



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 10, 2014

Supervisor: Robert Sablatnig

Contents

1	Introduction	1
1.1	Motivation	1
1.1.1	Interface für Kreshmoi	1
1.1.2	Interaktion mit einer Bildsuchmaschine	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	XYZ - PACS Workstation	5
2.4	Osirix	5
2.5	Funktionsumfang von Betrachtungstools	5
2.5.1	Zoom und Scroll	5
2.5.2	Fensterung und Level	5
2.5.3	Histogram Modifikation	5
2.5.4	Negativ bilden	5
2.5.5	Distanz und Flächenmaß	5
2.5.6	Collagen	5
3	Methodology	6
3.1	Verwendete Technologien und Protokolle	6
3.1.1	HTTP	6
3.1.2	REST	6
3.1.3	JSON	7
3.1.4	AJAX	7
3.1.5	Objectiv J	7
3.1.6	Cappuccino	8
3.1.7	WebGL	8
3.2	Funktionalität und Aufbau der Benutzeroberfläche	8
3.3	Kommunikation mit KRESHMOI	8
3.3.1	Query nach Bildern	8
3.3.2	Laden der Bilder	9
3.4	Architektur und Komponenten	10
3.4.1	Domänen Model	10
3.4.2	Architektur und Aufteilung in Komponenten	10
3.4.3	2DView	10
3.4.4	Kommunikations Module	10
3.5	Usability	10

3.5.1	Workflow bei der Befundung	10
4	Results	11
4.1	Geschwindigkeit	11
4.2	Usability	11
5	Conclusion	12

Abstract

1 Introduction

1.1 Motivation

1.1.1 Interface für Kreshmoi

Das Ziel von KHRESMOI ist das Durchsuchen und der Zugang zu medizinischen Informationen für verschiedene Benutzergruppen 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 Ergebnisliste navigieren kann. [?]

1.1.2 Interaktion mit einer Bildsuchmaschine

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 Referenz-Bild 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 Berachtung der Bilder bereitstellen. [?]

Diese Arbeit spezialisiert sich auf 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 Betrachtungs-Tools 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

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 möglich 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 Mobiledevices gar nicht möglich ist. [?]

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. [?]

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. [?]

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 Action-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.[?]

2 Related Work

Software für die Bildsuche in radiologischen Daten gibt es bis jetzt noch nicht, allerdings decken sich die Anforderungen größtenteils 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 Information System (KIS) oder Radiologie Information 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 weitergeben.

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-conforme 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 ausreichende Möglichkeiten zur Erweiterung durch einen Definitionsumschlüssel für alle DICOM Objekte, die sogenannte Image Object Definition.

2.3 XYZ - PACS Workstation

2.4 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 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 Protokoll die Daten auch von einem PACS-Server geladen werden.

2.5 Funktionsumfang von Betrachtungstools

2.5.1 Zoom und Scroll

2.5.2 Fensterung und Level

2.5.3 Histogram Modifikation

2.5.4 Negativ bilden

2.5.5 Distanz und Flächenmaß

2.5.6 Collagen

3 Methodology

3.1 Verwendete Technologien und Protokolle

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 retourniert.

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.

3.1.2 REST

REST ist im eigentlichen Sinn mehr ein Architekturstil als ein Protokoll welches mit HTTP umgesetzt wird. Die Idee von REST ist dass eine URL genau eine Resource auf einem Server adressiert, wobei eine Resource eine statische Datei oder das Ergebnis einer Aktion auf dem Server sein kann.

Der Architekturstil lässt sich durch fünf Prinzipien zusammenfassen:

Resource mit eindeutiger Identifikation:

Jede Resource wird durch eine URI (Uniform Resource Identifier) weltweit eindeutig identifiziert. Diese adressiert unter anderem den Server auf den sich die Resource befindet sowie Resource auf dem Server selbst.

Hypermedia

Verknüpfungen zu anderen Entitäten und werden als Links auf die jeweiligen Resources 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 Resource zur Verfügung gestellt werden muss. Dieses umfasst einen relativ kleinen Satz von Operationen welche auf die Resource ausgeführt werden können.

Unterschiedliche repräsentation der Ressourcen

Die Ressourcen können unterschiedliche Darstellungsformen haben. Ein Client kann also eine Resource 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 Resource umgewandelt werden.

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 vorallem 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.

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 erhalten 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.

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 Rückgaben 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. [?]

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 sowol 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.

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 anlehnt.

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.

3.2 Funktionalität und Aufbau der Benutzeroberfläche

3.3 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 Resoure über eine URL zugegrifen werden kann.

3.3.1 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 weitere interessante Bereiche sogenannte ROIs (Region of Interest) in den einzelnen Schnittbildern definiert. Die Übergabe einer ROI erfolgt als Polygon in Form einer Liste von Punkten im drei Dimensionalen 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 }

```

3.3.2 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 ein 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 Wagrecht 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

3.4 Architektur und Komponenten

3.4.1 Domänen Model

3.4.2 Architektur und Aufteilung in Komponenten

3.4.3 2DView

3.4.4 Kommunikations Module

3.5 Usability

3.5.1 Workflow bei der Befundung

4 Results

4.1 Geschwindigkeit

4.2 Usability

5 Conclusion