .....	13
3.5.1.	vtkDICOMImageReader.....	13
3.5.2.	vtkRenderer .....	13
3.5.3.	vtkRenderWindow .....	13
3.5.4.	vtkActor.....	13
3.5.5.	vtkLookupTable .....	14
3.5.6.	vtkImageMapToColors .....	14
3.5.7.	vtkCamera .....	14
3.5.8.	vtkMarchingCubes .....	14
3.5.9.	vtkMapper.....	16
3.5.10.	vtkWindowToImageFilter .....	16
3.5.11.	vtkPointPicker.....	16
3.6.	vtkFIWindowInteractor .....	16
3.7.	Modularer Aufbau .....	17

4.	Diagramme .....	18
4.1.	Ablaufdiagramm.....	18
4.2.	Use-Case Diagramm .....	20
4.3.	Funktionen und Testdrehbuch .....	22
4.3.1.	Programm starten .....	22
4.3.2.	Hilfefenster öffnen.....	23
4.3.3.	Bilder laden.....	24
4.3.4.	Grauwert ändern.....	25
4.3.5.	Bilder blättern.....	26
4.3.6.	Bilder automatisch blättern/Punkt selektieren .....	27
4.3.7.	Bilder zoomen.....	28
4.3.8.	Bilder drehen.....	29
4.3.9.	Bilder bewegen .....	30
4.3.10.	Bild-Informationen öffnen .....	30
4.3.11.	Opazität ändern .....	31
4.3.12.	Sättigung (Farbe) ändern.....	32
4.3.13.	3D-Objekt öffnen .....	33
4.3.14.	3D-Objekt im vordefiniertem HUs öffnen.....	34
4.3.15.	Bildschirmfoto speichern .....	35
4.3.16.	Größere Ansicht öffnen .....	36
4.3.17.	Film Ansicht .....	37
4.3.18.	Bilder entfernen .....	41
4.3.19.	Bilder erneut landen .....	42
4.3.20.	3D Modell speichern .....	43
4.4.	Klassendiagramm.....	44

5.	Benutzerhandbuch.....	45
5.1.	Installation .....	45
5.1.1.	Installation der Bibliotheken .....	45
5.1.2.	Programm installieren.....	48
5.2.	Programm starten .....	49
5.3.	Bilder einlesen und anzeigen.....	50
5.4.	Bilder entfernen .....	51
5.5.	Bilder neu laden.....	51
5.6.	Hilfefenster öffnen .....	51
5.7.	Tools .....	52
5.8.	3D-Ansicht öffnen .....	53
5.9.	Darstellung der Trachea .....	54
5.10.	Bilder größer darstellen.....	55
5.11.	3D-Modell speichern .....	57
5.12.	Historie Fenster.....	57
5.13.	Bild Informationen Fenstern .....	57
5.14.	Mausfunktionen.....	58
5.14.1.	Linke Maustaste .....	58
5.14.2.	Rechte Maustaste.....	59
5.14.3.	Mittlere oder linke + rechte Taste.....	59
5.15.	Programm beenden .....	59
6.	Danksagung .....	60
7.	CD Anhang Dateien .....	61
8.	Lebenslauf.....	63
9.	Literaturverzeichnis .....	64

## 1. Einleitung

### 1.1. Projektbeschreibung

Die Aufgabe dieser Bakkalaureatsarbeit ist darzustellen, wie Computertomographie Bilder eingelesen und mit dem „Marching Cubes“<sup>1</sup> Algorithmus als 3D-Bild der Trachea segmentiert werden und dann als 3D-Modell (als vtkDaten<sup>2</sup>) gespeichert werden.

Dabei sollen über visualisierende 2D oder segmentierte 3D-Modelle verschiedene Verarbeitungen durchgeführt werden. Als Grundlage dienen reale CT-Bilder, in denen die Knochen- oder Weichgewebestrukturen zunächst identifiziert und über die Hounsfield-Skala<sup>3</sup> (HNU-Skala) als 2D Bilder in medizinische dargestellt werden.

Hierfür wird ein Programm entwickelt, das die CT-Bilder einlesen kann und dem Benutzer die erwähnten Aufgaben ermöglicht. Da es sich bei diesem Programm um ein Visualisierung-, Verarbeitungsprogramm mit HNU-Skala Tabelle (auf Seite 7) und 3D-Darstellung der Trachea mit *Marching Cube* Algorithmus von CT-Bildern handelt, und heißt es „BlueEye“.

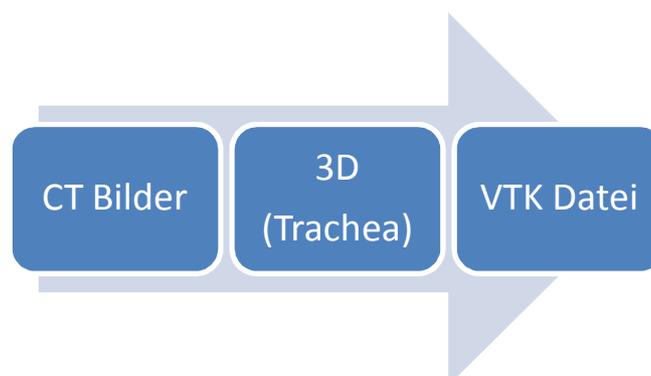


Abbildung 1: Die Aufgabe des Programms ist:

CT-Bilder einlesen, 3D-Bild segmentieren, darstellen und das segmentierte Bild als VTK-Datei zu speichern.

## 2. Grundlagen

### 2.1. CT Bilder (Computertomographie)

Computertomographie (CT)<sup>4</sup> ist eine rechnerbasierte Analyse von verschiedenen Richtungen aufgenommenen Röntgenaufnahmen eines Objektes, um ein dreidimensionales Bild zu erzeugen.

Der Computertomographie besteht aus einer rotierenden „Röntgenröhre“, in die der Patient an der zu untersuchenden Stelle hineingebracht wird. Die Abbildung 1 gibt einen Überblick.

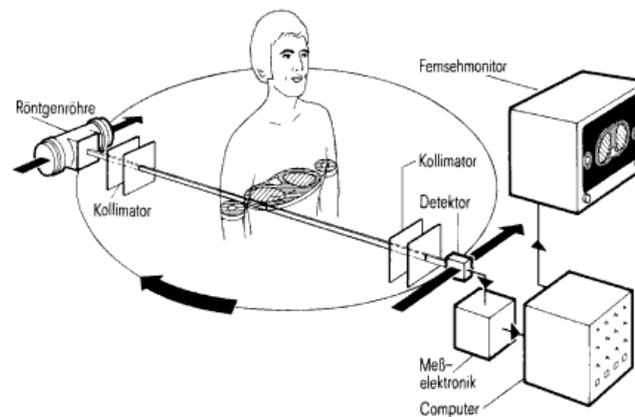


Abbildung 2: Überblick der CT<sup>5</sup>. Die Röntgenröhre und der Detektor mechanisch miteinander verbunden. Sie arbeiten mit einem einzelnen Röntgenstrahl, die Aufnahmen entstehen durch eine Verschiebe- und Drehbewegung (Translation-Rotations-Scanner). Bei den Rotate-Rotate-Geräten wird ein Fächer von Strahlen ausgesendet und von einem Kreissegment von Detektorzellen aufgenommen. Dadurch entfällt die Verschiebebewegung<sup>6</sup>.

Die Computertomographie liefert dabei nicht nur einfache Schattenbilder (wie beim Röntgenbild), sondern ein Schnittbild des entsprechenden Körperteils. Die einzelnen Schnitte entstehen beim CT durch das Prinzip, dass der gewünschte Körperbereich einmal rundherum durchleuchtet wird. Die Bilder werden digital im Computer dargestellt. Flächen mit hoher Dichte (Knochen und Kontrastmittel) werden weiß, Flächen mit geringer Dichte (z.B. Weichteile) werden grau oder schwarz dargestellt. Das Ergebnis kann dann zwei- oder dreidimensional dargestellt werden. Das Bild entspricht einem anatomischen Querschnitt.<sup>7</sup>

## 2.2. DICOM Format

DICOM<sup>8</sup> (**D**igital **I**maging and **C**ommunications in **M**edicine) ist ein offener Standard zum Austausch von Informationen in der Medizin. Diese Informationen können beispielsweise digitale Bilder, Zusatzinformationen wie Segmentierungen (Bildverarbeitung), Oberflächendefinitionen oder Bildregistrierungen sein.

## 2.3. Hounsfield Skala

Die Einheit der CT-Zahl heißt Hounsfield-Einheit (HU). diese Zahl ist sehr geeignet, da die HU-Werte der meisten körpereigenen Substanzen sich nur wenig vom HU-Wert des Wassers unterscheiden. Mit der Einheit wird die Abweichung in Promille von Wasser ausgedrückt. Die Hounsfield Skala beginnt bei -1000 für Luft und endet obwohl sie eigentlich im Prinzip nach oben offen ist, bei +3000 für Knochen. Die Lage der übrigen Gewebe auf der Skala ist in der Abbildung 3 dargestellt.

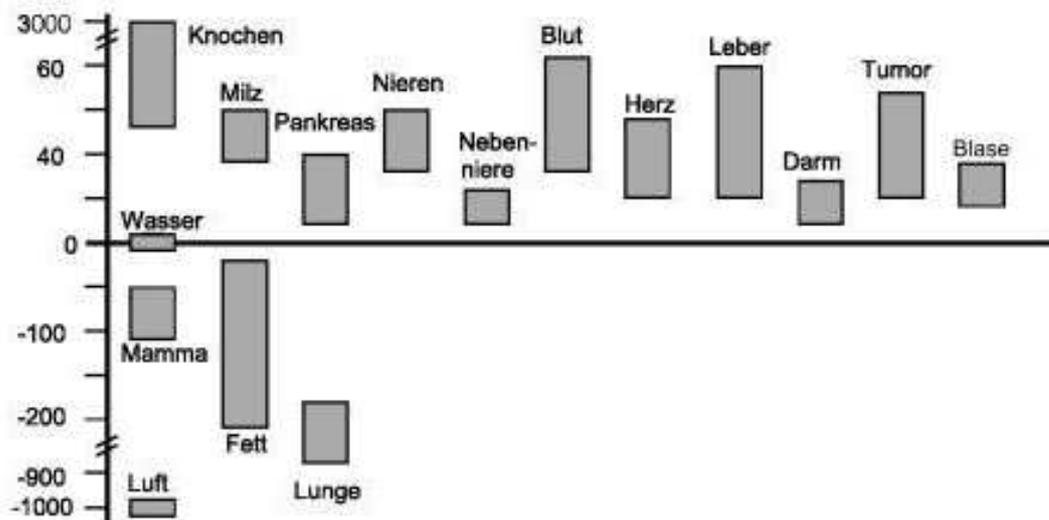


Abbildung 3: Hounsfield Skala

### 3. Material und Methoden

#### 3.1. C++

Das Programm wurde mit der Programmiersprache C++<sup>9</sup> implementiert.

#### 3.2. CMAKE

CMake<sup>10</sup> ist ein plattformunabhängiges Programmierwerkzeug für die Entwicklung und Erstellung von Software. Cmake ist in C++ geschriebene Open Source Software, die von der Firma Kitware<sup>11</sup> entwickelt wurde.

```
PROJECT (myPro1)
#FIND VTK
FIND_PACKAGE(VTK)
IF(USE_VTK_FILE)
INCLUDE( ${USE_VTK_FILE} )
ELSE (USE_VTK_FILE)
MESSAGE( FATAL_ERROR "VTK not found. Please set VTK_DIR.")
ENDIF (USE_VTK_FILE)

ADD_EXECUTABLE (myPro1 pro.cxx pro.h main.cxx)
TARGET_LINK_LIBRARIES(myPro1 vtkImaging)
```

Projekt Name

Benutzte Toolkits  
zB. VTK, FTLK, ITK

Geschriebene  
Programme Dateien

Benutzte Bibliotheken

Abbildung 4: CMakeList.txt Kode Beispiel

### 3.3. FLTK

Das Fast Light ToolKit (FLTK, fulltick)<sup>12</sup> ist eine Open Source Software und enthält einen eigenen GUI (Grafische Benutzeroberfläche)-Designer *fluid* (**FLTK User Interface Designer**)<sup>13</sup>.

FLTK ist ein GUI Toolkit für 3D-Grafikprogrammierung. Es ist eine sogenannte Cross-Plattform C++ Applikation. Diese erlaubt es, ein Programm mit nur einem Quellcode für verschiedene Betriebsplattformen wie *Linux*<sup>14</sup>, *Unix*<sup>15</sup>, *Mac OS X*<sup>16</sup> und *Windows*<sup>17</sup> zu entwickeln. Die FLTK Bibliotheken liefern plattformunabhängige Schnittstellen für alle Arten von C++ Programmen. FLTK bietet auch eine große Zahl vorgefertigter grafischer Kontrollelemente (Widgets), wie Inputs- und Outputsfelder, Buttons, Textfelder usw., die nach den eigenen Bedürfnissen angepasst werden können. Ein Layout Manager erlaubt eine leichte Entwicklung von grafischen Oberflächen.

## FLTK Code Beispiel:

```
#include <FL/Fl.H>
#include <FL/Fl_Window.H>
#include <FL/Fl_Box.H>
int main(int argc, char **argv) {

    Fl_Window *window = new Fl_Window(300,180);
    Fl_Box *box = new Fl_Box(20,40,260,100,"Hello, World!");
    box->box(FL_UP_BOX);
    box->labelsize(36);
    box->labelfont(FL_BOLD+FL_ITALIC);
    box->labeltype(FL_SHADOW_LABEL);
    window->end();
    window->show(argc, argv);
    return Fl::run();

}
```

Alle notwendigen Bibliotheken werden von Fluid Tool automatisch eingefügt.

Abbildung 5: FLTK Code Beispiel



Abbildung 6: FLTK „Hello World“ Beispiel

### 3.4. VTK

Das Visualization ToolKit (VTK)<sup>18</sup> ist ein Open Source Softwaresystem für 3D-Computergraphik, Bildverarbeitung und Visualisierung. VTK enthält eine C++ Klassenbibliothek und verschiedene Schnittstellen, z.B. Tcl<sup>19</sup>, Java<sup>20</sup> und Python<sup>21</sup>. Das Graphikmodell in VTK hat einen höheren Abstraktionsgrad als OpenGL<sup>22</sup>. Das heißt, es ist einfacher, Graphik- und Visualisierungsapplikationen zu erstellen.

In VTK können Applikationen in C++, Tcl, Java oder Python geschrieben werden. Mit Tcl oder Python und Tk, aber auch mit Java und seinen GUI Klassenbibliotheken, sind nützliche Programme schnell erstellt. VTK ist ein echtes Visualisierungssystem und unterstützt eine große Zahl von Visualisierungsalgorithmen, wie z.B. *scalar, vector, texture und volumetric methods*. Ferner sind viele Bildbearbeitungsalgorithmen direkt integriert, so dass 2D-Bilder und 3D-Graphikalgorithmen gemischt verwendet werden können.

Das Toolkit VTK arbeitet nach dem Pipelineprinzip. Es übergibt also das Ergebnis des einen Arbeitsschrittes, an den nächsten weiter.

```
#include "vtkSphereSource.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkRenderWindow.h"
#include "vtkRenderer.h"
#include "vtkRenderWindowInteractor.h"

// create sphere geometry with VTK
void main ()
{
    // create sphere geometry
    vtkSphereSource *sphere = vtkSphereSource::New();
    sphere->SetRadius(1.0);
    sphere->SetThetaResolution(18);
    sphere->SetPhiResolution(18);

    // map to graphics library
    vtkPolyDataMapper *map = vtkPolyDataMapper::New();
```

```
map->SetInput(sphere->GetOutput());

// actor coordinates geometry, properties, transformation
vtkActor *aSphere = vtkActor::New();
aSphere->SetMapper(map);
aSphere->GetProperty()->SetColor(0,0,1); // sphere color blue

// a renderer and render window
vtkRenderer *ren1 = vtkRenderer::New();
vtkRenderWindow *renWin = vtkRenderWindow::New();
renWin->AddRenderer(ren1);

// an interactor
vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
iren->SetRenderWindow(renWin);

// add the actor to the scene
ren1->AddActor(aSphere);
ren1->SetBackground(1,1,1); // Background color white

// render an image (lights and cameras are created automatically)
renWin->Render();

// begin mouse interaction
iren->Start();
}
```

Abbildung 7: VTK Code Beispiel (C++)

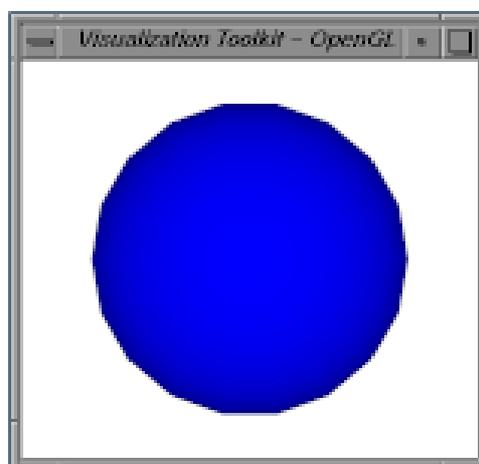


Abbildung 8: Ausgabe (Kugel)

### **3.5. VTK Methoden**

Alle in BlueEye benutzten VTK Klassen werden im folgenden Kapitel präsentiert.

#### **3.5.1. vtkDICOMImageReader**

Die DICOM Schichtbilder werden durch vtkDICOMImageReader eingelesen.

#### **3.5.2. vtkRenderer**

vtkRenderer bietet eine abstrakte Spezifikation für die Renderer. Ein Renderer ist ein Objekt, das den Rendering-Prozess für Objekte überprüft. Rendering ist der Prozess der Umwandlung von Geometrie, eine Spezifikation für die Beleuchtung und ein Blick der Kamera zu einem Bild. VtkRenderer führt auch die Koordinaten Transformationen zwischen Welt-Koordinaten, Blick-Koordinaten (der Computergrafik Rendering Koordinatensystem), und Display-Koordinaten durch.

#### **3.5.3. vtkRenderWindow**

VtkRenderWindow ist ein abstraktes Objekt um das Verhalten eines Renderings-Fensters zu spezifizieren. Es erstellt ein Fenster, in dem vtkRenderer eine Szene setzt und die geladenen Bilder darstellt.

#### **3.5.4. vtkActor**

Actors (vtkActor, vtkImageActor) erben von der Klasse vtkProp3D, welche für die 3D-Eigenschaften eines Objektes zuständig sind. Alle Bewegungen des Objektes werden in der vtkActor Klasse behandelt und ein Objekt in einer Render Szene repräsentiert. Ein vtkActor bietet Eigenschaften für Oberflächen eines Objektes (z.B. Ambient, Diffus oder Specular) durch eine assoziierte Klasse vtkProperty an. vtkProperty ist ein Objekt, das für die Beleuchtung und andere Oberflächeneigenschaften (Opazität, Spiegeln, Farbe usw.) eines geometrischen Objekts zuständig ist.

### **3.5.5. vtkLookupTable**

vtkLookupTable ist ein Objekt, welches von mapper Objekten auf der Karte in skalare Werte in RGBA (Rot, Grün, Blau, Transparenz) Farbspezifikationen oder in RGBA skalare Werten dargestellt wird. Die Farbtabelle kann durch direktes Einfügen von Farbwerten erstellt werden, oder durch Einfügen von Farbton, Sättigung, Wert und Alpha-Bereich und der Generierung in einer Tabelle.

### **3.5.6. vtkImageMapToColors**

Der vtkImageMapToColors Filter nimmt das Eingangsbild, das vom Reader gelesen wird, und bildet den ersten Bestandteil des Bildes durch eine vtkLookupTable ab. Das Resultat ist ein VTK\_UNSIGNED\_CHAR-artiges Bild. Wenn die vtkLookupTable nicht definiert ist oder auf NULL ersetzt wird, dann werden die Eingangsdaten übermittelt, wenn sie bereits von Typ VTK\_UNSIGNED\_CHAR-Art sind.

### **3.5.7. vtkCamera**

vtkCamera ist eine virtuelle Kamera für 3D-Rendering. Es bietet Methoden, um Position und Orientierung der Ansicht, Punkt- und Kontaktstelle und die Blickrichtung der Ansicht zu definieren.

### **3.5.8. vtkMarchingCubes**

vtkMarchingcubes ist ein Filter, der als Eingang ein Volumen (z.B., den 3D strukturierten Punkt eingestellt) nimmt und auf Ausgang eine oder mehrere Isosurfaces erzeugt.

Der Algorithmus ist:

Aus einem 3D-Raster ergeben sich Würfel, welche individuell betrachtet werden um eine Oberfläche zu ermitteln und die den Würfel teilen soll. Ein Würfel besitzt durch Querbildung acht Eckpunkte, die sich entweder innerhalb oder außerhalb der Kontur befinden und jeweils einen Index ( $v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8$ ) besitzen. Somit ergeben sich  $2^8 = 256$  mögliche Zustände für den Würfel.

Durch die Symmetrie des Würfels lassen sich die Zustände mit Hilfe von Rotation, Spiegelung und Inversion auf 15 reduzieren. In einer Triangle Lookup Table sind die 15 unterschiedlichen Polygone zur Teilung des Würfels gespeichert. Die Polygone werden nach dem gleichen Muster der Eckpunkte, welche nach einem Innen- und Außenbereich aufgeteilt wurden, dem Raster des Würfels zugewiesen. Anschließend wird sukzessiv zum nächsten "marschiert". Der Algorithmus terminiert, wenn der Anfangspunkt wieder erreicht wurde. Im Gesamtbild ergibt sich dann eine Oberfläche. vtkMarchingCubes ist ein Filter, der als Eingabe ein Volumen nimmt und eine oder mehrere Isoflächen erzeugt. Um eine Isofläche erhalten zu können, müssen als Eingabe ein oder mehrere Konturwerte angegeben werden. Als zusätzliche Alternative können ein min/max. skalarer Bildbereich (z.B. der Bildbereich von beliebigem Grauwert) und die Anzahl der Konturen eingegeben werden.

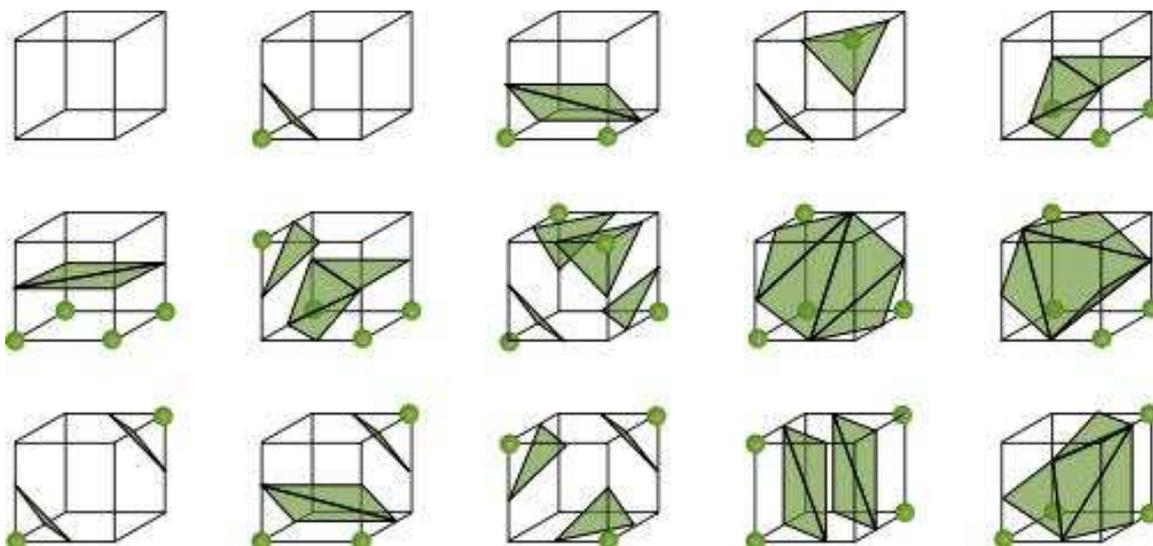


Abbildung 9: Es gibt 15 Grundkonfigurationen

### **3.5.9. vtkMapper**

Die Abbildung zwischen den Visualisierungsdaten und den grafischen Daten wird von Mappern (`vtkMapper`, `vtkPolyDataMapper`) übernommen. Sie lesen Daten aus dem Datenobjekt und machen sie für das Grafikmodell verfügbar, erstellt also geometrische Primitive. Ein Mapper ist gleichzeitig die geometrische Repräsentation für einen `vtkActor`.

### **3.5.10. vtkWindowToImageFilter**

`vtkWindowToImageFilter` bietet notwendige Methoden an, um die Daten (Bilder, Objekte usw.) von einem `vtkRenderWindow` oder `vtkWindow` zu lesen und sie zur Imaging-Pipeline als Eingabe zu übergeben. Damit können die Fensterinhalte als RGB (Rot, Grün, Blau) oder RGBA-Pixel gelesen und in einer Datei als Bild (z.B. jpg, png usw.) gespeichert werden.

### **3.5.11. vtkPointPicker**

Die Klasse `vtkPointPicker` erlaubt durch Positionierung der Maus und durch Drücken der linken Maustaste das Wählen eines Punktes. Durch „Picking“ erhält man  $x$ ,  $y$ ,  $z$ -Koordinaten (Identifikation eines Punktes), welche dem gewählten Punkt entsprechen.

## **3.6. vtkFIWindowInteractor**

Diese Klasse wurde von *Charl P. Botha* geschrieben<sup>23</sup>, damit VTK- und FLTK-Anwendungen miteinander integrieren können. Sie stellt einen plattformunabhängigen Interaktionsmechanismus für Maus, Tastatur, Zeitereignisse usw. zur Verfügung. `vtkFIRenderWindowInteractor` reagiert auf die Bewegungen des Input Devices, Buttons des Input Devices und Tastatureingaben.

### 3.7. Modularer Aufbau

Die Methoden können in anderen Programmen aufgerufen und dort die Ergebnisse in einem separaten Fenster angezeigt werden.

Hier steht ein Beispiel für die Benutzung der Aufrufe.

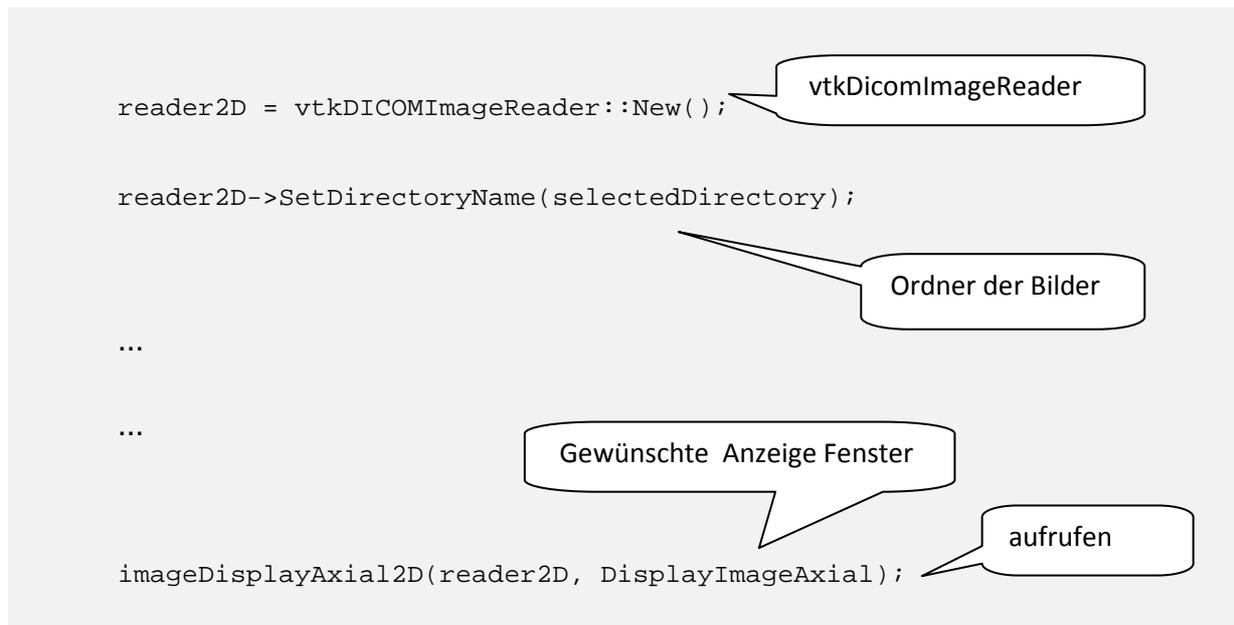


Abbildung 10: Beispiel von Modularer Aufbau

## **4. Diagramme**

### **4.1. Ablaufdiagramm**

In diesem Abschnitt wird der Ablauf des Programms mit textueller Beschreibung des Ablaufdiagramms vorgestellt. Ablaufdiagramme werden verwendet, um die Funktionsweise eines Computerprogramms oder eines Algorithmus zu veranschaulichen. Ablaufdiagramme bilden Prozesse aus unterschiedlichen Bereichen ab und ermöglichen so, eine schnelle Übersicht über komplexere Sachverhalte. Auf dem Pipelineprinzip aufbauend wird ein softwaretechnischer Entwurf eines Ablaufdiagramms erstellt, um zu zeigen, wie das zu entwickelnde Programm funktionieren soll und wie die Daten darin fließen.

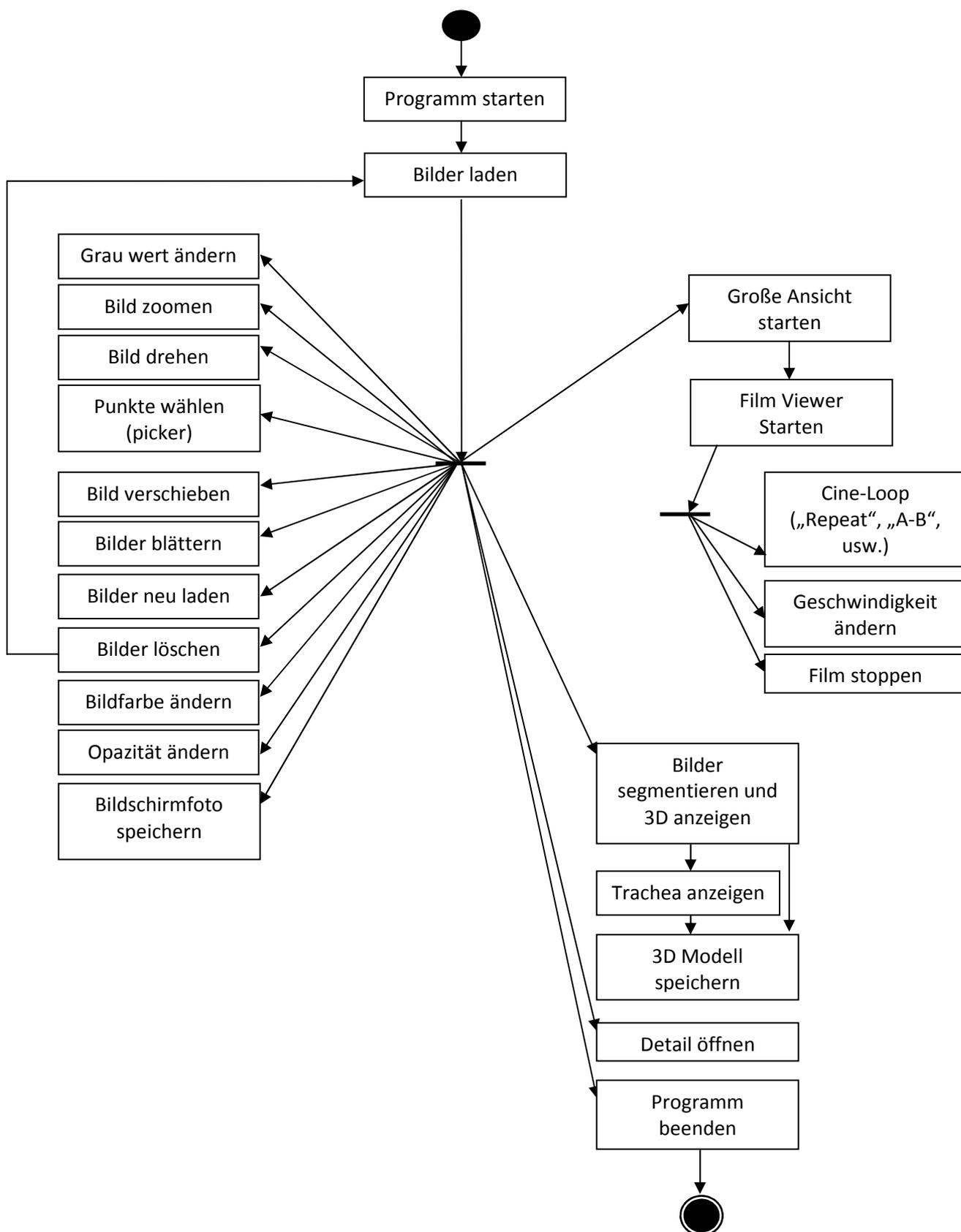


Abbildung 11: Ablaufdiagramm

## **4.2. Use-Case Diagramm**

In diesem Abschnitt werden die Use Cases mit ausführlicher textueller Beschreibung und Use Case Diagrammen vorgestellt. Use Cases definieren eine Interaktion zwischen Akteuren und dem betrachteten System, die stattfindet, um ein bestimmtes fachliches Ziel zu erreichen.

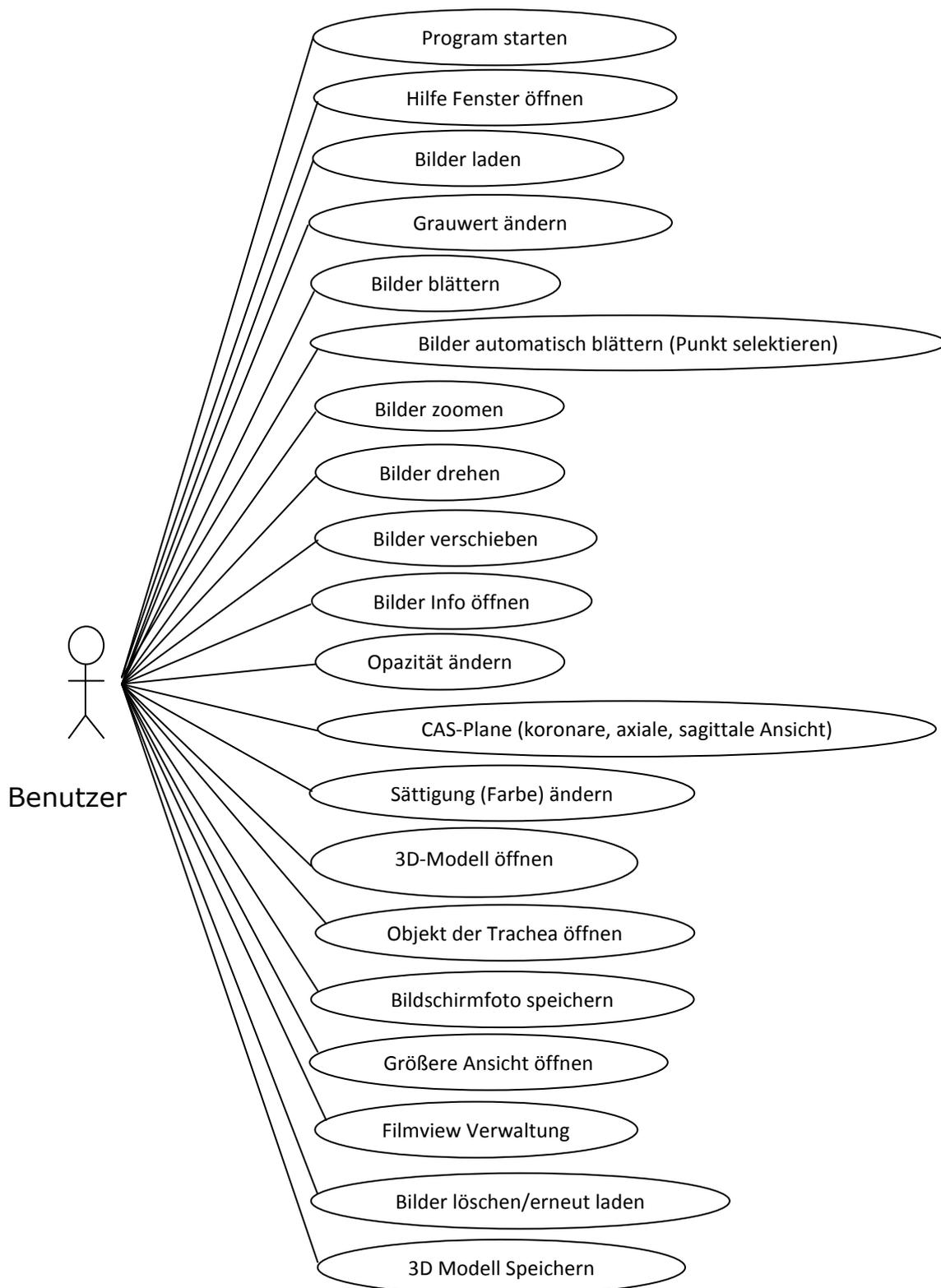


Abbildung 12: Use Case Diagramm

### 4.3. Funktionen und Testdrehbuch

#### 4.3.1. Programm starten

<b>Kurz Beschreibung</b>	Der Benutzer startet das Programm.
<b>Akteure/Auslöser</b>	Benutzer/Benutzer.
<b>Vorbedingung</b>	Installation erfolgreich beendet, das Programm ist startbereit.
<b>Nachbedingung</b>	GUI Fenster hat sich geöffnet.
<b>Basisablauf</b>	Nach der erfolgreichen Installation gibt man in der Konsole im Ordner des Programms den Befehl „./BlueEye“ ein. Alternative: Linke Maus klicken auf ausführbare Datei „BlueEye“.
<b>Anmerkung</b>	Siehe für Installation „Benutzerhandbuch“.
<input checked="" type="checkbox"/> <b>OK</b> <input type="checkbox"/> kosmetische Abweichungen <input type="checkbox"/> mittlere Abweichungen <input type="checkbox"/> große Abweichungen <input type="checkbox"/> System unbenutzbar	

### 4.3.2. Hilfefenster öffnen

<b>Kurz Beschreibung</b>	Der Benutzer öffnet das Hilfefenster.
<b>Akteure/Auslöser</b>	Benutzer/Benutzer.
<b>Vorbedingung</b>	Programm ist bereits gestartet.
<b>Nachbedingung</b>	Neues Fenster hat sich geöffnet.
<b>Basisablauf</b>	Benutzer klickt auf den Button „Help“. Alternative: Button „F1“ drücken.
<b>Anmerkung</b>	Im Hilfefenster befindet sich das „Benutzerhandbuch“.
<input checked="" type="checkbox"/> <b>OK</b> <input type="checkbox"/> kosmetische Abweichungen <input type="checkbox"/> mittlere Abweichungen <input type="checkbox"/> große Abweichungen <input type="checkbox"/> System unbenutzbar	

### 4.3.3. Bilder laden

<b>Kurz Beschreibung</b>	Der Benutzer liest die Schichtbilder ein.
<b>Akteure/Auslöser</b>	Benutzer/Benutzer.
<b>Vorbedingung</b>	Programm ist bereits gestartet.
<b>Nachbedingung</b>	Bilder werden auf den Fenstern (axial, sagittal und koronar) angezeigt und geeignete Buttons, Sliders und Funktionen werden aktiviert.
<b>Basisablauf</b>	Der Benutzer klickt auf den Button „Insert Images“ und wählt vom geöffneten Fenster die gewünschten CT-Bilder Ordner aus. Alternative: Strg + I.
<b>Anmerkung</b>	In Ordner müssen die CT-Bilder sich fortlaufend nummeriert und mit „.dcm“ Endung sein.
<input checked="" type="checkbox"/> <b>OK</b> <input type="checkbox"/> kosmetische Abweichungen <input type="checkbox"/> mittlere Abweichungen <input type="checkbox"/> große Abweichungen <input type="checkbox"/> System unbenutzbar	

#### 4.3.4. Grauwert ändern

<b>Kurz Beschreibung</b>	Der Benutzer ändert den Grauwert der Dargestellten 2D-Bilder.
<b>Akteure/Auslöser</b>	Benutzer/Benutzer.
<b>Vorbedingung</b>	Die Bilder werden erfolgreich geladen und/oder Große Ansicht ist geöffnet.
<b>Nachbedingung</b>	Der Grauwert eines Bildes bei den Ansichten („Axial“, „Sagittal“, „Koronar“) ist nach Wahl für „Level“ von -1000 bis 3000 und für „Width“ von 0 bis 4096 geändert.
<b>Basisablauf</b>	Der Benutzer führt den Mauszeiger auf irgendein Anzeigefenster („Axial“, „Sagittal“, „Koronar“) und drückt auf die linke Maustaste, bewegt die Maus gedrückt haltend nach rechts, links, hinten und nach vorne.
<b>Anmerkung</b>	Es funktioniert nur in den 2D-Fenstern.
<input checked="" type="checkbox"/> <b>OK</b> <input type="checkbox"/> kosmetische Abweichungen <input type="checkbox"/> mittlere Abweichungen <input type="checkbox"/> große Abweichungen <input type="checkbox"/> System unbenutzbar	

### 4.3.5. Bilder blättern

<b>Kurz Beschreibung</b>	Der Benutzer blättert die dargestellten Bilder durch.
<b>Akteure/Auslöser</b>	Benutzer/Benutzer.
<b>Vorbedingung</b>	Die Bilder wurden erfolgreich geladen.
<b>Nachbedingung</b>	Auf dem Fenster wird das Bild angezeigt, welches mit Hilfe des Sliders gewählt wurde.
<b>Basisablauf</b>	Der Benutzer hält mit der linken Maustaste den unter dem Anzeigefenster liegenden „Value Slider“ und bewegt ihn zwischen seinen Grenzen.
<b>Anmerkung</b>	Funktioniert nur in den 2D-Fenstern, „CAS-Plane Display“ wird automatisch aktualisiert.
<input checked="" type="checkbox"/> <b>OK</b> <input type="checkbox"/> kosmetische Abweichungen <input type="checkbox"/> mittlere Abweichungen <input type="checkbox"/> große Abweichungen <input type="checkbox"/> System unbenutzbar	

#### 4.3.6. Bilder automatisch blättern/Punkt selektieren

<b>Kurz Beschreibung</b>	Der Benutzer selektiert in irgendeiner Ansicht („Axial“, „Sagittal“, „Koronar“) einen Punkt/Bereich und alle anderen Ansichten zeigen dann denselben Punkt/Bereich.
<b>Akteure/Auslöser</b>	Benutzer/Benutzer.
<b>Vorbedingung</b>	Die Bilder wurden erfolgreich geladen.
<b>Nachbedingung</b>	In allen 2D-Displays werden die Bilder vom geklickten Punkt angezeigt.
<b>Basisablauf</b>	Linken Mausklick auf irgendein 2D-Fenster.
<b>Anmerkung</b>	Es funktioniert nur in den 2D-Fenstern, „CAS-Plane Display“ wird automatisch aktualisiert.
<input checked="" type="checkbox"/> <b>OK</b> <input type="checkbox"/> kosmetische Abweichungen <input type="checkbox"/> mittlere Abweichungen <input type="checkbox"/> große Abweichungen <input type="checkbox"/> System unbenutzbar	

### 4.3.7. Bilder zoomen

<b>Kurz Beschreibung</b>	Der Benutzer ändert die Größe eines dargestellten Bildes.
<b>Akteure/Auslöser</b>	Benutzer/Benutzer.
<b>Vorbedingung</b>	Die Bilder werden erfolgreich geladen und/oder „Larger View“ geöffnet.
<b>Nachbedingung</b>	Das Bild wird größer oder kleiner dargestellt.
<b>Basisablauf</b>	Die rechte Maustaste geklickt halten und nach oben (Bild vergrößern) bzw. nach unten (Bild verkleinern) bewegen.
<input checked="" type="checkbox"/> <b>OK</b> <input type="checkbox"/> kosmetische Abweichungen <input type="checkbox"/> mittlere Abweichungen <input type="checkbox"/> große Abweichungen <input type="checkbox"/> System unbenutzbar	

### 4.3.8. Bilder drehen

<b>Kurz Beschreibung</b>	Der Benutzer dreht irgendeine Ansicht („CAS Plane“ und „3D-Fenster“).
<b>Akteure/Auslöser</b>	Benutzer/Benutzer
<b>Vorbedingung</b>	Die Bilder wurden erfolgreich geladen und/oder „Larger View“ geöffnet.
<b>Nachbedingung</b>	Der Blickwinkel einer Ansicht (Bild) ist geändert.
<b>Basisablauf</b>	Die linke Maustaste geklickt halten und kreisförmig drehen.
<b>Anmerkung</b>	Nur in der CAS-Plane und dem 3D-Fenster möglich.
<input checked="" type="checkbox"/> <b>OK</b> <input type="checkbox"/> kosmetische Abweichungen <input type="checkbox"/> mittlere Abweichungen <input type="checkbox"/> große Abweichungen <input type="checkbox"/> System unbenutzbar	

### 4.3.9. Bilder bewegen

<b>Kurz Beschreibung</b>	Der Benutzer ändert die Position eines Bildes im Fenster.
<b>Akteure/Auslöser</b>	Benutzer/Benutzer.
<b>Vorbedingung</b>	Die Bilder werden erfolgreich geladen und/oder „Larger View“ geöffnet.
<b>Nachbedingung</b>	Die Position des Bildes ist geändert.
<b>Basisablauf</b>	Der Benutzer führt den Mauszeiger auf ein beliebiges Anzeigefenster, drückt auf die mittlere Maustaste und bewegt die Maus nach rechts, links, oben und nach unten.
<input checked="" type="checkbox"/> <b>OK</b> <input type="checkbox"/> kosmetische Abweichungen <input type="checkbox"/> mittlere Abweichungen <input type="checkbox"/> große Abweichungen <input type="checkbox"/> System unbenutzbar	

### 4.3.10. Bild-Informationen öffnen

<b>Kurz Beschreibung</b>	Der Benutzer bekommt Detail nformationen der CT-Bilder.
<b>Akteure/Auslöser</b>	Benutzer/Benutzer.
<b>Vorbedingung</b>	Die Bilder wurden erfolgreich geladen.
<b>Nachbedingung</b>	Informationen werden angezeigt.
<b>Basisablauf</b>	Klicken auf den „Infos“ Button. Alternative: Alt + I
<input checked="" type="checkbox"/> <b>OK</b> <input type="checkbox"/> kosmetische Abweichungen <input type="checkbox"/> mittlere Abweichungen <input type="checkbox"/> große Abweichungen <input type="checkbox"/> System unbenutzbar	

### 4.3.11. Opazität ändern

<b>Kurz Beschreibung</b>	Der Benutzer ändert die Opazität eines/aller Bildes.
<b>Akteure/Auslöser</b>	Benutzer/Benutzer.
<b>Vorbedingung</b>	Die Bilder wurden erfolgreich geladen und/oder „Larger View“ geöffnet.
<b>Nachbedingung</b>	Die Transparenz des Bildes ist geändert.
<b>Basisablauf</b>	Der Benutzer wählt einen Fenster („Axial“, „Sagittal“, „Koronar“, „Alle“) aus und ändert den Wert der Opazität.
<b>Anmerkung</b>	Der Opazitätswert von einem Fenster ist unabhängig von einem anderen Fenster. Es funktioniert nur in den 2D-Fenstern, das CAS-Plane Fenster wird automatisch aktualisiert.
<input checked="" type="checkbox"/> <b>OK</b> <input type="checkbox"/> kosmetische Abweichungen <input type="checkbox"/> mittlere Abweichungen <input type="checkbox"/> große Abweichungen <input type="checkbox"/> System unbenutzbar	

### 4.3.12. Sättigung (Farbe) ändern

<b>Kurz Beschreibung</b>	Der Benutzer ändert die Farbe eines Bildes.
<b>Akteure/Auslöser</b>	Benutzer/Benutzer.
<b>Vorbedingung</b>	Die Bilder wurden erfolgreich geladen und/oder „Larger View“ geöffnet.
<b>Nachbedingung</b>	Der Farbe der Bilder ist geändert.
<b>Basisablauf</b>	Der Benutzer markiert einen Fenster („Axial“, „Sagittal“, „Koronar“) und ändert damit die Sättigung des entsprechenden Fensters.
<b>Anmerkung</b>	Der Startwert der Farbe ist SW (Schwarz und Weiß). Eine zweite Möglichkeit ist SR (Schwarz und Rot). Funktioniert nur in den 2D-Fenstern, das CAS-Plane Fenster wird automatisch aktualisiert.
<input checked="" type="checkbox"/> <b>OK</b> <input type="checkbox"/> kosmetische Abweichungen <input type="checkbox"/> mittlere Abweichungen <input type="checkbox"/> große Abweichungen <input type="checkbox"/> System unbenutzbar	

### 4.3.13. 3D-Objekt öffnen

<b>Kurz Beschreibung</b>	Der Benutzer segmentiert die geladenen 2D-Bilder.
<b>Akteure/Auslöser</b>	Benutzer/Benutzer
<b>Vorbedingung</b>	Die Bilder wurden erfolgreich geladen und/oder „Größe Ansicht“ geöffnet.
<b>Nachbedingung</b>	Je nach geändertem Grauwert wird ein entsprechendes neues 3D-Modell (Bild) erzeugt.
<b>Basisablauf</b>	Klicken auf das „View 3D“ Button. Alternative: Strg + 3.
<b>Anmerkung</b>	CAS-Plane Fenster und 3D-Fenster benutzen das gleiche Display. Sie können das 3D-Fenster mit dem „3D“ TAB-Button aktivieren (Alternative: Alt + d) und mit „Alt + c“ wieder auf die CAS-Plane zurückschalten. Nach der Aktivierung der 3D-Modell in Hauptfenster, werden die „3D“ und „Trachea“ Buttons in „Larger View“ aktiviert.
<input checked="" type="checkbox"/> <b>OK</b> <input type="checkbox"/> kosmetische Abweichungen <input type="checkbox"/> mittlere Abweichungen <input type="checkbox"/> große Abweichungen <input type="checkbox"/> System unbenutzbar	

#### 4.3.14. 3D-Objekt im vordefiniertem HUs öffnen

<b>Kurz Beschreibung</b>	Der Benutzer erhält die Luftröhren-Ansicht
<b>Akteure/Auslöser</b>	Benutzer/Benutzer
<b>Vorbedingung</b>	Die Bilder wurden erfolgreich geladen, 3D-Ansicht klicken und/oder „Larger View“ geöffnet.
<b>Nachbedingung</b>	Es wird ein 3D-Trachea Modell (Bild) erzeugt, das durch die eingelesenen 2D-Bilder entsteht mit bei eingestellten Scannwerten von (-750,0) HU mit dem Marching Cubes Algorithmus erstellt wird.
<b>Basisablauf</b>	Klicken auf dem „Trachea“ Button auf Alternative: Strg + t.
<b>Anmerkung</b>	Nach der Aktivierung der 3D-Modell in Hauptfenster, werden die „3D“ und „Trachea“ Buttons in „Larger View“ aktiviert.
<input checked="" type="checkbox"/> <b>OK</b> <input type="checkbox"/> kosmetische Abweichungen <input type="checkbox"/> mittlere Abweichungen <input type="checkbox"/> große Abweichungen <input type="checkbox"/> System unbenutzbar	

### 4.3.15. Bildschirmfoto speichern

<b>Kurz Beschreibung</b>	Der Benutzer speichert ein Bildschirmfoto (Screenshot) von irgendeinem Fenster.
<b>Akteure/Auslöser</b>	Benutzer/Benutzer.
<b>Vorbedingung</b>	Die Bilder wurden erfolgreich geladen und/oder unter „Larger View“ geöffnet.
<b>Nachbedingung</b>	Ein „.jpg“ Bild wird von Hauptfenster 300x300 Pixel von „Larger View“ 500x500 Pixel erstellt.
<b>Basisablauf</b>	Der Benutzer wählt einen Fenster („Axial“, „Sagittal“, „Koronar“, „CAS-Plane“, „3D“) aus und speichert dessen Screenshot ab.
<b>Anmerkung</b>	Screenshots vom 3D-Fenster können erst nach Erstellung eines 3D Objektes erstellt werden.
<input checked="" type="checkbox"/> <b>OK</b> <input type="checkbox"/> kosmetische Abweichungen <input type="checkbox"/> mittlere Abweichungen <input type="checkbox"/> große Abweichungen <input type="checkbox"/> System unbenutzbar	

### 4.3.16. Größere Ansicht öffnen

<b>Kurz Beschreibung</b>	Der Benutzer stellt die im Hauptfenster angezeigten Bilder größer dar.
<b>Akteure/Auslöser</b>	Benutzer/Benutzer.
<b>Vorbedingung</b>	Die Bilder werden erfolgreich geladen.
<b>Nachbedingung</b>	Die im Hauptfenster angezeigten Bilder werden im „Larger View“ Fenster größer dargestellt.
<b>Basisablauf</b>	Klicken auf „Larger“ Button. Alternative: Strg + I (klein „L“).
<b>Anmerkung</b>	Die „3D“ und „Trachea“ Buttons wird aktiviert, die in der „Larger View“ Fenster, nachdem „3D View“ Button, das in Hauptfenster steht, gedrückt wurde.
<input checked="" type="checkbox"/> <b>OK</b> <input type="checkbox"/> kosmetische Abweichungen <input type="checkbox"/> mittlere Abweichungen <input type="checkbox"/> große Abweichungen <input type="checkbox"/> System unbenutzbar	

### 4.3.17. Film Ansicht

<b>Kurz Beschreibung</b>	Der Benutzer blättert irgendeine Ansicht automatisch durch. („Cine-Loop“)
<b>Akteure/Auslöser</b>	Benutzer/Benutzer.
<b>Vorbedingung</b>	Die Bilder wurden erfolgreich geladen und „Larger View“ geöffnet.
<b>Nachbedingung</b>	Es wird eine Serie von angezeigten Bildern durchgeblättert und gestoppt.
<b>Anmerkung</b>	<p>Alle Buttons (außer „Stop“ Button) sind deaktiviert, nachdem eine Funktion der Animation aktiviert wurde („Vorwärts Play“, „Rückwärts Play“, „Repeat“ oder „A-B Play“).</p> <p>Um die Buttons wieder zu aktivieren, klicken Sie „Stop“ Button.</p> <p>„Film View“ wird nur bei der 2D-Ansicht („Axial“, „Sagittal“ oder „Koronar“) aktiviert.</p>
<input checked="" type="checkbox"/> <b>OK</b> <input type="checkbox"/> kosmetische Abweichungen <input type="checkbox"/> mittlere Abweichungen <input type="checkbox"/> große Abweichungen <input type="checkbox"/> System unbenutzbar	

### 4.3.17.1. Film View starten

<b>Kurz Beschreibung</b>	Der Benutzer blättert irgendeine Ansicht automatisch durch.
<b>Nachbedingung</b>	Es wird eine Serie von irgendeiner Ansicht („Axial“, „Sagittal“ oder „Koronar“) durchgeblättert.
<b>Basisablauf</b>	Die gewünschte Ansicht („Axial“, „Sagittal“ oder „Koronar“) markieren und Filmfunktionen („Sprung zu erstem Bild“, „Sprung zu letztem Bild“, „Vorwärts Play“, „Rückwärts Play“, „Repeat“ oder „A-B Film“) auswählen.
<input checked="" type="checkbox"/> <b>OK</b> <input type="checkbox"/> kosmetische Abweichungen <input type="checkbox"/> mittlere Abweichungen <input type="checkbox"/> große Abweichungen <input type="checkbox"/> System unbenutzbar	

### 4.3.17.2. A-B Film starten

<b>Kurz Beschreibung</b>	Der Benutzer blättert irgendeine Ansicht zwischen ein ausgewähltem Startpunkt (Bild) und ein ausgewähltem Endpunkt (Bild) automatisch durch.
<b>Nachbedingung</b>	Es wird eine Serie zwischen ausgewählte Bilder durchgeblättert.
<b>Basisablauf</b>	Bringen Sie den Slider auf die von Ihnen ausgewählte Position und drücken Sie auf die A-B Button um die Startpunkt zu bestimmen. Im nächsten Schritt bringen Sie den Slider auf die von Ihnen ausgewählte Endposition auf der Leiste und drücken Sie auf die „Repeat“ Button um die Animation zu starten.
<input checked="" type="checkbox"/> <b>OK</b> <input type="checkbox"/> kosmetische Abweichungen <input type="checkbox"/> mittlere Abweichungen <input type="checkbox"/> große Abweichungen <input type="checkbox"/> System unbenutzbar	

### 4.3.17.3. Geschwindigkeit ändern

<b>Kurz Beschreibung</b>	Der Benutzer ändert die Geschwindigkeit der Animation.
<b>Vorbedingung</b>	Das „Larger View“ Fenster ist geöffnet und/oder automatisches Blättern von irgendeiner Ansicht ist gestartet. („Vorwärts Play“, „Rückwärts Play“, „Repeat“ oder „A-B Play“).
<b>Nachbedingung</b>	Mit dem Slider wird die Wiedergabegeschwindigkeit geändert.
<b>Basisablauf</b>	Bringen Sie den Slider auf die von Ihnen gewünschte Zahl (1-10).
<b>Anmerkung</b>	„1“ ist langsamste Geschwindigkeit. „10“ ist schnellste Geschwindigkeit.
<input checked="" type="checkbox"/> <b>OK</b> <input type="checkbox"/> kosmetische Abweichungen <input type="checkbox"/> mittlere Abweichungen <input type="checkbox"/> große Abweichungen <input type="checkbox"/> System unbenutzbar	

### 4.3.17.4. Animation stoppen

<b>Kurz Beschreibung</b>	Der Benutzer stoppt die Animation.
<b>Vorbedingung</b>	Die Animation wurde gestartet.
<b>Nachbedingung</b>	Die Animation wird angehalten.
<b>Basisablauf</b>	Klicken auf „Stop“ Button.
<input checked="" type="checkbox"/> <b>OK</b> <input type="checkbox"/> kosmetische Abweichungen <input type="checkbox"/> mittlere Abweichungen <input type="checkbox"/> große Abweichungen <input type="checkbox"/> System unbenutzbar	

### 4.3.18. Bilder entfernen

<b>Kurz Beschreibung</b>	Der Benutzer entfernt die in Hauptfenster angezeigten Bilder.
<b>Akteure/Auslöser</b>	Benutzer/Benutzer.
<b>Vorbedingung</b>	Die Bilder wurden erfolgreich geladen.
<b>Nachbedingung</b>	Alle Anzeigefenster werden geleert und alle veränderten Einstellungen werden zurückgesetzt.
<b>Basisablauf</b>	Klicken auf „Clear“ Button. Alternative: Strg + c.
<input checked="" type="checkbox"/> <b>OK</b> <input type="checkbox"/> kosmetische Abweichungen <input type="checkbox"/> mittlere Abweichungen <input type="checkbox"/> große Abweichungen <input type="checkbox"/> System unbenutzbar	

### 4.3.19. Bilder erneut laden

<b>Kurz Beschreibung</b>	Der Benutzer lädt die von Hauptfenster entfernten Bilder neu.
<b>Akteure/Auslöser</b>	Benutzer/Benutzer.
<b>Vorbedingung</b>	Die Bilder werden erfolgreich entfernt.
<b>Nachbedingung</b>	Die vom Hauptfenster entfernten Bilder werden erneut aufgerufen und angezeigt.
<b>Basisablauf</b>	Klicken auf „Reload“ Button. Alternative: Strg + r.
<input checked="" type="checkbox"/> <b>OK</b> <input type="checkbox"/> kosmetische Abweichungen <input type="checkbox"/> mittlere Abweichungen <input type="checkbox"/> große Abweichungen <input type="checkbox"/> System unbenutzbar	

### 4.3.20. 3D Modell speichern

<b>Kurz Beschreibung</b>	Der Benutzer speichert das 3D-Modell (z.B. Trachea) als vtkData.
<b>Akteure/Auslöser</b>	Benutzer/Benutzer.
<b>Vorbedingung</b>	Die Bilder wurden erfolgreich geladen und das 3D Modell wurde erfolgreich dargestellt.
<b>Nachbedingung</b>	Ein vtkData Objekt mit Endung „.txt“ wird aus dem 3D-Modell erstellt und gespeichert.
<b>Basisablauf</b>	Klicken auf „Save“ Button. Alternative: Strg + s.
<input checked="" type="checkbox"/> <b>OK</b> <input type="checkbox"/> kosmetische Abweichungen <input type="checkbox"/> mittlere Abweichungen <input type="checkbox"/> große Abweichungen <input type="checkbox"/> System unbenutzbar	

#### 4.4. Klassendiagramm

Ein Klassendiagramm ist eine graphische Darstellungsart, um die Programmklassen und Objekte eines Bereiches, die Eigenschaften (Attribute, Methoden, usw.), das Verhalten (Operationen) und die Beziehungen zwischen den Klassen deutlich zu modellieren<sup>24</sup>.

In diesem Abschnitt wird das Klassendiagramm von BlueEye mit textuellen Beschreibungen und mit einem Klassendiagramm vorgestellt.



Abbildung 13: Klassendiagramm

## 5. Benutzerhandbuch

### 5.1. Installation

Bevor sie mit der Installation dieses Programms beginnen, sollten Sie vorerst noch einige Bibliotheken installieren. Diese Bibliotheken sind CMake, VTK und FLTK. Code Beispiele sind für Linux Fedora 7<sup>25</sup> geschrieben.

#### 5.1.1. Installation der Bibliotheken

##### 5.1.1.1. Installation von CMake

Es gibt zwei Möglichkeiten für die CMake Installation. Die Erste ist eine Portable Edition und die zweite ist es, eine Installer Edition herunterzuladen.

#### ***Installer Edition:***

Laden Sie unter dem Link <http://www.cmake.org/cmake/resources/software.html> die Cmake Bibliothek (Version 2.4) herunter. Nachdem Sie es heruntergeladen haben, entpacken Sie es und öffnen Sie nun unter Linux eine Konsole oder ein Terminal und wechseln Sie (z.B. mit `cd /Desktop/cmake/`) in dieses Verzeichnis. Wenn Sie in dieses Verzeichnis gelangen sind, müssen Sie für die Installation bei Ihrem Benutzerkonto „superuser“ sein. Dafür geben sie den Befehl „su“ und Ihr „rootpassword“ ein. Ausschließlich folgen Sie schrittweise den unten aufgelisteten Codebefehlen.

```
$ su
$ ./bootstrap
$ make
$ make install
```

Abbildung 14: Cmake Install Code

### **Portable Edition:**

Laden Sie unter dem Link <http://www.cmake.org/cmake/resources/software.html> die Cmake Bibliothek (Version 2.4) Portable Edition herunter. Nachdem Sie es heruntergeladen haben, entpacken Sie es.

Nachteil von dieser Edition ist, dass man immer ganze gespeicherte Path schreiben muss.

Zum Beispiel:

```
Installierung des Programms:  
  
Portable Edition:   
$ /home/user1/Desktop/cmake/bin/cmake .  
  
Installer Edition:  
$ cmake .
```

Abbildung 15: Cmake Ausführungscode

### **5.1.1.2. Installation von VTK**

Laden Sie unter dem Link <http://www.vtk.org/get-software.php> VTK Bibliothek (Version 5.0.3) herunter. Nachdem Sie es heruntergeladen haben, entpacken Sie es und wechseln Sie (z.B. mit `cd /Desktop/vtk/`) in dieses Verzeichnis. Wenn Sie in dieses Verzeichnis gelangt sind, folgen Sie den unten aufgelisteten Codebefehlen wieder mit den „superuser“ Rechten.

```
$ su  
  
$ cmake .  
  
$ make  
  
$ make install
```

Abbildung 16: VTK Install Code

### 5.1.1.3. Installation von FLTK

Laden Sie unter dem Link <http://www.fltk.org/software.php> FLTK Bibliothek (Version 1.1.7) herunter. Nachdem Sie es heruntergeladen haben, entpacken Sie es und wechseln Sie (z.B. mit `cd /Desktop/fltk/`) in dieses Verzeichnis. Wenn Sie in dieses Verzeichnis gelangt sind, geben Sie wieder als Superuser folgende Codebefehle ein:

```
$ su  
  
$ cmake .  
  
$ make  
  
$ make install
```

Abbildung 17: FLTK Install Code

### 5.1.2. Programm installieren

Nachdem Sie oben genannte Bibliotheken installiert haben, öffnen Sie die CD und kopieren Sie die Datei „BlueEye.zip“ in Ihr gewünschtes Verzeichnis und entpacken es. Wechseln Sie in das Verzeichnis, in dem sich das Programm BlueEye befindet. (z.B. `cd /Desktop/BlueEye`).

Nun müssen Sie das Programm mit CMake konfigurieren. Für die Konfigurierung des Programms geben Sie in der Konsole oder im Terminal „`ccmake.`“ ein. Es erscheint jetzt ein CMake Konfigurationsmenü. Drücken Sie jetzt für die Konfiguration auf die „c“ Taste. Wenn auf dem CMake-Menü die Option „g“ (generate) erscheint, drücken Sie für die Generierung und Beendigung der Konfiguration auf die „g“ Taste. Somit haben Sie das Programm konfiguriert und es wird im Verzeichnis BlueEye ein „Makefile“ erstellt. Nun können Sie das Programm mit dem Befehl „`make`“ kompilieren und mit „`./BlueEye`“ starten.

**Hinweis:** Wenn bei der Konfiguration des Programms in der „`CmakeLists.txt`“ Datei gelistete Bibliotheken nicht gefunden wurden, wird es eine Fehlermeldung auswerfen, in dem fehlende Bibliotheksnamen erscheinen. Um diesen Fehler zu beheben, drücken Sie auf die „e“ Taste und gehen Sie auf die Zeile, auf der „... NOTFOUND“ Message erscheint. Drücken Sie auf die „Enter“ Taste und geben Sie den richtigen Pfad der nicht gefundenen Bibliothek ein (z.B. `/home/user1/libs/ftk`) und verlassen Sie den Eingabemodus mit „Enter“, Anschließend drücken Sie wieder auf die „c“ Taste, bis im Menü die Option „g“ erscheint.

## 5.2. Programm starten

Nach der erfolgreichen Installation erhalten Sie in BlueEye Verzeichnis eine ausführbare Datei, die heißt „BlueEye“. Mit der Konsole schreiben Sie „./BlueEye“ oder doppelklicken Sie einfach mit der linken Taste der Maus.

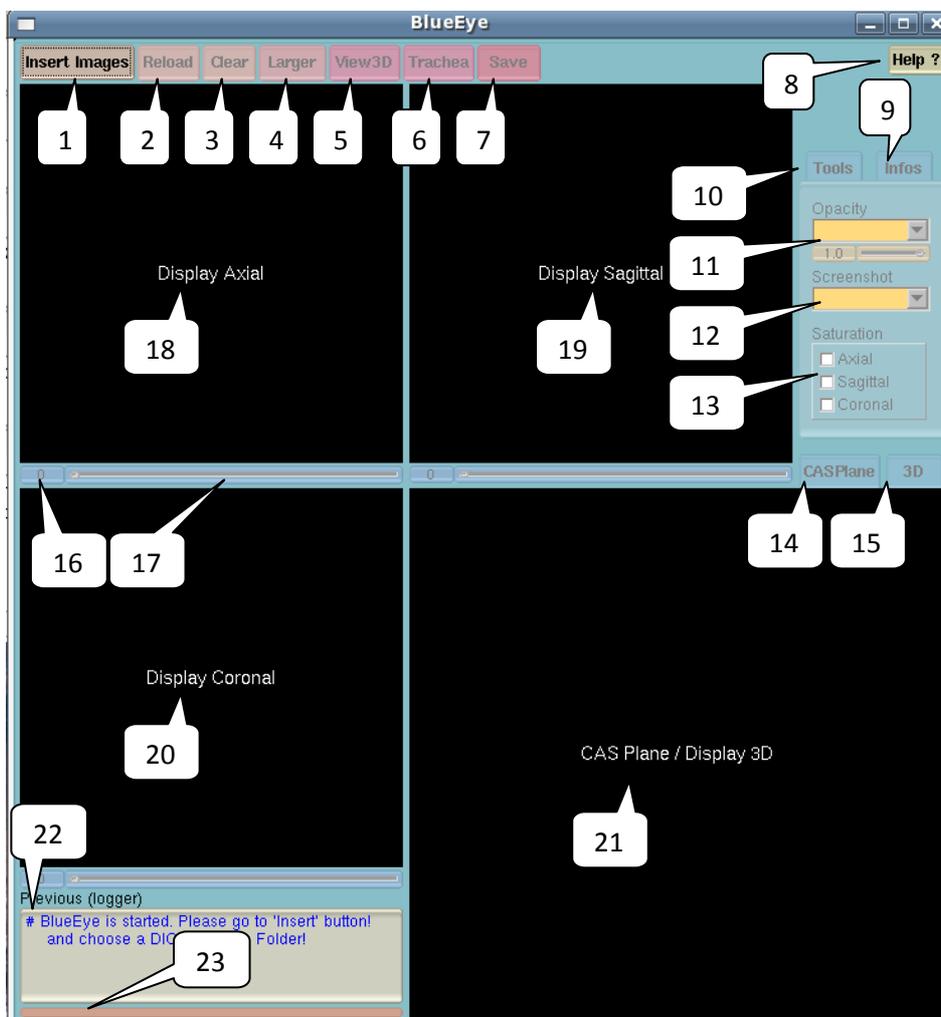


Abbildung 18: BlueEye Hauptfenster

### Die Buttons des Programs

- |                           |                                    |                        |
|---------------------------|------------------------------------|------------------------|
| 1) Einfügen die Bilder    | 9) Bild Informationen Seite öffnen | 15) 3D Display öffnen  |
| 2) Erneut laden           | 10) Tools Seite öffnen             | 16) Wert der Slider    |
| 3) Löschen                | 11) Opazität Einstellung           | 17) Slider Leiste      |
| 4) Größere Ansicht öffnen | 12) Bildschirmfoto speichern       | 18) Axial Fenster      |
| 5) 3D-Bild öffnen         | 13) Sättigung Einstellung          | 19) Sagittal Fenster   |
| 6) Trachea öffnen         | 14) CAS-Plane Display öffnen       | 20) Koronar Fenster    |
| 7) 3D-Objekt Speichern    |                                    | 21) CAS und 3D-Fenster |
| 8) Hilfe                  |                                    | 22) Protokoll          |
|                           |                                    | 23) Status Bar         |

### 5.3. Bilder einlesen und anzeigen

Nachdem Sie das Programm gestartet haben, gehen sie auf den „Insert Images“ Button und klicken Sie darauf oder drücken Sie auf die „Strg + i“ Taste. In der Folge dieser Aktionen öffnet sich ein Directory Chooser (Verzeichnis Wähler). Wählen Sie dort einen Ordner aus, in dem sich die DICOM („.dcm“) formatierte CT-Bilder befinden. Um das Laden der Bilder abzuschließen, klicken sie auf dem File Chooser Fenster auf „OK“.

**Hinweis:** Das Programm akzeptiert nur CT-Bilder, die mit .dcm Endung formatiert und fortlaufend nummeriert sind. Die geladenen Bilder werden in den Anzeigefenster, „Display Axial“, „Display Sagittal“, „Display Koronar“ und „Display CAS Plane“ als 2D in den Ansichten „Axial“, „Sagittal“, „Koronar“ und „Multiplanar“ angezeigt. Die Buttons: „Reload“, „Clear“, „View 3D“, „Larger“, „Infos“, die Progress Bar und alle Slider werden aktiviert.

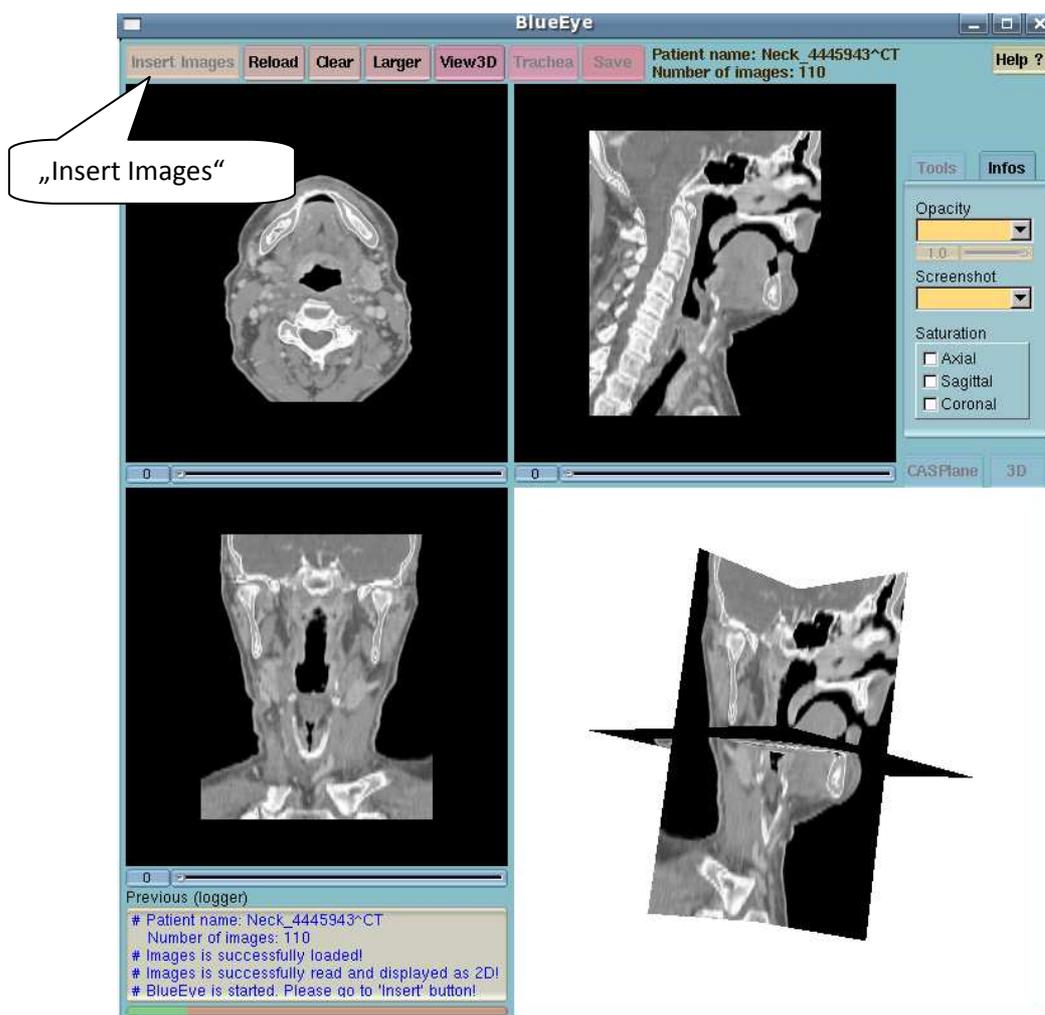


Abbildung 19: Nachdem die Bilder eingelesen wurden, wird der Button „Insert Images“ deaktiviert.

#### **5.4. Bilder entfernen**

Um auf dem Anzeigefenster dargestellte Bilder vom Hauptprogramm zu entfernen und alle von vorher übernommenen Einstellungen oder Veränderungen zurück zu setzen klicken Sie auf den „Clear“ Button (Button Nummer 3 von der Abbildung 18) oder drücken Sie auf die „Strg + c“ Tastenkombination.

**Hinweis:** Wenn das „Larger View“ Fenster auch geöffnet ist, wird es durch den Klick auf den „Clear“ Button geschlossen. Sie können gleiche Bilder wieder mit dem „Reload“ Button aufrufen oder mit dem „Load“ Button neue Bilder laden und anzeigen lassen. Nach dem Klick auf den „Reload“ Button wird das „Larger View“ Fenster nicht wieder geöffnet.

#### **5.5. Bilder neu laden**

Neues Laden von entfernten Bildern erfolgt durch den Klick auf den „Reload“ Button (Button Nummer 2 von der Abbildung 18) oder Druck auf die „r“ Taste. Nach der neuen Ladung werden alle übernommenen Einstellungen und Veränderungen zurückgesetzt.

#### **5.6. Hilfefenster öffnen**

Wenn Sie während der Benutzung Informationen erhalten möchten, klicken Sie auf das „Help“ Button (Button Nummer 8 von der Abbildung 18) oder drücken Sie die F1 taste.

## 5.7. Tools

Um die Opazität irgendeiner Ansicht zu ändern, wählen Sie in der Liste (Button Nummer 11 von der Abbildung 18) eine Ansicht und bewegen Sie die unten stehenden und nach der Auswahl aktivierten Slider zwischen den Werten 0.1 und 1.0.

Durch die „Screenshot“ (Button Nummer 12 von der Abbildung 18) können die Combobox irgendeiner Ansicht der angezeigten Bilder ausgewählt und als Bildschirmfoto gespeichert werden.

Um die Farbe von irgendeiner Ansicht zu ändern, markieren Sie das gewünschte Fenster (Button Nummer 14 von der Abbildung 18).

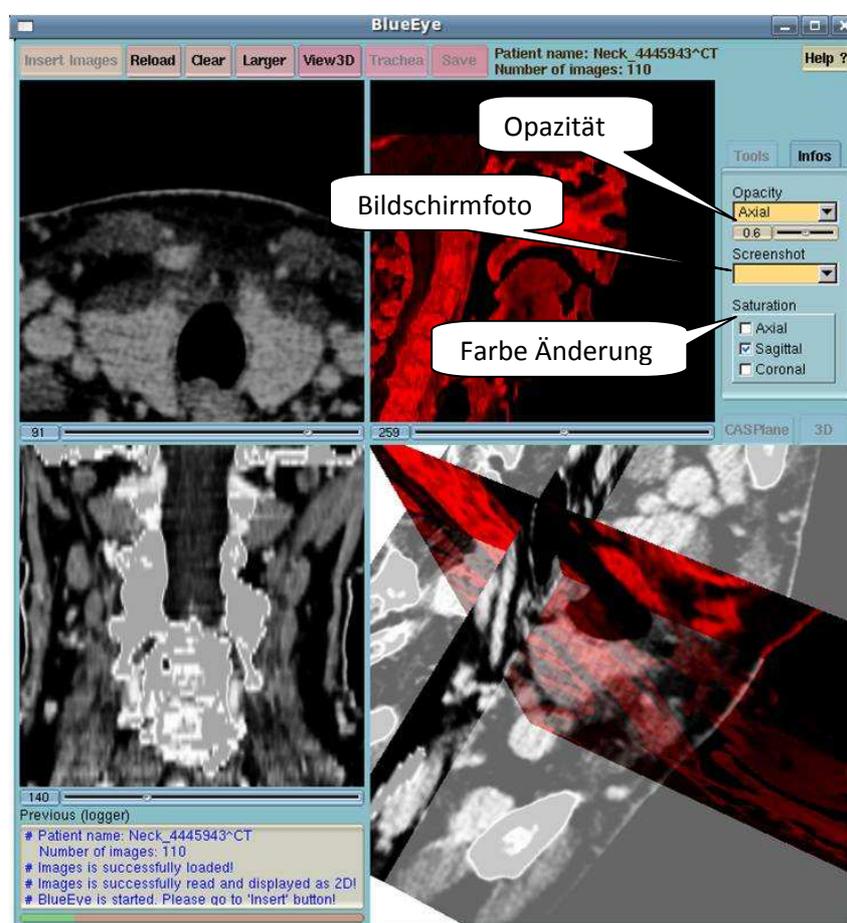


Abbildung 20: Tools

## 5.8. 3D-Ansicht öffnen

Als 2D angezeigte Bilder können durch einen Klick auf den „View 3D“ Button (Button Nummer 5 von der Abbildung 18) oder durch drücken auf die „Strg + 3“ Tastenkombination, drei-dimensional dargestellt werden. Die verschiedenen Ansichten (Knochenstruktur, Weichgewebe oder beide) eines 3D-Bildes erfolgen durch die Änderung des Grauwertes. Sie können gleich nach der Darstellung der 2D-Bilder auf „View 3D“ klicken und ein 3D-Bild mit vordefinierten Grauwerten zwischen -250 und 1200 im Anzeigefenster „Display 3D“ darstellen lassen oder zuerst den Grauwert ändern und wieder auf „View 3D“ klicken.

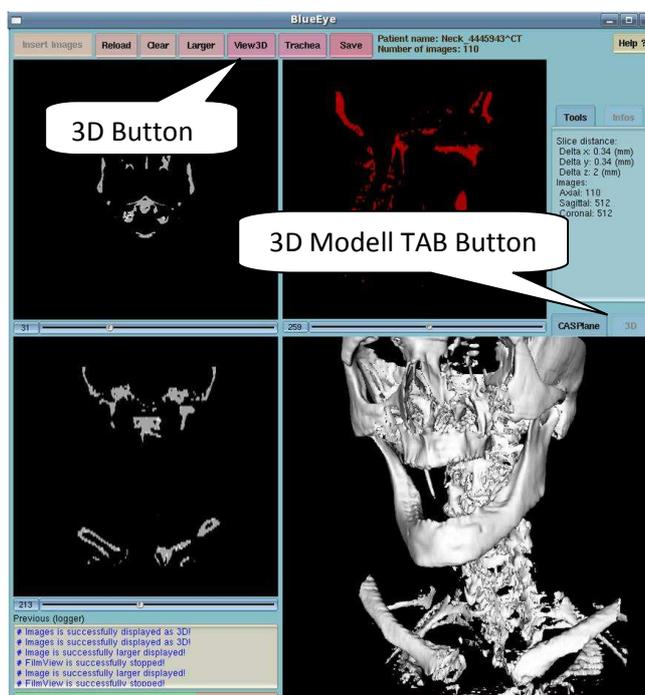


Abbildung 21: CAS- und 3D-Ansicht benutzen gleiche Fenster. Klicken Sie die TAB Button, welche Fenster aktivieren möchten.

CASPlane und 3D Fenstern sind mit der TAB Buttons getrennt. Wenn Sie wieder die CAS Plane Ansicht öffnen möchten, klicken Sie wieder auf das „CASPlane Tab“. Alternative Tastenkombinationen für das „CASPlane Tab“ (Button Nummer 14) bzw. „3D Modell Tab“ (Button Nummer 15 von der Abbildung 18) sind „Alt + C“ bzw. „Alt + 3“.

**Hinweis:** Wenn Sie in einem anderen Programm ein 3D-Modell darstellen und diese mit dem 3D-Modell von BlueEye vergleichen möchten, berücksichtigen Sie die Scan-Richtung und damit das Koordinatensystem der Bilder (Drei-Finger-Regel, Rechte-Hand-Regel oder Linke-Hand-Regel), ablesbar in der Headerdatei der CT-Bilder.

## 5.9. Darstellung der Trachea

Um die Trachea darzustellen, müssen Sie zuerst die 3D-Ansicht öffnen und dann das „Trachea“ Button (Button Nummer 6 von der Abbildung 18) aktivieren. Klicken Sie auf das „Trachea“ Button oder drücken sie auf die „Strg + t“ Tastenkombination.



Abbildung 22

**Hinweis:** um die Luftröhre zu modellieren, wurde eine vorbestimmte Grauwerte benutzt. Die Werte sind „-750“ und „0“.

## 5.10. Bilder größer darstellen

Die im Hauptfenster angezeigten Bilder („Axial“, „Sagittal“, „Koronar“, „CAS Plane“ und „3D“) können auch größer dargestellt werden. Dafür klicken Sie auf den „Larger“ Button (Button Nummer 4 von der Abbildung 18) oder drücken Sie auf die „Strg + I“ Taste. „Larger View“ kann auch genutzt werden, um schnell durch eine Serie von Bildern zu blättern.

Es kann der im Hauptfenster oder im „Larger View“ Fenster unter den Anzeigefenstern liegende Slider benutzt werden, um die angezeigten Bilder „Axial“, „Sagittal“ und „Koronar“ durch die Bewegung der Slider vorwärts oder Rückwärts manuell (per Hand) zu blättern.

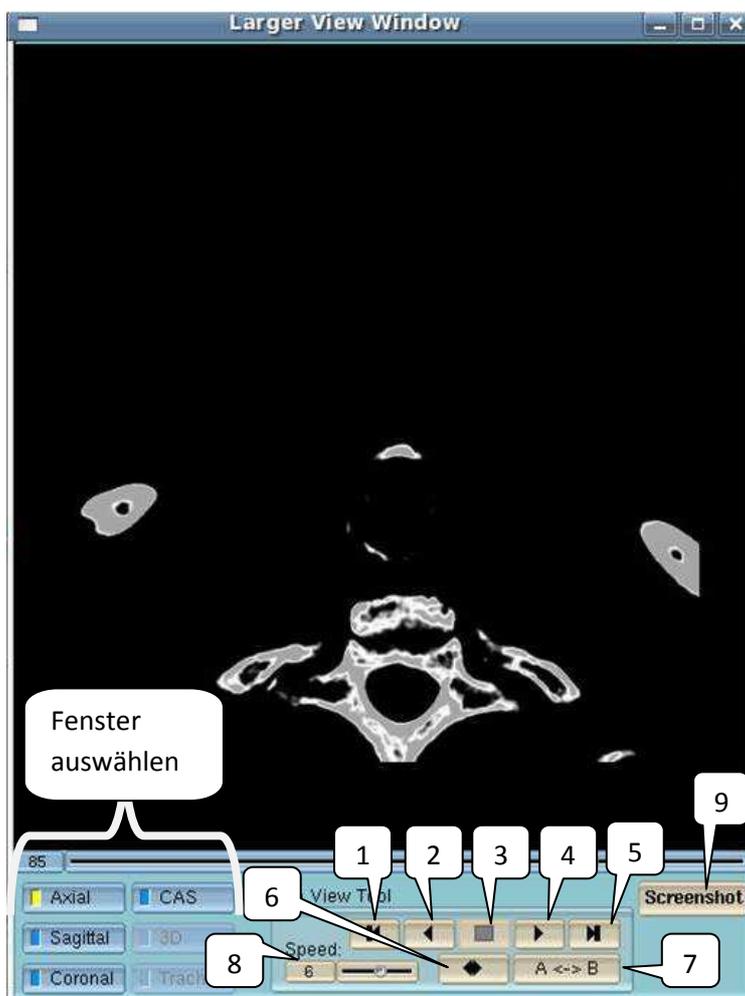


Abbildung 23: Überblick der „Larger View“

**Hinweis:** „3D“ und „Luftröhre“ Buttons werden aktiviert, nachdem im Hauptfenster „3D View“ aktiviert ist.

*Film View Buttons:*

- 1) Sprung zum ersten Bild.
- 2) Rückwärts Spiel: von letztem Bild zu erstem Bild durchblättern.
- 3) Automatische Filmvorführung anhalten.

**Hinweis:** bei der Filmvorführung sind alle Buttons deaktiviert (Display Buttons, Slider, Screenshot Button und Film Buttons), außer dem „Stop“ Button.

- 4) Vorwärts Spiel: von erstem Bild zu letztem Bild automatischen durchblättern.
- 5) Sprung zum letzten Bild.
- 6) Unendliche Vorwärts und Rückwärts Schleife: endlose Schleife von hintereinander Vorwärts und Rückwärts spielen.
- 7) „A <-> B“ Button ermöglicht die „vorwärts – rückwärts“ wiederholenden Blätter der Bilder durchzuspielen, die zwischen ausgewählten Bildern durchgeblättert wird.

**Hinweis:** um Start-Bild (A Punkt) auszuwählen verschieben die Slider an gewünschte Bild, dann „A <-> B“ Button drücken, danach wieder verschieben die Slider an gewünschte End-Bild (B Punkt) und drücken Sie „Repeat“ Button (Film View Button Nr. 6).

- 8) „Speed“ Slider ermöglicht die Filmvorführung Geschwindigkeit zu ändern.
- 9) „Screenshot“ (Bildschirmfoto) Button funktioniert wie im Hauptfenster.

**Hinweis:** Die Werte sind: 1 ist 10 Bild f.p.s. (Frame per second - Bild pro Sekunde) und 10 ist 1 Bild fps.

**Hinweis:** Film View Buttons sind bei 3D-Ansichten deaktiviert (die CAS-Plane, 3D und Trachea)

### 5.11. 3D-Modell speichern

Ein als 3D dargestelltes Modell kann durch den Klick auf den „Save“ Button (Button Nummer 7 von der Abbildung 18) oder Druck auf die „Strg + S“ Tastenkombination als vtkData gespeichert werden. Nachdem Sie auf den „Save“ Button geklickt haben, wird ein „File Chooser“ geöffnet. Um die Speicherung der Datei abzuschließen, wählen Sie einen Ort/Ordner, in den die Datei gespeichert werden soll und geben Sie einen Namen ein und klicken Sie auf den „OK“ Button.

### 5.12. Historie Fenster

Alle Aktionen, die bei der Benutzung des Programms durchgeführt wurden, werden in dem „General Info“ Fenster (Nummer 22 von der Abbildung 18) als Text protokolliert.

### 5.13. Bild Informationen Fenstern

Nach dem Laden der CT-Bilder kann man Bildgröße, Bildanzahl, Abstände zwischen den Bildern und den Patientnamen sehen. Das Rechtsseitige Info-Fenster öffnet sich auch mit der Tastenkombination „Alt + i“.

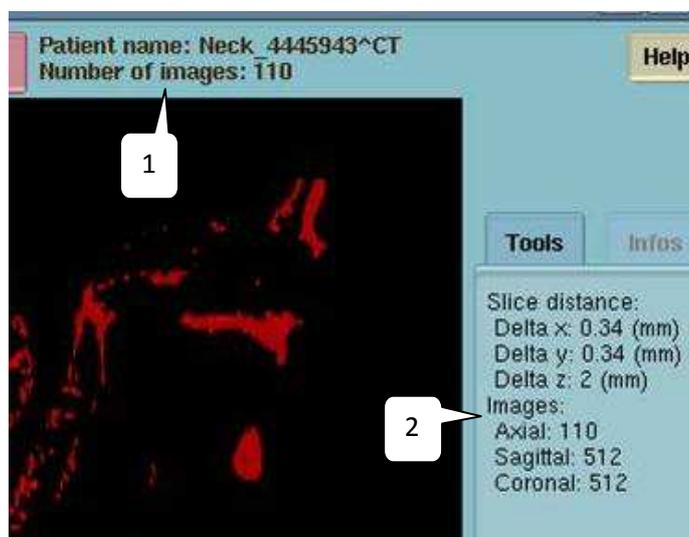


Abbildung 24: #1 zeigt zeigt Patientname und Anzahl von Bildern an, #2 zeigt die weiteren Detailsinformationen (Auflösung, Voxeldimension, ...) an.

## 5.14. Mausfunktionen

### 5.14.1. Linke Maustaste

Es gibt 3 Funktionen der linken Maustaste.

- **Grauwerte ändern**

Klicken und Halten Sie auf irgendein 2D Fenster („Axial“, „Sagittal“ und „Koronar“) und bewegen Sie den Zeiger zum gewünschten Grauwert.

**Hinweis:** Das funktioniert in einem x-y Koordinatensystem. X-Achse (horizontale Richtung) ist der sogenannte Level, die Y-Achse (vertikale Richtung) ist der sogenannte Width.

- **Punkte auswählen**

Um einen gewünschten Punkt in irgendeiner Ansicht (z.B. bei der „Axial“ Ansicht) oder auch in anderen Ansichten („Sagittal“ und „Koronar“) sehen zu können, klicken Sie mit der linken Maustaste auf den gewünschten Bereich beim „Axial“ Anzeigefenster. Somit sehen Sie den gleichen Punkt, den Sie bei der „Axial“ Ansicht selektiert haben auch in den „Sagittal“ und „Koronar“ Ansichten (seitlich und von vorne).

- **3D-Modell drehen**

Um das Bild zur gewünschten Position bringen zu können, kann eine Ansicht gedreht werden. Dafür führen Sie den Mauszeiger auf das Anzeigefenster „Display CAS Plane“ oder auf „Display 3D“ und drücken sie auf die linke Maustaste und bewegen Sie die Maus gedrückt haltend in x-y Koordinatenrichtungen (nach rechts, links, oben und nach unten).

### 5.14.2. Rechte Maustaste

Die angezeigten Bilder können vergrößert oder verkleinert werden. Für die Vergrößerung drücken Sie auf die rechte Maustaste und bewegen Sie die Maus gedrückt haltend in +y Koordinatenrichtung. Für die Verkleinerung, bewegen Sie die Maus gedrückt haltend in -y Koordinatenrichtung.

### 5.14.3. Mittlere oder linke + rechte Taste

Die Position eines dargestellten Bildes kann im Rahmen des Anzeigefensters geändert/verschoben werden. Während der Mauszeiger auf irgendeinem Anzeigefenster steht, drücken Sie auf die mittlere Maustaste und bewegen Sie die Maus gedrückt haltend in  $\pm x, y$  Koordinatenrichtungen (nach rechts, links, oben und nach unten Richtungen) oder bestimmen Sie einen Ort im Anzeigefenster, führen Sie den Mauszeiger hin und drücken Sie auf die mittlere Maustaste.

**Hinweis:** Alle Mausoptionen, außer die Punktselektion, können auch im „Larger View“ Fenster angewendet werden.

## 5.15. Programm beenden

Wenn Sie das Programm verlassen möchten, klicken Sie im Hauptfenster oben rechts auf den „x“ Button, drücken auf die „Esc“ Taste oder „Alt + F4“ Tastenkombination und klicken Sie im geöffneten Popup Fenster auf den „Yes“ Button.

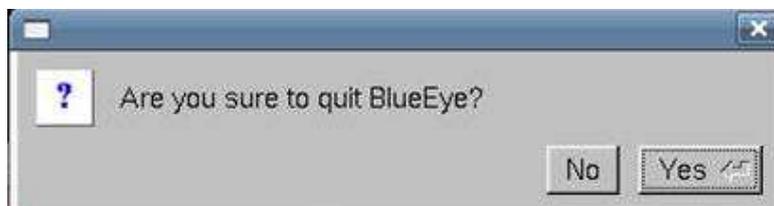


Abbildung 25: um Programm zu beenden, klicken Sie auf den Button „Yes“

## **6. Danksagung**

Ich möchte mich an dieser Stelle recht herzlich bei Herrn Prof. Wolfgang Freysinger für die Thema, seine unkomplizierte und hilfreiche Unterstützung bei Fragen und die Bewilligung des freundlichen Arbeitsraums bedanken.

Des Weiteren möchte ich mich meiner Familie und meinen Freunden, besonders Herren Özgür Güler, Yusuf Özbek, Arif Ergün, danken, die mir zur Seite stehen.

## 7. CD Anhang Dateien

Die CD besteht aus folgenden Dateien.

### 7.1. Das Programm (Source Code)

- Main.cpp

Die Hauptklasse des Programms.

- Info.cpp

Alle Status Nachrichten werden in dieser Klasse definiert.

- CMakeList.txt

Alle benutzte Toolkits (VTK, FLTK), Bibliotheknamen, deren Pfade und die Dateinamen, die bei der Konfigurierung und Kompilierung des Programms nötig sind, werden in dieser Klasse definiert.

- Blueeye.cpp

Enthält alle Ressourcen (Präprozessoren, Funktionen, Variablen, Instanzen, Zugriffskontrolle, Konstruktoren, Konstanten, Anweisungen, Schleifen, Objekte, Zeiger, usw.), die für das Programm nötig sind.

- Blueeye.h

Die Header-Datei von der Klasse *Blueeye.cpp* enthält alle Deklarationen von inline und statische Funktionen, virtuelle Funktionendeklarationen, Präprozessoren, die in der Klasse *Blueeye.cpp* benutzt und aufgerufen werden.

- BlueEyeGUI.cpp

Enthält alle Ressourcen (Funktionen, Deklarationen usw.) die für die Benutzeroberfläche des Programms nötig sind.

- BlueeyeGUI.h

Die Header-Datei von der Klasse *BlueeyeGUI.cpp*.

- BlueeyeGUI.fl

Fluid Datei um BlueEyeGUI.cpp zu erstellen.

- vtkFIWindowInteraktor.cpp

Enthält alle Präprozessoren, Funktionen, Variablen, Konstruktoren, Konstanten, Anweisungen, Objekte, Zeiger, usw., die der Benutzer bei den Mausoptionen braucht (Steuerung für Bildpositionierung, Bildumdrehung, Änderung der Bildgröße usw.).

- vtkFIWindowInteraktor.h

Die Header-Datei von der Klasse vtkFIWindowInteraktor.cpp.

- README

Installation Anleitung des Programms.

## **7.2. Doxygen**

## **7.3. Dokumentation**

## **7.4. Bibliotheken**

- CMAKE
- VTK
- FLTK

## 8. Lebenslauf

### **Persönliche Daten:**

Abdullatif Yesil  
Bahnhofstrasse 28/38  
6850 Dornbirn  
E-mail: abdullatif.yesil@student.uibk.ac.at  
Geburtsdatum und -Ort: 20. November 1983  
in Kütahya/Türkei  
Staatsbürgerschaft: Türkei

### **Schulbildung:**

1989 – 1995 Volks- und Hauptschule  
in Kütahya/Türkei  
1995 – 1997 Volksschule in Istanbul/Türkei  
1997 – 1998 ABG für Verwaltung  
Gymnasium für Handel in Istanbul/Türkei  
1998 – 2001 Atatürk Gymnasium  
in Kütahya/Türkei Matura im Juni 2001 in Türkei

### **Ausbildung:**

seit 2003 Studium der Informatik in Innsbruck

### **Berufliche Tätigkeiten:**

2001 - 2008 Firma Otantik  
Reisebüro und Geschenksartikel

seit 2006 Bäckerei in Dornbirn

seit 2008 Projektmitarbeiter  
HNO-Klinik Innsbruck

## 9. Literaturverzeichnis

---

<sup>1</sup> William E. Lorensen, Harvey E. Cline "MARCHING CUBES: A HIGH RESOLUTION 3D SURFACE CONSTRUCTION ALGORITHM" (Computer Graphics, Volume 21, Number 4, July 1987)

<sup>2</sup> VTK User's Guide (VTK 4.4)  
"Install, Use and Extend The Visualization ToolKit" 2005  
ISBN: 1-930934-13-0, Kitware, Inc.  
VTK Data File Format: <http://www.vtk.org/pdf/file-formats.pdf>

<sup>3</sup> Röntgen Computertomographie (CT) Hounsfield, 1969  
Klinik für Epileptologie, Universität Bonn  
Lehrnetz CT1  
<http://www.meb.uni-bonn.de/epileptologie/cms/upload/homepage/lehnertz/CT1.pdf>

<sup>4</sup> Computer Tomographie, „<http://de.wikipedia.org/wiki/Computertomographie>“  
und „<http://www.portal-der-zahnmedizin.de/behandlungen/diagnose-und-geraete/computertomographie/computertomographie.html>“

<sup>5</sup> Erich Wintermantel, Suk-Woo Ha, „*Medizintechnik – Life Science Engineering*“, ISBN 3540749241, 2008

<sup>6</sup> Monks - Ärzte im Netz GmbH,  
[http://www.internisten-im-netz.de/de\\_prinzip\\_362.html](http://www.internisten-im-netz.de/de_prinzip_362.html)

<sup>7</sup> Portal der Zahnmedizin,  
<http://www.portal-der-zahnmedizin.de/behandlungen/diagnose-und-geraete/computertomographie/computertomographie.html>

<sup>8</sup> Digital Imaging and Communications in Medicine , *Medical Imaging & Technology Alliance*, <http://medical.nema.org/>

---

<sup>9</sup> C++, *Bjarne Stroustrup*, <http://de.wikipedia.org/wiki/C++>

<sup>10</sup> CMake, Kitware Inc. , Andy Cedilnik, Bill Hoffman, Brad King, Ken Martin,  
<http://www.cmake.org/>

<sup>11</sup> Kitware Inc. , <http://www.kitware.com>

<sup>12</sup> Fast Light ToolKit, Bill Spitzak, <http://www.fltk.org/>

<sup>13</sup> Fast Light ToolKit, Bill Spitzak, <http://www.fltk.org/>

<sup>14</sup> Linux, <http://de.wikipedia.org/wiki/Linux>

<sup>15</sup> Ken Thompson, Dennis Ritchie, Douglas McIlroy, u. a. , <http://www.unix.org>

<sup>16</sup> Apple Inc., <http://www.apple.com>

<sup>17</sup> Microsoft Inc. <http://www.microsoft.com/>

<sup>18</sup> VTK, *Kitware Inc.* , <http://www.vtk.org/>

<sup>19</sup> Tool command language, John Ousterhout, <http://de.wikipedia.org/wiki/Tcl>

<sup>20</sup> JAVA, Sun Microsystems, Inc., <http://java.sun.com/>

---

<sup>21</sup> Python, Python Software Foundation, <http://www.python.org/>

<sup>22</sup> OPENGL, Khronos Group, <http://www.opengl.org/>

<sup>23</sup> David Pont, Charl P. Botha, <http://www.cpbotha.net/>, 2002-2006

<sup>24</sup> Dan Pilone & Neil Pitman, „UML 2.0 in a Nutshell“, O'Reilly Verlag GmbH & Co.KG, 1. Auflage, ISBN: 9783897213425, Seite: 12-35, (2006)

<sup>25</sup> Fedora, Red Hat & Community, <http://www.fedoraproject.org/>