```
01000011 01101000 01100001 01110000
01110100 01100101 01110010 00001010
00110011 00101110 00110011 00101110
00110010 00001010
```

Chapter 3.2

‘COMPUTER ASSISTED SURGICAL ANATOMY MAPPING’ Advanced description of the web-based version

S.E.F. van Berkel, B. Bijl. J.Y.T. den Hollander, S. Rabbelier, B.M.W. Sedee, N.N. Smit.

Bachelor project nr. IN3405. Supervisors: A.L.A. Kerver, C.P. Botha, G.J. Kleinrensink, M. Sepers.

INTRODUCTION

The benefits of a web-based version of CASAM are numerous. Most importantly it gives the researcher full control over the algorithms used for averaging and warping. When commercial software, such as MagicMorph is used the exact algorithms and computations cannot be legally looked at directly and the final renditions can only be verified using conventional anatomy research. Furthermore, when all steps of CASAM are included in the web-site it provides a fully integrated system that allows for quick computations and rendition making without the need for any commercial software such as Photoshop and MagicMorph. This means that all aspects of CASAM are fully controlled by algorithms chosen by the developers and/or researchers.

A further obvious benefit of a web-based application is that it is accessible from anywhere. Working on your anatomy research from the comforts of your home whilst not needing a high-end computer would be possible since all calculations are done server-side. Also, all data is saved encrypted and backed-up server-side. A globally accessible anatomy database, with standardized ways for adding new data, would in theory be the necessary basis for international collaboration between anatomy labs.

Therefore the aim of this Bachelor project is to integrate each step necessary for CASAM into one web-based application, using only open-access libraries and algorithms. The principle idea is that the user can create a clinically relevant anatomy project. After dissecting and photographing multiple specimens in the dissection room he can add the images to that project. When the images are uploaded the user is guided through the different steps of CASAM; averaging, warping and making renditions.

IMPLEMENTATION.

The system was build using so called “spike solutions”(quick and simple solutions to subproblems) to build the different basic functions such as uploading images and placing landmarks on images using drag & drop. Spike solutions were also written for more specific functions such as the warping of photos and visualizing a point distribution model. Well-working spike solutions were then added to the existing system and basic functions.

Interface.

The design of the interface is of great importance as it mainly facilitates the interaction with the users; A well designed interface enhances the user`s productivity. JavaScript and its libraries are mainly used for the interface. This allowed for easy implementation of visual effects such as drag & drop, fade, and the use of sliders. By using JavaScript it is also possible to use AJaX techonologies, allowing server request and page updates to be send without actual user input.

The main interface entails a left and right menu and centrally there is an area which contains the images (Figure 1). Both menu`s are divided into specific areas in which different tabs are loaded. Since most of the tabs are not used simultaneously the tabs are foldable.

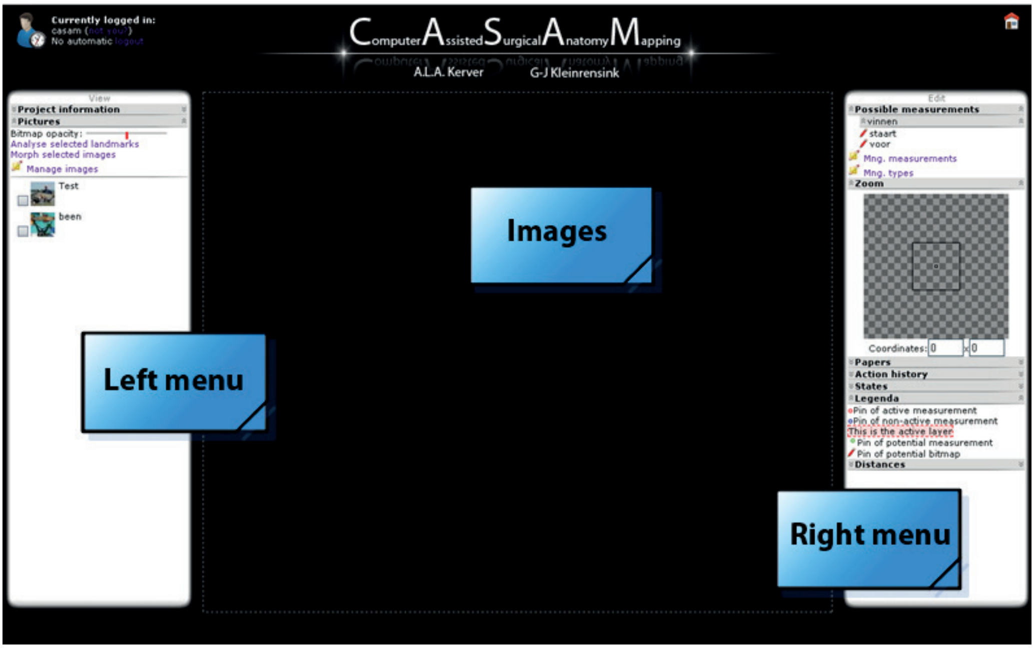


Figure 1. main interface design

The left menu shows project related settings and a list of project linked images with accompanying bitmaps and landmarks. The right menu shows tools for adjusting the images that are loaded in the left menu. The upper right tab contains measurements that can be applied to the currently active image. The next tab shows a magnified part of the active image, allowing for accurate landmark placement and measurements. Then there are tabs such as "action history" to undo changes and tabs for "relevant papers", "States", "Legend" and "Distances".

The interface also provides several popups in which certain objects are managed. One window allows for adding and deleting images. There also are popup windows for “Measurements” and “Classifying Measurements”. Popup windows need to be closed before work in the original window can be continued.

Visual effects were added to the interface for a fluid workflow. The effects provided by Scriptaculous are used for effects such as slide down tabs and fade-in popups. Also, in the left menu, added landmarks are briefly colored yellow to show that they have been added.

To manage different projects a project selection page was made (Figure 2) in which current projects can be selected and new projects can be added. Small thumbnails are shown for each project which change every few seconds, giving a slideshow of all images of the project.

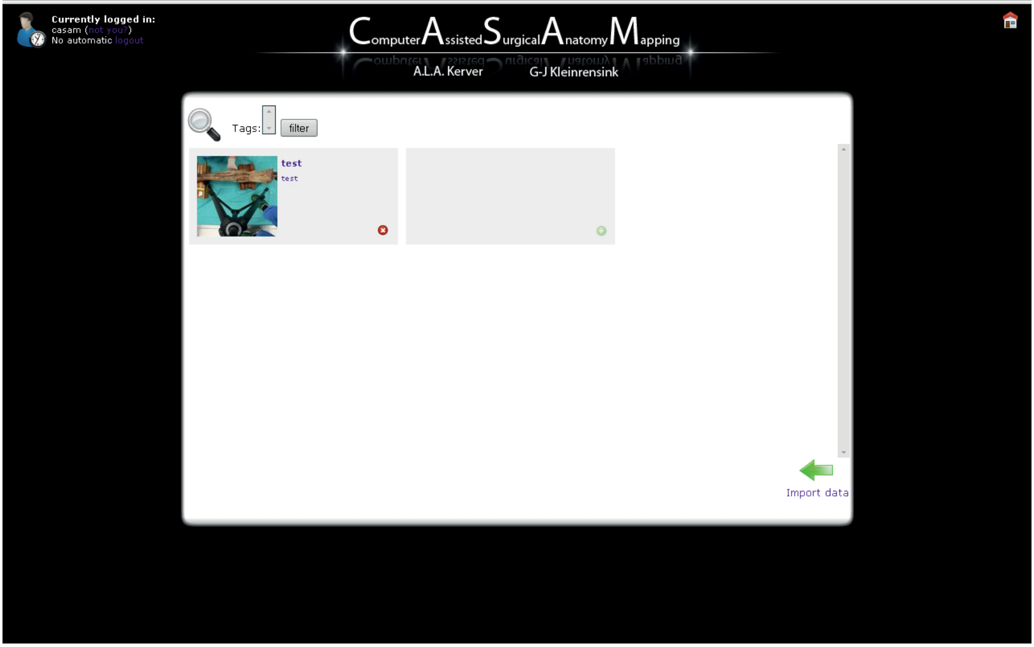


Figure 2. Project Selection page.

DATABASE SYSTEMS

First the database model needed to be made (Figure 3). Since CASAM needs to be an easy to implement web-application and Python libraries are needed for image processing Django was chosen as the framework for this application. Django is a python framework that can be used to built websites and web applications. It is a Model View Controller designed language, which means that the datamodel and functionality regarding this datamodel are separated from the user-interface and functionality regarding the user interface. Django also has a clear abstraction layer for the database component, the Object-Relational Database Mapping. This allows for object-oriented programming with a direct mapping of created objects to a relational structure. As created objects are mapped to the relational database, implementation of these objects is simplified significantly. The class diagram of the database system is shown in figure 3.

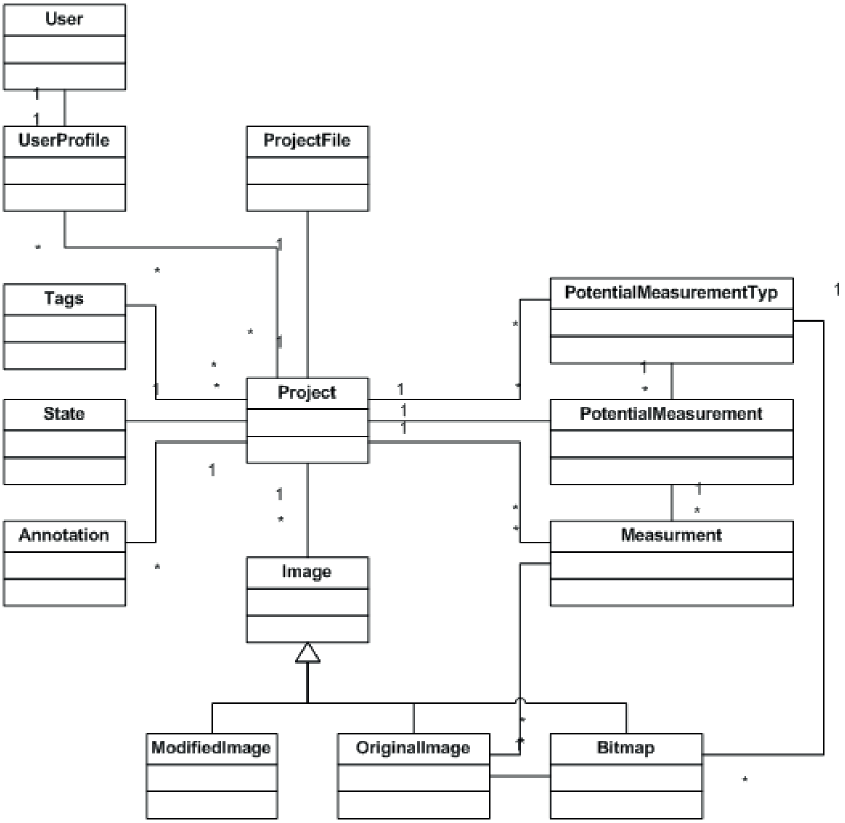


Figure 3. Class diagram of the database

ASPECTS OF THE DATABASE SYSTEM

Show multiple transparent images simultaneously

To show different highlighted renditions of anatomy it is imperative that multiple images, showcasing these renditions can be visualized simultaneously. The user needs to be able to change the order of images and their transparency individually. Therefore a slider is created for each image in each project. However the slider only loads when the image is made visible by the user. The user can then, using the slider, change the transparency of the corresponding image. The slider is derived from the Scriptaculous library. To change the order of images, a Sorttable was created in the left frame of the interface. The order of images can be changed by drag & drop of each individual image. The order of the images is automatically updated when a new image is added to the list. The image on top of the list also automatically is the "active image", meaning that computations are done on this image.

Landmarks

Landmarks are stored separately on another model in the database. This model is linked to three parts of the database: "potential measurements", the image the landmarks is part of and the project the image itself is part of. To save a landmark it fist needs to be created. When an image is loaded a listener is linked to it by JavaScript. The listener itself is removed when a new image is loaded and thus becomes the "active image".

The listener is activated when the image is clicked on (to create a landmark) and a popup window shows information on the landmark. In this popup window a PotentialMeasurement needs to be chosen to which that landmark belongs to, and an invisible ID of the image is added. When the landmark is saved (click on Save) the JavaScript code is checked for that specific landmark. If the landmark already exists a 'repositioning'-change is made in which the new coordinates are saved. Then an AJaX call is made to the server checking if that specific landmark of that specific image already exists. If it does exist it is retrieved and changed. If it does not exist, the landmark is added. In both cases the landmark is given back to JavaScript using a JSON-string. When the landmark is saved to the server the change is also saved in JavaScript. Finally, for an existing landmark, the position of the landmark is adjusted and a new "measurement" is created and placed on the correct new coordinates of the image.

Highlighting relevant parts of an image; Bitmaps

The flash-application is used to draw bitmaps over existing images to highlight relevant anatomy and anatomical areas. Figure 4 shows the sequence diagram of the flash application.

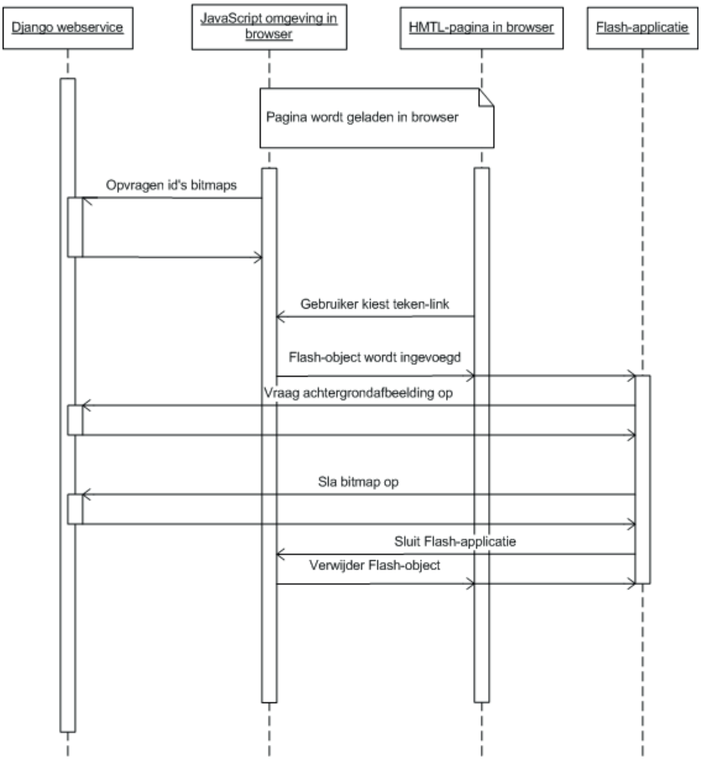

Django webservice
JavaScript omgeving in browser
HMTL-pagina in browser
Flash-applicatie
Pagina wordt geladen in browser
Opvragen id's bitmaps
Gebruiker kiest teken-link
Flash-object wordt ingevoegd
Vraag achtergrondafbeelding op
Sla bitmap op
Sluit Flash-applicatie
Verwijder Flash-object

Figure 4. Sequence diagram of flash application.
The first part of the diagram shows how existing bitmaps are loaded in the JavaScript-environment. The rest of the diagram shows how the system reacts when the user chooses to either draw or adjusts a bitmap.

First, in the HTML-page, an object is added in which the flash application is loaded. Then the flash-application loads the image upon which is to be drawn. Both the size of the brush (the drawing tool) and the color can be adjusted. This allows not only for accurate highlighting of relevant anatomy such as nerves and arteries but also complete areas of for instance muscles can be colored.

When the user chooses to save the bitmap, the drawing is send to the server (Figure 5). If the drawing is successfully saved the Flash-application is given a confirmation and asks the JavaScript-environment to close the Flash-application.

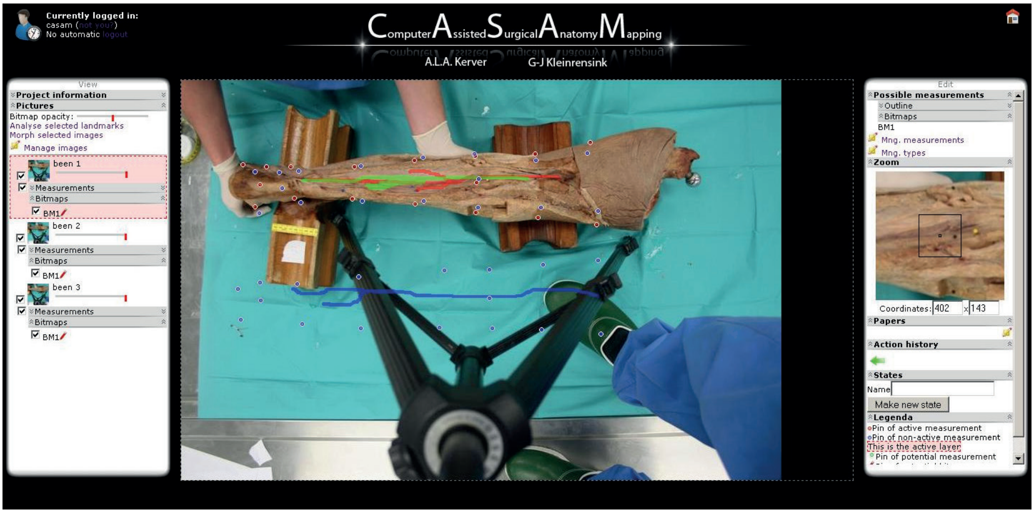


Figure 5. Bitmaps drawn using the Flash application.

Adding relevant papers and websites

Papers and websites relevant to each specific project are saved in a separate model of the database. Papers can be stored locally and the link to a website consists of an URL. All relevant background can be accessed by the user under the “paper” tab.

Adding relevant Tags to a project

In order to categorize projects, Tags or Keywords need to be added. Especially in larger databases with numerous projects tags or keywords help users to access specific information. Also, by using tags, relations between different projects can be identified; If multiple projects for instance focus on a specific nerve the projects can be merged. When a new project is created the user can choose from a selection of existing tags. New tags can only be added to the project when the project is already created and open. This entices users to apply existing tags, thus promoting uniform tags that help link multiple projects.

Static states

States are a way of saving certain interesting sets of images, landmarks, measurements or bitmaps. A researcher can create a state, showing the nerves and arteries of 20 dissected specimen. A surgeon can then for example instantly load this state over the ankle of his patient; this makes data easily accessible to certain groups of users (surgeons/course participants) whilst other users (researchers) determine what exactly is shown. Currently when requested by the user, a part of the HTML structure is saved as a state. This static state can then be opened as a pop-up.

Export datasets

As projects need to be interchangeable between different research departments (and databases) created objects are saved with Universal Unique ID`s. Since each object now has a unique ID it is possible to exchange datasets between multiple databases with a minimal chance of conflict. Administrators of databases are able to export and import these datasets. Datasets are exported to Zip-files and can either be imported automatically using the web-application or manually, using the script in the ZIP-package.

IMAGE PROCESSING

With CASAM variations of multiple specimens can be compared by warping individual specimen to match the average shape. One of the biggest aspects of the project therefore is Image Processing. The required techniques for a web-based version of CASAM could be implemented in multiple ways; Numpy[1], or bigger libraries such as Insight Segmentation and Registration ToolKit (ITK)[2] or the Visualization ToolKit (VTK)[3]. Also Python Imaging Library (PIL)[4] could be used for simple image processing. To combine the best of both worlds PIL was used in combination with the VTK library. The authors decided that to warp anatomical structures in a mathematically sound manner using part of the method for Active Shape Modelling[5,6], namely Point Distribution Models, in combination with Thin Plate Spline transformations[7].

Point distribution model

The goal of a point distribution model is to create an average and show the two main modes of variation, from a certain set of points. The user can select landmarks for each specimen (=image) and give each landmark a specific number and type. When all landmarks are selected for all specimen the user can click “Analyse selected landmarks” and the point distribution model of the selected landmarks is computed (Figure 7). JavaScript verifies which images and landmarks are selected and the ID`s of each image and all landmarks are passed on through an AJaX request. Using these ID`s the associated objects are then retrieved from the database. Then it is analysed if landmarks are suitable for a point distribution model. Points are not suitable for a point distribution model if not enough points are selected or if points are of a different landmark type.

When the correct landmarks are selected a vtkUnstructuredGrid is made using the coordinates of the landmarks in each image. The vertices (a special kind of point that describes the corners or intersections of geometric shapes) represent the landmarks selected by the user. Then a Generalized Procrustes Analysis (a statistical analysis that compares the shapes of objects) is conducted using the vtkProcustesAlignmentFilter. By using the RigidBody mode, only rigid transformations (Chapter 2, translation, rotation, reflection) are used for the alignment of landmarks between each image.

The aligned grids are then put through a Principal Component Analysis (PCA) using the vtkPCAAnalysisFilter. This analysis reduces the variation of multiple landmarks into both an average point and linearly uncorrelated variables called principle components (Figure 6).

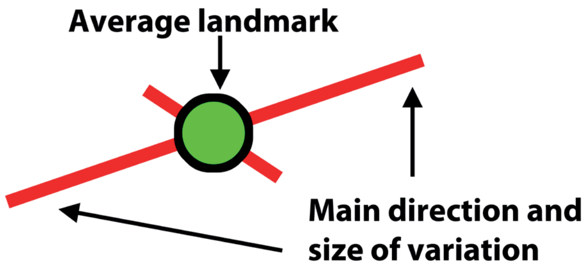
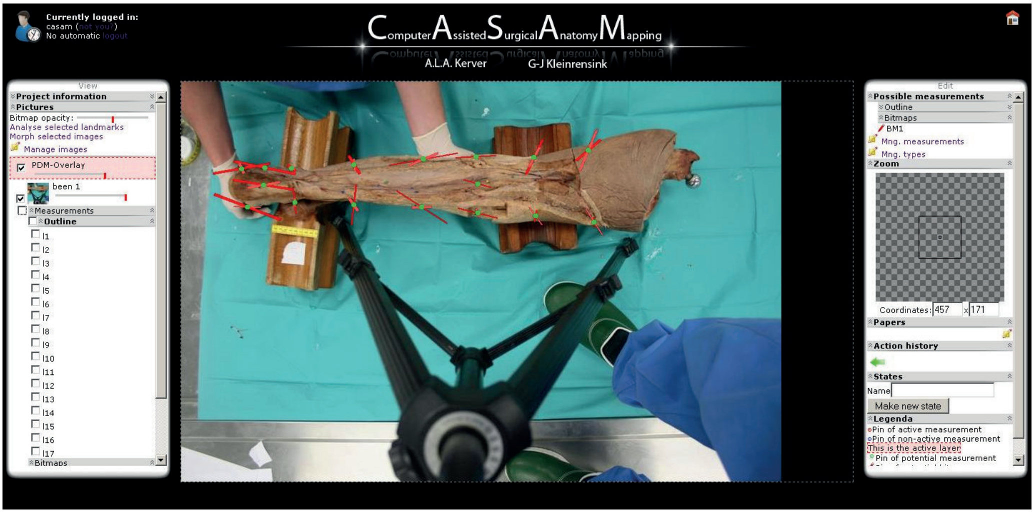


Figure 6. Point distribution model.
Average landmark and both the first and second principal component.

The first principal component has the largest possible variance and accounts for as much of the data variability as possible. The second principle component in turn has the highest variance possible as long as it is uncorrelated with previous components. In other words the green point represents the average of all landmarks of that type (for instance the lateral malleolus of 20 specimen) and the largest line represents both the direction and size of the largest (principle) variance (Figure 6 and 7). To compute these principal components the GetParameterisedShape is calculated (= three standard deviations from the mean). Visualisation of the computed data is done in PIL. Using the dimensions of the original image and the computed coordinates an ellipse is drawn. Then a line is drawn between the extremes of the ellipse, representing the two main modes of variation (principle components). The drawn lines and average landmarks are then projected on a transparent background so they can be visualized over one of the original images (Figure 7).



Currently logged in:
casam (not you?)
No automatic logout

Coordinates: 457 x 171

Figure 7. Point distribution model of all landmarks of three different specimen.
Green dots: average landmark. red lines: first and second principle component (first and second direction and value of variation).

Thin plate spline transformation.

Similarly to the Point Distribution Model landmarks are loaded and read from the interface. Besides the normal landmarks checks as described for a PDM, for a warp all selected landmarks need to be shape-defining (ie. not measurement landmarks, or landmarks representing specific anatomical locations not used for warping). The shapedefining landmarks can be either bony or non-bony. The images and the accompanying coordinates of the landmarks are instantly transformed to vtkPoints objects. First al images and landmarks need to be aligned;l The vtkPoints are used in the vtkLandmarkTransform. vtkLandmarksTransform distinguishes between "source"and "target". The "source"are the original coordinates of the landmarks of each image. The "target" can either be an average shape (created by the point distribution model) or a specific image (of a patient for example). Using vtkImageReslice the actual transformation is computed. To keep the quality of drawn structures the Nearest Neighbor method is chosen as the interpolation-mode. Similar to the Point Distribution model RigidBody mode is selected, so that images are aligned using only rigid transformations. The resulting grids are then used for the actual warp. The warp is implemented using vtkThinPlateSplineTransform and the “source” are the images and landmarks that have been aligned previously. Again the “target” can either be an average shape (created by the point distribution model) or a specific image (of a patient for example). The end-result is shown using vtkImageReslice: all images, drawn bitmaps and landmarks are warped to match the “target” (Figure 8).

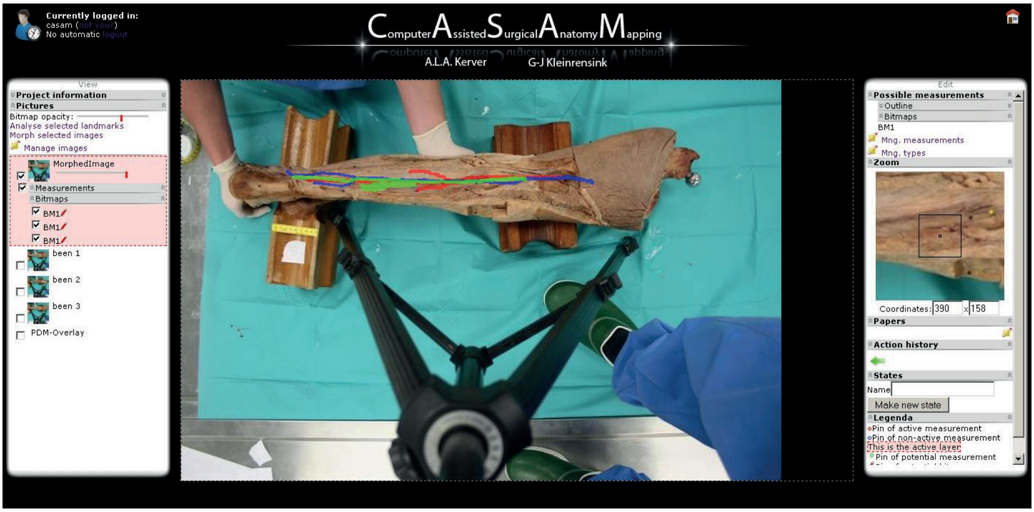


Figure 8. Warping.
“Morphed image” shows the warped bitmaps of three different specimen to match the shape of the first specimen. (the course of the short saphenous vein of three specimen is visualized over specimen nr.1)

CONCLUSION

With the web-based version of CASAM it is currently possible to perform all the necessary steps for CASAM online, using just your internet-browser; The images of multiple, photographed and dissected specimen are uploaded into a certain project. Then predefined landmarks can be placed which are either used for measurements or the warping process. Relevant anatomy can be highlighted in each individual specimen using the flash drawing application. An average shape of all the specimen can be computed using the Point Distribution Model. All drawn bitmaps, representing highlighted and relevant anatomy, can be warped to either match the average shape or the shape of a specific specimen (such as a patient). The technologies used in the web-based version of CASAM are summarized in figure 9.

Computations are done server-side, meaning that users with low-end computers will still be able to use CASAM and no commercial software is needed. If necessary, researchers from all over the world can be allowed to access and contribute to a project and the Export function allows for streamlined export and import of datasets between different databases.

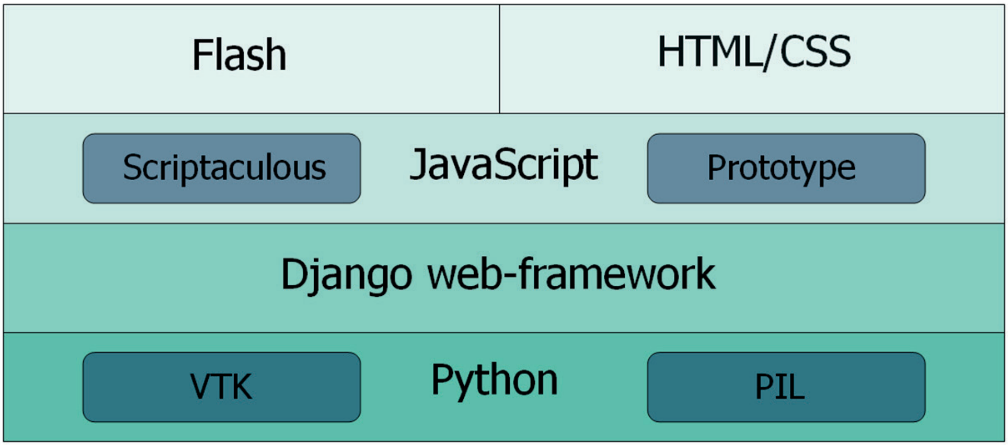


Figure 9. Technologies used for the web-based version of CASAM.
VTK: Visualization ToolKit[3] . PIL: Python Imaging Library[4].

APPLICATION IN PERSONALISED TEACHING

The web-based version of CASAM is especially well suited for an application in one of the main Pillars of modern surgery ; Personalized teaching. It is currently used in Anatomy and dissection programs at the Erasmus MC department of Anatomy and Neuroscience and benefits both residents and experienced surgeons. Using the web-based version of CASAM over the original CASAM means that whole process is much faster and the end-result is obtained quickly. Participants of a course on calcaneus (heel-bone) fractures for instance draw an incision line for a lateral approach of the ankle. The incision is either drawn using a skin-marker or made using a scalpel on an actual specimen of an ankle . The ankles with all drawn incision lines are then quickly photographed and loaded into the web-based version of CASAM. Since the dissected relevant anatomy of nerves and arteries is already loaded into the project their anatomy can be easily related

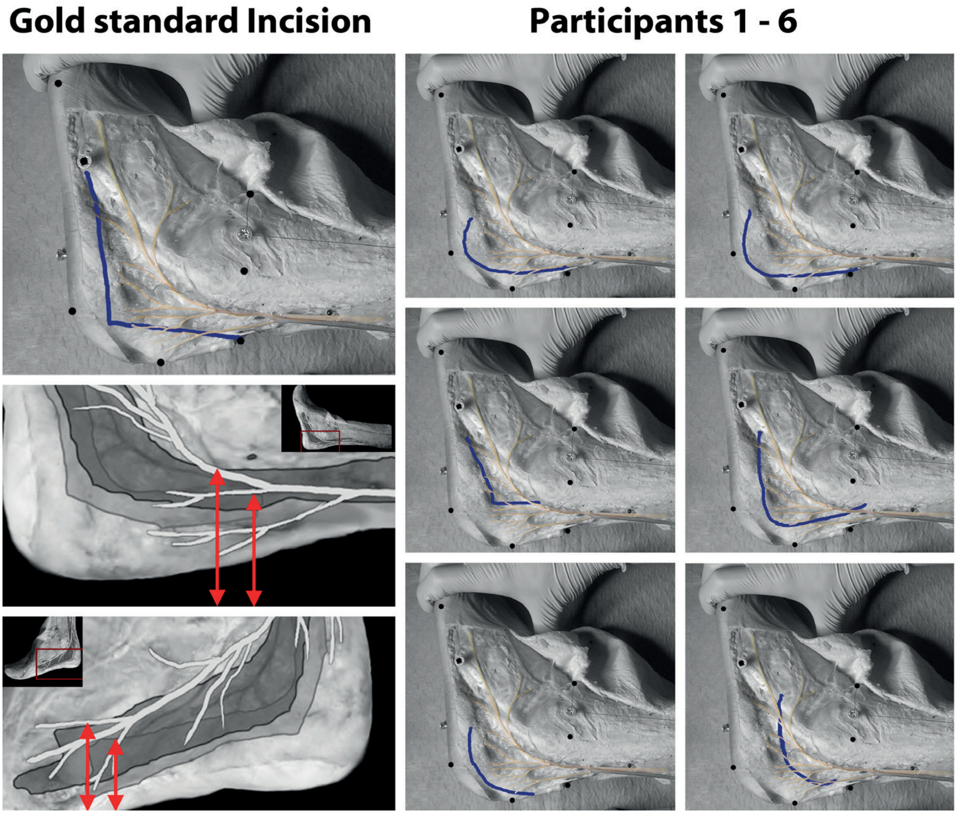


Figure 10. Incision lines of 6 participants related to the “gold standard”.

to each individually drawn incision line. Within an hour (and in the future within minutes) each individual course-participant has personal feedback on the incision line he drew himself on an actual ankle. The feedback can not only consist of the "gold standard incision line" (Figure 10) but also the location of 20 dissected nerves and or 20 dissected arteries (Figure 11).

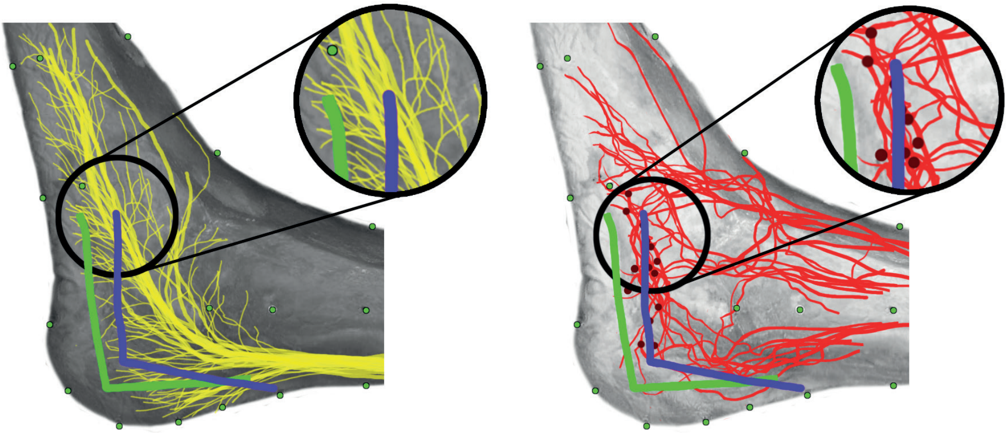


Figure 11. Examples of personalized feedback.
incisions of two participants in relation to the sural nerver (yellow) and main arterial blood supply of the lateral ankle (red). participant 1: good incision line (green). participant 2, bad incision line (purple). Notice how the incision line of participant two has more probability of damaging the sural nerve or the perforating artery (dark red dots).

REFERENCES

1) Numpy, 2009. http://numby.xcipy.org/ [Online; accessed 15-april-2009]

2) Itk insight toolkit, 2009. http://www.itk.org/ [Online; accessed 15-april-2009]

3) Vtk Visualization Toolkit, 2009. http://www.vtk.org/ [Online; accessed 15-april-2009]

4) Python imaging library (pil), 2009. http://www.pythonware.com/products/pil/ [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.

7) Bookstein, F.L., 1989. Principal warps: Thin-plate splines and the decomposition of deformations. In IEEE Transactions on pattern analysis and machine intelligence 11-6.

3