

## C.A.S.A.M.-project Eindverslag

|                                |                        |
|--------------------------------|------------------------|
| S.E.F. van Berkel (1262882)    | B. Bijl (1312405)      |
| J.Y.T. den Hollander (1268538) | S. Rabbelier (1308211) |
| B.M.W. Sedee (1263153)         | N.N. Smit (1286455)    |

June 8, 2009

# Contents

|  |           |
|--|-----------|
| <b>Voorwoord</b>   | <b>4</b>  |
| <b>1 Samenvatting</b>  | <b>5</b>  |
| <b>2 Inleiding</b>   | <b>6</b>  |
| <b>3 Probleemstelling en Analyse</b>                         | <b>7</b>  |
| 3.1 Probleemstelling . . . . .                               | 7         |
| 3.2 Analyse . . . . .  | 7         |
| <b>4 Planning</b>  | <b>9</b>  |
| 4.1 Voorgenomen planning . . . . .                           | 9         |
| 4.2 Werkelijke planning . . . . .                            | 10        |
| <b>5 Aanpak</b>  | <b>12</b> |
| 5.1 Project gerelateerd . . . . .                            | 12        |
| 5.2 Database systeem . . . . .                               | 12        |
| 5.3 Image Processing . . . . .                               | 13        |
| <b>6 Implementatie</b>                                       | <b>14</b> |
| 6.1 Project gerelateerd . . . . .                            | 14        |
| 6.2 Database systeem . . . . .                               | 14        |
| 6.3 Image Processing . . . . .                               | 14        |
| <b>7 Aanbevelingen</b>                                       | <b>17</b> |
| 7.1 Bestandstypen van foto's . . . . .                       | 17        |
| 7.2 Verschillende kleuren in flash applicatie . . . . .      | 17        |
| 7.3 Opslaan van een lege bitmap . . . . .                    | 18        |
| 7.4 Te grote foto's inladen in de flash-applicatie . . . . . | 18        |
| 7.5 statistische data bij landmarks . . . . .                | 18        |
| 7.6 uitwerken afstand meet applicatie . . . . .              | 18        |
| 7.7 uitgebreid user management en arts-view mode . . . . .   | 19        |
| 7.8 verwijderen measurements e.d. zonder admin . . . . .     | 19        |
| 7.9 morphen naar standaard AnatomicalView . . . . .          | 20        |
| 7.10 zoom op meerdere niveau's . . . . .                     | 20        |
| 7.11 schaalweergave in image . . . . .                       | 21        |
| 7.12 export on unix import on windows . . . . .              | 21        |
| <b>8 Conclusie</b>   | <b>22</b> |
| <b>9 Acknowledgement</b>                                     | <b>22</b> |

|   |           |
|---|-----------|
| <b>10 Bijlagen</b>                            | <b>23</b> |
| 10.1 Oriëntatieverslag . . . . .              | 23        |
| 10.2 Plan van Aanpak . . . . .                | 23        |
| 10.3 Requirements Analysis Document . . . . . | 23        |

## Voorwoord

# 1 Samenvatting

## 2 Inleiding

De afgelopen tien weken hebben wij met elkaar gewerkt aan een webapplicatie voor het C.A.S.A.M.-project . Voor het vervullen van deze opdracht hebben wij op diverse terreinen verschillende uitdagingen moeten overwinnen. Dit ging van de communicatie met experts van het Erasmus Medisch Centrum die geen verstand hadden van de IT-wereld, tot aan het werken met programmeertalen waar niemand ervaring mee had. Uiteindelijk hebben we een product gebouwd dat vele verschillende programmeertalen bij elkaar brengt en op deze manier gebruiken we de verschillende krachten in een nieuwe combinatie. Ook hebben wij het idee dat het product voldoet aan de eisen van het C.A.S.A.M.-project . Dit resultaat hebben wij ook te danken aan het Agile programmeerprincipe waardoor wij veel tussentijds contact hebben gehad met de opdrachtgever en de uiteindelijke gebruikers.

In dit verslag besteden wij aandacht aan de verschillende uitdagingen die voor ons lagen, de oplossingen die we daarvoor hebben gevonden en het research wat we daarvoor hebben moeten doen. In sectie 2 leggen we uit wat de probleemstelling is en daaraan gekoppeld wordt in sectie 3 een analyse gemaakt van het probleem en een oplossingsrichting gedefinieerd. In sectie 4 zullen we ingaan op de planning die we hebben gemaakt, om in sectie 5 in te gaan op de aanpak van ons probleem. Daarna zullen we in sectie 6 een toelichting geven op onze implementatie en een aantal bijzondere aspecten uitlichten. Om daarna in sectie 7 een analyse te maken over de mogelijkheden om het product uit te breiden en te verbeteren.

Bij dit verslag horen enkele documenten welke zijn toegevoegd als bijlage. Voorbeelden hiervan zijn het Plan van Aanpak, het orientatieverslag en het RAD-document van de database applicatie. In het verslag zal op verschillende momenten worden verwezen naar deze documenten.

## 3 Probleemstelling en Analyse

### 3.1 Probleemstelling

Het CASAM-project was op zoek naar een applicatie die hun kon helpen bij de anatomische vraagstukken die zij hadden. Het ging hierbij vooral om hulp bij het warpen en morphen van foto's en de mogelijkheden om op verschillende foto's bijzonderheden aan te kunnen geven en deze met elkaar te vergelijken. Al in het eerste gesprek bleek dat er op dat moment nog geen enkele applicatie lag en dat alle foto's gewoon werden opgeslagen op een harde schijf en dat er out-of-the-box software werd gebruikt voor de foto manipulatie. Op dat moment hebben wij voorgesteld om naast de foto manipulatie ook aandacht te besteden aan de administratie en opslag van het materiaal. Tenslotte kwamen zij ook met de vraag of het mogelijk was om bepaald materiaal dat in het ziekenhuis werd gemaakt, zoals rntgenfoto's, direct te importeren in het systeem. Ook wilde zij graag publicaties die hoorde bij een project erbij willen opslaan.

### 3.2 Analyse

In overleg met het CASAM project hebben we toen afgesproken om het project in drie delen op te delen: een database applicatie, Image processing en tenslotte de koppeling met andere systemen. Daarnaast werd meteen afgesproken dat de project groep op de TU Delft zou werken, dit omdat de facilitaire mogelijkheden in Delft beter zijn dan op het EMC. Om wel voldoende feedback te blijven ontvangen is er gekozen voor een Agile programmeerprincipe, op deze manier zou de opdrachtgever voldoende betrokken blijven bij het proces.

In de eerste week van het project hebben wij een lijst opgesteld van de verschillende functionaliteiten die in het systeem moesten komen. Van te voren was al duidelijk dat niet alle functionaliteiten gebouwd konden worden binnen de scope van dit project. Daarom zijn in samenspraak met de opdrachtgever de functionaliteiten van prioriteiten voorzien, uiteindelijk is daar een diagram uitgekomen, zie hiervoor bijlage A.

Mede op basis van het prioriteiten diagram hebben wij de verschillende uitdagingen gedefinieerd:

- Project gerelateerd
  - Gebrek aan medische en anatomie kennis
  - Gebrek van IT kennis aan de kant van de opdrachtgever
  - Samenwerking in een grote groep
- Database systeem
  - Gebruikersvriendelijkheid
  - Van diverse locaties bereikbaar
  - Eenvoudige manier om data toe te voegen aan het systeem

- Schaalbaarheid
- Image processing
  - Variaties in landmarks
  - Morphen van verschillende foto's naar n gemiddelde foto
  - Verplaatsen en roteren van foto's naar standard AnatomicalView

In de volgende secties lichten wij toe hoe we met deze uitdagingen zijn omgegaan.



## 4 Planning

We begonnen het project met het maken van een planning voor de komende tien weken zodat er voor zowel de opdrachtgever als onszelf duidelijk was wanneer er wat gedaan zou worden. In verband met de Agile methoden zijn de eerste paar weken in meer detail gepland dan de latere weken met de bedoeling om de planning steeds bij te werken voor de komende (paar) weken. Deze voorgenomen planning wordt besproken in paragraaf 4.1. Voor een uitgebreidere beschrijving van de diverse fasen kan het Plan van Aanpak uit de bijlage worden bekeken. Zoals verwacht is deze ruwe planning aangepast in de weken daarna, deze realisatie wordt besproken in 4.2.

### 4.1 Voorgenomen planning

Doordat we in het begin slechts een grove planning hebben gemaakt waren vooral de eerste paar weken in detail uitgewerkt. Er moesten twee documenten af zijn voordat er met de implementatie van het product kon worden begonnen, dit zijn het plan van aanpak (PvA) en het requirement analysis document (RAD). Tevens hebben we ons direct voorgenomen een demo in de eerste weken van het project te maken zodat de opdrachtgever snel feedback kon geven op de richting waar het project in gaat. Aangezien de tentamens op 16/06 beginnen is afgesproken voor deze datum het project af te ronden. In de onderstaande Ganttchart wordt de planning zichtbaar gemaakt zoals wij het aan het begin van het project voorstelden.

#### 4.1.1 Plan van Aanpak

Tijdens ons eerste gezamenlijk overleg is het volgende schema voor het PvA afgesproken, waarbij de deadlines zijn gemaakt met de gelegenheid om verbeteringen door te voeren in het achterhoofd.

09/04: Concept PvA mailen 14/04: Verwerken feedback PvA 15/04: Meeting in erasmus 17/04: PvA af

#### 4.1.2 Requirement Analysis Document

Tevens is voor het RAD een (minder granulaire) planning gemaakt, deze overlapt met de planning voor het PvA gezien het schrijven van het tweede document en het verwerken van verbeteringen aan een ander document tegelijk kunnen worden uitgevoerd.

Nadat de tweede ronde van feedback op het PvA was verwerkt werd begonnen met het RAD, zodat deze zo snel mogelijk afgerond kon worden en aan de implementatie kon worden begonnen. Eventuele feedback op het PvA hierna wordt eerst verwerkt.

27/04: RAD af

#### **4.1.3 Database Applicatie**

Om een demo te kunnen geven waarin zo veel mogelijk geëvalueerd kan worden over de gewenste functionaliteiten was besloten om de twee weken na het inleveren van het RAD te besteden aan het maken van spike solutions en deze te integreren in een simpele applicatie die de basisfunctionaliteiten aan de opdrachtgever presenteert.

08/05: Database Applicatie af

#### **4.1.4 Active Shape Modelling**

Een belangrijk deel van de applicatie is het weergeven van de hoofdmodi van variatie per landmark type en het weergeven van de gemiddelde landmarks.

25/05: Active Shape Modelling af

#### **4.1.5 Image Processing**

Aanvankelijk zagen we deze fase als het projecteren van resultaten op bijvoorbeeld röntgenfoto's. In de loop van het project hebben we de betekenis van deze fase echter verandert in het tekenen van gebieden en morphen van deze gebieden naar een gemiddeld been. Dit is een cruciale fase voor de aangeboden oplossing.

11/06: Image Processing af

#### **4.1.6 Eindverslag**

Het schrijven van het eindverslag moet ruim voor de eindpresentatie afgerond zijn zodat het verslag goedgekeurd kan worden voor aanvang van de eindpresentatie.

12/06 Eindverslag af

#### **4.1.7 Eindpresentatie**

De eindpresentatie is de uiteindelijke afronding van het project.

19/06: Eindpresentatie gegeven

### **4.2 Werkelijke planning**

In deze sectie geven wij aan hoe de opgestelde planning aansluit op de uiteindelijke realisatie.

Zowel het PvA als het RAD zijn op tijd af gekomen, de andere milestones zijn wat gewijzigd. Aangezien verschillende leden van de projectgroep tentamens hebben in de weken vanaf 15 juni is besloten om het project rond deze datum af te ronden. Hierdoor schoven de deadlines voor het schrijven van het eindverslag en het geven van de eindpresentatie op naar voren.

In de originele planning is hierdoor naar 10 juni verschoven en de eindpresentatie zal op 15 juni plaatsvinden. Door een goede scheiding van de verschillende functionaliteiten bleek het goed mogelijk om een werkverdeling te maken

waar iedereen zijn of haar expertisegebieden in kwijt kon. Hierdoor liepen de Database Applicatie, Active Shape Model en Image Processing fasen parallel aan elkaar en werden de laatste werkzaamheden tegelijk afgerond.

In de latere weken van het project bleek ook nog enige tijd nodig om de code te refactoren, bugs te verwijderen en functionaliteiten verder te verbeteren. Dit zorgde ervoor dat de eerdergenoemde drie parallel-lopende fasen wat uitliepen.

In de originele planning was geen rekening gehouden met de tijd die nodig is om de applicatie ook te laten draaien op een server in het Erasmus MC. Hierdoor moet dit tegelijk plaatsvinden met het schrijven van het verslag.

In de onderstaande Ganttchart is te zien hoe de werkelijke planning verliep:

## 5 Aanpak

Een van de eerste beslissingen die moest worden genomen is de architectuur van het systeem. Al snel bleek dat de opdrachtgever een voorkeur had voor een webapplicatie die door het hele Erasmus MC op het netwerk simpel en snel gebruikt kon worden. Voor onze Image Processing kregen wij het advies van Dr. Botha om een aantal Python bibliotheken te bekijken en deze indien mogelijk te gebruiken.

De combinatie van deze feiten leidde ons richting Django, dit is een python framework voor het bouwen van websites in Python. Na wat research en de ervaringen die één van de projectleden al had met Django hebben we ervoor gekozen om Django te gaan gebruiken.

Voor wat betreft de visuele interactie hebben we gekeken naar verschillende Javascript bibliotheken, mede door de uitgebreide ervaringen van diverse leden hebben we uiteindelijk gekozen voor de combinatie van Prototype en Scriptaculous.

Tijdens het project bleek uiteindelijk dat er behoefte was aan een tekenapplicatie binnen de omgeving en deze was niet te realiseren via Javascript en is er uiteindelijk ook nog een Flash applicatie geschreven welke communiceert met Javascript.

### 5.1 Project gerelateerd

In de begin periode hebben wij ons ingelezen op de basis anatomische kennis die nodig was om de experts van het Erasmus MC te kunnen volgen. Hierbij hebben we vooral veel gebruik gemaakt van de kennis van één van de groepsleden, die voor ze aan deze studie begon drie jaar. Ook zijn we een dag langs geweest bij het EMC en hebben daar een rondleiding gehad door het laboratorium en de snijzalen waar onderzoek wordt gedaan.

Een andere uitdaging was de relatief grote groep studenten voor dit project. Om te voorkomen dat we elkaar te veel in de weg zouden zitten hebben we de diverse delen van het project opgesplitst en de taken duidelijk verdeeld. Daarnaast hebben we gebruik gemaakt van software zoals Trac, dit is een Project Management Systeem. Daarnaast was een locatie geregeld op de TU Delft waar we iedere dag hebben gewerkt en waar we elke ochtend een werkoverleg konden hebben voor wat er die dag zou gebeuren. Hierdoor bleef iedereen goed op de hoogte van de status van het gehele project.

### 5.2 Database systeem

Zoals hierboven al aangegeven zouden we gaan werken met Django en moest het beschikbaar worden als webapplicatie. Django is een Model-View-Controller (MVC) opgebouwde taal met een duidelijk abstractie laag voor het database gedeelte. Na diverse brainstorm sessies zijn we uiteindelijk uitgekomen op een klasse diagram wat de basis zou gaan vormen voor het database systeem. Het diagram staat in bijlage B. Voor de interface hebben we ervoor gekozen om veel

gebruik te maken van Javascript en zijn bibliotheken. Dit gaf ons eenvoudige toegang tot diverse visuele effecten, zoals drag en drop, fade en de sliders). Daarnaast konden we met behulp van Javascript gebruik maken van de AJAX techniek, hiermee konden server requests sturen en de pagina updaten zonder dat de gebruiker hier last van had.

### 5.3 Image Processing

De doelen op Image Processing gebied waren het in kaart brengen van de variaties per landmark en het morphen van getekende structuren (bijvoorbeeld zenuwen of venen) naar een standaardview waarop bijvoorbeeld structuren die van verschillende benen afkomen toch op één been kunnen worden afgebeeld.

Om het eerste doel te bereiken hebben we gebruik gemaakt van een subonderdeel van een volledig Active Shape Model, namelijk het Point Distribution Model. Het Active Shape Model is ontwikkeld door Tim Cootes, in de paper *An Introduction to Active Shape Models* licht hij de theorie toe.[6]

Waar het Active Shape Model zich bezighoudt met het automatisch herkennen van structuren uit een foto, is het Point Distribution Model[5] in staat gegeven een aantal landmarks (van hetzelfde type) op verschillende foto's het gemiddelde en de hoofdassen van variatie weer te geven. Een landmark is hierbij een punt dat zo is gekozen dat het in iedere foto te vinden is en duidelijk op ieder fotovoortbeeld te vinden is. De landmarks die nodig zijn voor het Point Distribution Model moeten hierbij informatie geven over het verloop van een structuur, bijvoorbeeld het aangeven van het verloop van een bepaalde vene.

Om het tweede doel te bereiken, het morphen van getekende structuren is er gebruik gemaakt van een Thin Plate Spline transformatie. Om de Thin Plate Spline transformatie uit te voeren zijn landmarks nodig die bepalend zijn voor de vorm van een object. Hierbij kan gedacht worden aan een set van landmarks die bestaat uit bony landmarks (landmarks die voortkomen uit de anatomie van het menselijk lichaam en voelbaar zijn vanaf de buitenkant), landmarks die de grenzen van het object aangeven en landmarks die zich op gelijke afstanden tussen voorgaande landmarks bevinden.

Impliciet zit er in dit doel nog een Image Processing doel verwerkt, namelijk het aangeven van lijnen, gebieden of punten die interessant zijn voor de onderzoeker. We hebben ervoor gekozen om de onderzoeker dit real-time te laten doen in de applicatie door gebruik van Flash.

Over de theoretische achtergrond van de verschillende gebruikte technieken is meer te vinden in het oriëntatieverslag. De implementatie komt aan bod in de volgende sectie.

## 6 Implementatie

In de tweede week van het project zijn we echt begonnen met het bouwen van het systeem. We zijn begonnen met zogenaamde spike solutions (snelle simpele oplossingen om een deelprobleem op te lossen) van verschillende basis functionaliteiten, zoals het uploaden van images en het plaatsen van landmarks geplaatste images met behulp van drag & drop. In een later stadium hebben we voor de grotere uitdagingen spike solutions geschreven, hieronder vallen bijvoorbeeld het morphen van plaatjes en het visualiseren van het Point Distribution Model. Als een spiked solutions werkte werd deze aan het bestaande systeem gekoppeld, op deze manier hadden we in een vrij korte tijd een basis systeem met haar basis functionaliteiten. Zo konden we ons vervolgens concentreren op de Javascript functies welke zouden gaan zorgen voor de AJAX afhandeling en de visuele effecten.

### 6.1 Project gerelateerd

Daar waar het mogelijk en handig was hebben we geprogrammeerd in tweetallen. Tijdens de implementatie fase hebben we veel gebruik gemaakt van de beschikbare whiteboards, deze waren handig voor het brainstormen en als iemand vast liep dan kon het probleem getekend en uitgelegd worden. Dit alles zorgde voor een goede synergie en hield iedereen de motivatie die hij had goed vast. Ook tijdens de implementatie fase hebben we veel contact gehad met de leden van het C.A.S.A.M.-project en zijn er verschillende tussentijdse presentaties geweest. Steeds hebben we na deze presentaties een uitgebreid overleg gehad over de stand van zaken en geëvalueerd over hoe het project er voor stond. In sommige gevallen hebben we ook onze prioriteitenlijst aangepast aan de hand van deze vergaderingen.

### 6.2 Database systeem

Lorem ipsum

Voorbeeld van de implementatie van de bitmap creator python-javascript-html-flash-javascript-python

Op het eind van het project hadden we veel bereikt. Maar ook is er werk blijven liggen. Voorbeelden hiervan zijn: de ViewMode voor artsen en de analyse van statische data.

### 6.3 Image Processing

Dit was een van de grootste uitdagingen van het project. Niemand had ervaring met de verschillende bibliotheken of uitgebreide ervaring met dit aspect van Image Processing. Er waren verschillende opties om de benodigde technieken te implementeren, zo konden we ze zelf implementeren met behulp van Numpy[2], of gebruik maken van grotere libraries zoals Insight Segmentation and Registration Toolkit (ITK)[1] of de Visualization Toolkit (VTK)[4]. Verder was er nog de

Python Imaging Library (PIL)[3], die geschikt is om eenvoudige beeldbewerking te doen. Uiteindelijk is er, mede op advies van Dr. Botha, gekozen voor PIL in combinatie met de VTK library. Om aan de vraagstelling te kunnen voldoen (het in kaart brengen van de variaties per landmark en het kunnen projecteren van getekende gebieden op n gemiddeld plaatje) hebben we gekozen om een deel van de methode Active Shape Modeling, genaamd Point Distribution Models, toe te passen in combinatie met Thin Plate Spline transformaties. Hierdoor is het mogelijk voor de onderzoeker om op een wiskundige verantwoorde manier anatomische structuren op een gemiddelde te projecteren en de variaties per landmark te visualiseren. Al snel bleek dat er veel research nodig was om de theorie achter Principal Component Analysis, Active Shape Models, Point Distribution Models en de Thin Plate Spline Transformaties te doorgronden. Ook kostte het enige extra tijd om met de VTK bibliotheek zelf bekend te raken.

### 6.3.1 Point Distribution Model

Om het Point Distribution Model te maken is een set nodig van verschillende landmarks waarvan de variaties en de gemiddelden interessant zijn om weer te geven. Het is hierbij van belang om de landmarks zo te kiezen dat ze makkelijk zijn te vinden in elke foto en belangrijke punten aangeven op de structuur waar het om gaat. De gebruiker kan landmarks in de foto aangeven door eerst een landmarktype aan te maken, dan een landmark van een bepaald type en hierbij aan te geven of het een shapedefining landmark is of niet. Nadat de gebruiker in de foto's de gewenste landmarks heeft aangegeven kan hij door 'Analyse selected landmarks' aan te klikken het Point Distribution Model van de geselecteerde landmarks uitrekenen en weergeven. Op dit moment wordt in javascript gecontroleerd welke images en landmarks geselecteerd zijn en wordt via een ajax request de id's van de image en de geselecteerde landmarks per image doorgegeven. Aan de hand van de id's van de image en de landmarks worden vervolgens de bijbehorende objecten uit de database gehaald. In de view wordt getest of de geselecteerde landmarks wel geschikt zijn om een Point Distribution Model van te maken. Dit is bijvoorbeeld niet zo als er geen landmarks geselecteerd zijn, als er te weinig landmarks geselecteerd zijn van een type of als de geselecteerde landmarks van een geheel ander type zijn. In elk van deze gevallen wordt er een foutmelding aan de gebruiker geretourneerd in de vorm van een JavaScript alert met de aard van de fout, zodat de gebruiker zijn selectie kan aanpassen. Als echter de juiste landmarks zijn geselecteerd worden allereerst via de coördinaten van de landmarks per image een vtkUnstructuredGrid aan gemaakt met de desbetreffende punten erin als vertices. Vervolgens wordt over deze grids de Generalized Procrustes Analysis uitgevoerd via het vtkProcrustesAlignmentFilter. Dit zorgt ervoor dat de gevonden modi van variatie onafhankelijk van de positie en de rotatie van de objecten zijn. Door ervoor te kiezen om de mode op RigidBody te zetten blijft echter wel de grootte van de structuren behouden. De opgeslagen grids zijn nu iteratief getransleerd en geroteerd naar elkaar toe. Nu is het tijd om de Principal Component Analysis (PCA) op de aligned grids uit te voeren, dit is gedaan met

behulp van het `vtkPCAAnalysisFilter`. Uit het resultaat kunnen de gemiddelde posities per landmark gehaald worden. We hebben er voor gekozen om de eerste twee modi van variatie te berekenen, de hoofdmodus van variatie (eerste mode) zorgt altijd voor de grootste verandering en is de eigenvector die uit de PCA komt met de grootste eigenwaarde. Om deze variaties te berekenen vragen we de `GetParameterisedShape` en berekenen we hiermee voor de eerste en tweede modus de extremen, dat wil zeggen plus en min 3 standaarddeviaties van het gemiddelde. Nu we de benodigde coördinaten hebben (gemiddelde en extremen) kunnen we beginnen aan de visualisatie van de landmarks. Dit wordt gedaan met behulp van de PIL. Aan de hand van de afmetingen van de originele images en de gegeven coördinaten word voor het gemiddelde een ellipse getekend. Tussen de extremen van de twee hoofdmodi van variatie wordt een lijn getekend. De lengte van deze lijn geeft de grootte van de mogelijke variatie aan op basis van de voorbeelden uit de gegeven landmarks. Door de ellipsoids en lijnen weer te geven met een doorzichtige achtergrond is het vervolgens mogelijk om deze als een overlay over de images te projecteren. De overlay word opgeslagen in de database is beschikbaar voor weergave vanuit de interface.

### **6.3.2 Thin Plate Spline Transformatie**

#### **6.3.3 Teken**



## 7 Aanbevelingen

Hieronder een korte toelichting op werkzaamheden waar wij niet aan toe zijn gekomen. Sommige van deze werkzaamheden zijn niet gebeurd omdat ze buiten de scope van het project vielen of tijdens het project bedacht werden. In enkele gevallen hadden we wel gepland om het werk te doen maar zijn we er niet aan toe gekomen. In de toelichting geven we ook een korte samenvatting van wat er in brainstorm sessies en/of vergaderingen over is afgesproken. Wij hopen op deze manier dat toekomstige groepen op een eenvoudige en snelle manier kunnen verder werken aan het product.

### 7.1 Bestandstypen van foto's

Op dit moment kan de morph-functie alleen foto's aan die het bestandstype jpeg hebben. Alhoewel dit het meest voorkomende type is zou het een verbetering zijn als het systeem ook met andere typen om kan gaan. Een probleem hierbij is, dat de foto's bij het uploaden wel de extensie 'jpg' krijgen, maar het eigenlijk nog geen jpeg foto's zijn. Het kan nu bijvoorbeeld gebeuren dat er een gif-afbeelding wordt geupload, die vervolgens de extensie jpg krijgt. Hierdoor geeft de vtkJPEGReader een error, en kan er met dit plaatje niet gemorphed worden. Het weergeven van de afbeelding in de browser gaat echter wel goed, waardoor het voor de gebruiker niet duidelijk is dat er eigenlijk iets mis is met de afbeelding.

Een oplossing zou kunnen zijn om bij het uploaden van de afbeelding de Python Imaging Library te laten kijken naar de foto, en deze om te laten zetten naar een echt jpeg bestand. Het gevaar hierbij is echter, dat er door de compressie van het jpeg formaat het een en ander aan scherpte van de foto verloren gaat. Een eventuele oplossing voor dit probleem zou kunnen zijn om helemaal geen jpeg-bestanden meer op te slaan, en alle foto's te laten omzetten naar het PNG-formaat. Bij deze omzetting gaat vrijwel geen informatie in de foto verloren, en voor dit formaat zijn dezelfde VTK-library's beschikbaar als voor het jpeg formaat.

Dit is echter een probleem waar wij binnen het project zodanig laat tegenaan liepen, dat er geen tijd meer was om dit op een fatsoenlijke manier op te lossen.

### 7.2 Verschillende kleuren in flash applicatie

Binnen onze applicatie is het op dit moment al mogelijk om een bitmap te tekenen in een willekeurige kleur. Wat echter nog niet mogelijk is, is om in één bitmap meerdere kleuren op te slaan. Voor de gebruikers zou het echter intuïtiever zijn om dit wel te kunnen, vooral omdat er bij het tekenen van de bitmap wel met verschillende kleuren gewerkt kan worden. Het moet de gebruiker dan expliciet verteld worden dat de hele bitmap wordt opgeslagen in de kleur waarmee het laatst is gewerkt.

### 7.3 Opslaan van een lege bitmap

\* alle correcties weggooien en geen wijziging opslaan? \* lege bitmap opslaan?

### 7.4 Te grote foto's inladen in de flash-applicatie

### 7.5 statistische data bij landmarks

Een bekende wens vanuit de gebruikers bij het EMC is dat er statistische data opgeslagen kan worden bij de aangegeven landmarks. Deze data heeft dan bijvoorbeeld betrekking op de diepte waarop een vene op verschillende punten in het been ligt. In de huidige situatie wordt deze data apart in een Excel-sheet bijgehouden, en naar een statisticus gebracht. Deze maakt hier vervolgens in aparte software (SPSS) een aantal grafieken van, die de gebruikers eigenlijk bij de andere data op zouden willen slaan. In de toekomst zou het mogelijk moeten zijn om deze data bij de landmarks zelf op te slaan. In een ideale situatie zouden zelfs de grafieken in het systeem moeten kunnen verschijnen, waarna de gebruikers van een willekeurig punt op de grafiek de waarde op kunnen vragen.

Vanwege de complexiteit van het opslaan van de statistische data, het later genereren van de bijbehorende grafieken, en het kunnen opvragen van de waarden op een willekeurig punt, hebben wij hier nog geen stappen voor ondernomen. Een volgende groep zou kunnen beginnen met het enkel toevoegen van de data, en deze overzichtelijk weer te geven, zonder hier grafieken van te maken. De analyse van deze data, en het hieruit produceren van bruikbare data, is daarna weer een probleem apart, wat zelfs voor een eerste opzet waarschijnlijk een heleboel onderzoek benodigt. Het enige licht wat wij nu al een beetje op de zaak kunnen laten schijnen, is dat er een PyCha module bestaat, waarmee in Python grafieken gemaakt kunnen worden.

### 7.6 uitwerken afstand meet applicatie

Vanuit het EMC was de wens opgekomen om het mogelijk te maken om in foto's afstanden te meten. Dit zou dan gebruikt kunnen worden bij een deel van de statistische data, zoals dit hierboven al beschreven is. Aangezien een van de eerste vragen van het EMC was hebben we hier al even naar gekeken, en een zgn. 'spiky solution' voor gebouwd. Hier zat echter nog een niet acceptabele meetfout in, waardoor de functie niet in de uiteindelijke interface terecht is gekomen. Er waren echter wel een aantal manieren waarop de nu aanwezige meetfout waarschijnlijk teruggedrongen kon worden:

**verplaatsen lineaal** Op de foto's van het EMC is een lineaal zichtbaar. Echter deze lineaal ligt op de verkeerde positie ten op zichte van waar gemeten wordt, er is namelijk een diepte verschil. Ook ligt de lineaal niet recht onder de camera, dit zorgt voor een hoek in de kalibratie stap.

**fout zichtbaar maken** Op dit moment wordt er wel gerekend en wordt de fout op een basis manier uitgerekend. Deze fout zou zichtbaar moeten gemaakt worden aan de gebruiker, zodat hiermee rekening kan worden gehouden.

**opslaan kalibratie** Elke keer als een foto wordt geopend moet de kalibratie stap opnieuw worden uitgevoerd. Hierdoor ontstaat mogelijk onzorgvuldigheid en eigenlijk moet de kalibratie stap worden opgeslagen.

**automatische herkenning** Als we de lineaal kunnen laten herkennen door een algoritme dan kan de kalibratie stap geautomatiseerd worden en wordt de menselijke factor verkleind.

## 7.7 uitgebreid user management en arts-view mode

Wat betreft het huidige user management zijn er nog vele verbeteringen mogelijk. Momenteel zijn er namelijk 3 gebruikersgroepen aanwezig, die niet altijd evengoed gecontroleerd worden. Zo is het nu voor een arts ook mogelijk om landmarks toe te voegen aan een project, terwijl dit niet zou moeten mogen. Ook moet het toekennen van rechten aan projecten beter en duidelijker gaan verlopen. Daarnaast zou een gebruiker ten alle tijden zijn eigen wachtwoord aan moeten kunnen passen, terwijl dat nu alleen door een 'beheerder' gedaan kan worden.

Zoals al aangegeven is er voor het user management al wel een begin gemaakt, maar dit moet eigenlijk nog uitgebreid worden. Een deel van deze uitbreiding zou ook in kunnen houden dat er nauwer wordt samengewerkt met de permissions die in Django aan gebruikers toegekend kunnen worden, omdat hier eigenlijk nog niets mee gebeurt.

Zelf zijn wij door tijdsdruk hier niet goed meer aan toegekomen, en is het eigenlijk blijven liggen, ondanks dat een degelijke user management in eerste instantie wel een van de eisen het EMC was.

## 7.8 verwijderen measurements e.d. zonder admin

Wat betreft het verwijderen van een aantal objecten uit het systeem, is er ook nog grote winst te behalen. Zo wordt voor het verwijderen van de volgende objecten nu nog de Django-admin-interface gebruikt:

- Bitmaps
- Landmarks
- PDMs
- Projects
- Tags

Van deze objecten wordt er nu alleen voor projecten ook de mogelijkheid geboden om deze via de normale interface te verwijderen, terwijl het eigenlijk het netst is, om alle objecten via de interface te kunnen verwijderen.

Toen wij hier zelf over aan het nadenken waren, kwamen wij tot de conclusie dat een aantal oplossingen waren die allemaal eigenlijk niet zo goed gingen werken. De oplossingen waar wij aan gedacht hebben, staan hier onder beschreven.

**Inline delete** Een van de mogelijkheden was om in de interface achter elke landmark en elke bitmap een rood kruisje te plaatsen. Hier hebben wij vanaf gezien, omdat wij vonden dat de gewone gebruikersinterface dan eigenlijk veel te vol werd. Verder staan er eigenlijk nergens in de interface kruisjes, en hebben we overal zogenaamde 'managers' voor

**Bij de image manager** Omdat de landmarks en bitmaps onderdeel zijn van de afbeeldingen, zou het misschien bij de image manager erbij kunnen. Dit zou dan in kunnen houden dat er eerst op de afbeelding geklikt moet worden, waarna er een lijstje met landmarks en bitmaps verschijnt die verwijderd kunnen worden. Nadeel hiervan is dat de image manager eigenlijk niet de plek is om dit soort dingen in te regelen, omdat hij puur bedoeld is voor de afbeeldingen.

**Delete mode** De radicaalste oplossing was om een 'delete mode' toe te voegen aan de interface. Het idee hierbij was dat je op een knop kon drukken, waarna er eigenlijk een nieuwe interface geladen werd. In deze nieuwe interface kon je dan eigenlijk alles naar een prullenbak slepen, waarna het verwijderd werd op het moment dat je deze 'delete mode' weer verliet. Dit klonk op zich als een heel leuk idee, ware het niet dat dit heel veel tijd zou gaan kosten. Daar kwam nog eens bij dat we niet zeker wisten of dit wel een idee was wat aan zou slaan bij de gebruikers, en of dit voor de gebruikers wel echt intuïtief was.

**Eigen manager** Een eigen manager voor de landmarks en bitmaps was waarschijnlijk het beste idee, maar door de tijdsdruk zijn we ook hier niet aan toe gekomen. Het enige wat eigenlijk tegen was op deze oplossing was het feit dat er dan nog een link bij zou komen in de interface. Gekscherend werd er gezegd dat er bijna behoefte was aan een manager voor de managers, omdat het er inmiddels al zo veel zijn.

## 7.9 morphen naar standaard AnatomicalView

### 7.10 zoom op meerdere niveau's

Een leuke en handige uitbreiding voor de nu aanwezige zoom-functie in de interface, is om het niveau van de zoom in te kunnen stellen. In het zoom scherm wordt het huidige plaatje nu twee keer zo groot weergegeven als dat het in werkelijkheid eigenlijk is, maar idealiter is dit in te stellen naar een willekeurige grootte. Om dit te bereiken is het wellicht het makkelijkst om te beginnen met het zoomen op een aantal vastgestelde niveau's, zoals 150%, 200%, 250% en 300%. Dit kan dan later altijd nog uitgebreid worden naar een de gebruiker in te stellen niveau, binnen een aantal vastgestelde grenzen.

De reden waarom wij dit nog niet hebben aangepakt, is dat het voor ons een te groot project zou worden. Ook zitten er aan de huidige implementatie nog wat haken en ogen voordat het op deze manier kan werken, omdat nu het huidige plaatje in zijn geheel op 200% wordt ingeladen in dat frame. Als je dit uit gaat

breiden naar een aantal vast niveau's is het misschien wat betreft de load nog wel haalbaar de afbeelding op deze niveau's alvast op te halen, maar voor een willekeurig niveau is dit onhaalbaar. Een ander alternatief is om de afbeelding met het gewenste zoomniveau op te halen nadat het niveau is ingesteld door de gebruiker. Het nadeel hiervan is dat één afbeelding dan heel vaak opgehaald moet worden, voordat de gebruiker het gewenste niveau heeft gevonden. Een derde optie was om de afbeelding zichzelf te laten vergroten in het zoom-kader, afhankelijk van het niveau. Het grote nadeel van deze oplossing was dat een afbeelding momenteel wordt ingeladen met een grote die afhankelijk is van de grote van zijn container. Als deze dus klein is, wordt er standaard een foto met heel weinig detail ingeladen.

### **7.11 schaalweergave in image**

### **7.12 export on unix import on windows**

## 8 Conclusie

In de afgelopen tien weken hebben wij een systeem gebouwd dat gebruikt kan worden door het CASAM-project binnen het EMC ten behoeve van hun onderzoek naar de anatomie. In onze ogen voldoet het systeem aan de eisen die de opdrachtgever ons gegeven heeft. In dit project hadden wij op verschillende terreinen grote uitdagingen. Van project management tot het verwerken van de Image processing in onze applicatie. Wij zijn van mening dat de meeste van deze uitdagingen gewonnen zijn en dat zij hebben meegedragen aan een bijzonder product. Zoals eigenlijk bij elk ICT-project is er zeker nog ruimte over voor uitbreiding en verbetering van het huidige product. Deze vervolg werkzaamheden zouden in dezelfde of andere samenstelling kunnen plaatsvinden. Een volgende groep zou zelf moeten kijken in welke richting zij door willen, uiteraard kunnen de punten die genoemd worden in dit verslag als een uitgangspositie worden gebruikt voor de planning van deze werkzaamheden.

## 9 Acknowledgement

Wij willen graag een aantal mensen en organisaties bedanken voor hun hulp en bijdrage aan het welslagen van dit project: Dr. Botha voor de begeleiding van de kant van de TU Delft en zijn hulp met de visualisatie bibliotheken. Dr. KleinRensink en Dhr. Kerver voor hun begeleiding van de kant van het CASAM project Ing. Slats voor het beschikbaar maken van een zaal en andere faciliteiten aan de Drebbelweg waar wij tien weken konden werken.

## References

- [1] Itk insight toolkit, 2009. <http://www.itk.org/> [Online; accessed 15-april-2009].
- [2] Numpy, 2009. <http://numpy.scipy.org/> [Online; accessed 15-april-2009].
- [3] Python imaging library (pil), 2009. <http://www.pythonware.com/products/pil/> [Online; accessed 07-june-2009].
- [4] Vtk visualisation toolkit, 2009. <http://www.vtk.org/> [Online; accessed 15-april-2009].
- [5] Tim Cootes. Statistical shape models. <http://personalpages.manchester.ac.uk/staff/timothy.f.cootes/Models/pdms.html> [Online; accessed 07-june-2009].
- [6] Tim Cootes. An introduction to active shape models. In *Image Processing and Analysis*, pages 223–248. Oxford University Press, 2000.

## 10 Bijlagen

### 10.1 Oriëntatieverslag

### 10.2 Plan van Aanpak

### 10.3 Requirements Analysis Document