Online Medical Imaging Platform

Willinger Christin

Computer Vision Lab
Institute of Computer Aided Automation
Vienna University of Technology

September 4, 2014

Supervisor: Robert Sablatnig

Contents

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 Kentnissen
- Ärzte
- Radiologen

Dazu verknüpft KHRESMOI Daten aus verschiedenen heterogenen Resourcen wie Bildern aus PACS, Bildern und Text aus Publikationen in Journalen oder Daten von Webseiten. Da sich die verschiedenen Resoucen qualitativ sehr stark voneinander unterscheiden können wird einen 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 hiebei 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 duch die Ergebnissliste navigieren kann. [?]

1.1.2 Interaktion mit einer Bildsuchmaschiene

Ein Teilprojekt von KHRESHMOI ist das Druchsuchen 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 Bildsuchmaschiene muss daher sowohl Tools zum markieren von interessanten Bereichen, als auch die Funktionalität zur verfnünftigen Berachtung der Bilder bereitstellen. [?]
Diese Arbeit spezialisiert sich auf das Durchsuchen von radioloischen 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 radionlogische Aufnahmen produziert werden, bietet eine effizientes Durchsuchen dieser die möglichkeit Sie für Ausbildung und Forschug wieder zu verwenden.
Dazu muss das User-Interface die grundliegenden Funktionen eines Betrachtungs-Tools für Röngten- und Computertomographie-Aufnahmen zur verfüfung Stellen:

- Zoom
- Schnelles anpassen von Kontrast und Helligkeit

- Navigation durch die Schnitte eins 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.
- Betrachen 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 verschidenen Ansichten.
- Umsetzung der Applikation im Webbrowser.

1.3 Möglichkeiten zur Umsetzung

Aufgrund der Anforderung dass, das Programm in einem Webbrowser ausgeführter 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 Plichtenheft untersucht.

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 möglicht 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 deisem Ansatz wird die Anwendung in HTML, CSS und JavaScript oder einem Framework welches auf diesen Technologien aufsetzt entwickelt.
Da diese Technologien von fast allen neuen Browsern unterstütz 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 unterstüzt Canvas Elemente für 2D Bilder jedoch ist eine Verarbeitung der Bilder durch die Grafikhardware nur begrenzt möglich. Für die Verwendung der Grafikarte durch den Browser gibt es die Schnittstelle WebGL welche auf OpenGL ES aufbaut. Diese wurde aber noch nicht in allen gängigen Browsern implementiert bzw ist in einigen nocht nicht stabil und muss extra aktiviert werden. [?]

1.3.3 Flash

Flash Anwendungen sind Programme in einem properitären Format deren Lizenz mitlerweile Adobe gehört. Diese werden von einem Interpreter dem Flash-Player ausgeführt, welcher es über einen Browserplugin ermöglich Flashcode in Webseiten einzubinden und mit der Webseite zu interagieren.
Die Erstellung von Anwendung erfolgt in Entwicklungsumgebung Flash wobei Code in der Objektorientieren 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 properitäres Format handelt ist der Support und die Weiterentwickung nicht Sichergestellt.[?]

2 Related Work

2.1 Funktionsumfang von Betrachtungstools

2.1.1 Zoom und Scroll

2.1.2 Fensterung und Level

2.1.3 Histogram Modifikation

2.1.4 Negativ bilden

2.1.5 Distanz und Flächenmaß

2.1.6 Collagen

2.1.7 Beispiel Osirix

3 Methodology

3.1 Verwendetet 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 Kommunikationspartner findet in der From von Nachrichten statt, wobei der Client eine Anfrage an einen Server stellt und dieser die Anfrage bearbeitet und eine Antwort retuniert.
Eine Nachricht setzt sich aus einem Header und einen 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 Sin mehr ein Architekturstiel als ein Protokoll welcher mit HTTP umgesetzt wird. Die Idee von REST ist dass eine URL genau eine Resource auf einem Server addressiert, wobei eine Resource eine statische Datei oder das ergebniss einer aktion auf dem Server sein kann.
Der Architekturstiel ist lässt sich durch fünf Prizipien zusammenfassen:

Resource mit eindeutiger Identifikation:
Jede Resrouce wird durch eine URI (Uniform Resouce Identifier) weltweit eindeutig identifiziert. Diese addressiert unter anderem den Server auf den sich die resource befindet sowie Resource auf dem Server selbst.

Hypermedia
Verknüfungen zu anderen Entitäten und werden als Links auf die jeweiligen Resourcen dargestellt. Weiters kann die Steuertung des Applikationszustandes durch Links auf weiter Aktionen durch Hypermedia umgesetzt werden.

Standard-Opperationen
Es gibt ein definiertes Interface welches von jeder Resource zur verfügung gestellt werden muss. Dieses umfasst einen relativ kleinen Satz von Opperationen welche auf die Resource ausgeführt werden können.

Unterschiedliche repräsentation der Resourcen
Die Resourcen 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 unterstüzt wird. In HTTP wird die gewünschte Dartstellung im Header angegeben.

Zustandslose Kommunikation
Der Server hällt 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 Autausch 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 Resourcensparender sind. Wie bei XML werden Listen und Objekte in einer von Menschen lesbaren Form darestellt. 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 Appliketion Daten vom Server wird an diesem eine HTTP Anfrage gesendet und Callback-Funktionen für den Fall einer Antwort oder eines Fehlers beim Browser registriert. Erhällt der Browser eine Antwort auf sine 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 unterstüzt. 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 eine Datentyp zu definieren, werden diese aufgrund von schwacher Typisierung vom Interpreter nicht auf ihre Einhaltung überprüft. In der aktullen 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 haupsächlich 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 kentnisse 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 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.

3.2.1 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 Resourcen 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 Interesst) in den einzelnen Schnittbildern definiert. Die Übergabe einer ROI erfolgt als Polygon in Form einer Listen von Punkten im drei Dimensionalen Raum des Volumes.

Listing 1: Query

```
{
    "queryid":"",
    "text":"",
    "imageid":"",
    "roi":
    [
        "polygon":
        [
            "point":{"x":0, "y":0, z:0},
        ],

    ]
}
```

Listing 2: Result

```
{
    "rankedImageID":
    [
        {
            "imageID": "9587336b132b127b936ad0afd80ca862",
            "normalDimX": 1,
            "normalDimY": 380,
            "normalDimZ": 488,
            "relevance": 1,
            "report": "",
            "title": "Some Image"
        },
    ]
}
```

3.2.2 Laden der Bilder

Die Bildaten eines Volumes können über eine GET-Operation geladen werden. Dabei wird jedes Volume als eine Resource im Unterverzeichniss */image/* indentifiziert. Über Parameter in der URL werden die Schnittrichtung und und die Nummer des Bildes in der jeweiligen Schnittebene angegeben. Wir 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 entrsprechende der Terminologie für die Bildgebung in der Anatomischen die Optionen:

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

`http://baseurl/image/IMAGE_ID.jpg?slice=SLICE_NO&direction=DIRECTION_KEY`

3.3 Architektur und Komponenten

3.3.1 Domänen Model

3.3.2 Architektur und Aufteilung in Komponenten

3.3.3 2DView

3.3.4 Kommunikations Module

3.4 Usability

3.4.1 Workflow bei der Befundung

4 Results

4.1 Geschwindigkeit

4.2 Usability

5 Conclusion