



FAKULTÄT
FÜR INFORMATIK
Faculty of Informatics



Online Medical Imaging Platform

Willinger Christin

Computer Vision Lab
Institute of Computer Aided Automation
Vienna University of Technology
September 25, 2014

Supervisor: Robert Sablatnig

Contents

1	Introduction	1
1.1	KRESHMOI	1
1.2	Pflichtenheft	2
1.3	Möglichkeiten zur Umsetzung	2
1.3.1	JavaApplet	2
1.3.2	HTML5	3
1.3.3	Flash	3
2	Related Work	4
2.1	PACS-Systeme	4
2.2	DICOM	4
2.3	Osirix	4
2.4	XYZ - PACS Workstation	6
3	Methodology	7
3.1	Verwendete Technologien und Protokolle	7
3.1.1	HTTP	7
3.1.2	REST	7
3.1.3	JSON	8
3.1.4	AJAX	8
3.1.5	Objectiv J	8
3.1.6	Cappuccino	9
3.1.7	WebGL	9
3.1.8	PixiJS	9
3.2	Funktionalität und Aufbau der Benutzeroberfläche	10
3.2.1	Funktionalität des 2D-Betrachters	10
3.2.2	Funktionalität der Ergebniss-Liste	11
3.3	Technische Umsetzung	11
3.3.1	2D Betrachter	11
3.3.2	Berechnung der Fensterung	11
3.4	Kommunikation mit KRESHMOI	12
3.4.1	Query nach Bildern	12
3.4.2	Laden der Bilder	13
3.5	Architektur und Komponenten	14
3.5.1	Domänen Model	14
3.5.2	Architektur und Aufteilung in Komponenten	14
3.5.3	2DView	14
3.5.4	Kommunikations Module	14
3.6	Usability	14
3.6.1	Workflow bei der Befundung	14

4	Results	15
4.1	Geschwindigkeit	15
4.2	Usability	15
5	Conclusion	16
	Bibliography	17

Abstract

1 Introduction

Das Ziel dieser Projektarbeit war die Entwicklung eines Webbasierten Frontends für eine Suchmaschine welche das Durchsuchen von radiologischen Bildern aus der Medizin ermöglicht. Die Suchmaschine selbst ist ein Teil des KRESHMOI-Projekts welches sich generell mit der Aufbereitung und dem Durchsuchen von medizinischen Informationen beschäftigt. Das System KRESHMOI wird in diesem Kapitel erklärt und im speziellen auf die Suche für radiologische Bilder eingegangen. Weiters werden die Anforderungen an das Frontend angeführt und auf die Möglichkeiten zur Umsetzung mit den am aktuellen Entwicklungsstand verfügbaren Technologien diskutiert.

1.1 KRESHMOI

Das Ziel von KHRESMOI ist das Durchsuchen und der Zugang zu medizinischen Informationen für verschiedene Benutzer-Gruppen mit unterschiedlichem medizinischen Vorwissen. Die Einteilung der Benutzer erfolgt in 3 Kategorien:

- Personen ohne speziellen medizinischen Kenntnissen
- Ärzte
- Radiologen

Dazu verknüpft KHRESMOI Daten aus verschiedenen heterogenen Ressourcen wie Bildern aus Radiologie-Archiven, Bildern und Text aus Publikationen in Journalen oder Daten von diversen Webseiten. Da sich die verschiedenen Ressourcen qualitativ sehr stark voneinander unterscheiden können wird eine Bewertung ihrer Glaubwürdigkeit durchgeführt und dem Benutzer angezeigt. Die Suchanfrage kann in textueller Form oder als Bild-Query sowie als Kombination von beidem gestellt werden. Ein weiteres wichtiges Feature hierbei ist die multilinguale Suche, da die Menge an verfügbaren medizinischen Informationen nicht in alle Sprachen gleich ist. Dies bedeutet dass die Suchanfrage in mehrere Sprachen übersetzt wird und somit auch anderssprachige Quellen durchsucht werden können. Die Zusammenfassungen der Suchergebnisse werden anschließend in die Anfragesprache rückübersetzt, wodurch der Benutzer schnell durch die Ergebnissliste navigieren kann [kre].

Ein Teilprojekt von KHRESMOI ist das Durchsuchen von medizinischen Bilddaten wobei diese in 2D, 3D oder 4D (Video) vorliegen können. Um eine Suchanfrage auf ein Bild stellen zu können müssen an einem Ausgangsbild Positiv- oder Negativ-Bereiche eingezeichnet werden welche dann die Anfrage formen. Aus der Textur eines markierten Bereiches wird ein Feature-Vektor extrahiert mit dem anschließend eine Datenbank von zuvor indizierten Bildern durchsucht wird. Ein Frontend einer Bildsuchmaschine muss daher sowohl Tools zum markieren von interessanten Bereichen, als auch die Funktionalität zur vernünftigen Betrachtung der Bilder bereitstellen [kre].

Diese Arbeit spezialisiert sich auf die Interaktion mit dem Teilsystem für das Durchsuchen von radiologischen Aufnahmen in 2D und 3D, welche in einem PACS (Picture Archiving and Communication Systems) abgelegt sind. Da in einem Krankenhaus täglich große Mengen an Daten durch radiologische Aufnahmen produziert werden, bietet eine effizientes Durchsuchen dieser die Möglichkeit Sie für Ausbildung

und Forschung wieder zu verwenden. Dazu muss das User-Interface die grundlegenden Funktionen eines Betrachtungstools für Röntgen- und Computertomographie-Aufnahmen zur Verfügung stellen:

- Zoom
- Schnelles anpassen von Kontrast und Helligkeit
- Navigation durch die Schnitte eines 3D-Körpers in einer Schnittachse

[Hua10]

1.2 Pflichtenheft

- Kommunikation mit KRESHMOI über HTTP Anfragen. Dies beinhaltet das Senden von Suchanfragen, Auswerten der Ergebnisse und Laden der zugehörigen Bilder.
- Betrachten von CT Volumes. Navigation durch die einzelnen Schnitte eines Volumes entlang einer auswählbaren Achse.
- Schnelles anpassen von Kontrast und Helligkeit bei den einzelnen Schnittbildern mit der Maus (Fensterung).
- Zoomen und Scrollen des Bildausschnittes in einem Bild oder Volume.
- Tools zum Anotieren von Bereichen innerhalb der Bildern welche zur Interaktion mit der Bildsuche dienen.
- Präsentation der Suchergebnisse.
- Anzeige der den Bildern oder Volumes zugehörigen Reports.
- Ansicht zum vergleichen von verschiedenen Ergebnissen.
- Modulare Komposition der verschiedenen Ansichten.
- Umsetzung der Applikation im Webbrowser.

1.3 Möglichkeiten zur Umsetzung

Aufgrund der Anforderung dass, das Programm in einem Webbrowser ausgeführt werden soll, stehen derzeit drei Technologien zur Verfügung um die Anwendung umzusetzen. Diese werden in den folgende Absätzen kurz angeführt und ihre Vor- und Nachteile im Bezug auf das Pflichtenheft untersucht. Dabei spielt neben der Portabilität der Software, die Performance bei der Verarbeitung der Bilder einen große Rolle, da sowohl das Fenster als auch das Navigieren durch die einzelnen Bilder einer 3D Aufnahme relativ flüssig funktionieren müssen.

1.3.1 JavaApplet

Bei einem Java Applet wird ein Java Programm in eine Webseite eingebunden und vom dem Browser des Clients geladen. Der Browser übergibt das Applet dem Java Interpreter wofür aber ein spezieller Plugin notwendig ist. Der große Vorteil dieses Ansatzes ist dass die Anwendung sehr performant ist. Dies ergibt aus den beiden Punkten dass Java Code compiliert wird und dass es möglichst ist die Grafikhardware

des Clients zum bearbeiten der Bilder zu verwenden, welche für diese Aufgabe besser geeignet ist als die CPU. Dafür müssen die notwendigen Plugins sowie ein aktueller Java Interpreter auf dem Client installiert sein, was bei manchen Betriebssystemen für Mobilgeräte wie Tablets gar nicht möglich ist [Ora].

1.3.2 HTML5

Bei diesem Ansatz wird die Anwendung in HTML, CSS und JavaScript oder einem Framework welches auf diesen Technologien aufsetzt entwickelt. Die Entwicklung von Programmcode erfolgt bei Webapplikationen in JavaScript oder in Code für einen Interpreter welche in der JavaScript Laufzeitumgebung ausgeführt wird. Da diese Technologien von fast allen neuen Browsern für Desktop und Mobilgeräte unterstützt werden, kann damit ein sehr hoher Grad an Portabilität erreicht werden. Weiters sind Webapplikationen für den Benutzer sehr einfach zu verwenden da anstatt einer Installation um die Anwendung nutzen zu können nur eine Webseite geöffnet werden muss [W3C14].

HTML5 unterstützt Canvas Elemente für 2D Bilder jedoch ist eine Verarbeitung der Bilder durch die Grafikkarte nur begrenzt möglich. Für die Verwendung der Grafikkarte durch den Browser gibt es die Schnittstelle WebGL welche auf OpenGL ES aufbaut. Diese wurde jedoch noch nicht in allen gängigen Browsern implementiert oder ist in einigen noch nicht stabil und muss extra aktiviert werden [Gro14].

1.3.3 Flash

Flash Anwendungen sind Programme in einem proprietären Format deren Lizenz mittlerweile Adobe gehört. Diese werden von einem Interpreter dem Flash-Player ausgeführt, welcher es über einen Browserplugin ermöglicht Flashcode in Webseiten einzubinden und mit der Webseite zu interagieren. Die Erstellung von Anwendung erfolgt in Entwicklungsumgebung Flash wobei Code in der objektorientierten Sprache Aktion-Script erstellt werden kann. Flash bietet eine umfangreiche Grafik-API mit Unterstützung von Beschleunigung durch die Grafikkarte über OpenGL oder DirectX. Die Darstellung von Flash-Inhalten im Browser setzt den Flash-Player-Plugin voraus. Da es sich um ein proprietäres Format handelt ist der Support und die Weiterentwicklung nicht sichergestellt [Sys14].

2 Related Work

Software für die Bildsuche in radiologischen Daten gibt es bis jetzt noch nicht, allerdings decken sich die Anforderungen großteils mit Betrachtungstools für 2D und 3D Daten aus der Radiologie und Nuklearmedizin. Solche Softwareprodukte finden sich in den Betrachtungs-Workstations von PACS-System in Krankenhäusern oder als Betrachtungstools für Datensätze des offenen Standards DICOM. Die beiden Konzepte PACS und DICOM werden im folgenden Kapitel kurz erklärt, sowie die Umsetzung der benötigten Funktionalität in zwei konkreten Softwareprodukten diskutiert.

2.1 PACS-Systeme

Ein PACS-System (Picture Archiving and Communication System) dient zum Speichern und Austausch von medizinischen Bilddaten. Obwohl es prinzipiell für alle bildgebenden Verfahren verwendet werden kann wird es vorwiegend für Daten aus der Radiologie und Nuklearmedizin genutzt. Das System setzt sich aus dem PACS Server und den Workstations zusammen. Der Server sammelt Daten von den bildgebenden Geräten, verknüpft Sie mit Daten aus einem Krankenhaus Informations System (KIS) oder Radiologie Informations System (RIS) und sorgt für ihre Archivierung in einem Kurz- oder Langzeitarchiv. Die Kommunikation mit den bildgebenden Geräten erfolgt meist durch ein Protokoll welches den DICOM Standard implementiert. Die Befundung erfolgt auf den PACS Workstations welche die Daten vom Server laden und anzeigen. Die Workstation stellt die Funktionalität zur Betrachtung und zum Nachbearbeiten der Bilder zur Verfügung. Änderungen der Daten werden von der Workstation zurück auf den Server geladen. Je nach Funktionsumfang stehen auch Tools zur Befundung zur Verfügung welche die Daten an das RIS oder KIS weiter geben [Hua10].

2.2 DICOM

DICOM steht für Digital Imaging and Communication in Medicine und ist ein offener Standard welcher die Übertragung und das Speichern von medizinischer Bildinformation spezifiziert. Die wesentlichen Teile der Spezifikation sind die Datenstruktur für die Bilddaten und zugehörige Metainformationen, Services welche auf diesen Daten operieren, Anforderung an DICOM konforme Hard- und Software-Produkte und das Ablegen der Informationen auf einem Datenträger. Das Datenmodell setzt sich in Anlehnung an die reale Welt grundlegend aus den Entitäten Patient, Studie, Serie und Image zusammen zwischen denen jeweils eine 1:n oder 0:n Beziehung besteht. Es bietet weiters ausreichend Möglichkeiten zur Erweiterung durch einen Definitionsmechanismus für alle DICOM Objekte die sogenannte Image Object Definition [Hua10].

2.3 Osirix

OsiriX ist eine Software zur Betrachtung und Nachbearbeitung von DICOM Bilddaten. Sie wird als freie Open-Source-Software unter der GPL für das Betriebssystem Mac OS X entwickelt. OsiriX ist nur für die Forschung und dem privaten

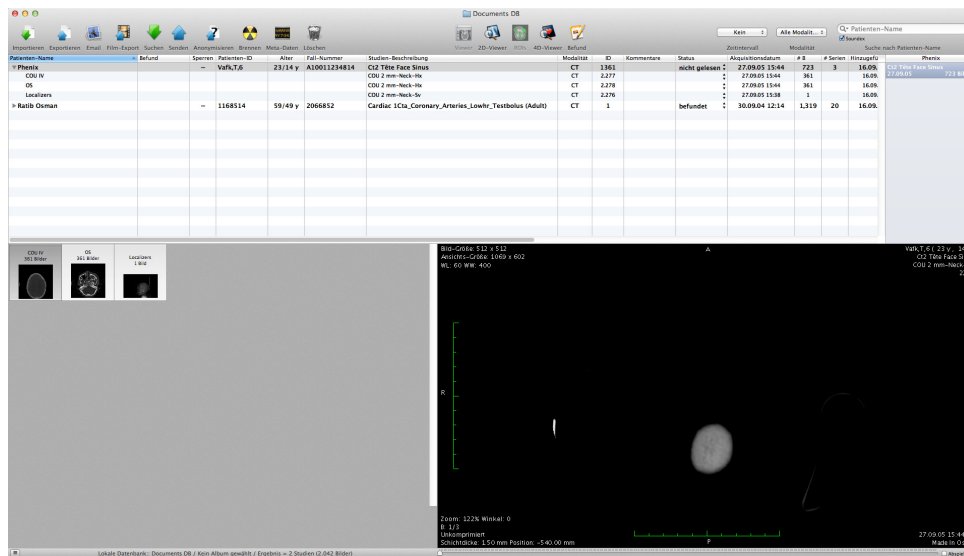


Figure 1: Datenbankansicht von OsiriX.

Gebrauch zugelassen, für einen diagnostischen Einsatz in der Medizin steht die kostenpflichtige Version OsiriX MD zur Verfügung. Das Programm führt eine Datenbank von DICOM Datensätzen, welche von DICOM-Dateien importiert bzw auch wieder als solche exportiert werden können. Weiters können über das DICOM-Protokoll die Daten auch von einem PACS-Server geladen werden [Ros14].

Von der Navigationsstruktur unterteilt sich OsiriX in eine Datenbankebene und eine Betrachtungsebene welche jeweils durch verschiedene Fenster umgesetzt wurden. Der Vollständigkeit halber sei erwähnt das es auch eine Ebene für 3D Rendering der Volumes gibt welche aber für diese Arbeit keinen weiteren Nutzen darstellt. In der Datenbankebene können die Datensätze importiert, exportiert und durchsucht werden, wobei jede Studie für einen Patienten einen Eintrag in der Liste darstellt. Weiters stehen Unterlisten zum Anzeigen der Serien einer Studie zur Verfügung. Unter der Listenansicht werden die ausgewählten Einträge als Thumbnails und in einem minimalen 2D Betrachter, welche die wichtigsten Funktionen der Betrachtungsebene implementiert zur Vorschau dargestellt. In der Betrachtungsebene finden sich die eigentlichen Funktionen und Tools welche für die Interaktion mit den Bildern notwendig sind. Dafür bietet OsiriX eine große Funktionspalette, wobei hier nur auf die Kernfunktionalität eingegangen wird [Ros14].

Funktionsumfang

Die Steuerung für die Interaktionstools erfolgt auch wie bei den PACS Workstations mit der Maus, welcher die spezifischen Funktionen zugewiesen werden können. Die jeweils zugewiesene Funktion wird durch das Drücken der Maustaste aktiv.

- Bei der **Fensterung** wird mit der X-Achse wird die Fensterbreite (Kontrast) und mit der Y-Achse das Fensterzentrum (Helligkeit) angepasst.
- **Positionierung** des Bildes innerhalb der Anzeigefläche. Geht das Bild über die Anzeigefläche wird dies durch eine farbige Linie an der jeweiligen Kante signalisiert.

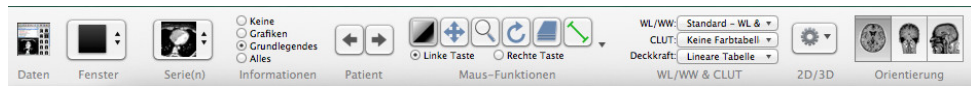


Figure 2: Toolbar für die 2D Betrachtungsebene.



Figure 3: Darstellung eines Volumes aus zwei verschiedenen Orientierungen. Die Schnittposition des linken Bildes wird im rechten als grüne Linie dargestellt.

- **Zoom** durch das Verschieben der Maus entlang einer Achse.
- **Rotation** um das Bildzentrum durch das Verschieben der Maus entlang der X-Achse.
- **Navigation** durch das Volume. Die Achse in der die Maus nach einem Klick zuerst verschoben wird, wird für die Navigation gewählt, die andere bleibt inaktiv.
- Einzeichnen von **ROIs**. Hierbei werden Punkte, Linien, Polygone, Winkel und noch weitere Geometrien unterstützt. Sind in dem Datensatz die notwendigen Informationen vorhanden, erfolgt eine automatische Vermessung der ROIs.

Weiters können mehrere Betrachtungs-Fenster nebeneinander angeordnet werden, wobei sich diese zur Darstellung unterschiedlicher Serien bzw. Orientierungen eines Datensatzes nutzen lassen. Die Orientierung gibt an, entlang welcher Achse die Bilder eines Volumes geschnitten werden. Wird in verschiedenen Fenstern eine unterschiedliche Orientierung gewählt, so wird beim Navigieren durch das Volume in einem Fenster die Position der Schnittebene in den anderen als Linie angezeigt [Ros14].

2.4 XYZ - PACS Workstation

3 Methodology

3.1 Verwendete Technologien und Protokolle

Bevor auf die eigentliche Umsetzung eingegangen wird werden noch kurz die verwendeten Frameworks und Technologien erklärt. Ein Teil davon wie HTTP und Rest wurden bereits durch das Interface für KRESHMOI festgelegt, ein weiter ergab sich durch die Anforderung einer Web-Applikation. Die Frameworks und Technologien für Grafik und GUI wurden gewählt weil mit Ihnen eine schnelle Umsetzung der Anforderungen möglich war.

3.1.1 HTTP

HTTP (Hyper Text Transfer Protokoll) ist ein Protokoll zur Übertragung von Daten über ein Netzwerk welches auf TCP aufsetzt. Der Datenaustausch zwischen zwei Kommunikationspartnern findet in der Form von Nachrichten statt, wobei der Client eine Anfrage an einen Server stellt und dieser die Anfrage bearbeitet und eine Antwort returniert. Eine Nachricht setzt sich aus einem Header und einem Body zusammen. Der Body enthält die Nutzdaten und der Header enthält Metadaten über die Nutzdaten. Vom Aufbau der Nachricht unterscheiden sich Anfrage und Antwort nur in der ersten Zeile:

- **Anfrage:** Enthält die HTTP-Methode, die URL welche auf die Resource am Server zeigt und die Protokollversion.
- **Antwort:** Enthält die Protokollversion und den Serverstatus. Der Serverstatus liefert eine Aussage ob der Request erfolgreich bearbeitet wurde bzw welche Art von Fehler bei der Bearbeitung aufgetreten ist.

HTTP ist ein zustandsloses Protokoll, daher wird nach jeder Anfrage die Verbindung vom Server wieder abgebaut. Für eine Zuordnung eines Clients muss dieser eine Session-ID mitsenden welche normalerweise im Header enthalten ist [Gro99].

3.1.2 REST

REST ist im eigentlichen Sinn mehr ein Architekturstil als ein Protokoll welcher mit HTTP umgesetzt wird. Die Idee von REST ist dass eine URL genau eine Ressource auf einem Server adressiert, wobei eine Ressource eine statische Datei oder das Ergebnis einer Aktion auf dem Server sein kann. Der Architekturstil lässt sich durch fünf Prinzipien zusammenfassen:

- **Ressource mit eindeutiger Identifikation:** Jede Ressource wird durch eine URI (Uniform Resource Identifier) weltweit eindeutig identifiziert. Diese adressiert unter anderem den Server auf den sich die Ressource befindet sowie die Ressource auf dem Server selbst.
- **Hypermedia:** Verknüpfungen zu anderen Entitäten und werden als Links auf die jeweiligen Ressourcen dargestellt. Weiters kann die Steuerung des Applikationszustandes durch Links auf weitere Aktionen durch Hypermedia umgesetzt werden.

- **Standard-Operationen:** Es gibt ein definiertes Interface welches von jeder Ressource zur Verfügung gestellt werden muss. Dieses umfasst einen relativ kleinen Satz von Operationen welche auf die Ressource ausgeführt werden können.
- **Unterschiedliche Repräsentation der Ressourcen:** Die Ressourcen können unterschiedliche Darstellungsformen haben. Ein Client kann also eine Ressource in einem bestimmten Format (z.B.: XML, HTML, JSON) anfordern sofern diese Darstellung vom Server unterstützt wird. In HTTP wird die gewünschte Darstellung im Header angegeben.
- **Zustandslose Kommunikation** Der Server hält keine Zustandsinformationen über den Client welcher über die Dauer eines Requests hinaus geht. Daher muss der Zustand einer Anwendung entweder am Client liegen oder vom Server in eine Ressource umgewandelt werden.

[Til09]

3.1.3 JSON

Bei JSON (JavaScript Object Notation) handelt es sich um ein Datenformat zum Austausch von Arrays und Objekt-Graphen. JSON findet neben XML vor allem in der Kommunikation zwischen Client und Server bei Webanwendungen Verwendung, wobei JSON Daten wesentlich kompakter und damit ressourcensparender sind. Wie bei XML werden Listen und Objekte in einer von Menschen lesbaren Form dargestellt. Dabei werden folgende Datentypen unterstützt welche wiederum beliebig tief ineinander verschachtelt werden können: NULL, Boolean, Zahl, String, Array und Objekt [Gam06].

3.1.4 AJAX

AJAX (Asynchronous JavaScript and XML) ermöglicht es einer Webanwendung kleinere Mengen von Daten nachzuladen und damit Teile der Webseite dynamisch zu ändern, statt bei jeder Aktion die Webseite neu zu laden. Benötigt die Web Applikation Daten vom Server wird an diesem eine HTTP Anfrage gesendet und Callback-Funktionen für den Fall einer Antwort oder eines Fehlers beim Browser registriert. Erhält der Browser eine Antwort auf seine Anfrage ruft er die Callback-Funktion auf und übergibt die erhaltenen Daten wodurch die Webanwendung mit der Verarbeitung dieser fortfahren kann. Dies ermöglicht die Entwicklung komplexer Webapplikationen, wobei die Webapplikation selbst mit der Seite geladen wird und die Daten die der Benutzer mit der Anwendung verarbeiten möchte dynamisch von dieser nachgeladen werden können [Gam06].

3.1.5 Objectiv J

Objective J ist eine Programmiersprache welche sich von der Syntax stark an Objective C anlehnt. Sie ist eine Erweiterung oder Obermenge von Javascript und wird von einem in Javascript geschriebenen Interpreter abgearbeitet. In Javascript können Objekte durch Prototyping erstellt werden, das Konzept von Klassen wird

aber nicht unterstützt. Obj J bietet zusätzlich zu den nativen JS Objekten die Definition von Klassen inklusive Vererbung und die Generierung von Objekten daraus. Obwohl es die Sprache erlaubt für Variablen, Methodenparameter und Rückgabe einer Funktion einen Datentyp zu definieren, werden diese aufgrund von schwacher Typisierung vom Interpreter nicht auf ihre Einhaltung überprüft. In der aktuellen Version wird die Übergabe von Referenzen als Parameter ähnlich einem Pointer in C unterstützt [Pro13].

3.1.6 Cappuccino

Bei Cappuccino handelt es sich um ein Web Application Framework für Objectiv J und Javascript, welches hauptsächlich der Erstellung komplexer Benutzeroberflächen dient. Das Framework lehnt sich sowohl vom Aussehen als auch von der Benennung der Komponenten sehr stark an das GUI-Framework Cocoa von Apple an. GUI-Elemente werden als Objekte erstellt welche von einer View-Klasse erben und innerhalb von anderen Views positioniert werden können. Das Interface wird von einem HTML5 fähigen Browser gerendert wobei für dessen Erstellung keinerlei HTML oder CSS Kenntnisse notwendig sind [Pro13].

3.1.7 WebGL

WebGL ist eine API für die Erstellung von 2D und 3D Grafiken in Browsern mit der Unterstützung der Grafikkarte. Im Gegensatz zur Canvas-2D API wo die Bilder in der CPU gerendert werden, ist WebGL aufgrund der Hardwarebeschleunigung wesentlich performanter. WebGL ist eine Shaderbasierte API welche sich sehr stark an OpenGL ES anlehnt. Diese bedeutet dass Code für die Recheneinheiten (Shadereinheiten) der Grafikkarte entwickelt wird, welchen der Treiber der Karte in Bytecode übersetzt und zur Ausführung in den Grafikchip lädt. Die Shaderprogramme werden in der Programmiersprache GLSL geschrieben welche sich sehr stark an C orientiert. Der Zugriff auf die Schnittstelle erfolgt über das HTML Canvas Element in welchem die Ausgabe der Grafikkarte dargestellt wird. Dies geschieht mittels JavaScript wo die API Funktionen zur Übergabe der Nutzdaten, Befehle und der Shaderprogramme bereitstellt [Gro14].

3.1.8 PixiJS

Da WebGL eine relativ Hardware nahe API ist wurden einige Frameworks entwickelt welche von der Komplexität einer solchen API abstrahieren, zu denen auch PixiJS zählt. Dies bietet Funktionen zum Zeichnen von Geometrischen Figuren, Füllen von diesen und Laden von Texturen. Eine Szene in PixiJS ist als Baum organisiert wobei jeder Knoten in dem Baum wiederum Operationen zur Manipulation wie Transformation, Translation oder Transparenz anbietet. Ein weiteres Feature ist dass auf einen Knoten ein Filter mit WebGL Fragment-Shader-Code gesetzt werden kann welcher zur Manipulation der Farbinformation in den einzelnen Bildpunkte dient [Ltd14].

3.2 Funktionalität und Aufbau der Benutzeroberfläche

Der Workflow beim Durchsuchen von radiologischen Aufnahmen mit KRESHMOI gliedert sich in folgende Schritte:

1. Auswahl eines Start-Datensatzes.
2. Betrachten des Datensatzes.
3. Anotieren einer ROI und eventuelle Eingabe von Text in die Suchzeile.
4. Suche absenden und Ergebnis listen.
5. Betrachten von Datensätzen aus der Ergebnisliste.
6. Um neue Suchanfrage von ein Ergebnissdatensatz aus zu stellen werden die Schritte von 3 an wiederholt.

Entsprechend dieses Ablaufs wird beim start der Applikation eine Liste mit allen Datensätzen geladen, aus welchen einer als Einstiegspunkt in die Suche ausgewählt werden kann. Dieser Datensatz wird in den Betrachter der Hauptansicht geladen. Die Hauptansicht setzt sich aus verschiedenen Interaktions-Elementen (Views) zusammen, welchen in mehreren möglichen Layouts mit unterschiedlicher Multiplizität und Anordnung miteinander kombiniert werden. Von diesen Elementen gibt es Vier verschiedene Typen:

- Betrachtungsansicht für Volumes (2D-Betrachter)
- Betrachtungsansicht für den Report (Report-Betrachter)
- Präsentationsansicht für die Ergebnisse (Ergebnis-Liste)
- Eingabezeile für Schlagwörter mit dem Suchbutton (Suchzeile)

Die Fläche welchen die einzelnen Views innerhalb des Browserfensters einnehmen kann zwischen ihnen beliebig verschoben werden, unter der Prämisse dass die Bedienelemente der einzelnen Views genügend Platz haben. Auf den Funktionsumfang des 2D-Betrachters und der Ergebnis-Liste wird in den folgenden Unterkapiteln etwas genauer eingegangen.

3.2.1 Funktionalität des 2D-Betrachters

Der 2D-Betrachter orientiert sich von seinen Funktionen sehr stark an der Betrachtungssoftware einer PACS-Workstaion oder eines DICOM-Betrachters. Er ermöglicht die räumliche Navigation durch die Schnitte eines Volumes in einer bestimmten Orientierung sowie das Umschalten zwischen den Orientierungen. Weiter kann wenn das Volume aus der Ergebnisliste ausgewählt wurde, also nicht der Startdatensatz ist, eine Anzeige mit den Deckungsbereichen der Suche hinzu geschaltet werden.

Zur Interaktion mit den Volume wird auch wie bei anderen Betrachtern der Maus eine bestimmte Funktion zugewiesen. Die Funktionen umfassen Fensterung, Zoom, Bewegen des Bildes und Zeichnen und Löschen von Polygonen (ROIs). Das Einzeichnen von ROIs ist um die Benutzung einfach zu halten auf ein einzelnes Schnittbild beschränkt. Wird das Werkzeug zum Zeichnen oder Löschen von Polygonen auf mehreren Schnitten verwendet springt der Betrachter bei deren Verwendung immer wieder auf das Schnittbild mit dem ersten Polygon zurück. Das Bild kann

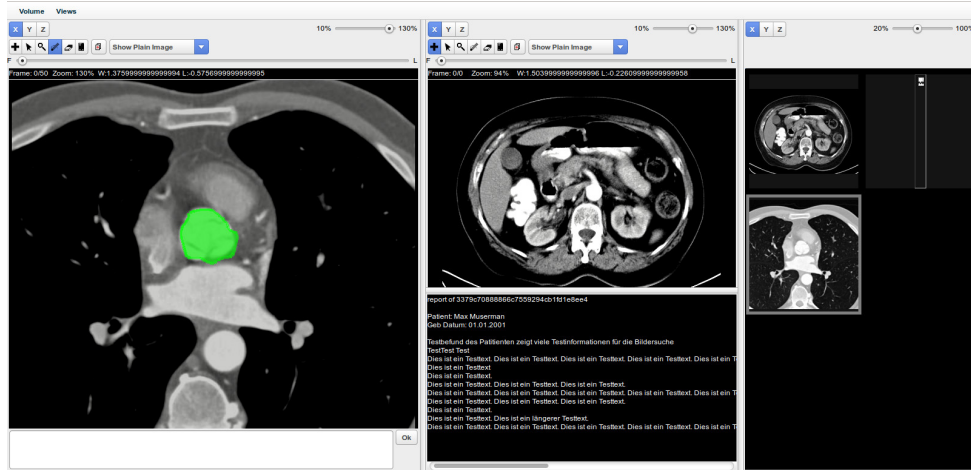


Figure 4: Hauptansicht mit Layout für zwei 2D-Betrachter, Report-Betrachter und Ergebnis-Liste

halb Transparent mit dem Report des Datensatzes überblendet werden, was aber die Mausfunktion auf das Scrollen des Reports reduziert. In allen anderen Funktionen ist das Scrollen immer an die Navigation durch die Schnitte gebunden.

3.2.2 Funktionalität der Ergebniss-Liste

Die Ergebnis-Liste präsentiert die Ergebnisse einer Suche welches sie über Drag and Drop in einen Betrachter laden lassen. Die Präsentation erfolgt über ein Vorschaubild welches von KRESHMOI als repräsentatives Bild für die Suche gewählt wurde. Um ein Suchergebniss im Vorfeld zu Beurteilen verfügt die Ergebnis-List ebenfalls über die Funktion die Orientierung der Schnitte des Volumes zu ändern und die Vorschaubilder zu zoomen.

3.3 Technische Umsetzung

3.3.1 2D Betrachter

3.3.2 Berechnung der Fensterung

Die Berechnung der Fensterung ist eine Anpassung von Kontrast und Helligkeit des Bildes, wozu für jedes Pixel der Farbwert mit einer Linear Transformation in einen neuen Zielwert überführt werden muss.

$$F(x) = c * x + b \quad (1)$$

Dabei ist der Kontrast durch c und die Helligkeit durch b gegeben. Für diese Berechnung wurde ein Ansatz auf Basis von JavaScript durch die Canvas 2D Api und eine Implementierung mit Web GL getestet.

Canvas 2D API

Für die Ausgabe im Browser wird ein HTML5 Canvas Element verwendet, auf welches über ein sogenannten Grafikkontext zugegriffen werden kann. Dieser Grafikkon-

text wird in JavaScript vom Browser als Objekt zur Verfügung gestellt. Durch das Kontext Objekt kann der Bildinhalt in ein Array und umgekehrt auch ein Array von Farbwerten in das Bild geschrieben werden. Dabei wird das Bild zeilenweise in ein eindimensionales Array Serialisiert, wo jeder Bildpunkt auf 4 Speicherstellen mit einem Integer-Wert zwischen 0 und 255 abgebildet wird. Die ersten drei Speicherstellen bilden den Farbwert durch die Anteile von Rot, Grün, Blau und die vierte den Alpha Kanal. Für die Fensterung müssen alle Speicherstellen für die Farbe transformiert und wieder auf einen Integer Wert gerundet werden. Um unnötige Berechnungen für jedes einzelne Pixel zu sparen, werden pro Fensterung alle Pixelwerte zwischen 0 und 255 einmal vorberechnet und in einer LookUp Tabelle als Array gespeichert. Anstatt den Wert für jedes Pixel neu zu berechnen und zu runden fungiert bei diesem Ansatz der aktuelle Wert eines Pixels als Schlüssel für die Tabelle welche den neuen Wert des Pixels enthält. Die Anzahl der Operationen wird dabei auf 255 Berechnungen und die Anzahl der Speicherzugriffe pro Bildpunkt reduziert.

```
function calcWindowForDomElement(domElement, contrast, brightness)
{
    var ctx = domElement.getContext("2d");
    var w = domElement.width;
    var h = domElement.height;

    var lut = getLookUpTable(contrast, brightness);
    var imageData = ctx.getImageData(0, 0, w, h);
    var imageDataSize = (w * h * 4);

    for(var pos = 0; pos < imageDataSize; pos = pos + 4){
        imageData.data[pos] = lut[imageData.data[pos]];
        imageData.data[pos + 1] = lut[imageData.data[pos + 1]];
        imageData.data[pos + 2] = lut[imageData.data[pos + 2]];
    }

    ctx.putImageData(imageData, 0, 0);
}

function getLookUpTable(contrast, brightness){
    var lut = new Array();

    for(var i = 0; i < 256; i++){
        var newValue = Math.round(contrast * i + brightness);
        if(newValue < 0) newValue = 0;
        if(newValue > 255) newValue = 255;
        lut[i] = newValue;
    }

    return lut;
}
```

3.4 Kommunikation mit KRESHMOI

Der Datenaustausch mit KRESHMOI basiert auf REST wobei sowohl auf die einzelnen Slices von einem Volume, als auch auf die Suche als Ressource über eine URL zugegriffen werden kann.

3.4.1 Query nach Bildern

Der Zugriff auf die Suchfunktion erfolgt über eine POST-Operation in welcher die Anfrage und die Antwort in JSON codiert werden. Zum Ausführen einer Suchanfrage stehen zwei Ressourcen zur Verfügung:

/index

Liefert ein Array von allen Verfügbaren Datensätzen zurück

/query

Liefert ein Array von Datensätzen zurück welche anhand der Übergebenen Suchkriterien gefunden wurden. Eine Suchanfrage basiert immer auf einen Datensatz welcher in der Anfrage übergeben werden muss. In diesem Datensatz werden weiters Interessante Bereiche sogenannte ROIs (Region of Interest) in den einzelnen Schnittbildern definiert. Die Übergabe einer ROI erfolgt als Polygon in Form einer Listen von Punkten im Dreidimensionalen Raum des Volumes.

Listing 1: Query

```
1 {
2   "queryid": "",
3   "text": "",
4   "imageid": "",
5   "roi":
6   [
7     "polygon":
8     [
9       "point": {"x": 0, "y": 1, "z": 0},
10      "point": {"x": 1, "y": 1, "z": 0},
11      "point": {"x": 0, "y": 0, "z": 0}
12    ],
13
14  ]
15 }
```

Listing 2: Result

```
1 {
2   "rankedImageID":
3   [
4     {
5       "imageID": "9587336b132b127b936ad0afd80ca862",
6       "normalDimX": 52,
7       "normalDimY": 512,
```



```

8         "normalDimZ": 512,
9         "relevance": 1,
10        "report": "",
11        "title": "Some Image"
12    },
13 ]
14 }

```

3.4.2 Laden der Bilder

Die Bilddaten eines Volumes können über eine GET-Operation geladen werden. Dabei wird jedes Volume als eine Resource im Unterverzeichnis */image/* identifiziert. Über Parameter in der URL werden die Schnittrichtung und die Nummer des Bildes in der jeweiligen Schnittebene angegeben. Wird keine Nummer für das Bild angegeben wählt KRESHMOI eine repräsentatives Bild für die Suchanfrage, welches für die Präsentation der Ergebnisse verwendet wird. Für die Schnittebene gibt es entsprechende der Terminologie für die Bildgebung in der Anatomie die Optionen:

- Axial: Die Schnitte erfolgen Waagrecht in der Transversalebene
- Sagital: Die Schnitte erfolgen Senkrecht in der Sagitalebene
- Coronal: Die Schnitte erfolgen Senkrecht in der Frontalebene

Sollen die Regionen welche der Suchanfrage entsprechen in den einzelnen Bildern markiert werden, muss die Query-ID mit übergeben werden um die ursprüngliche Suchanfrage zu identifizieren.

Beispiel:

```

/image/9587336b132b127b936ad0afd80ca86.jpg?slice=1&direction=axial&query=
1

```

3.5 Architektur und Komponenten

3.5.1 Domänen Model

3.5.2 Architektur und Aufteilung in Komponenten

3.5.3 2DView

3.5.4 Kommunikations Module

3.6 Usability

3.6.1 Workflow bei der Befundung

4 Results

4.1 Geschwindigkeit

4.2 Usability

5 Conclusion

References

- [Gam06] Johannes Gamperl. *AJAX : Web 2.0 in der Praxis*. Galileo Press, Bonn, Mar 2006.
- [Gro99] Network Working Group. Hypertext transfer protocol – http/1.1. <https://www.ietf.org/rfc/rfc2616.txt>, Jun 1999.
- [Gro14] Khronos Group. WebGL - opengl es 2.0 for the web. <http://www.khronos.org/webgl/>, 2014.
- [Hua10] Dr. H. K. Huang. *PACS and Imaging Informatics*. Kluwer Academic Press, Wiley-Blackwell, Mar 2010.
- [kre] Khresmoi - description of work - proposal 257528.
- [Ltd14] Goodboy Digital Ltd. Pixi.js - 2d webgl renderer with canvas fallback. <http://www.pixijs.com/>, 2014.
- [Ora] Oracle. Java applets. <http://docs.oracle.com/javase/tutorial/deployment/applet/>.
- [Pro13] Cappuccino Project. Cappuccino web framework. <http://www.cappuccino-project.org/>, Jul 2013.
- [Ros14] Antoine Rosset. Osirix - dicom viewer. <http://www.osirix-viewer.com/>, Sept 2014.
- [Sys14] Adobe Systems. Funktionen | adobe flash player. <http://www.adobe.com/at/products/flashplayer/features.html>, 2014.
- [Til09] Stefan Tilkov. *REST und HTTP : Einsatz der Architektur des Web für Integrationsszenarien*. Heidelberg : dpunkt-Verlag, 2009.
- [W3C14] W3C. W3c candidate recommendation. <http://www.w3.org/TR/2014/CR-html5-20140731/>, Jul 2014.