

M.Sc. Thesis
Master of Science in Engineering

 **DTU Compute**
Department of Applied Mathematics and Computer Science

Classification of the anatomy on 3D scans of human heads

Nicolas Tiaki Otsu (s072254)

Kongens Lyngby 2014



DTU Compute

**Department of Applied Mathematics and Computer Science
Technical University of Denmark**

Matematiktorvet

Building 303B

2800 Kongens Lyngby, Denmark

Phone +45 4525 3031

compute@compute.dtu.dk

www.compute.dtu.dk

Abstract

In order to create full 3D surface models of human heads, a state-of-the-art method involves the use of 3D scanner systems, such as the Canfield Scientific Vectra M3 3D-scanner. By utilizing a built-in protocol, partial scans can be formed. The following process of stitching together the partial scans involves a *semi-automatic* alignment in which salient key points are manually annotated and used for alignment. This process is time consuming and involves a learning curve for the person doing the annotations.

The present work presents a method for automatically assigning anatomical feature labels to the surface vertices of frontal human face scans. Randomized decision forests with weak classifiers have been used as classification models. The models have been trained with a novel method for computing three dimensional vertex feature descriptors, called *tangent plane features*.

The author has been provided with an active shape model of frontal human faces which is based on scans from 641 test persons from the Danish Blood Donor Study. This has been utilized to generate a large dataset of plausible frontal surface shapes.

The results from the work indicate that the classification of the randomized decision forests is enhanced when the feature computations are based on an area around each surface vertex of up to 10% of the diagonal of the shape bounding box. Setting up a framework for multiple, single-scale investigation has proven to give good insights into the tuning parameters of the randomized decision forests. Cascading classifiers did not improve the results but heuristics for a method that could improve them have been made.

In one of the conducted experiments, the classification of anatomical regions on frontal human face scans by the use of tangent plane features as weak classifiers for training randomized decision forests has yielded an average accuracy of 95% on an independent test set.

Resumé

En moderne metode til dannelse af komplette 3D-overflademodeller af menneskehoveder involverer brugen af 3D-skanningssystemer, såsom Canfield Scientific Vectra M3 3D-skanneren. Ved at udnytte en indbygget protokol kan partielle skanninger skabes. Den efterfølgende proces med sammenklipping af de partielle skanninger involverer en semiautomatisk justering i hvilken fremstående pejlemærker manuelt annoteres og bruges i justeringen. Denne proces er tidskrævende og involverer en læringskurve for personen der udfører annoteringerne.

I den foreliggende afhandling præsenteres en metode til automatisk at kunne tildele anatomiske særprægsmærkater, eller *feature labels*, til de tredimensionelle overfladepunkter på frontale menneskelige ansigtsskanninger. Randomiserede beslutningsskove med svage sorteringsmaskiner, eller *classifiers* er blevet brugt som klassifikationsmodeller. Modellerne er blevet trænet med en ny metode til at beregne tredimensionelle punkt-særprægs-beskrivere, kaldet *tangent plane features*.

Til sin rådighed har forfatteren fået stillet en active shape model af frontale menneskeansigter som er baseret på skanninger fra 641 testpersoner fra Det Danske Bloddonorstudie. Dette er blevet anvendt til at generere et stort datasæt af plausible frontale ansigtsformer.

Resultaterne fra arbejdet indikerer at klassifikationen via de randomiserede beslutningsskove forbedres når featureberegningerne er baseret på et område omkring hvert overfladepunkt på op til 10% af diagonallængden af den mindste kube der kan indeholde formen. En omstrukturering af systemet til at kunne foretage multiple enkelt-skalaundersøgelser har vist sig at give god indsigt i parameterjusteringen af de randomiserede beslutningsskove. Gennemførslen af en kaskade af classifiers forbedrede ikke resultaterne, men der er i den sammenhæng blevet skabt en heuristik der muligvis ville kunne forbedre dem.

I et af de udførte eksperimenter har klassifikationen af anatomiske regioner på frontale menneskeansigtsskanninger ved brug af tangent plane features som svage classifiers i forbindelse med træning af randomiserede beslutningsskove i et tilfælde ført til en gennemsnitlig prædiktionsnøjagtighed på 95% på et uafhængigt testsæt.

Preface

This master thesis was prepared at the Image Analysis and Computer Graphics section at the DTU Compute Department of Applied Mathematics and Computer Science at the Technical University of Denmark in fulfilment of the requirements for acquiring the degree Master of Science in Engineering (M.Sc.Eng). The work amounts to 30 ECTS points.

Kongens Lyngby, April 14, 2014

A handwritten signature in black ink, appearing to read 'Nicolas Otsu', written in a cursive style.

Nicolas Tiaki Otsu (s072254)

Acknowledgements

I am very thankful to my three academic supervisors, associate professor Rasmus Reinhold Paulsen, associate professor Line Katrine Harder Clemmensen and PhD student Stine Harder for their inspiration and guidance throughout the project.

A warm thank you is also directed towards PhD student Jannik Boll Nielsen who has been a steady and supportive help at the weekly Friday meetings.

I thank Jens Fagertun for providing me with the active shape model that has been used in the analysis.

I also thank Abhishek Jaiantilal for the Matlab mex/standalone interface for classification based random forests which has been made available through GPLv2 license.

Lastly, I deeply thank my beloved partner, Astrid, for her serenity and for convincing me that, indeed, nothing was impossible.

Contents

Abstract	i
Resumé	iii
Preface	v
Acknowledgements	vii
Contents	ix
1 Introduction	1
1.1 Background	1
1.2 Motivation	2
2 Previous Work	7
3 Data	9
3.1 The Active Shape Model	9
3.2 Synthesizing data from the Active Shape Model	11
4 Methods	19
4.1 Anatomical labelling	19
4.2 The Tangent Plane Features are the 3D Vertex Features	19
4.3 Randomized Decision Forests as a Classification Approach	30
5 Results and Evaluation	39
5.1 Procedure for reporting the results from the analysis	39
5.2 Experiment overview	41
5.3 Results and evaluation of experiments	41
6 Discussion	65
7 Conclusion	67
8 Future Work	69
Bibliography	71

Introduction

The reader is introduced to the background and motivation that lays the foundation for the work in the thesis.

1.1 Background

By using a semi-automatic approach, today, three dimensional surface models of human heads can be created and represented as a triangular mesh. Typically, the person is exposed to a multiple camera set-up and a partial 3D surface model is created. By rotating the person and repeating the process for various degrees of rotation, the partial 3D surface models can be merged, or *stitched*, together into a very detailed, full 3D surface model.

In the process of stitching partial 3D surface models together, a crucial requirement is that there must be an overlap between neighbouring partial scans. In the context of aligning frontal head surface scans, overlaps can be determined on the basis of consistent *anatomical regions*, such as the nose, eyelids, chin, and so on. By manually annotating salient features that overlap on each partial scan, they can be stitched together.

Harder et al. [HPL⁺13] have proposed a framework for creating full 3D human head surface models based on partial scans. Their system set up involves the use of a Canfield Scientific Vectra M3 3D-scanner system, which can be seen in Figure 1.1. An important aspect of the approach is that it involves a stitching process. This involves manual annotation of so-called anatomical "sub-parts" that are consistent in that they do not vary much between partial scans.

In order to annotate the anatomical surface regions and points of interest on the human head, a program such as Sumatra [Pau] can be used. It is short for "The Surface Manipulation and Transformation Toolkit" and has been developed by Rasmus Reinhold Paulsen and DTU Compute. Figure 1.2 shows how manual annotation has been used to subdivide a triangular mesh into 24 facial regions, or classes.

In the present thesis, a leap is taken onwards from the manual annotation procedure by investigating the possibility of developing a system that is based on implementing randomized decision forests that enable automatic anatomical classification on frontal human surface models.



Figure 1.1: The Canfield Scientific Vectra M3 system set up.

1.2 Motivation

Manual annotations of 3D points on a head surface model requires both time, anatomical knowledge and human precision. Whereas the presumed lower fail rate may enable humans to compete with computers on the two last-mentioned requirements, the time factor is where they are outshined by the computational capacities of computers. The Microsoft Kinect has a body-part recognition system for single depth images that is based on the works of Shotton et al. [SSK⁺13]. Here, randomized decision forests are used to achieve object recognition in 5 ms on the Xbox GPU, which enables real-time body part segmentation. Figure 1.3 shows an example of a depth image in which darker grey scale colours signify low distance to camera and associated body-part segmentation.

Motivated by the fast body-part recognition system developed by Shotton et al. a frontal face region classification method will be the aim for the present thesis work. Taking offset in a frontal face active shape model, some significant differences between the approach of Shotton et al. [SSK⁺13] and the approach in the present thesis are tied to the following points:

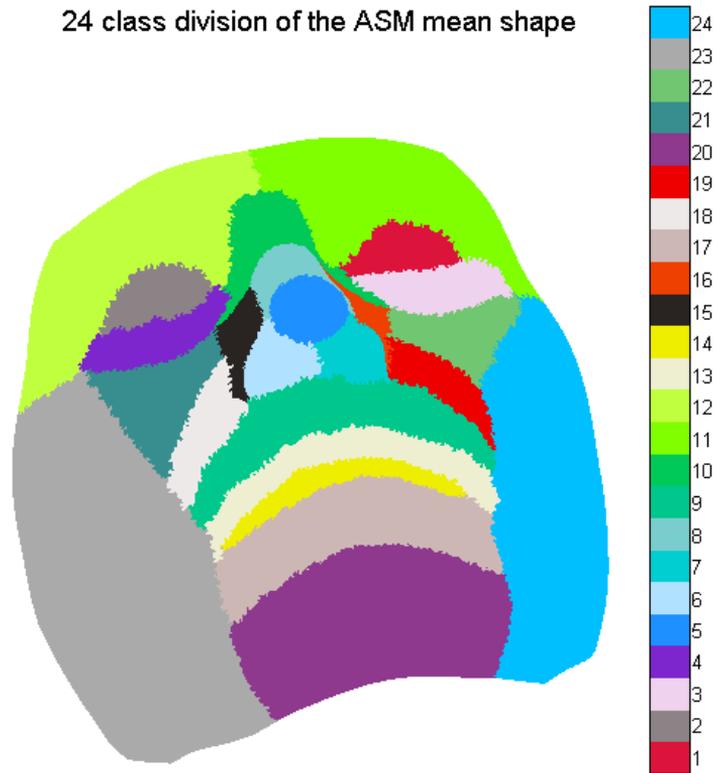


Figure 1.2: Subdivision of a triangular mesh into a total of 24 classes, or *ground truth labels* based on manual annotation in Sumatra [Pau].



Figure 1.3: Example of a depth image (left) and corresponding body-part segmentation based on randomized decision forests as described by Shotton et al. [SSK⁺13].

1. Shotton et al. are using single depth images to classify body part distribution. They are also using low-computational-cost per-pixel depth image feature responses as weak classifiers when training their randomized decision forest. The present work will work on 3D vertices and the weak classifiers will be focusing on computing various so-called *tangent plane features*, which is a novel concept.
2. Shotton et al. are synthesizing a large training data set of poses from a motion capture ("mocap") database. Each pose is randomized to simulate random span in weight, height, camera noise, camera pitch and roll, hair style and clothing. The present work will utilize a frontal face active shape model provided by the Image Analysis and Computer Graphics section at DTU to synthesize training data based on a mean shape whose facial regions will be manually annotated. By propagating these annotated vertices and randomizing the principal component perturbations, the resulting data set covers the range of all possible facial appearances and expressions that can possibly be created by the active shape model.

Ideally, the functionality that is achieved can be utilized within scientific areas within:

-
- Advanced hearing aid shell shape modelling based on simulation of sound wave collision with the intricate features of the human head surface.
 - Driver awareness programs for real-time detection of sudden attention deficits arisen within a vehicle driver.
 - Surgical planning via 3D-models of human faces or full heads to enhance surgical decisions or even aid surgical robots.

Previous Work

The main inspiration for the present work is founded upon the Microsoft Kinect body-part labelling framework by Shotton et al. [SSK⁺13]. The authors are using randomized decision forests to classify pixels in input depth images with respect to what body part they belong to. This approach is based upon 3D representation of shapes and do not suffer from the difficulties of various illumination conditions and texture and colour variations that image based methods are exposed to. They are training randomized decision forests based upon sampling from a synthesized training set of static *motion-capture* (mocap) poses. An important factor is that each pose has a body-part labelling that is consistent across the entire mocap database. Each sample is then subject to randomized variational perturbation based upon parameters such as character height, weight, clothing, pose, rotation and translation, as well as camera camera position and orientation. Each perturbed shape is then graphically rendered into depth images and a corresponding *ground-truth* body-part scheme, see Figure 1.3. By combining weak classifiers based on single pixel-wise depth image intensity readings, they train randomized decision forests that enables body-part classification with an accuracy that greatly supersedes the accuracy of each weak classifier.

Approaches towards localization of facial features in depth images acquired from facial 3D scans from the Xbox Kinect has been implemented by Fanelli et al. [FDG⁺13]. The authors are using a patch based depth image voting system to determine head pose and facial features. Fanelli et al. [FDVG13] have expanded this approach to incorporate active shape models, thus enabling them to align depth and intensity images of unseen faces by fitting a 3D model to them.

In the context of locating facial features in 2D images, several approaches are worth mentioning. Using Haar-like features, Cootes et al. [CILS12] are training a random forest based regression-voting scheme to locate feature points in 2D images. Cevikalp et al. [CTF13] are using a cascade of classifiers based on roots and parts detectors to locate faces and face parts in 2D images. By using head pose as a conditional *global face property*, Dantone et al. [DGFVG12] are training regression forests to locate facial features in 2D images. Zhu et al. [ZR12] are using mixture-of-tree models to detect faces, estimate pose and facial landmarks on both standard face image databases as well as "in the wild" Flickr images. Yang et al. [YP13] are using a star graph based structured-output regression forest approach to localize individual face parts.

In this section the origins of the data is described and how it is utilized to form the data that the present analytical work is based upon.

3.1 The Active Shape Model

Fagertun et al. [FAHP13] and Fagertun et al. [FHR⁺13] describe how they have recorded 3D facial scans of 641 volunteers of the Danish Blood Donor Study [PEK⁺12]. The Canfield Vectra M3 Imaging System, shown in Figure 1.1, has been used to capture and create the 3D facial scans. Each scan contains between 70,000 to 100,000 vertices consisting of Cartesian (x,y,z) coordinates with corresponding RGB (red-blue-green) colour intensity values. The scans have been cropped to only show the faces with the neck, ear and hair regions being excluded. By using a method described by Blanz and Vetter, [BV03], the cropped scans have been restructured such that there is so-called point-to-point correspondence between them. The cropping and restructuring processes have brought the common number of vertices for all scans to 39,653.

Fagertun et al. [FHR⁺13] have created both a 2D active appearance model of the texture information in the shapes (Cootes et al. [CET⁺01]) and also, a 3D active shape model (Cootes et al. [CTCG95]) describing the 3D vertex coordinate variations. The active appearance model framework originates from the active shape model architecture and is not utilized in the present work. Based on the description of how to build an active shape model from a set of 2D-hand images by Stegmann and Gomez [SG02], the following is a description of how the current model is built from the 3D face scans.

Generalized Procrustes Shape Alignment

Each of the $N = 641$ facial scans are built up as a vector \mathbf{x} of $(3 \times n, 1)$ elements, in which the number of vertices is $n = 39,653$:

$$\mathbf{x} = [x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n, z_1, z_2, \dots, z_n]^T \quad (3.1)$$

The centroids, or center-of-mass for each shape, are calculated as

$$(\bar{x}, \bar{y}, \bar{z}) = \left(\frac{1}{n} \sum_{j=1}^n x_j, \frac{1}{n} \sum_{j=1}^n y_j, \frac{1}{n} \sum_{j=1}^n z_j \right), \quad (3.2)$$

and each shape is now translated and rescaled based on the centroids and the centroid sizes

$$S(x) = \sum_{j=1}^n \sqrt{(x_j - \bar{x})^2 + (y_j - \bar{y})^2 + (z_j - \bar{z})^2}. \quad (3.3)$$

After subtracting the mean shape from each shape (centering around zero) and dividing by the centroid size (scaling to unit length), what remains to complete the Procrustes alignment, is the removal of rotational artefacts. This can be done by selecting a reference shape, for instance, the first shape and aligning the remaining $N - 1$ shapes to this. Conceptually, a shape $x_j, j = \{2, 3, \dots, N - 1\}$ is aligned to the reference shape x_1 by computing the singular value decomposition (SVD) of $x_1^T x_j$, thus maximizing the correlation between the two shapes. By using the notation $UDV^T = SVD(x_1^T x_j)$, the rotation matrix $VU^T = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$ can be obtained. This is then used to superimpose x_j to x_1 in the most optimal way by $x_j \leftarrow x_j * VU^T$.

The *Procrustes mean shape* can now be computed as the mean of the Procrustes aligned shapes:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i. \quad (3.4)$$

Eigenvalue Decomposition of the Procrustes Mean Shape

A new set of variables y , as well as its mean \bar{y} , can now be made through a linear transformation of the data vector in Eqn. 3.1 by the formulas

$$y = Mx \quad (3.5)$$

$$\bar{y} = M\bar{x}. \quad (3.6)$$

The shape covariance matrix for the Procrustes mean shape in Eqn. 3.1 and the covariance matrix for y in Eqn. 3.5 can be represented by

$$\Sigma_x = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(x_i - \bar{x})^T \quad (3.7)$$

$$\Sigma_y = \frac{1}{N} \sum_{i=1}^N (y_i - \bar{y})(y_i - \bar{y})^T. \quad (3.8)$$

By inserting the expressions for y from Eqn. 3.5 and Σ_x from Eqn. 3.7 into the expression for Σ_y and rearranging, we arrive at

$$\Sigma_y = M\Sigma_x M^T. \quad (3.9)$$

Restricting the vectors of M to be of unit length and orthogonal, M becomes an orthogonal matrix, meaning that $M^{-1} = M^T$, and by multiplying M^T on the left side of Eqn. 3.9, and utilizing that $M^T M = I$, we get the expression

$$M^T \Sigma_y = \Sigma_x M^T. \quad (3.10)$$

By denoting M^T by Φ and rearranging, we get the equation for the square matrix Σ_x that can be eigen decomposed:

$$\Sigma_x = \Phi \Sigma_y \Phi^{-1}, \quad (3.11)$$

such that the diagonal values of Σ_y becomes eigenvalues and Φ , the orthogonal matrix of column eigenvectors.

3.2 Synthesizing data from the Active Shape Model

The data that was made available for the current thesis comes in the following form:

- A $(3 * 39653, 1)$ sized vector representing the mean shape \bar{x} in Eqn. 3.4.
- A $(3 * 39653, 397)$ size matrix representing the 397 eigenvectors, or principal components Φ .
- A $(397, 1)$ sized vector representing the square roots of the diagonal eigenvalues corresponding to the eigenvectors.
- A $(77872, 3)$ sized triangular connectivity matrix. Each row corresponds to a face and the three columns holds the index k of the three vertices in the mean shape in Eqn. 3.4. Notice that k can take on the values $\{1, 2, \dots, n\}$ and $n = 39,653$.

In order to synthesize frontal face models with plausible variation, an initial choice has been made to focus on the shape variation and not use the texture information. The variational model poses are described by Paysan et al. [PKA⁺09] and are generated from the formula:

$$x(\alpha) = \bar{x} + \Phi \sigma \alpha \quad (3.12)$$

The elements of the formula in Eqn. 3.12 are:

- x is a new shape with a scalar factor α that determines the amount of perturbation of the vertex coordinates along each principal component direction. The values are spread with even intervals of 0.25 between -2 and 2, amounting to 17 scalar values.

- \bar{x} is the mean shape and is organized in the form: $\bar{x} \in R^{3m,1}$. Here, $m = 39.653$ signifies the number of vertices and the x , y and z coordinates are stacked in one column vector.
- Φ is the principal component matrix consisting of the orthonormal eigenvectors.
- σ_s are square roots of the eigenvalues associated with the eigenvectors.

In order to minimize the number of principal components to use for generating the training data set, the eigenvalues of the shape model have been examined. In Figure 3.1, the top sub-figure shows the eigenvalue numbers plotted against their value, while the bottom sub-figure is the accumulated eigenvalue sum as a fraction of the total amount of variation described by the entire active shape model. A threshold of 80% has been chosen in this connection and thus, by retaining the 119 highest principal components, the number of orthogonal principal directions are reduced by 278, down from 397.

In order to save the generated faces as vtk files, the Matlab functions `readVTK` and `writeVTK` created by Mario Richtsfeld have been utilized, for which the author directs his appreciations. Also, the function `explorer3dMM` by Jens Fagertun has been utilized with the aim of obtaining insight into the composition and functionality of vtk files, to whom appreciations are also directed.

By varying over the principal component scores derived from perturbing the mean shape by the 119 largest eigenvalues and for the 17 scalar values as described in Eqn. 3.1, a total of 2023 vtk files are generated and saved in a separate *training data* labeled folder. Each vtk files takes up 2.77 MB, summing up to a total of 5.5 GB.

Visual inspection of the vtk files is redundant due to the variations being quite small, as Figures 3.2 witnesses. However; very small perturbations are visible.

In order to ensure that the scales of each shape can be compared, the bounding box is computed as the diagonal of the smallest box that can contain all vertices in a shape. This has been shown for the mean shape in Figure 3.3.

When calculating the diagonal lengths of the bounding box for all the 2023 shapes in the data set, it can be seen that the lengths are normal distributed with mean $\mu = 221.388$ pixels and standard deviation $\sigma = 0.466$ pixels, as can be seen in Figure 3.4.

By investigation, the shape formed by PC31 and $\alpha = -2.0$ has the smallest bounding box diagonal, while the shape formed by the exact same principal component but with $\alpha = 0.50$ has the largest bounding box diagonal. This is not surprising when taking into consideration that the principal component directions describe orthogonal axes with maximal variation. Figure 3.5 show the two resulting shapes in various views.

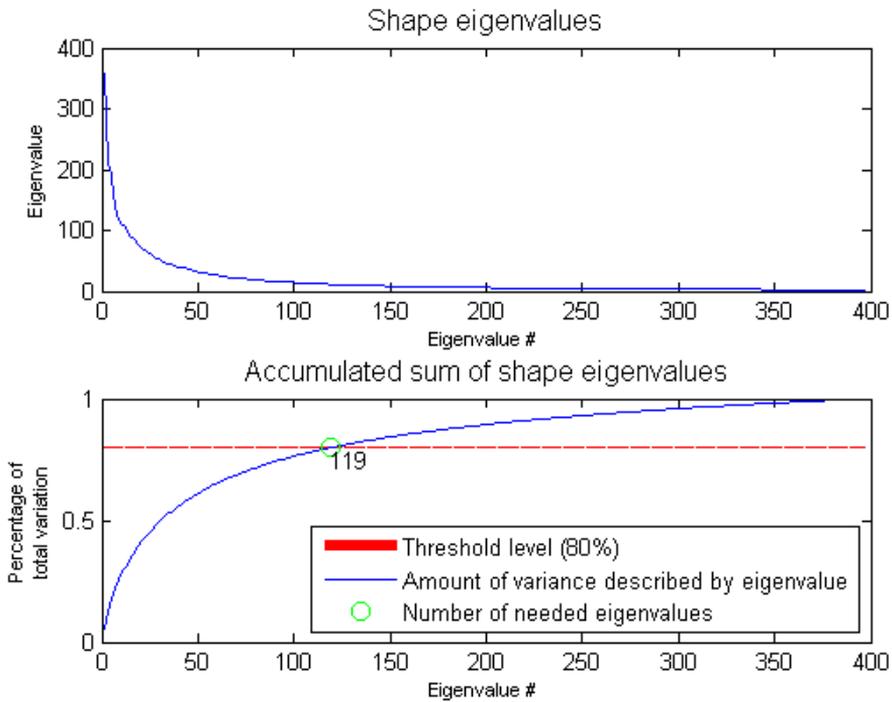


Figure 3.1: Visualization of the procedure for selecting the number of principal components required to account for the desired amount of the total variation in the active shape model coordinates. See text for description.

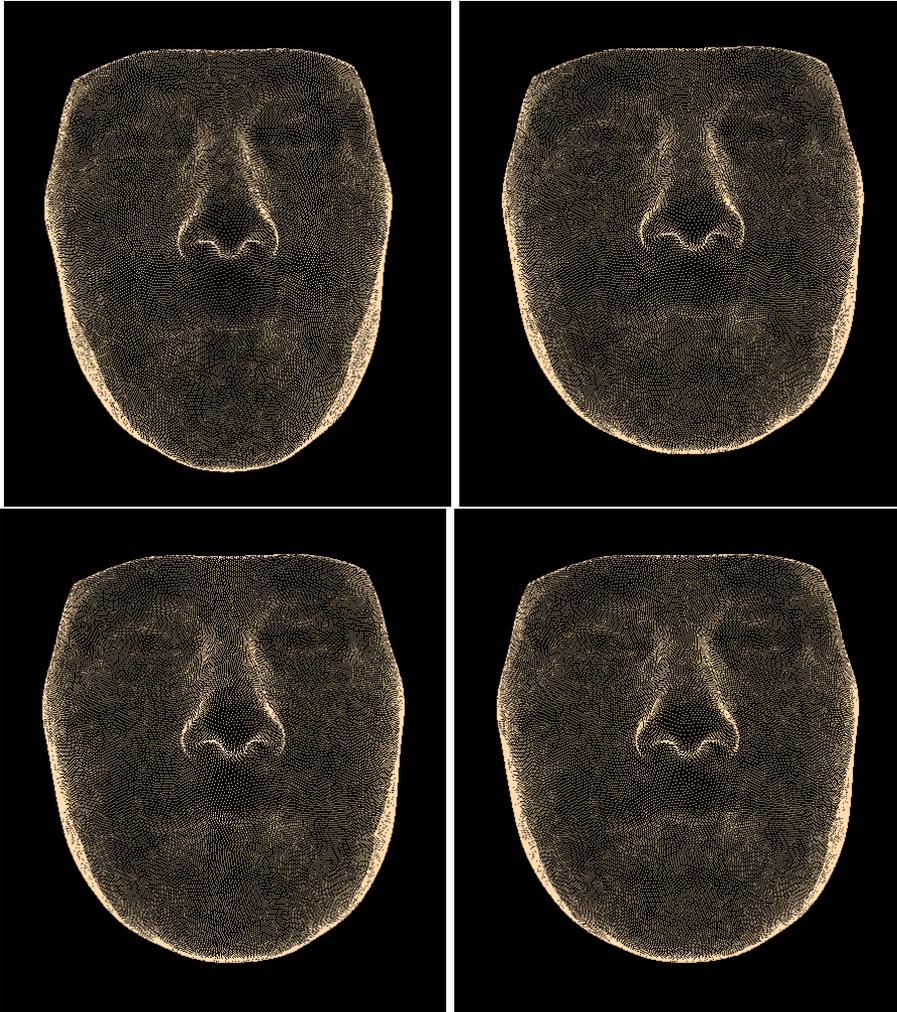


Figure 3.2: From left to right, top to bottom: Shapes generated from (1) PC 1 and $\alpha = -2.0$, (2) PC 36 and $\alpha = -1.0$, (3) PC 71 and $\alpha = 0.25$ and (4) PC 119 and $\alpha = 2.0$. See Eqn. 3.1.

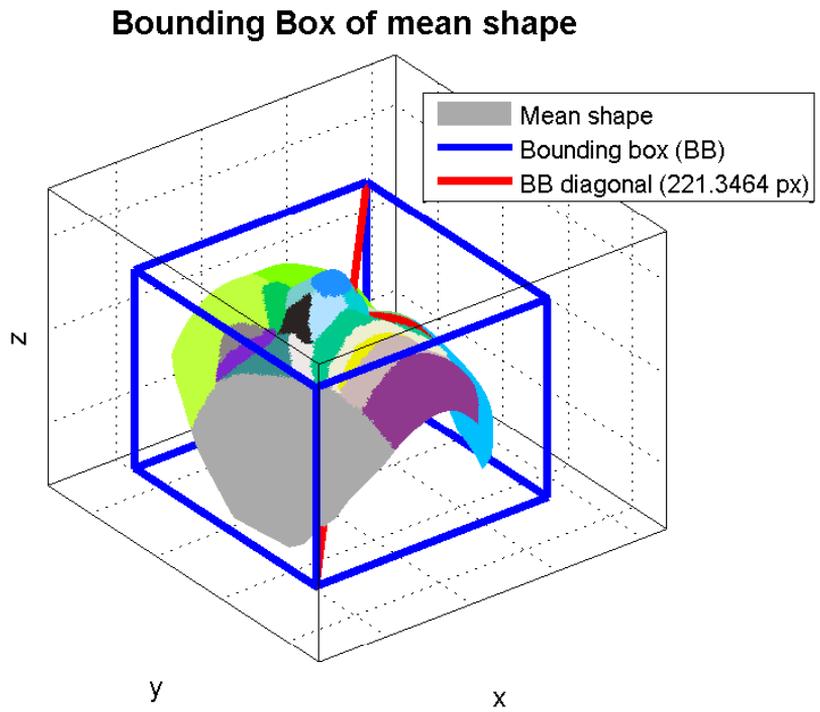


Figure 3.3: Bounding box shown for the mean shape.

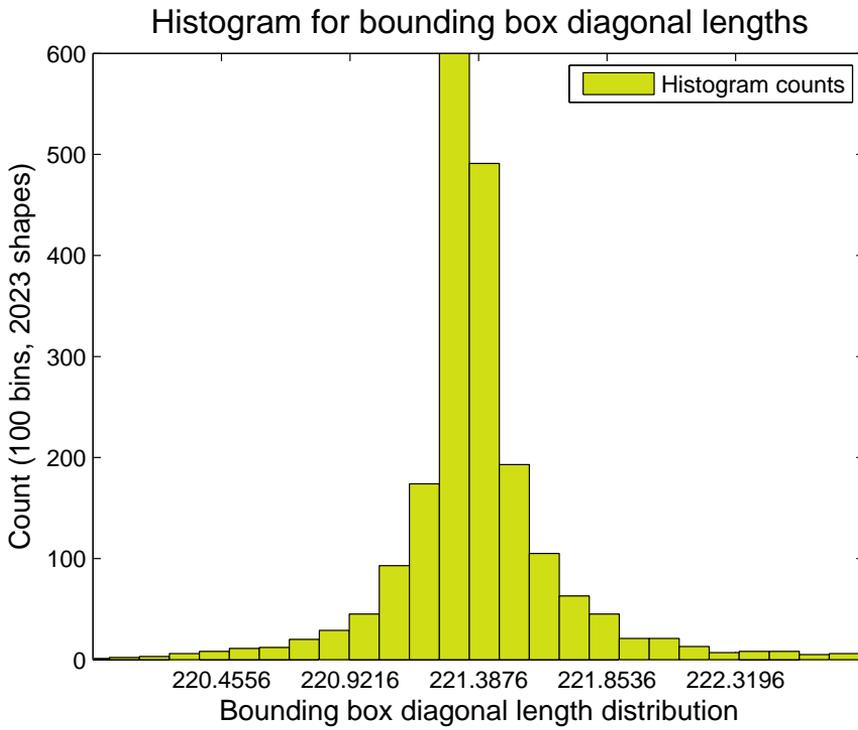


Figure 3.4: 100 bin histogram showing the distribution of bounding box diagonal length for the 2023 shape data set. The values for $\mu \pm \beta \cdot \sigma$ is shown for β scalars $\{-2, -1, 0, 1, 2\}$ on the abscissa.

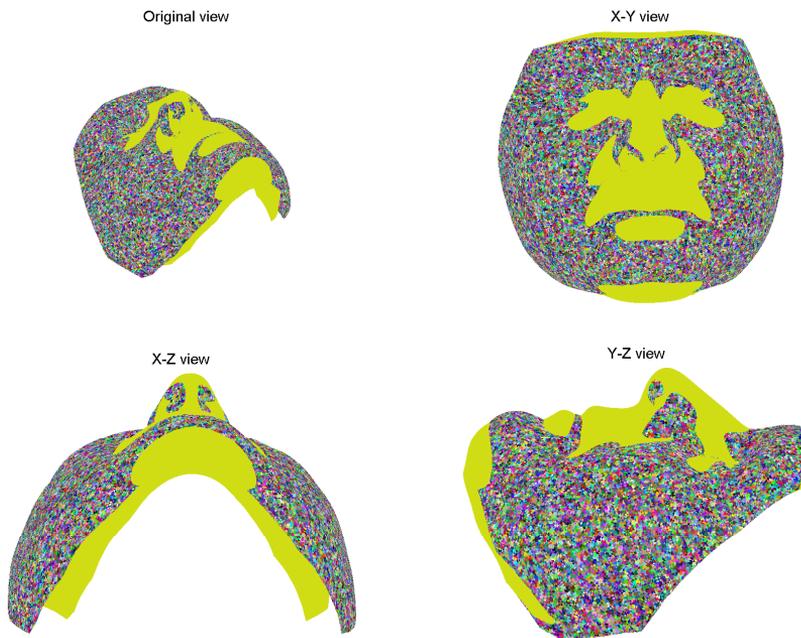


Figure 3.5: The shapes with the smallest (yellow) and largest (multicoloured) bounding box diagonals (216.8 pixels and 227.7 pixels), seen from various view points. The shapes both stem from PC31, but with $\alpha = -2.0$ and $\alpha = 0.50$.

As stated in Chapter 1, the works of Shotton et al. [SSK⁺13] impose the main inspiration towards the present work, which is why the methods of analysis are comparable to this work. Namely, the data synthetization process, the body-part class labelling approach, simple features as weak learners and classification by use of randomized decision forests are being scrutinized. As described in Chapter 3, 2023 triangular vertex meshes have been synthesized from an active shape model. This chapter describes the methods that the present work is analyzing.

4.1 Anatomical labelling

Since all the shapes are generated from the same active shape model, there is point-to-point correspondence between the vertices of all shapes in the data set, and therefore, the mean model shape has been used for the manual class annotation. The shape has been loaded into Sumatra [Pau] and a nominal class label has been assigned to each vertex. One of steps in the manual annotation process is shown in Figure 4.1. Adding artificial colours to each class, the representation is seen in Figure 1.2. It must be made clear that for the purpose of defining prominent facial characteristics, an exact correspondence with well-defined facial regions is not needed. The reason for that is that often, anatomical surface regions are named after non-surface structures, such as bones, muscles, other tissue, such as glands, or even vessels or nerve fibre bundles. However, for descriptive purposes, the annotated regions and their location in relation to anatomical regions are described in Table 4.2.

4.2 The Tangent Plane Features are the 3D Vertex Features

The present work presents a novel feature representation that is inspired by the depth image features by Shotton et al. [SSK⁺13]. This section describes the depth image features and how they inspired to the creation of the tangent plane features.

From Depth Image Features to Tangent Plane Features

Shotton et al. [SSK⁺13] are using depth image (I) features which are visualized in Figure 4.3. The response of the features for a given pixel (x) marked with a yellow cross are computed as the pixel intensity difference between the pixels marked with red circles. The position of these pixels are determined by offset vectors u and v

Label #	Left/ Right/ Center	Anatomical region	Abbreviation	# of vertices	% of total
1	Left	eye	L-E	373	0.9
2	Right	eye	R-E	474	1.2
3	Left	infraorbital area	L-IA	733	1.8
4	Right	infraorbital area	R-IA	613	1.5
5	Center	apex of nose	AN	423	1.1
6	Right	nostril and nose wing	R-NNW	362	0.9
7	Left	nostril and nose wing	L-NNW	369	0.9
8	Center	dorsum of nose	DN	922	2.3
9	Center	skin part of upper lip	SUL	1429	3.6
10	Center	root of nose and glabella	RNG	1049	2.6
11	Left	upper eyelid, eyebrow and lower forehead	L-UEELF	2891	7.3
12	Right	upper eyelid, eyebrow and lower forehead	R-UEELF	3028	7.6
13	Center	mucosal part of upper lip	MUL	889	2.2
14	Center	mucosal part of lower lip	MLL	777	2.0
15	Right	side of nose	R-SN	651	1.6
16	Left	side of nose	L-SN	725	1.8
17	Center	skin part of lower lip	SLL	1927	4.9
18	Right	area above nasolabial sulcus	R-AANS	394	1.0
19	Left	area above nasolabial sulcus	L-AANS	331	0.8
20	Center	chin	C	3608	9.1
21	Right	area below palpebromalar sulcus	R-ABPS	902	2.3
22	Left	area below palpebromalar sulcus	L-ABPS	859	2.2
23	Right	cheek	R-C	8378	21.1
24	Left	cheek	L-C	7546	19.0

Table 4.2: Description of correspondence between class labels and anatomical regions. For each class is also shown its name abbreviation, number of vertices and the percentage of total vertices (39,653).

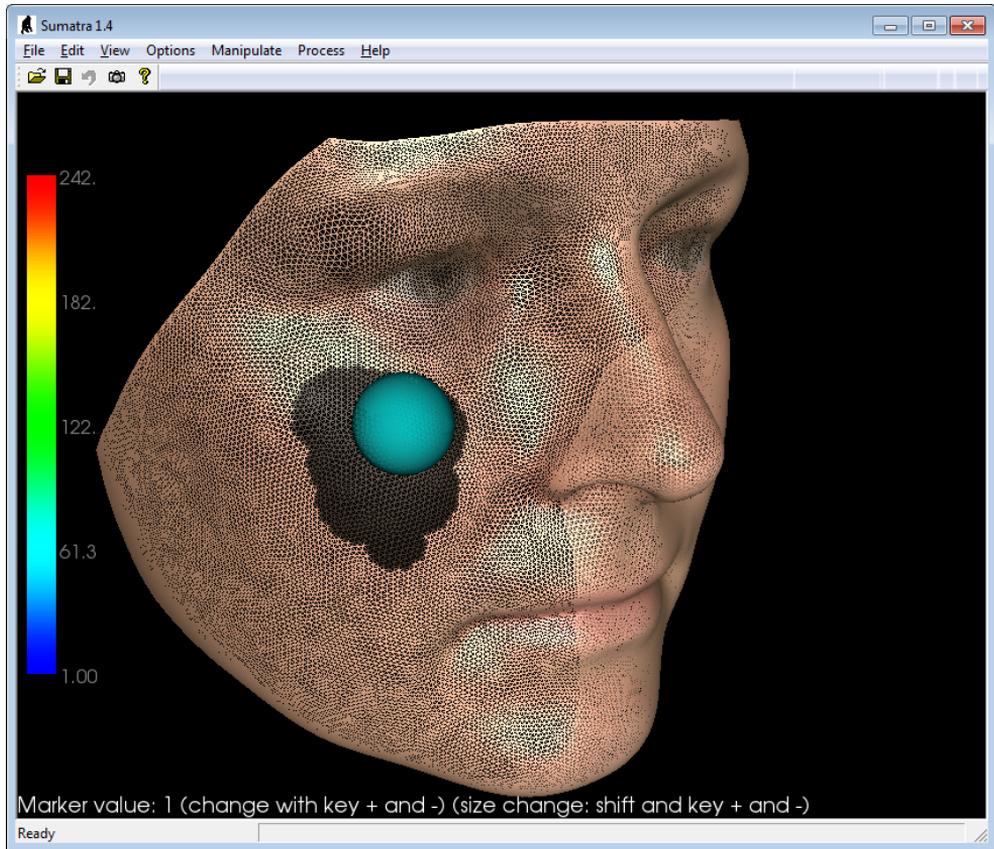


Figure 4.1: Manual annotation procedure carried out in Sumatra [Pau].

which are parameters in a candidate feature list $\phi(u, v)$. The pixel intensity $d_I(y)$ in a pixel y is also called the depth. With this notation, the feature response f for a pixel x in image I and a candidate feature $\phi(u, v)$ can be computed by Eqn. 4.1:

$$f_\phi(I, x) = d_I\left(x + \frac{u}{d_I(x)}\right) - d_I\left(x + \frac{v}{d_I(x)}\right). \quad (4.1)$$

Each offset is normalized according to the pixel intensity in the pixel x .

The width of an object, such as for example the arm with the yellow cross in the right image in Figure 4.3, will vary with the distance of that object to the camera. This varying distance has been taken into account by Shotton et al. [SSK⁺13] when the feature response f is computed from Eqn. 4.1. When an object is moved away from the camera (i.e. if the intensity $d_I(x)$ of a pixel x on that object is increased),

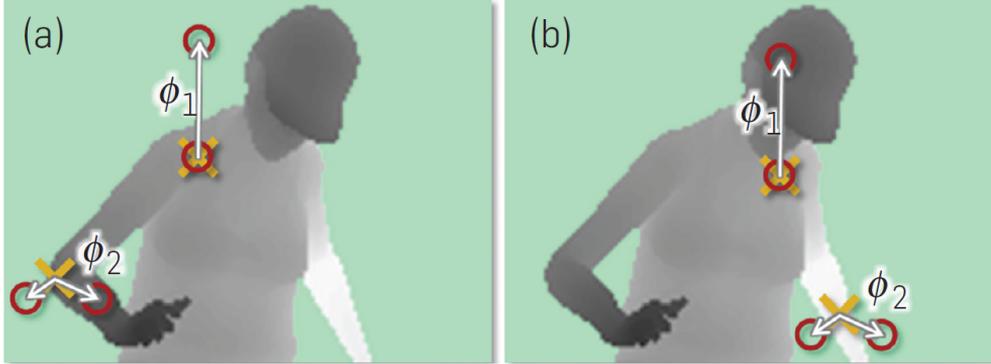


Figure 4.3: 2 types of depth image features shown for different pixels x (yellow crosses). From Shotton et al. [SSK⁺13].

the smaller the object width will be in the image plane. The scaling of the length of the displacement vectors u and v by $\frac{1}{d_I(x)}$ compensates for this effect.

In contrast to the pixel intensity of 2D depth images leading to varying scaling of offset vectors, the vertex response $d_I(y)$ considered in the present work is based on offset vectors that are spanning the tangent plane to the vertex. This means that the offset plane is always in level with the vertex and therefore, the offsets do not need to be scaled by $\frac{1}{d_I(x)}$. Removing this scaling from Eqn. 4.1, Eqn. 4.2 substitutes this:

$$f_\phi(I, x) = d_I(x + u) - d_I(x + v). \quad (4.2)$$

In the present work, there are no 2D depth intensity images. Instead, the "image" I is a 3D triangular mesh, also denoted as a shape. Figure 4.4 shows a close up of vertex 7416 (black) in an arbitrary shape along with its blue normal vector \vec{n} and scaled unit tangent vectors \vec{a} (red) and \vec{b} (purple). The tangent plane they span is shown in transparent red and by combining the tangent vectors, 8 points have been visualized in yellow and green.

In Eqn. 4.1, the offsets u and v are normalized by the pixel intensity in a pixel marked with a yellow cross in Figure 4.3. The pixel range in a depth image starts at a low number for the objects that are closest to the camera and ends at a high, fixed number which describes the "general" background distance. Since the closest object cannot be at a distance of zero to the camera, this normalization always allows for some degree of displacement by the u and v vectors.

In the present work, instead of pixel intensity, shortest distance-to-mesh will be used to describe the vertex response $d_I(y)$. Where Shotton et al. used the offset vectors in Eqn. 4.1 to describe 2D displacement in depth intensity images, in this work, they will represent displacements along scaled unit \vec{a} and \vec{b} tangent plane vectors.

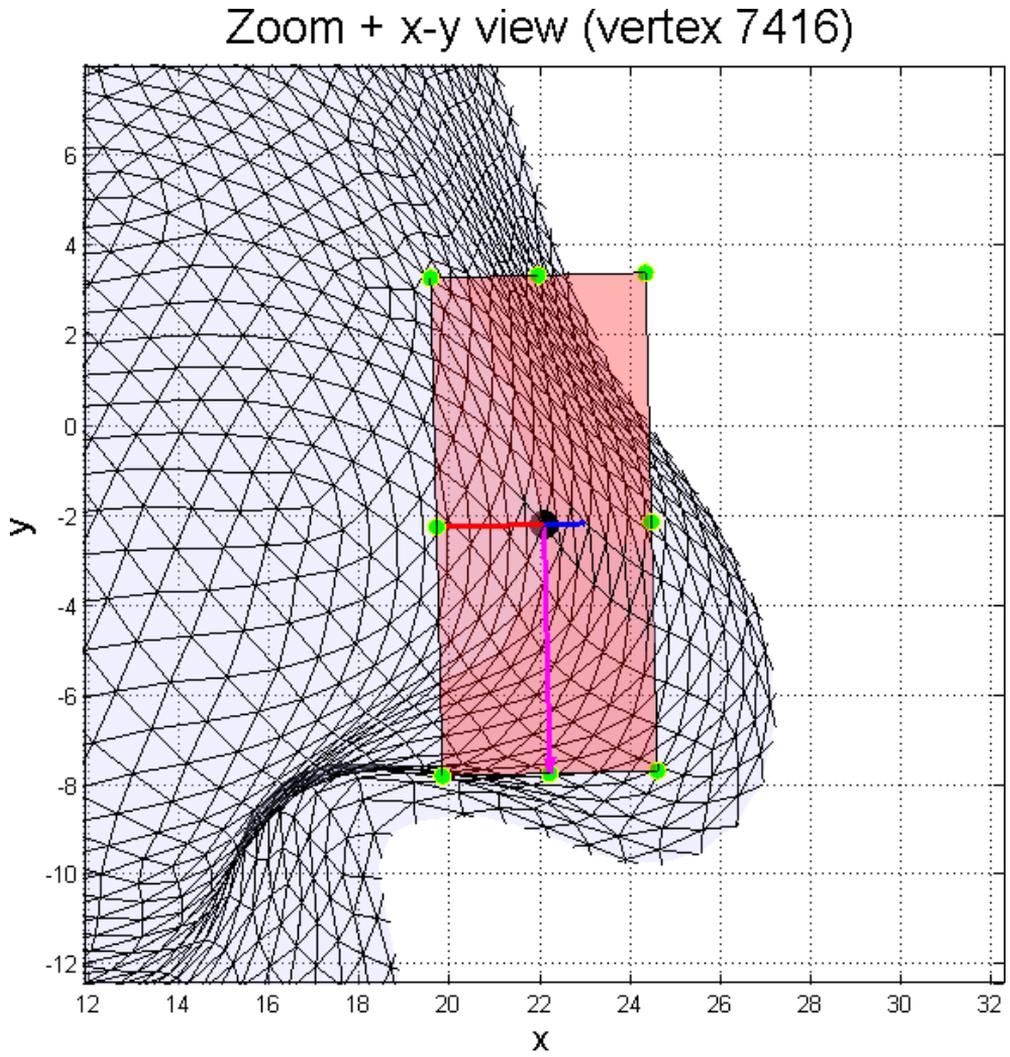


Figure 4.4: Tangent plane (pink) of vertex 7416(black) with 8 evenly spread out points (yellow and green).

The displacements are computed by scaling the unit tangent vectors with a scalar, denoted by `pFactor`, which is computed by this formula:

$$pFactor = scaleFactor * boxDiag. \quad (4.3)$$

In Eqn. 4.3, `scaleFactor` is a scalar that can take the values between 0 and 1, and `boxDiag` is the length of the diagonal of the bounding box of the shape that the vertex belongs to. As a reference, `boxDiag` is the length of the red line in Figure 3.3.

In the examples in this chapter, the tangent vectors are scaled by a `scaleFactor` of 0.025, which means that the length of these vectors are 2.5 % of the diagonal length of the bounding box of the entire shape.

The naming convention for the 9 points that are shown in green and yellow in Figure 4.4 are visualized in Figure 4.5. In this figure, `v` refers to the vertex on the shape, `ta` and `tb` are the `pFactor` scaled tangent vectors, and the points are made in the following way:

- a is $v - t_a - t_b$,
- b is $v - t_b$,
- c is $v + t_a - t_b$,
- d is $v - t_a$,
- v is the vertex,
- f is $v + t_a$,
- g is $v - t_a + t_b$,
- h is $v + t_b$, and
- i is $v + t_a + t_b$.

Based on such 9 points in this example (the ninth point being `x` itself), $8 + 7 + 6 + 5 + 4 + 3 + 2 + 1 = 36$ combinations of candidate $\phi(u, v)$ features can be chosen.

Figure 4.6 shows 120 tangent planes for a training shape. Five vertices have been sampled at random from each class such that there is an equal distribution of samples from each of the 24 classes. Figure 4.5 shows how labels are assigned to the 9 points lying in each vertex tangent plane that lay the foundation for the feature rule calculations. Noticing that the middle point is in fact the vertex `v` itself, the rules can be derived from Eqn. 4.2 and will take the form as shown in Table 4.7.

The computations of feature rule responses for one of the 120 sampled vertices shown in Figure 4.6 are shown in Table 4.8.

The tangent plane feature calculations include the following steps:

- The active shape model contains a connectivity list that describes how the vertices are connected into a triangle mesh.

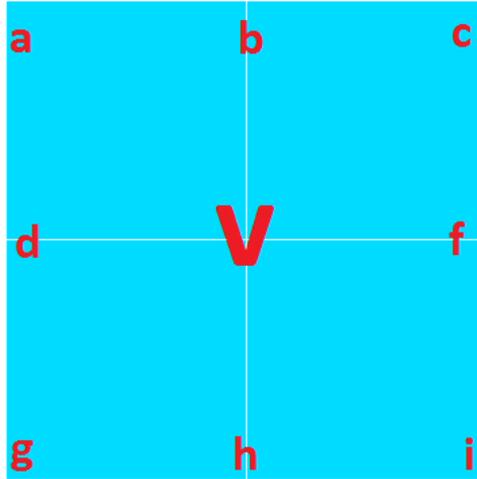


Figure 4.5: Visualization of tangent plane point association. The blue plane corresponds to the pink tangent plane in Figure 4.4. v is the vertex and a is $v - t_a - t_b$, b is $v - t_b$, c is $v + t_a - t_b$, and so forth. (t_a, t_b) are the tangent vectors that have been scaled by $pFactor$.

- Vertex normals can be computed from the neighbouring points $\{p_i\}$ in the vicinity, as shown in Figure 4.9, as described by Bærentzen et al. [BGAA12], chapter 17. Conceptually, a principal component analysis is applied to the (x, y, z) coordinates of the neighbouring points $\{p_i\}$ and the normal vector n is then the principal direction that shows the least variation in $\{p_i\}$.
- Based on the normals, the normal plane and tangential vectors can be used to define the offsets for which the shortest distance to the surface can be computed. This computation will be used in the computation of the feature based vertex descriptors.

Based on the model vertex triangulation connectivity list `t1` and the vertices `pts` that make out the triangulated mesh, a triangulation class `TR` is made in Matlab via the built-in command `TR = triangulation(t1,pts)`. By using the built-in method call `vn = vertexNormal(TR)`, the unit normal vectors to each vertex is computed. In Figure 4.9, a unit normal vector is shown for the mean shape of the active shape model.

In order to compute the tangent vectors for the triangulated mesh based on the normal vector, the method described in chapter 8.5 in Bærentzen et al. [BGAA12] is used. By denoting by \mathbf{n} the normal vector, $\tilde{\mathbf{a}}$ a random vector and \mathbf{a} and \mathbf{b} two vectors spanning the tangent plane, the calculations in Eqn. 4.4 can be made:

PC_100_SD_1.vtk. 120 random vertex tangent planes

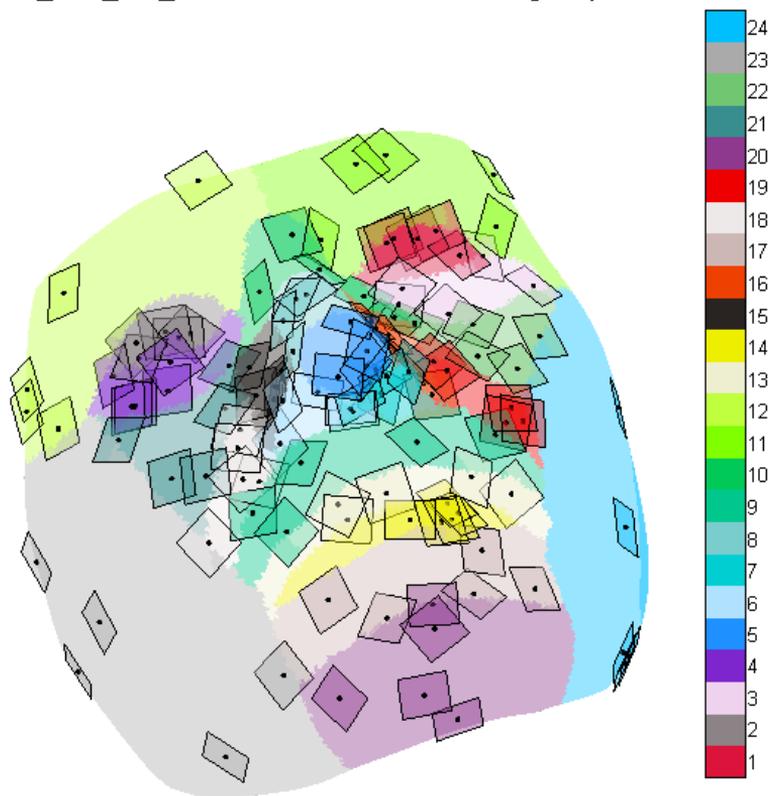


Figure 4.6: Random sub selection of five vertices from each of the 24 classes in a training mesh. The tangent planes associated with each vertex is colour matched with the vertex class colour.

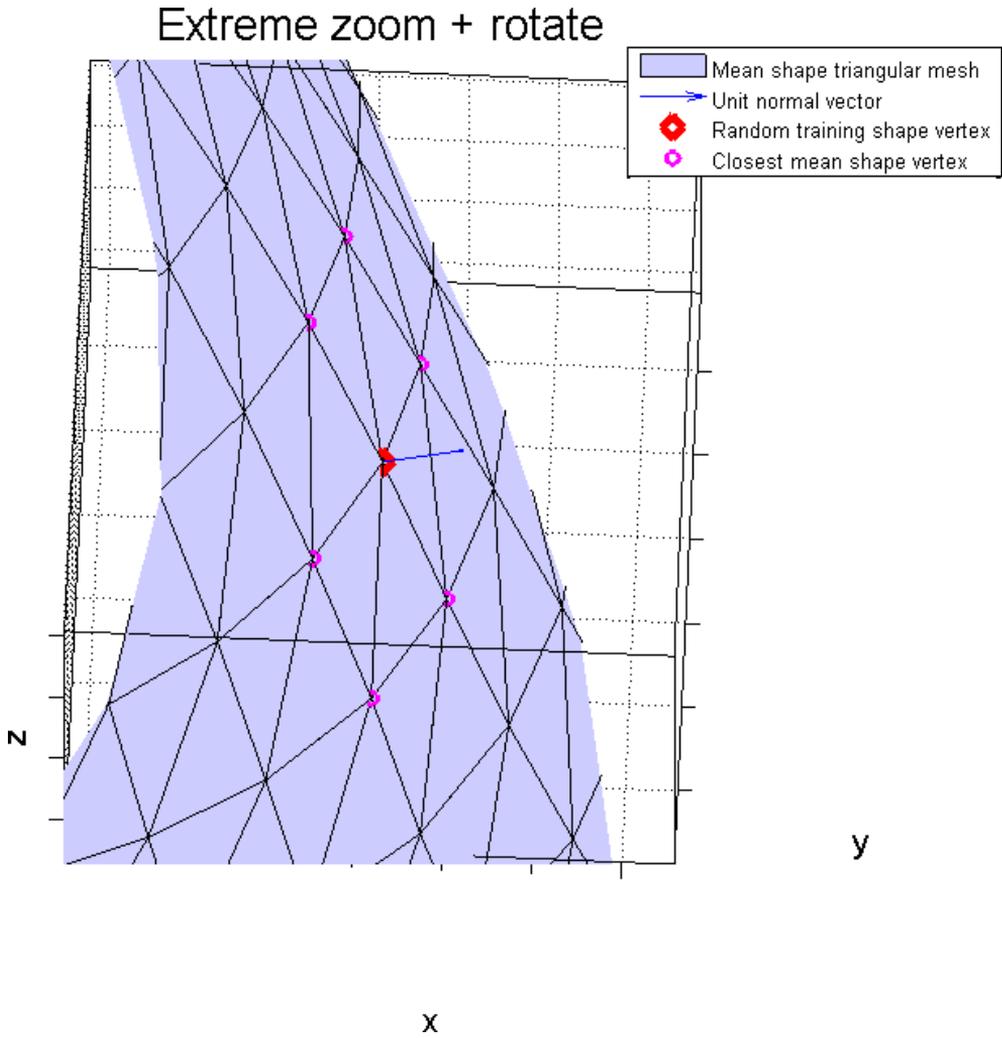


Figure 4.9: The red diamond marks a highlighted random mesh vertex and its unit normal vector. The neighbouring vertices marked by magenta circles are derived from the highlighted vertex via the connectivity list associated with the active shape model.

$(x + u, x + v)$	$d_I(x + u)$	$d_I(x + v)$	$f_\phi(I, x)$
(a,b)	dist(I,a)	dist(I,b)	dist(I,a)-dist(I,b)
(a,c)	dist(I,a)	dist(I,c)	dist(I,a)-dist(I,c)
(a,d)	dist(I,a)	dist(I,d)	dist(I,a)-dist(I,d)
(a,v)	dist(I,a)	dist(I,v)	dist(I,a)-dist(I,v)
(a,f)	dist(I,a)	dist(I,f)	dist(I,a)-dist(I,f)
\vdots	\vdots	\vdots	\vdots
(h,i)	dist(I,h)	dist(I,i)	dist(I,h)-dist(I,i)

Table 4.7: Correspondence table of the elements in Eqn. 4.2 and the labelling scheme for a single vertex in Figure 4.6. In the table, a, b, c, d, v, f, g, h and i refer to the tangent plane points in Figure 4.5. The first column contains the 36 possible point pairs $(x + u, x + v)$. When comparing the notion of x , u and v from Eqn. 4.2 with Figure 4.6, x is the 3D coordinate of the vertex v , and u and v are the shifts by the combinations of the scaled tangent vectors t_a and t_b . The second and third columns represent the vertex responses and are computed as the closest distance from the point to the shape surface. The last column contains the actual **tangent plane features**. For example, the four entries in the first line can be interpreted as follows. The first entry (a,b) means that the combination of the two points (a,b) from Figure 4.5 are considered. The second entry is the first vertex response in Eqn. 4.2, namely, the closest distance from the point a to the shape surface. The third entry is the vertex response for point b. The fourth entry is the subtraction of point b’s vertex response from point a’s feature response. Notice that this is not the absolute difference and therefore, the feature response can have a negative value.

$$\mathbf{a} = \frac{\tilde{\mathbf{a}} - \mathbf{n}(\tilde{\mathbf{a}} \cdot \mathbf{n})}{\|\tilde{\mathbf{a}} - \mathbf{n}(\tilde{\mathbf{a}} \cdot \mathbf{n})\|}, \quad (4.4)$$

$$\mathbf{b} = \mathbf{n} \times \mathbf{a}$$

The random vector $\tilde{\mathbf{a}}$ has been chosen as the upward (z-axis) unit vector $[0 \ 0 \ 1]^T$. By making these calculations on the point in focus in Figure 4.9, the unit tangent vectors are shown in Figure 4.10.

For a given triangular mesh, the following calculations are made to establish coordinates of points lying in the tangent plane that will be used for feature computations:

- Determine bounding box corner coordinates and diagonal length, as shown in Figure 3.3.

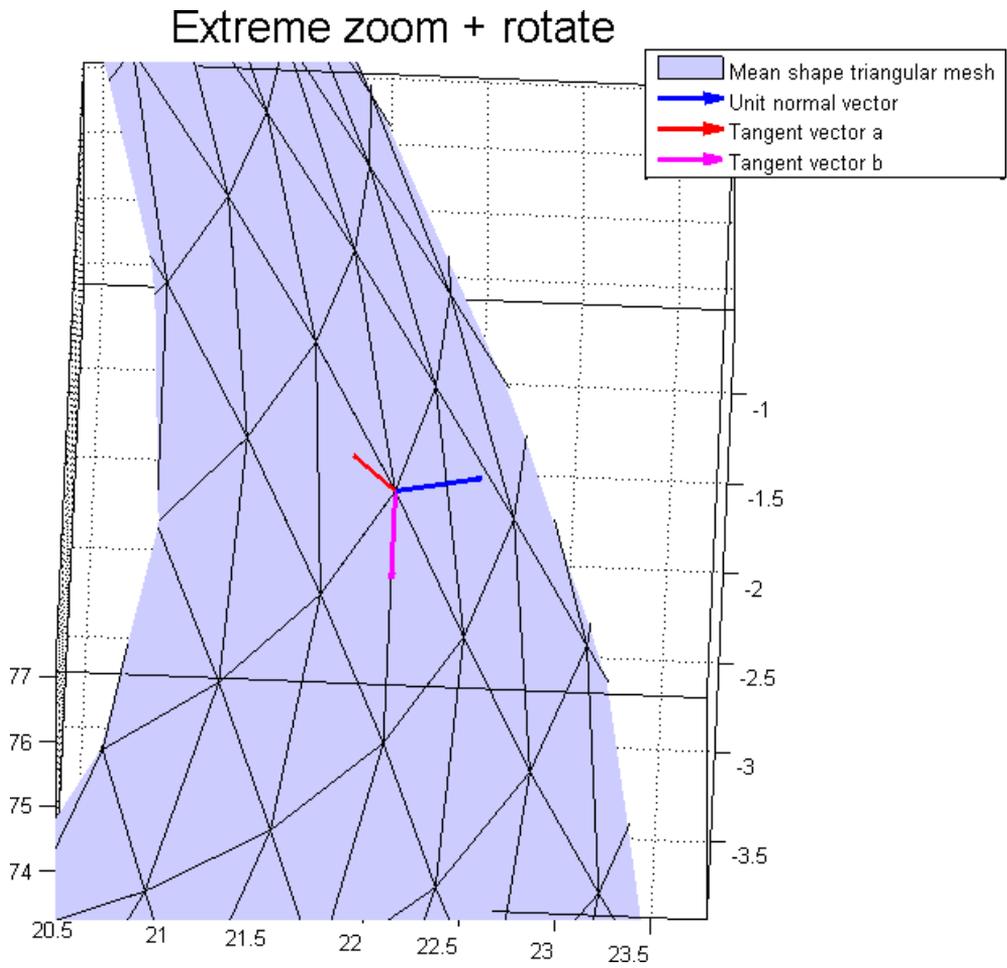


Figure 4.10: Unit length vectors to the vertex marked with a red diamond in Figure 4.9. Red vector **a** and magenta vector **b** are computed via Eqn. 4.4.

$(x + u, x + v)$	$d_I(x + u)$	$d_I(x + v)$	$f_\phi(I, x)$
(a,b)	0.4356	0.4296	0.0060
(a,c)	0.4356	0.9591	-0.5234
(a,d)	0.4356	0.5011	-0.0655
(a,v)	0.4356	0	0.4356
(a,f)	0.4356	0.5191	-0.0834
\vdots	\vdots	\vdots	\vdots
(h,i)	0.3507	0.5952	-0.2444

Table 4.8: Feature response table for one of the 120 sampled vertices shown in Figure 4.6. The right column contains the tangent plane features and the reader is referred to Table 4.7 for an elaboration of the table entries. If all features are computed for the 120 vertices, the data matrix X will have the size $(n\text{Obs}, n\text{Features}) = (120, 36)$. Table 4.13 shows the entries of the data matrix X .

- Triangulate and compute unit vertex normals (the blue vector in Figure 4.9 is an example).
- Based on Eqn. 4.4, compute tangent vectors of unit normal based on alignment with the z-axis. Also, scale tangent vectors based on diagonal length (red and magenta vectors in Figure 4.4).
- Compute the tangent plane spanned by the tangent vectors as can be seen in Figure 4.4.

4.3 Randomized Decision Forests as a Classification Approach

This section describes the theory behind randomized decision forests and how it is manifested within the methods.

The concept of randomized decision forests was invented by Breiman [Bre99] and was influenced by Amit and Geman [AG97] who classified handwritten digits and Latex symbols in 2D images by optimizing binary decision trees based on simple spatial relationship features. Criminisi et al. [CSK11] are influenced by Breiman’s classification random forests and one of the authors, Shotton, is also one of the key persons in the main inspiration to the present work [SSK⁺13].

A decision forest is built up of a number of decision trees, thus, making it an *ensemble method*. Within a random forest, also referred to as a *randomized decision forest*, each single tree is trained from a random subset of input variables, or in the present case, tangent plane features. The random subset sampling is performed by creating vectors that are iid, which is important, since this lowers the correlation between the trees. Figure 4.12 shows the structure of a decision tree.

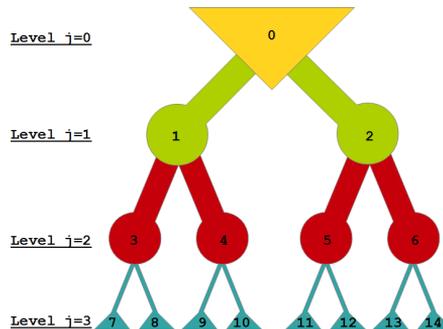


Figure 4.11: Basic concept of a decision tree structure. All observations are considered at the $j = 0$ level. Based on the splitting criterion, the observations are split into the two child nodes and so forth. If the trees are trained with a minimum node size of 1, the trees are grown until the leaf nodes (blue) only contain one kind of class.

An important aspect is that by growing deep trees, each tree fits very well to the random of subset of data that it was created from. The single tree bias is therefore, very small, but the variance among the trees is high. By averaging over the trees, the forest is formed and the predictive power of each tree keeps the bias low but lowers the variance. It is important to keep in mind that randomness is only injected into the system during the training phase, and not during testing.

The following describes the random forests theory and how it is applied in the present work.

The Theory behind Randomized Decision Forests

Criminisi et al. [CSK11] describe a *decision tree* as a tool for deciding what class a certain observation belongs to. In the present context this can be interpreted as *what anatomical region a vertex belongs to*. A tree is constituted of a root node that branches out via binary splits into several layers of internal nodes ultimately ending in leaf nodes that holds the classification labels. In each node a question is asked and answered via *rules of thumb* (weak learners). By randomly growing a vast amount of trees in this order, each internal node and each leaf node has a trained distribution associated with it. By computing the information gain at each node split, the split with the highest confidence can be chosen.

Decision trees are randomly trained in an ensemble to form a *decision forest*. The key model parameters to this procedure, are:

- The size of the forest (the number of trees, `nTrees`)

- The stopping criterion for training each tree (in this context, the minimum number of elements in a leaf node, `minNodeSize`).
- The weak learner model choice (the number of features to sample from at each node, `mtry`). The total possible number is `nFactor = 36`, see Table 4.7.
- The objective function for determining how to split at each internal tree node (Θ_j^*).
- The function that is used for determining the information gain during the tree training (I in Eqn. 4.6).
- The feature choice (the tangent plane features described in Section 4.2).

If the number of tangent plane features for a given vertex \mathbf{v} is `nFeatures`, then a small sub sample of features is sampled at each split node, both during training and testing. This sub sampling can be formulated as a rule $\phi(\mathbf{v})$, that reduces the feature dimensionality in this way:

$$\phi : \mathbb{R}^{\text{nFeatures}} \rightarrow \mathbb{R}^{\text{mtry}}, \quad (4.5)$$

in which $\text{mtry} \ll \text{nFeatures}$.

The following notation is used:

- $c \in C$ is a class label.
- C is the full list of class labels: $C = \{c_k\} = \{1, 2, \dots, \text{nClass}\}$, where the default class label type is nominal and the default number of classes `nClasses` is 24, as shown in the case with manually annotated class labels in Figure 1.2.
- S^{right} and S^{left} are the left and right child nodes at an internal node S_j . When using binary trees as in the present work each split will be dual and inevitably creating 2 offspring nodes that can then either become split nodes themselves or end as leaf nodes.

Both during the training and testing process, at each internal tree node S_j , a random subset `mtry` is sampled from the training data and the split with the highest information gain I_j is chosen. By defining the Shannon entropy as $H(S) = -\sum_{c \in C} p(c) \log(p(c))$, the information gain I is described by Eqn. 4.6:

$$I = H(S) - \sum_{i \in \{\text{left}, \text{right}\}} \frac{|S^i|}{|S|} H(S^i). \quad (4.6)$$

At each split node, a weak learner model contains a binary split function

$$h(\mathbf{v}, \Theta_j) \in \{0, 1\}. \quad (4.7)$$

The domain values 0 and 1 relate to the outcome of a threshold evaluation at the split node. This is associated with the parameters for the weak learner model at the internal tree node S_j which is denoted by

$$\Theta_j = (\phi, \tau), \quad (4.8)$$

where ϕ is the sub sample rule described in Eqn. 4.5 and τ is a parameter vector of linearly spaced threshold values that are considered during the information gain computations I_j .

The weak learner model is defined by

$$h(\mathbf{v}, \Theta_j) = [\phi(\mathbf{v}) > \tau], \quad (4.9)$$

which takes on the two possible values 0 and 1 as described in Eqn. 4.7.

Noting that the formula for the information gain I in Eqn. 4.6 is generic and is generally applied to all non-leaf nodes in a tree. In order to compute the specific information gain I_j for node j , the following sets are required for consideration:

- S_j , S_j^{left} and S_j^{right} . These are the sets of training points in the j 'th node and its two child nodes.
- The parameter list Θ_j in Eqn. 4.8.

The specific information gain in node j then becomes

$$I_j = I(S_j, S_j^{\text{left}}, S_j^{\text{right}}, \Theta_j), \quad (4.10)$$

and the aim is then to maximize the specific information gain via the **objective function**

$$\Theta_j^* = \operatorname{argmax}_{\Theta_j} I_j. \quad (4.11)$$

It must be kept in mind that the parameter list Θ_j in Eqn. 4.8, and therefore, also, the objective function Θ_j^* in Eqn. 4.11 are governed by the number `mtry` of sampled tangent plane features as described above.

When testing for the class of an unseen observation, the following procedure is followed. For all $t = \{1, 2, \dots, T\}$ trees in the forest, each leaf node is associated with a probabilistic leaf predictor model

$$p_t(c|v), \quad (4.12)$$

and all trees can be averaged together to obtain the full prediction for the forest by the formula:

$$p(c|v) = \frac{1}{T} \sum_{t=1}^T p_t(c|v). \quad (4.13)$$

In order to decide how the trees should be grown, three popular choices of stopping criteria are evident:

1. Grow the trees until a desired depth level is reached.
2. Grow the trees until the information gain for further growing falls under a desired value.
3. Grow the trees until the minimum number of observations in a node is reached.

In the present work, the third criterion has been chosen. In particular, deep trees are sought for in the pursuit of low bias - high variance trade-off.

Implementation

Liaw et al. [LW02] have implemented a C based code for creation of randomized decision forests in R which is based on a Fortran implementation by Breiman and Cutler [BC04]. A mex/standalone interface to this package has been created for Matlab by Jaialtilal [Jai13]. The author directs his thanks to the creators.

The following steps are required in order to train the model:

1. Load desired number of random training meshes into workspace.
2. Randomly sample desired number of vertices from each class.
3. Compute features for each of these samples and assemble into one matrix.
4. Collect ground truth labels for each sample and assemble into one vector.
5. Subdivide feature matrix and ground truth vector into bootstrapping training and evaluation sets. These are also referred to as "in-bag" and "out-of-bag" sets.
6. Run the Random Forest software with the training data as input as well as the desired number of trees and number of classes to sample from at each split.

The Random Forest software has a functionality for predicting the class of an input feature test matrix.

Preparing the data

In order to set up a few examples, five vertices are picked at random from each class from an arbitrary training shape (adding up to a total of 120 vertices), which has been visualized in Figure 4.6. In Figure 4.12, the same has been done, only in this instance, 100 random vertices from each class have been sampled. Table 4.13 shows how the features of the sub-partitioned data in Figure 4.6 are assembled into a data matrix X and Table 4.14 shows the corresponding vector of class labels.

Prior to training a randomized decision forest, the following parameters have to be determined:

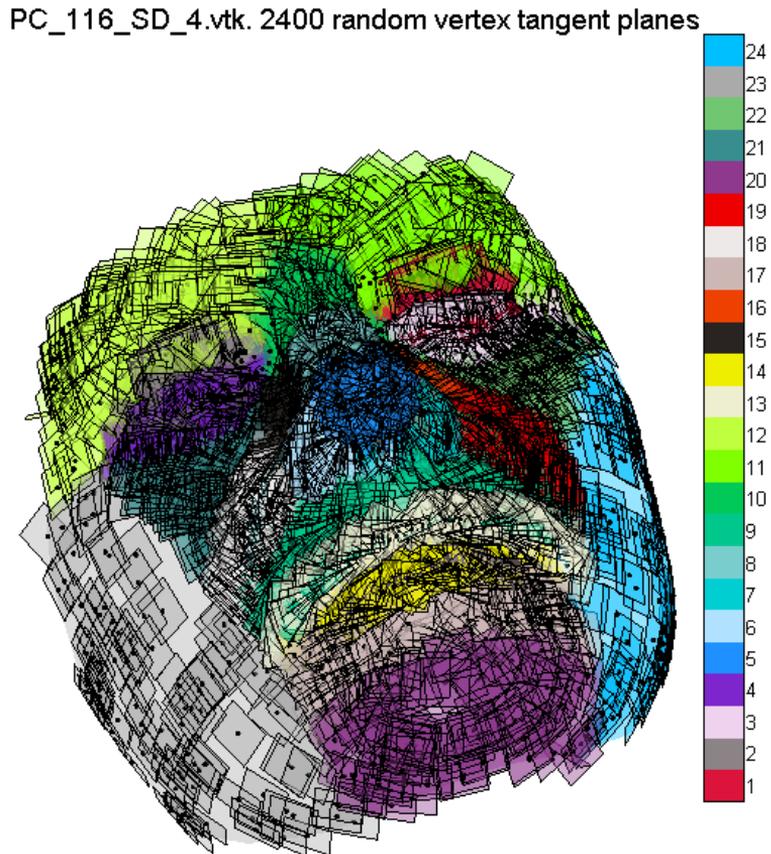


Figure 4.12: Random sub selection of 100 vertices from each of the 24 classes in a random training mesh. The tangent planes associated with each vertex is colour matched with the vertex class colour.

vertex ID \ $f_\phi(I, x)$	(a,b)	(a,c)	...	(g,i)	(h,i)
3613	1.0971	0.8992	...	-0.7631	-0.5983
16859	0.3909	-1.1980	...	0.3053	0.1749
⋮	⋮	⋮	⋮	⋮	⋮
8995	-0.6271	1.0901	...	-2.6292	-1.3669
9010	-0.0149	-0.6473	...	-0.1002	-0.8294
⋮	⋮	⋮	⋮	⋮	⋮
27584	-0.0458	0.0058	...	0.4416	0.3027
10783	-0.0482	0.2137	...	3.3530	-0.0371

Table 4.13: Table of the entries in the data matrix X associated with the 120 random samples shown in Figure 4.6. The rows are the 120 observations, or vertex IDs and the columns are the 36 features. Each row in this table is the result of a computation of the right column in Table 4.7 for the respective vertex with the vertex IDs in the left column on this table.

class
1
1
⋮
2
2
⋮
24
24

Table 4.14: Table of the entries in the ground truth response vector Y which corresponds with the data matrix X in Table 4.13. The rows contain the class assignment labels that was manually annotated in Sumatra [Pau], and the number of observations is the same for these two tables. The class labels in this table correspond with the coloured class labels in Figure 4.6.

- Number of observations, `nObs`, which is based upon:
 - Number of training meshes to include in the training set, `nShapes`.
 - Number of vertices to sample from each class, `nSamplesPerClass` (24 classes, number of vertices in each class varies, see Table 4.2).
- Fraction of "in-bag"/"out-of-bag" observations to keep for training/evaluating when doing bootstrap sampling during individual tree training (default is $5/6 * nObs$).
- Number of trees to train for the model, `nTrees`.
- Number of allowed features that can be sampled during each tree split, `mtry`.

Three scenarios have been carried out, and the parameters are as follows:

- Number of training meshes: 4, 4 and 50.
- Number of vertices from each class: 100, 150 and 150.
- Bootstrap training/test fraction: $5/6$ in all three cases.
- Number of trees: 500, 500 and 1000.
- Number of allowed features: 3 in all three cases.

In order to test the performance of the three models, a random mesh from the training data set is used. The same number of vertices from each class is picked at random and the true classes are then used as input to the model.

CHAPTER 5

Results and Evaluation

This chapter is built up of three sections. The initial section 5.1 describes the measures that are used to report the results. This is followed by a section 5.2 that provides the reader with an overview of the experiments that have been carried out as well as an interpretation and evaluation of the results. The final section 5.3 reports the results and analyses them.

5.1 Procedure for reporting the results from the analysis

In order to understand how well the randomized decision forest based classification utilizing tangent plane features approach works, the concepts of variable importance are described in the following.

Two measures of variable importance

The introduction of the novel tangent plane features necessitates an investigation of the variable importance. Two methods are used, namely the decrease in Gini impurity and the decrease in accuracy [Bre99]. Their individual functionality is described separately below.

Decrease in accuracy

As Breiman states [Bre99], this is a method of randomly permuting out-of-bag estimates in a certain fashion and comparing the model's prediction on this permuted out-of-bag set with that of the unpermuted set.

The following steps are carried through *during the training of each tree* in the randomized decision forest training:

- After training the tree on the in-bag samples, the out-of-bag samples are tested on the tree and the misclassification rate when comparing the prediction with the true class labels is computed and kept for reference.
- For each of the $n = 1, 2, \dots, n\text{Features}$ tangent plane features, or variables, their corresponding entries in the out-of-bag samples are now randomly permuted. This means that, if the out-of-bag set is a matrix of size $(n\text{Obs}, n\text{Features})$, the first column is randomly shifted around. For each of the $n\text{Features}$ permutations the following is done:

- The n 'th permuted out-of-bag set is tested on the tree and the prediction is saved.
- When all `nFeatures` predictions for the `nFeatures` perturbed out-of-bag data sets have been computed, they are combined into one common prediction by majority voting.
- The misclassification rate of this majority voted prediction is now computed by comparison with the true class labels.
- The misclassification rate of the perturbed out-of-bag reruns is now compared to the misclassification of the non-perturbed out-of-bag samples and the percent increase in misclassification is used as a measure of the importance of the `nFeatures` variables.

When interpreting the variable importance when using this method, there are two extremes that enhance the understanding. In one end, if, say, variable 3 has a very low value, it means that randomly perturbing the values of that specific variable for all observations and comparing the prediction with the unperturbed prediction did not lead to a lower prediction. This means that in this example, it does not really matter if variable 3 is even used, because no matter what values it takes, it does not enhance the predictions. In the other end, if, say, variable 12 has a very high value, this means that a random perturbation of that specific variable for all observations critically impacts and lower the predictions, which makes that variable highly important.

Decrease in Gini impurity

The method for using the Gini impurity to determine the variable importance as described by Deng et al. [DRT11] is elaborated in the following.

By denoting p_j^t as the proportion of class j in node t in a tree, the Gini impurity can be computed by $I_G(t) = \sum_{j=1}^{nClasses} p_j^t(1 - p_j^t)$. By inspection, if p_j^t approaches 1, it means that node t only contains observations from class j . The $(1 - p_j^t)$ part of the equation leads to the Gini impurity becoming zero.

Because a split takes place in the node, the Gini impurity in the two children nodes is lower than that of the parent node.

Breiman [Bre99] states that by averaging the decrease in Gini impurity for all variables over all trees, it is possible to get a total picture of how much each variable contributes to lowering the Gini impurity during the training of all trees in the forest.

The method for interpreting the variable importance based on the Gini impurity decrease is similar to that of the decrease in accuracy as described above. A high decrease for a variable means that the descriptive effect of that variable cannot be ascribed to random chance and therefore, that variable is deemed important. A low decrease for a variable implies that there is a higher risk that the variable is only making correct predictions due to random chance.

5.2 Experiment overview

Six investigations have been carried out and the overall parameters are shown in Table 5.1.

The first experiment was done to determine the effect of increasing number of trees used in training randomized decision forests. The characteristics of this experiment is a large number of trees.

The second experiment is the first of two approaches to enhance the separation between classes that was hard to separate during the first experiment. This approach involves oversampling of the problematic classes.

The third and fourth experiment deals with *cascades of classifiers*, and is an alternative approach to solving the problem of hard-to-separate classes.

The fifth experiment is a multi-scale approach to investigating how an expansion of the tangent plane feature dimensionality affects variable importance when trained on multiple randomized decision forests. A sub-sampling parameter `mtry` is also investigated in this experiment.

The sixth and final experiment is a multiple, single-scale approach to investigating a large range of varying tangent plane feature scales and their effect on the overall prediction accuracy. A large number of randomized decision forests is trained and a deeper investigation on the sub-sampling parameter `mtry` is carried out.

5.3 Results and evaluation of experiments

In this section the results from the six experiments are shown and evaluated.

Experiment 1

The first experiment was carried out with the following fixed parameter settings: `scaleFactor`, `nFeatures`, `nClasses`, `nShapes`, `mtry`, `nSamplesPerClass` and `min. node size`. A single randomized decision forest was trained.

This experiment gives some insight to the behaviour of the out-of-bag training error for increasing number of trees. Figure 5.2 shows that when averaging over an increasing number of randomized decision trees, the total predictive strength of the forest increases. However, once the error has fallen below 10%, the effect falls significantly for the number of trees exceeding 100.

The created randomized decision forest has been tested on a random shape from the data set with the constraint that the shape was not included in the training of the model. Figure 5.3 shows the confusion matrix between the predicted and actual classes. The average accuracy can be read as the percentage in the lower right block. The high value of 95% is partly due to the high number of shapes used in the model training. Also, 150 random vertex samples from each class ensures that there is an equal emphasis on each class. A lower accuracy for prediction of classes 23 and 24 is evident, and by inspection of the ground truth class labels in Figure 1.2, it turns out that these two classes are the left and right pair of cheeks. Looking these classes up

	Experiment 1		Experiment 2		Experiment 3	
	Value	Fixed?	Value	Fixed?	Value	Fixed
scaleFactor	0.025	yes	0.025	yes	0.025	yes
nFeatures	36	yes	36	yes	36	yes
nClasses	24	yes	24	yes	2 (23,24)	yes
nShapes	50	yes	50	yes	50	yes
nTrees	1000	yes	100	yes	1000	yes
mtry	3	yes	3	yes	3	yes
nSamplesPerClass	150	yes	varying	no	300	yes
min. node size	3	yes	5	yes	5	yes
# of random forests	1	yes	1	yes	1	yes

	Experiment 4		Experiment 5		Experiment 6	
	Value	Fixed?	Value	Fixed?	Value	Fixed?
scaleFactor	0.025	yes	[0.010, 0.025, 0.100]	no	20 values	no
nFeatures	36	yes	108	yes	36	no
nClasses	23	yes	24	yes	24	yes
nShapes	50	yes	50	yes	50	yes
nTrees	200	yes	150	yes	150	yes
mtry	3	yes	[2,4,8, 16,32, 64,128]f	no	36 values	no
nSamplesPerClass	150	yes	150	yes	150	yes
min. node size	5	yes	1	yes	1	yes
# of Random forests	1	yes	7	no	720	no

Table 5.1: Overview table of parameter settings for the experiments that were carried out.

in Table 4.2 reveals that they contain 21.1% and 19.0% of the total number of shape vertices, respectively, which makes them the largest classes by far. In comparison, the third largest class, the centred chin part, class 20, contains less than half the number of vertices.

Another reason for the high accuracy for this experiment lies in the training error graph in Figure 5.2, in which it was witnessed that the large number of trees in the forest increase the quality of the predictions.

Another interesting subject to investigate is the importance of the variables, or the *tangent plane features* that were used in the training of the forest and were described in Section 4.2. The features were scaled with `scaleFactor` = 0.025, which, as a reference, is the same scale that was used to create the tangent planes in Figure 4.4.

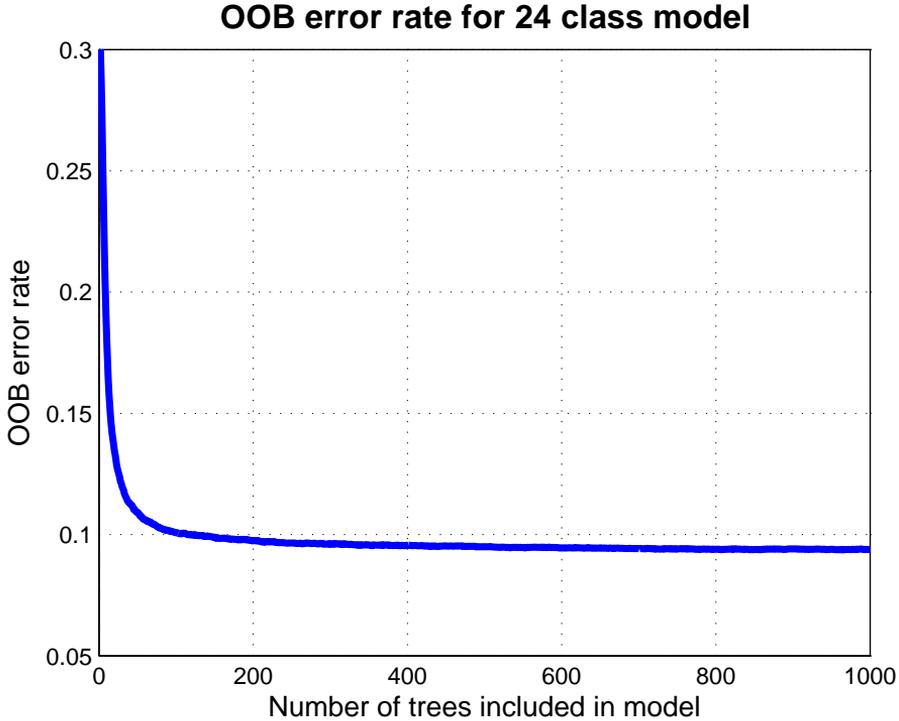


Figure 5.2: Experiment 1. Training error for the out-of-bag training data for increasing number of averaged trees in the forest.

Figure 5.4 shows the two measures of calculating the variable importance, the mean decrease in accuracy and Gini impurity as described in Subsection 5.1. Generally, what is important to notice here is that the range of importance for the most and least important variables is a factor 2, at most, which leads to the evaluation that all variables are sufficiently important to retain for further analysis.

Experiment 2

In order to investigate a way to increase the accuracy of the prediction of the paired classes 23 and 24, a new experiment was conducted. The same parameters as in Experiment 1 was kept, except for two parameters: the number of `nTrees` was lowered from 1000 to 100 and `nSamplesPerClass` was changed such that instead of evenly sampling 150 vertices per class, 300 samples were taken from each of classes 23 and 24 and 150 samples were taken from the remaining classes. Figure 5.5 shows the out-of-bag training error and variable importance plots. When comparing with Figures

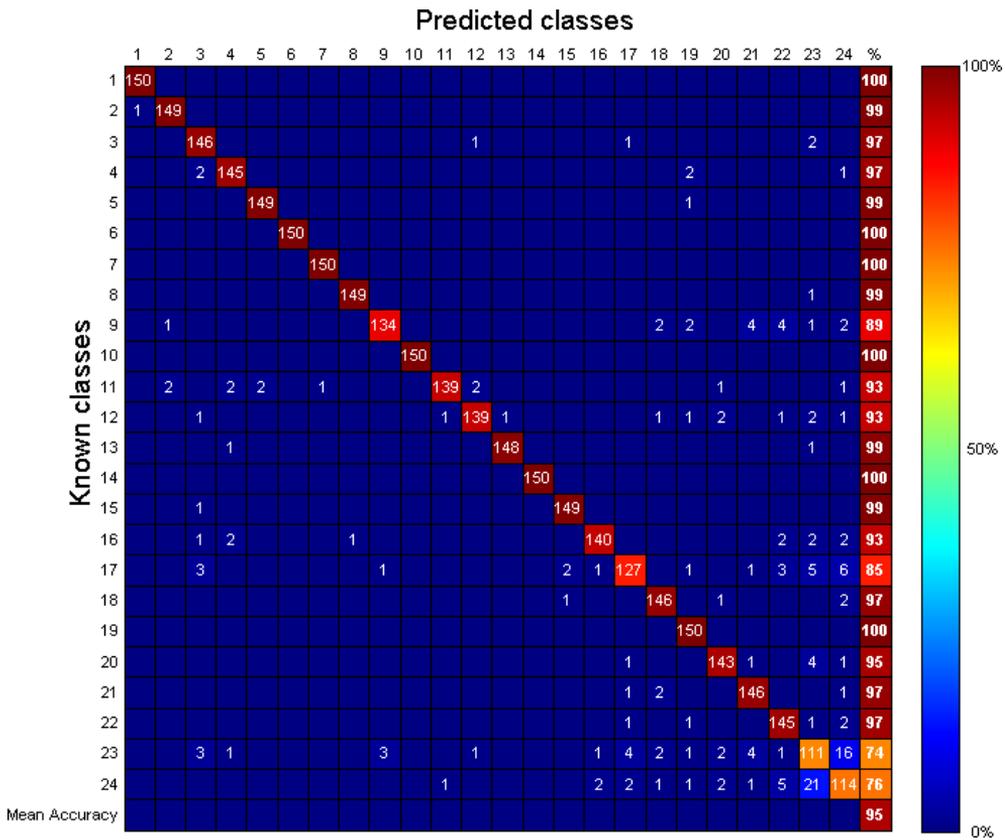


Figure 5.3: Experiment 1. Confusion matrix between the predicted classes for a subset of 150 vertices from each class in a random test shape and the manually annotated ground truth labels in Figure 1.2.

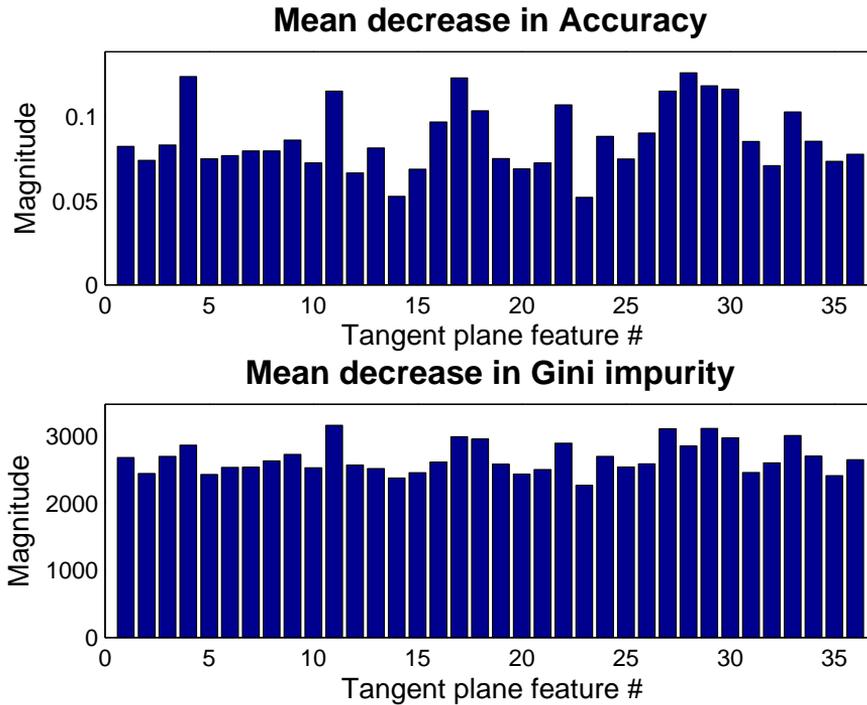


Figure 5.4: Experiment 1. Variable importance plots. For reference, the variables $f_\phi(I, x)$ are described in the most-right column of Table 4.7.

5.2 and 5.4, it can be seen that the training error is higher in Experiment 2 due to the lower number of trees in the forest, and also, the same variables tend to show slightly higher importance.

The most indicative conclusion to draw from this experiment come from investigating the confusion matrix between the model prediction on a random test shape that was not included in the model training, as seen in Figure 5.6. The random test shape was equally sampled from all 24 classes. When comparing to the accuracy of the model in Experiment 1 where all classes were equally sampled, it can be seen that the accuracy for class 23 increases from 74% to 78%, and for class 24, the accuracy increase is from 76% to 85%. The increase is largest for class 24 but both classes tend to be easier to predict by this experiment. However, there is a downside in the average accuracy that actually drops from 95% to 91%. When looking at the predictions of the third worst classified class, which was class 17, its accuracy actually dropped from 85% to 75%, which is a drastic drop. This side effect is caused by the way that

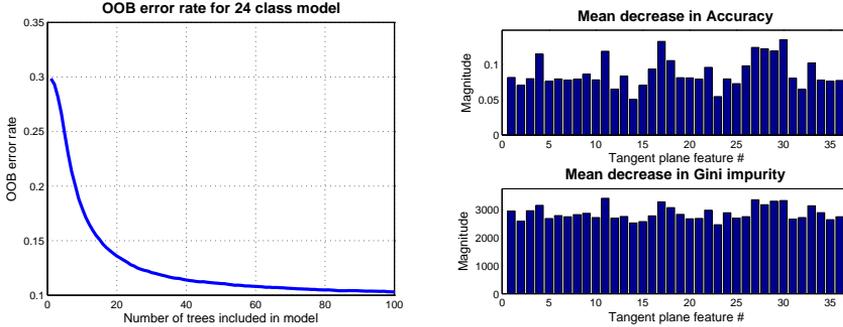


Figure 5.5: Experiment 2. Left: Training error for the out-of-bag training data for increasing number of averaged trees in the forest. Right: Variable importance plots. For reference, the variables $f_\phi(I, x)$ are described in the most-right column of Table 4.7.

the randomized decision forests are formed. At each split node, a higher amount of samples from classes 23 and 24 are evaluated at the expense of the remaining classes being under represented.

Experiments 3 and 4

In Experiment 2 it could be seen that oversampling the difficult classes 23 and 24 did not improve the average accuracy of the prediction of the randomized decision forest. Namely, a notable amount of observations that had the class 24 ground truth label were falsely predicted as class 23, and vice versa, which could be seen in the Experiment 1 and 2 confusion matrices in Figures 5.3 and 5.6. On the ground truth labelling Figure 1.2 it is obvious that these two classes possess two significant attributes: they're paired and they contain a relatively large amount of vertices. In order to enhance the separation of these classes, Experiments 3 and 4 constitute an approach of *cascading classifiers* that consists of two randomized decision forest models has been trained in order to let the two models complement each other. After a brief representation of the randomized decision forest models, a combined test will be described.

In Experiment 3 a randomized decision forest has been trained based only on sampling from classes 23 and 24, and in Experiment 4, another forest has been trained on a dataset in which the 24 ground truth class labels have been altered so that vertices with class label 24 are relabelled into class 23.

Experiment 3 has been based on 50 randomly sampled training shapes, each with 300 samples per class and a number of 1000 trees, and Experiment 4 has been based on 50 randomly sampled training shapes, each with 150 samples from each of the 23 classes and 200 trees. The out-of-bag training error is for the two experiments is

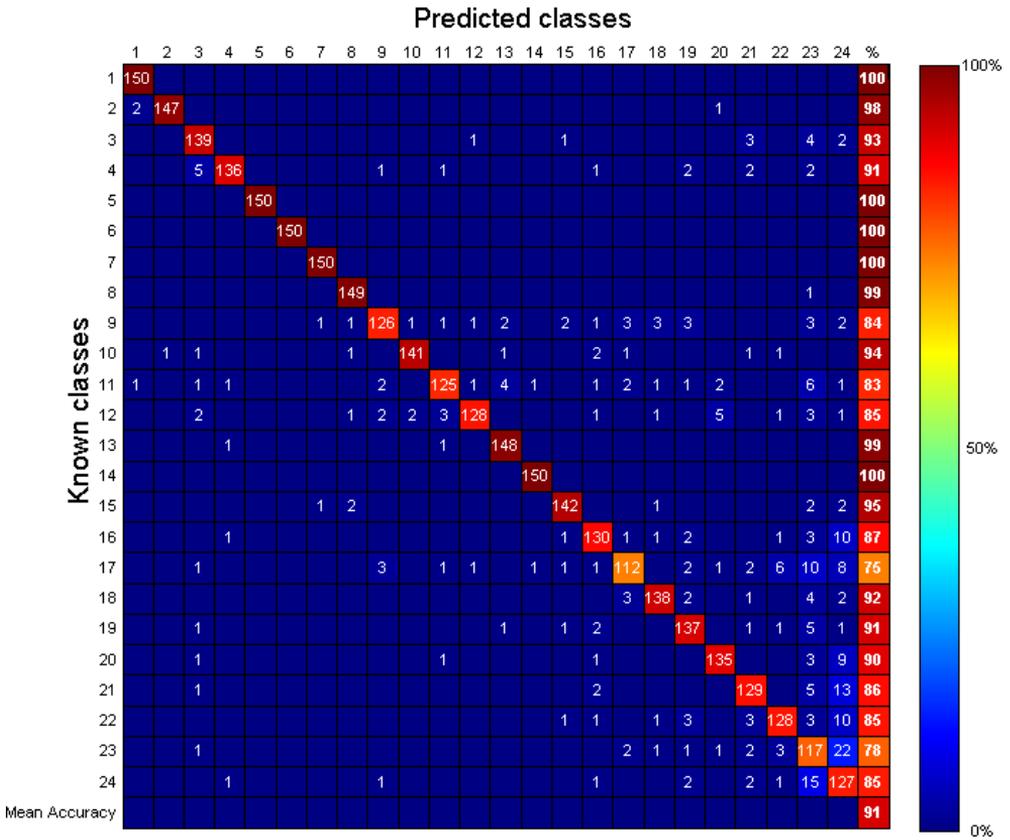


Figure 5.6: Experiment 2. Confusion matrix between the predicted classes for a subset of 150 vertices from each class in a random test shape and the manually annotated ground truth labels in Figure 1.2.

shown in Figure 5.7. The graphs show that in Experiment 3, the out-of-bag training error barely drops below 13%, even after 1000 trees, and in Experiment 4, the same error after 200 trees is below 9%, which leads to the expectation that the models should be able to predict if a vertex belongs to a cheek, but that it can be hard to distinguish if it belong to the left or the right cheek.

In order to utilize the two RF models, a random test shape has been sampled from the full data set with the criterion that the shape has not been used in creation any of the models. 300 vertices have been sampled and tangent plane features have been computed. The model from Experiment 4 has been used to predict the class labels of this test shape, and the confusion matrix between the 23 ground truth classes and the predicted classes has been shown in Figure 5.8.

The 317 samples that were predicted as label 23 have been retained and the RF model based on classes 23 and 24 (Experiment 3) has been used to predict which of these two classes they belong to. The tangent planes for these samples have been visualized in Figure 5.9. The resulting confusion matrix has been shown in Figure 5.10.

To conclude upon the observations by utilizing a cascade of classifiers, the two confusion matrices in Figures 5.8 and 5.10 will be scrutinized. In the first step of the cascading process, the average accuracy is at a relative high level of 94%. However, since only 75% are correctly predicted as belonging to the combined class 23/24 and only 88% of those are correctly predicted according to the left and right cheek, it means that only $75\% * 88\% = 66\%$ are correctly predicting the correct cheek.

Experiment 5

The fifth experiment stands out from the previous experiments in that it is the only experiment with more than 36 features. In Experiments 1 through 4, a single `scaleFactor` value of 0.025 has been used, which corresponds to the middle sized tangent plane shown in Figure 5.11. In Experiment 5, `scaleFactor` takes on the values [0.010, 0.025, 0.100]. For each value, the corresponding 36 tangent plane features are computed, and assembled into a 108-row vector.

Another variational feature of this experiment is that the various values [2, 4, 8, 16, 32, 64, 128] of the parameters `mtry` have been used. For each of these values, a randomized decision forest has been trained on 50 training shapes, 150 trees and `min. node size` of 1, which implies that all trees are grown fully (only one observation in each leaf node).

The target of the analysis in Experiment 5 is to investigate if it possible to deduce what values of `scaleFactor` and `mtry` yield the best randomized decision forest classification. The out-of-bag training errors have been shown in Figure 5.12 and by a close-up it appears that an `mtry` value of 4 is a good choice for the 108 element tangent plane feature scenario. The `mtry` value of 128 has been included, but it is obsolete, since this number is higher than the feature dimension of 108, which means that all features can be picked as random candidates at each node sampling. This im-

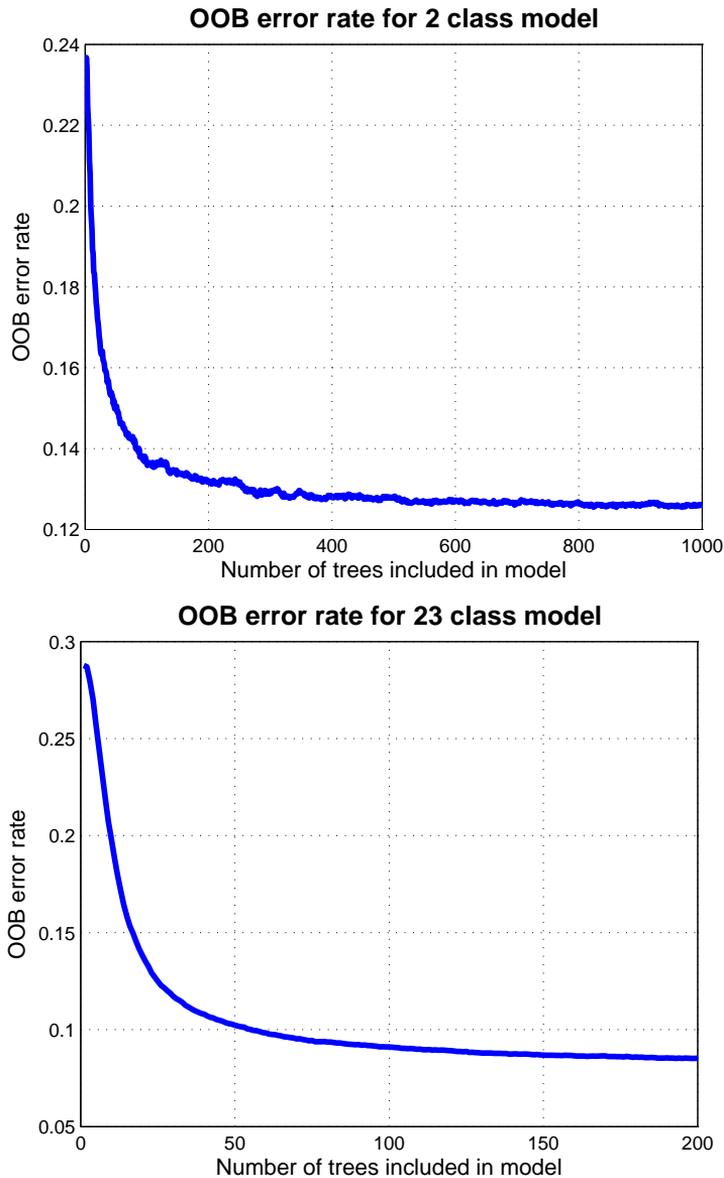


Figure 5.7: Experiment 3 (top) and 4 (bottom). Out-of-bag error computed on the same training set during training of an RF based only on the classes 23 and 24 (Experiment 3) and for an RF with classes 23 and 24 merged together (Experiment 4).

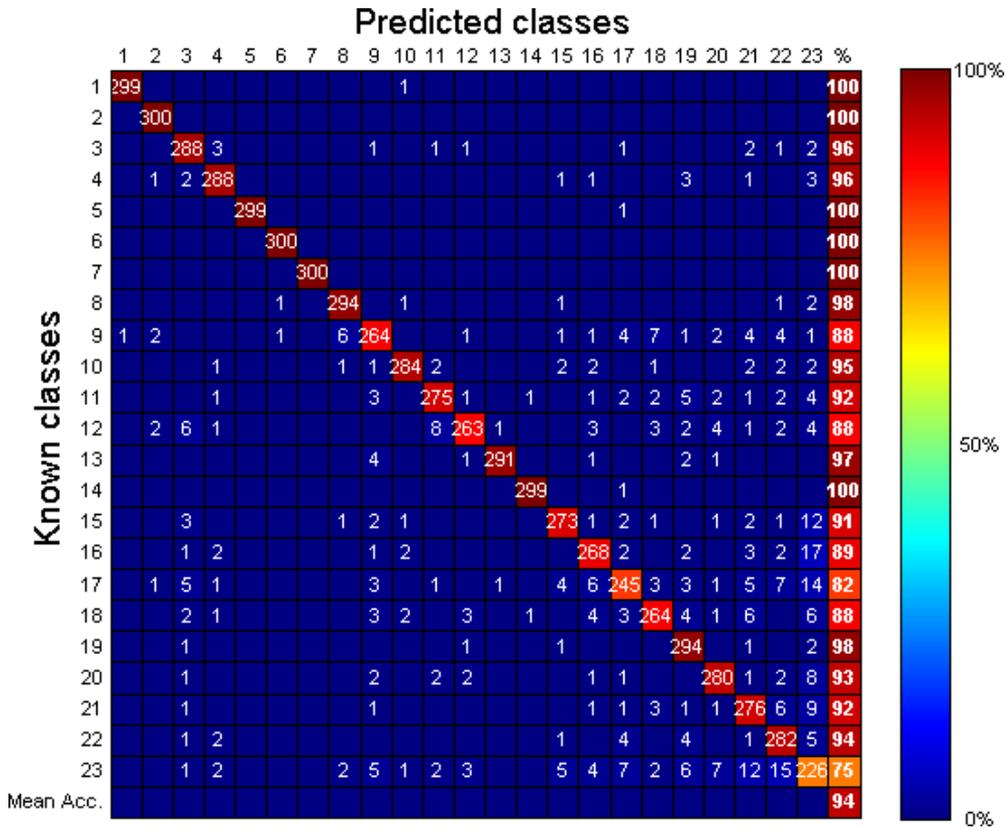


Figure 5.8: Confusion matrix between the ground truth class labels of a previously unseen test shape and predicted class labels for the RF with classes 23 and 24 merged together (Experiment 4). 300 samples from each class were made and 317 samples were predicted as belonging to class region 23. Average accuracy is 0.94.

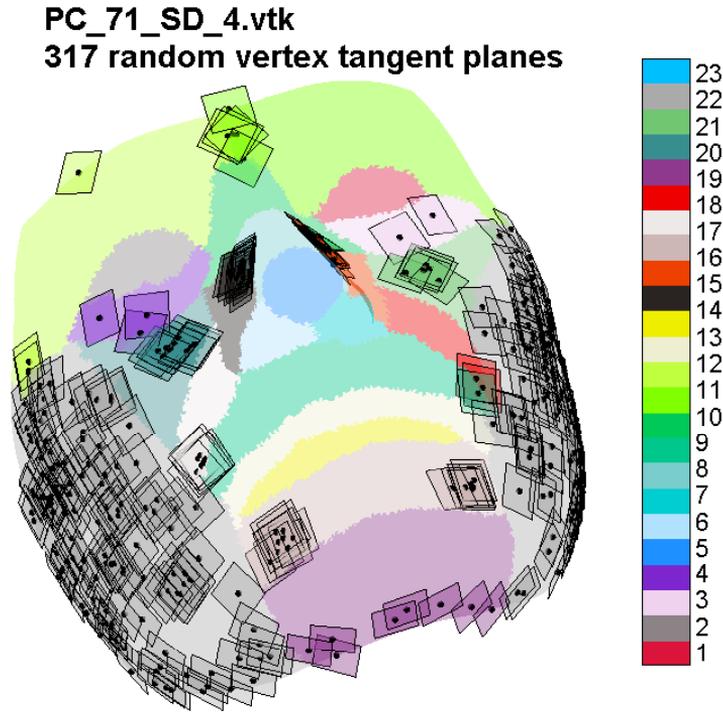


Figure 5.9: The 317 samples that were predicted to belong to the merged class 23/24 label. The labels can be seen in the confusion matrix in Figure 5.8.

plies that the concept of randomizing trees disappears. Therefore, this graph should not be used and is only there for reference.

The two measures for variable importance have been illustrated in Figures 5.13 and 5.14. The first figure shows the average decrease in accuracy for each of the 108 features during the training of the 150 trees for each of the 7 randomized decision forest models that corresponds with the 7 `mtry` values. The graph also depicts the average graph for the seven models as well as the average of this graph for each of the feature number intervals 1 to 36, 37 to 72 and 73 to 108 (green lines). These green lines correspond to the three scaling levels in Figure 5.11. The second figure shows the same computational procedures, only for the Gini impurity measure instead of accuracy decrease.

In order to test the performance of the randomized decision forests in Experiment 5 on previously unseen data, a test set consisting of 50 randomly sampled shapes with 150 random samples from each shape was created under the constraint that none of the shapes had been used in the model training. Upon computing the 108 features

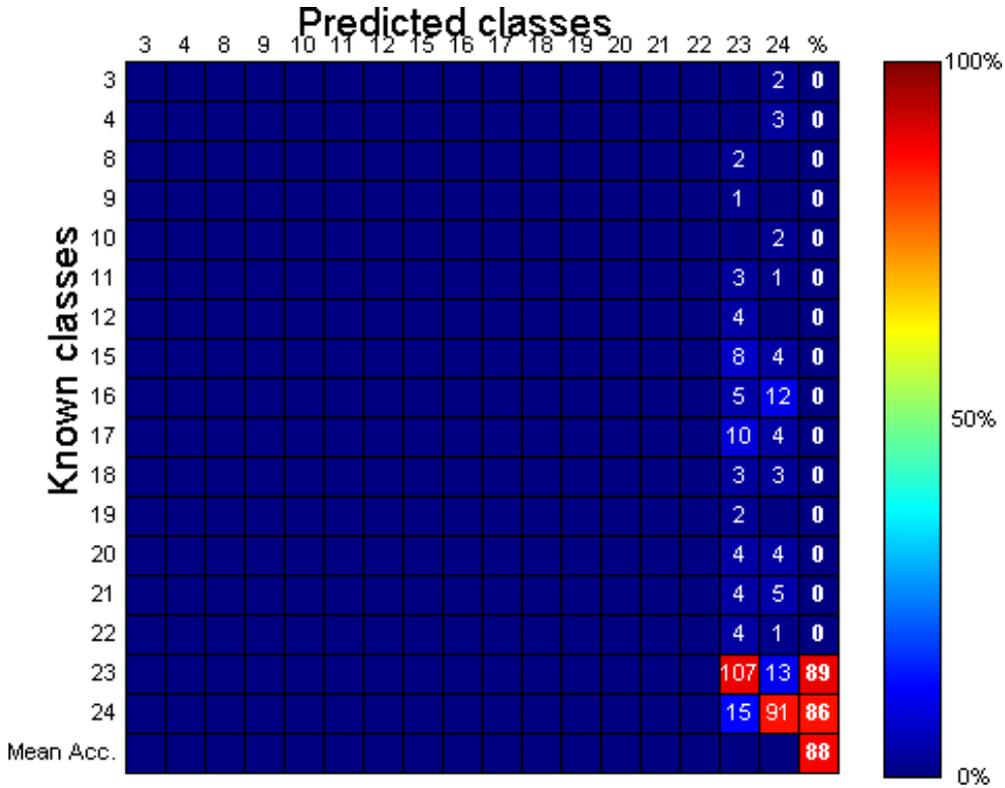


Figure 5.10: Confusion matrix between the ground truth class labels of a previously unseen test shape and predicted class labels for the RF that was based only on the classes 23 and 24 (Experiment 3). Average accuracy is 0.88.

for all the sampled vertices, the randomized decision forests corresponding to `mtry` values of 2,4,8,16,32 and 64 were tested and their average accuracy was computed. The last model was kept out due to the too large `mtry` value. Figure 5.15 depicts the average accuracy for the models. It can be seen that for a `mtry` value of 16, the test accuracy is highest.

The first conclusion that can be drawn from Experiment 5 is based on the bottom sub figure in Figure 5.12. The green graph that lies slightly above the thick, grey graph corresponds to the randomized decision forest that was trained for the `mtry` value of 4, which means that during the training of that model, for each node, 4 features out of 108 were considered at random. However, even though this was the case for the out-of-bag training error, the picture is another for the actual test data, which Figure 5.15 clearly witnesses. Here, it is evident that a `mtry` value of 16 yields

PC_110_SD_7.vtk 1 random vertex tangent planes

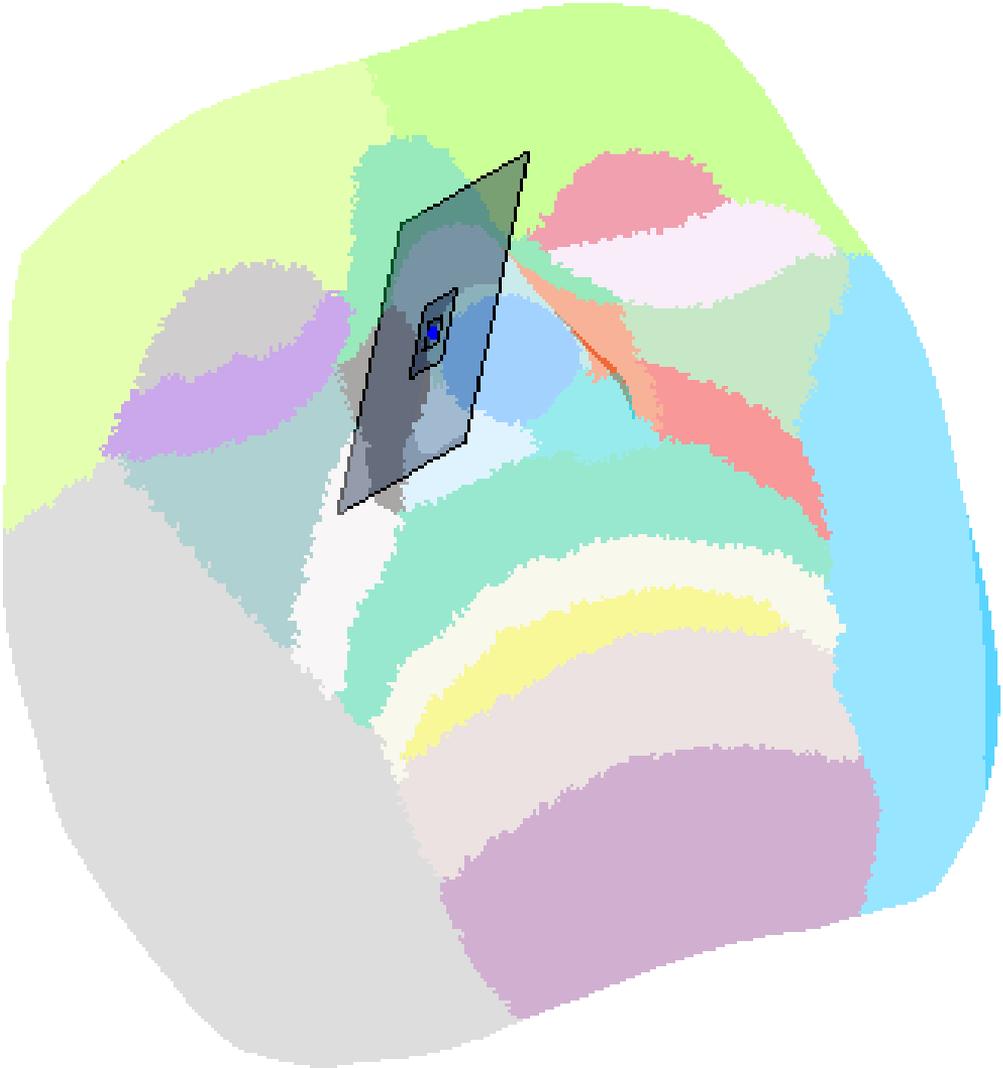


Figure 5.11: Experiment 5. Three example tangent planes for a random vertex based on the tangent plane scaling values 0.010, 0.025 and 0.100.

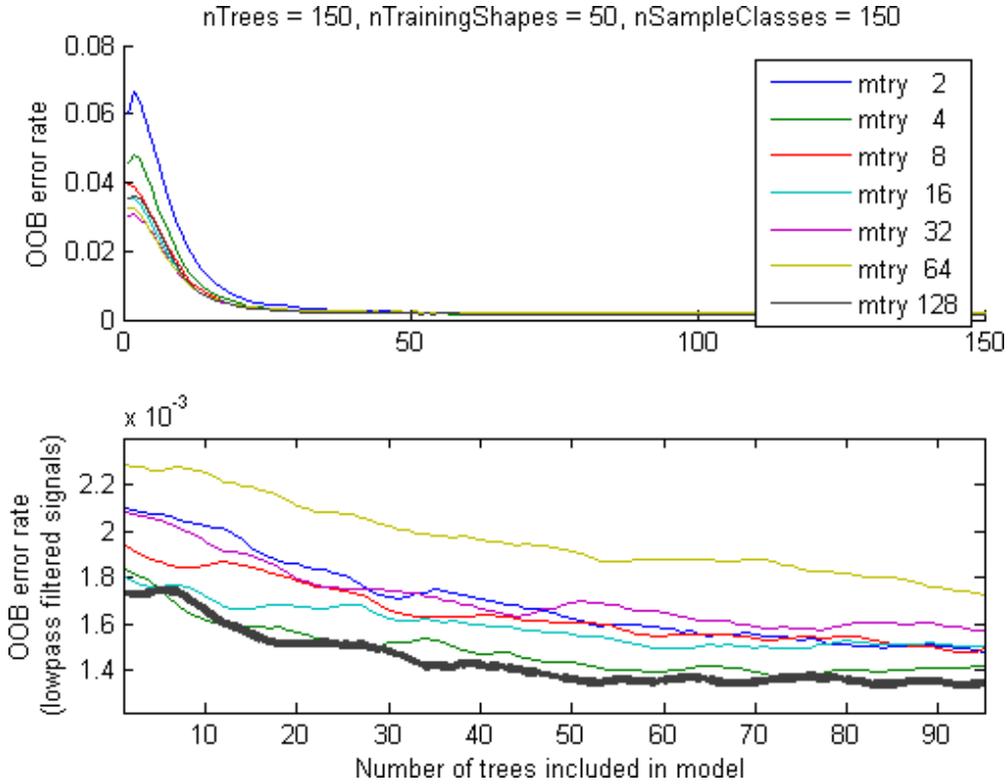


Figure 5.12: Experiment 5. Top: out-of-bag training error during RF training. Bottom: a close-up of the graphs with a low pass filter to smooth the graphs.

better test results on hitherto unseen data.

Another conclusive remark is based on the variable importance plots in Figures 5.13 and 5.14. Common for both measures of variable importance is that both signal a general "stair-case" tendency on the average graphs for the models, which have been shown with thick blue lines to enhance the perception of this effect. By averaging over the feature intervals that correspond to each scale, it is evident that the highest variable importances can be traced for the `scaleFactor` value 0.100. Due to the quite big difference in variable importance for a feature vector of length 108, further investigations is required to make more distinct analysis of the single scales.

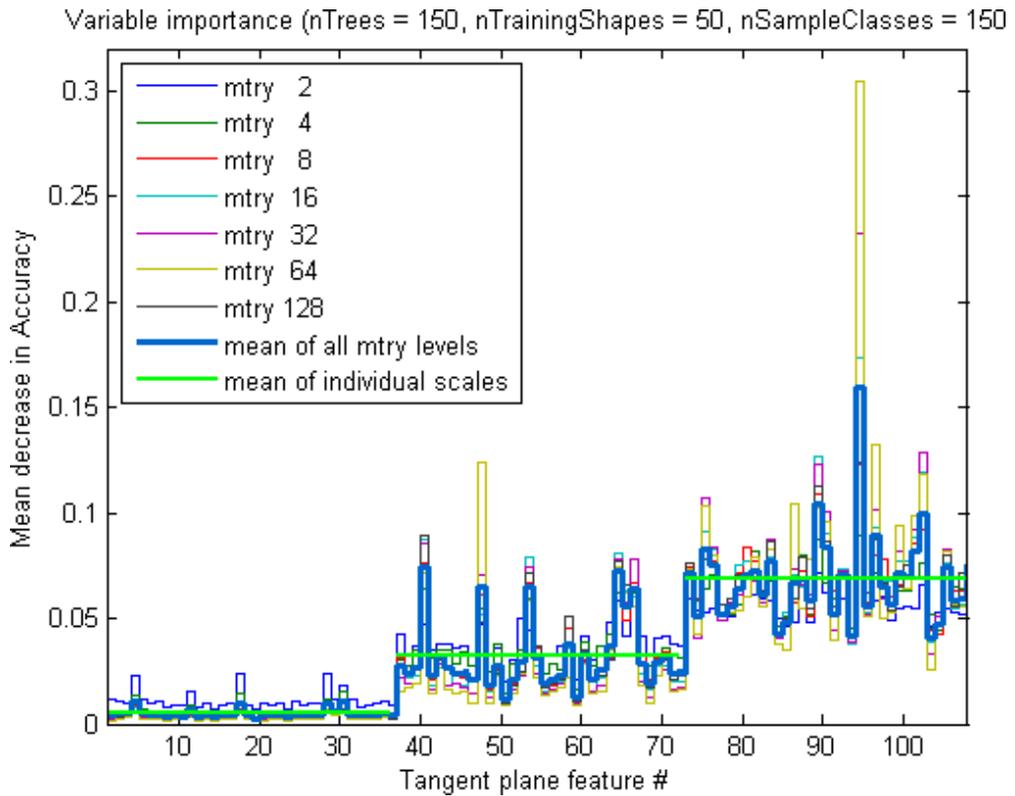


Figure 5.13: Experiment 5. The seven first stair graphs show the mean decrease in accuracy for the 108 features (average of the 150 trees). The thick blue stair graph is the average of the other stair graphs. The three green graphs are the averages of the thick blue graph in the three feature scale intervals 1 to 36, 37 to 72 and 73 to 108.

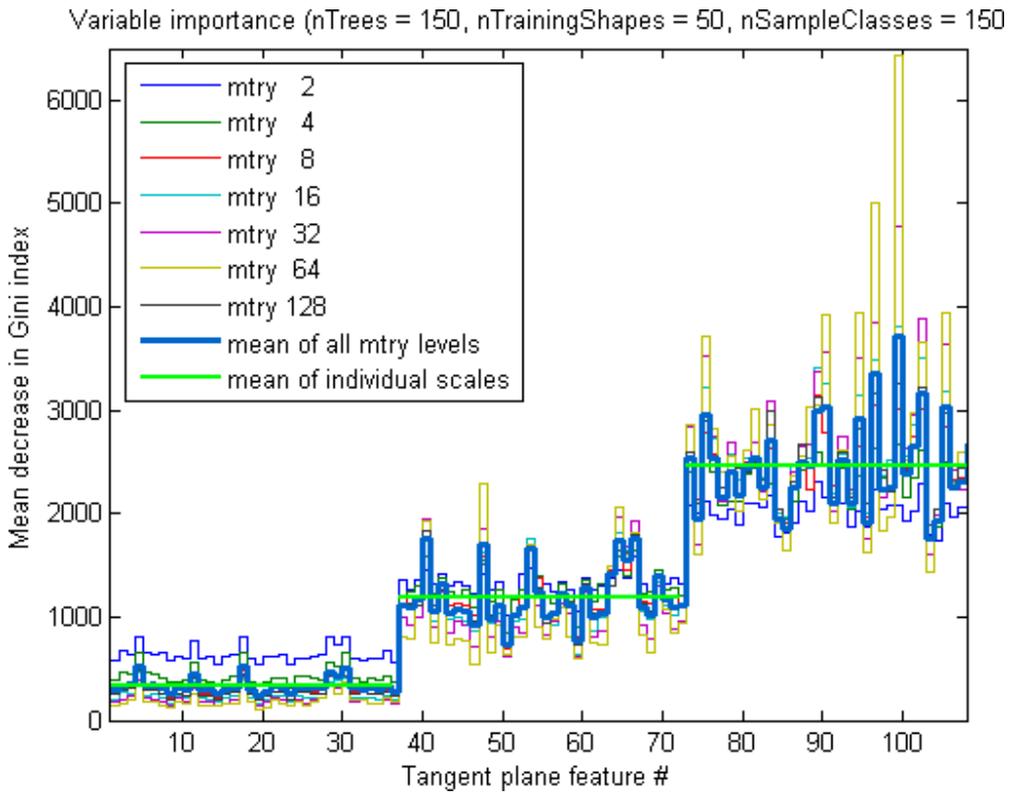


Figure 5.14: Experiment 5. The seven first stair graphs show the mean decrease in Gini impurity for the 108 features (average of the 150 trees). The thick blue stair graph is the average of the other stair graphs. The three green graphs are the averages of the thick blue graph in the three feature scale intervals 1 to 36, 37 to 72 and 73 to 108.

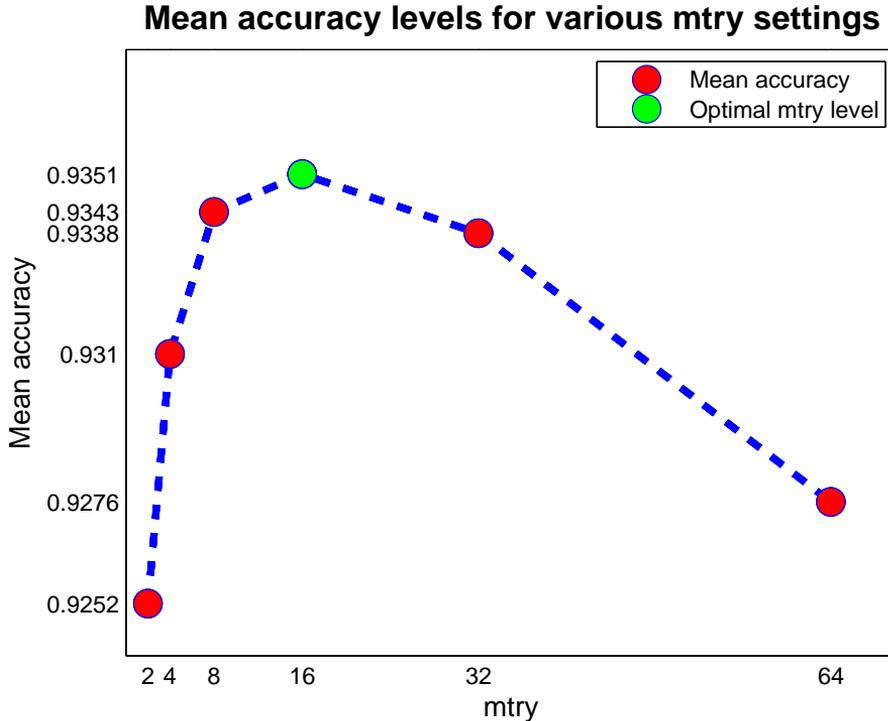


Figure 5.15: Experiment 5. The average accuracy for testing the 6 RFs corresponding to the `mtry` values 2,4,8,16,32 and 64. The test was performed on a large test set of 50 shapes and 150 random samples per class.

Experiment 6

The sixth and last experiment is inspired by the outcome of Experiment 5 that pointed towards the existence of some values for `scaleFactor` and `mtry` that could have potential influence on raising the average test accuracy. The main distinction between Experiments 5 and 6 is clear: `scaleFactor` is now varied in a different way from before. Instead of forming one long multi-scale feature vector, now, a series of multiple, single-scaled vectors are formed. This means that each feature vector now has the length 36 as was the case in Experiments 1 through 4. The `scaleFactor` values range from 0.005 to 0.100 in steps of 0.005, adding to a total of 20 values. Figure 5.16 shows an example of the corresponding tangent planes for a random vertex in a random shape.

Once again the `mtry` parameter is varied, but this time it is done for all possible values, which means that it will range between 1 and 36. This combination of 20 `scaleFactor` values and 36 `mtry` values leads to a system of 720 randomized decision

forests.

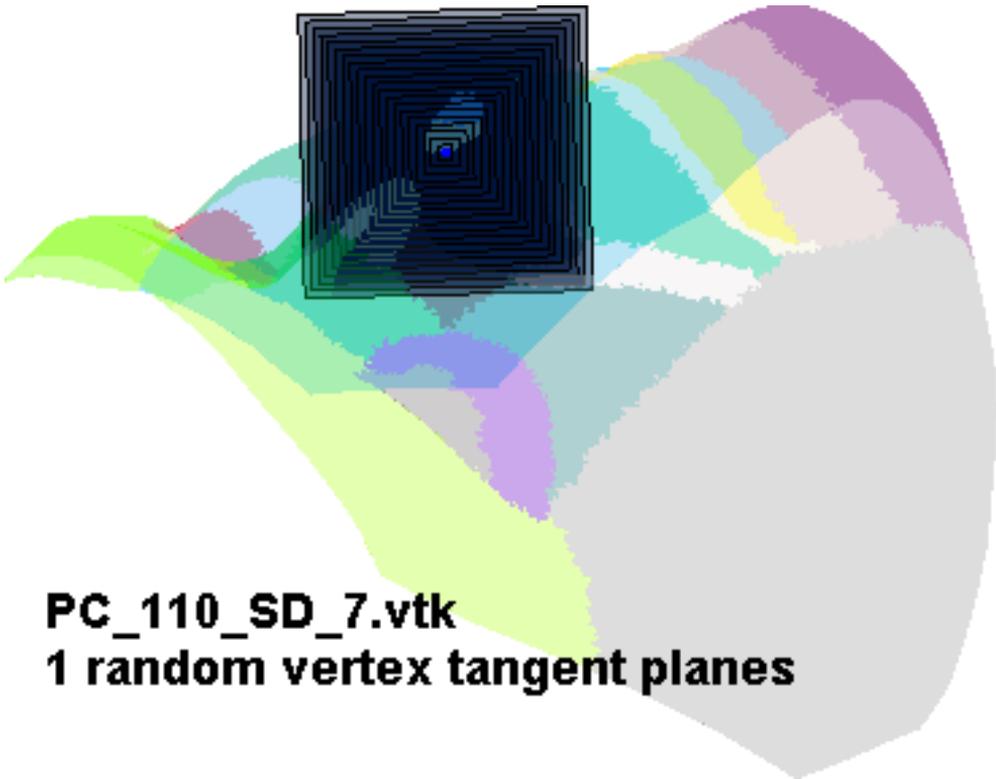


Figure 5.16: Experiment 6. Twenty example tangent planes for a random vertex based on the different tangent plane scaling values used in the experiment.

The first noticeable observation with this experiment comes from investigating the out-of-bag training error graphs that are shown in Figure 5.17. For each randomized decision forest corresponding to the 720 `mtry`, `scaleFactor` combination, the training error graph for increasing number of trees is shown. Two graphs have been highlighted, namely the best (yellow) and worst (red) models measured on their error rate at the inclusion of the 150th tree. The yellow graph corresponds to the `(mtry, scaleFactor)` setting `(14, 0.100)` and the red graph corresponds to the `(mtry, scaleFactor)` setting `(17, 0.005)`. The confusion matrices of the corresponding models are depicted in Figure 5.18. In this figure, it can be seen that the average accuracy is 84% in the best case and a mere 7% in the worst case. By investigation, it can be seen that the worst model has a very high tendency towards predicting observations to belong to either class 11, 7 or 24. By comparing to the ground truth Figure 1.2 and refer-

ence Table 4.2, it can be seen that this corresponds to the quasi-anatomical regions Left upper eyelid, eyebrow and lower forehead (L-UEELF), Left nostril and nose wing (L-NNW) and Left cheek (L-C), respectively.

OOB training error for all mtry - scaleFactor combo settings

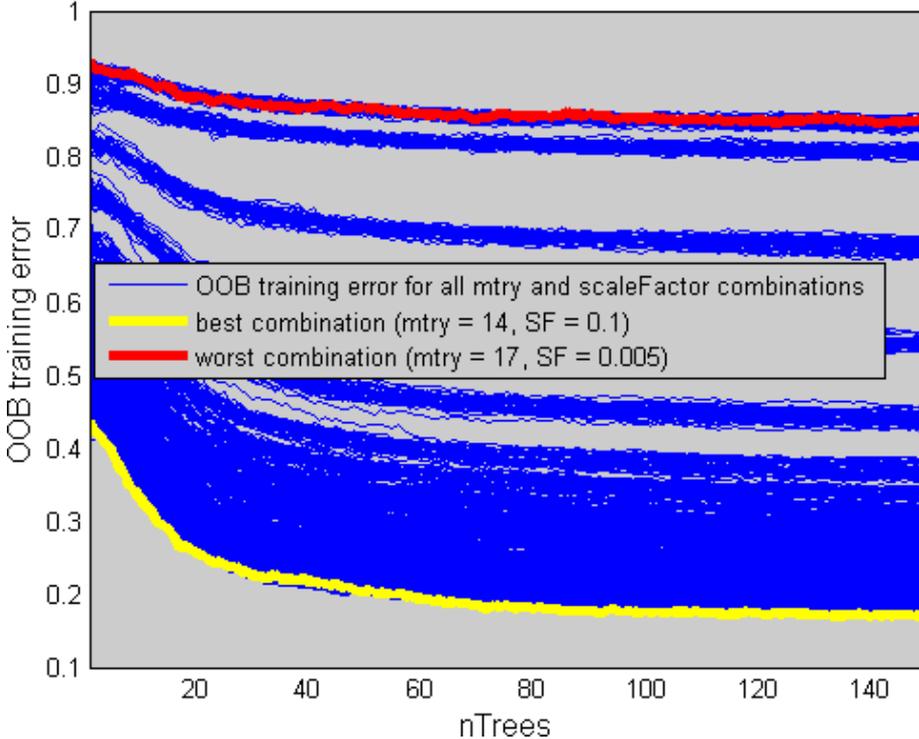


Figure 5.17: Experiment 6. 720 blue out-of-bag training error graphs, corresponding to all combinations of the `mtry` and `scaleFactor` parameter values. Yellow and red graphs show the models that exhibit the lowest training error for all 150 trees included. The best combination corresponds to a `mtry` value of 14 and `scaleFactor` of 0.1, which is the biggest tangent plane in Figure 5.16. The worst combination is at an `mtry` value of 17 and corresponding `scaleFactor` value of 0.005, which is the smallest tangent plane.

In order to test the performance of the 720 randomized decision forest models, a test set has been created in the following way: 50 random shapes have been sampled such that none of them have been included in the model training. Each shape has 150 samples taken from each class. For each model, the corresponding feature computation is governed by that model's `scaleFactor` value. While keeping the vertex

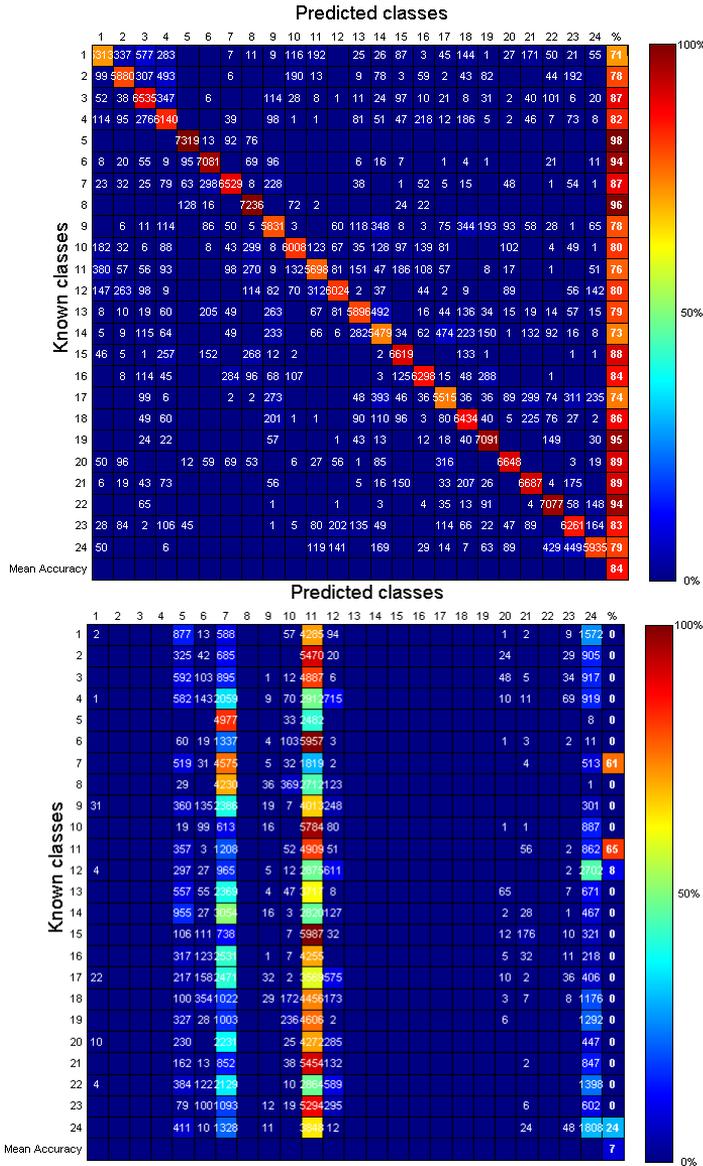


Figure 5.18: Experiment 6. Confusion matrices for the two RF models that exhibit the best (top) and worst (bottom) out-of-bag training error rate. See Figure 5.17 for the corresponding training error rates.

list and shape index fixed, the model is then tested for each of the 36 `mtry` settings. This is repeated such that all 720 randomized decision forests are tested on the same test data set for comparison.

In Figure 5.19, the average accuracy for the 720 randomized decision forests have been plotted. The abscissa is subdivided into 20 distinct regions that each comprise the 20 distinct `scaleFactor` values corresponding to the 20 different tangent plane feature scales that are visualized in Figure 5.16. Each of the subdivided regions contain the corresponding 36 `mtry` levels. On the figure is also marked the model that manifest the best performance with respect to the average accuracy (green). The model has the `(mtry,scaleFactor)` combination (10,0.100). The worst performing model has the `(mtry,scaleFactor)` combination (31,0.010), which is shown in red.

A different way of viewing the best and worst `(mtry,scaleFactor)` combinations is shown in Figure 5.20. The same nature of the `(mtry,scaleFactor)` combinations may be slightly more apparent in this figure. In Figure 5.19 it could easily be deduced that a high `scaleFactor` led to a better accuracy, but on Figure 5.20, besides that the same conclusion can also easily be deduced, in addition, it is revealed that lower `mtry` levels lead to a slightly better performance than higher. This is in accordance with the nature of the randomization functionality of the decision forests: if the `mtry` value is too high, too many features are considered at each node split, and this leads to similar conditions for the single trees of the forests. In the other end, if there is a low `mtry` setting, only very few features are sampled at each node split stage, which leads to very different trees and thus, higher randomization.

In order to look closer at the performance of the best and worst performing randomized decision forests in terms of their accuracy performance on the test set, Figure 5.21 shows the confusion matrices. In this figure, it can be seen that the average accuracy is 85% in the best case and 4% in the worst case. By investigation, it can be seen that the worst model has a very high tendency towards predicting observations to belong to either class 6 or 7. By comparing to the ground truth Figure 1.2 and reference Table 4.2, it can be seen that this corresponds to the paired quasi-anatomical regions `Right nostril and nose wing (R-NNW)` and `Left nostril and nose wing (L-NNW)`.

Mean accuracy levels for mtry - scaleFactor combo settings

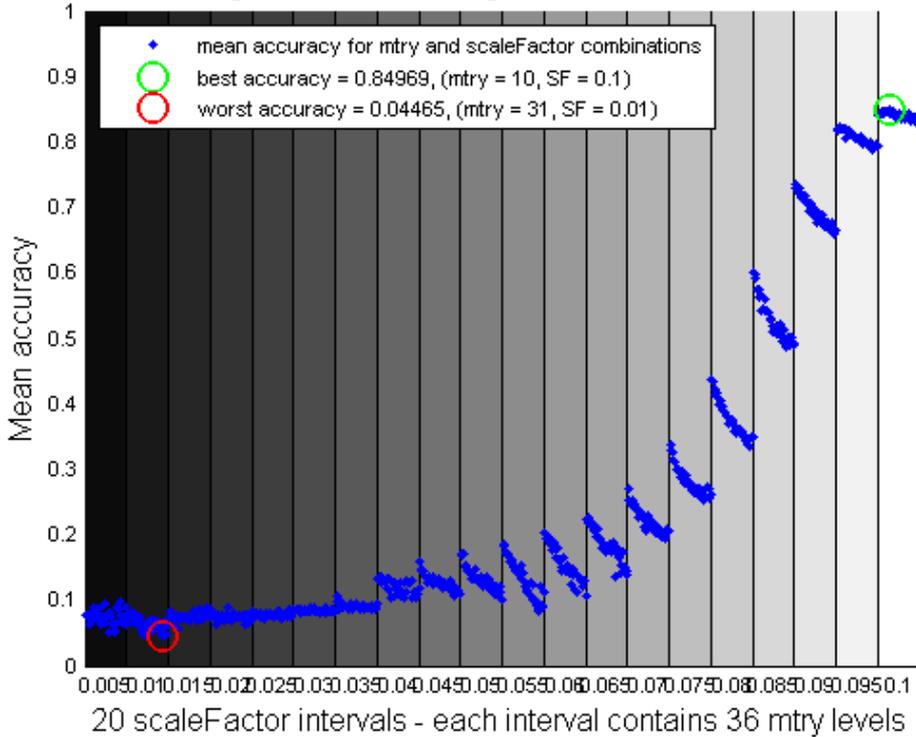


Figure 5.19: Experiment 6. Mean accuracies for the 720 RFs. Each of the 20 vertical grey-scaled subdivisions contain 36 models corresponding to 36 mtry values and illustrate each of the 20 scaleFactor values. The models that elicit the best and worst accuracies (green and red) are shown and correspond to the (mtry,scaleFactor) combinations (10,0.100) and (31,0.010) and associated mean accuracies of 85% and 4%. The corresponding tangent plane sizes are the biggest and the second-smallest in Figure 5.16.

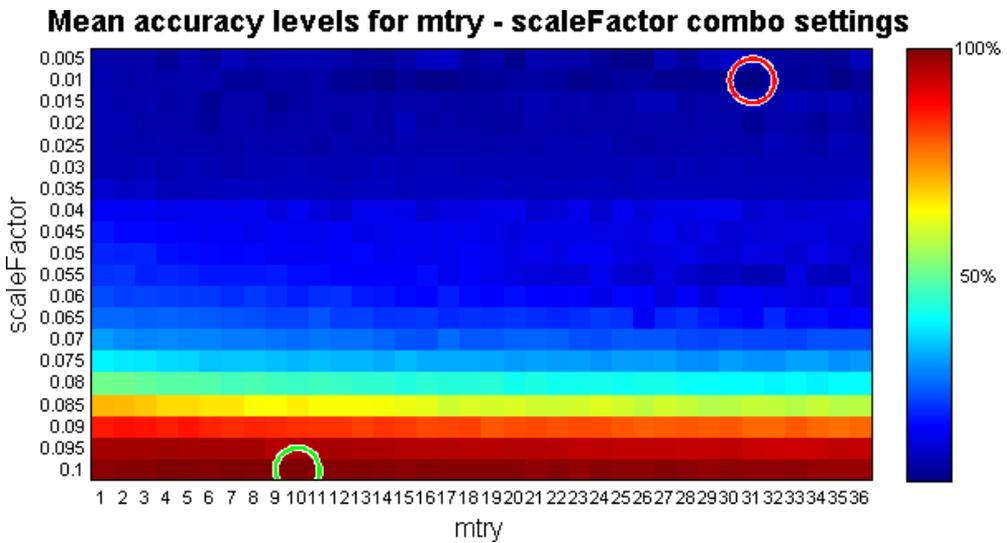


Figure 5.20: Experiment 6. Mean accuracies for the 720 RFs. This is an alternative approach to visualizing the results in Figure 5.20. The horizontal axis contains the mtry values and the vertical axis, the scaleFactor values. Each square in the image is color scaled according to its mean accuracy when subjected to the test data set. The best and worst RF model (mtry,scaleFactor) combination are marked with green and red rings, respectively.

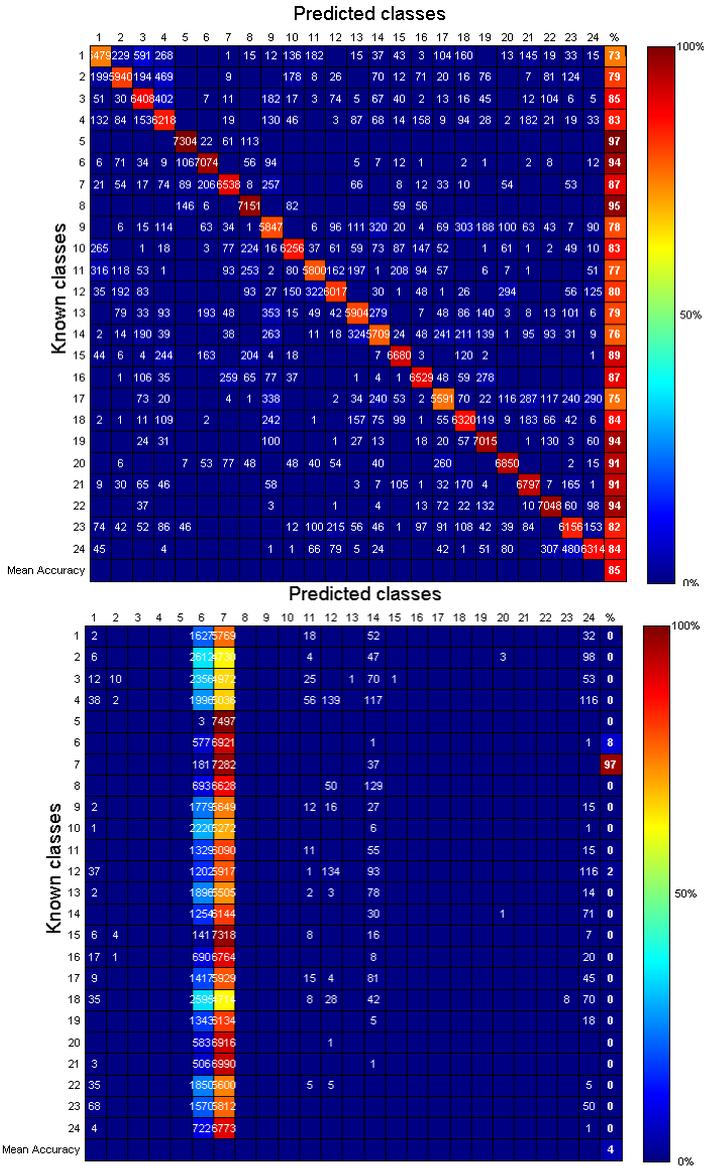


Figure 5.21: Experiment 6. Confusion matrices for the two RF models that exhibit the best (top) and worst (bottom) average test accuracies. When comparing to Figure 5.19, the following can be said about the two RF models. The models correspond to the (mtry,scaleFactor) combinations (10,0.100) and (31,0.010) and associated mean accuracies of 85% and 4%. The corresponding tangent plane sizes are the biggest and the second-smallest in Figure 5.16.

The main parts of the analysis has been:

- To generate a large plausible data set of frontal human faces from an active shape model.
- To create a set of weak classifiers.
- To use the weak classifiers to train randomized decision forests.
- To use the randomized decision forests and weak classifiers to classify fronto-facial anatomical regions on new scans.

The active shape model based data set consisting of 2023 shapes with plausible variation was formed and used in the training of various settings of randomized decision forests. The diagonal of the bounding boxes for all the shapes were computed and the variation enabled the creation of multiple forests that were different. Not only because of the randomization processes involved in the work of sampling the vertices and shapes but also because of the variability in the data set.

Inspired by Shotton et al. [SSK⁺13], the tangent plane features were developed as a tool for describing facial features. Because this is a novel idea, investigations on the features were done to analyze how they affected the performance of the randomized decision forests.

The experiments were focused upon investigations of the tangent plane feature parameters and also, upon the settings for training and testing the randomized decision forests.

Through the experiments it was revealed that some classes were difficult to classify properly. Namely, the two paired cheek regions spawned new experiments within the fields of uneven observation sampling, classifier cascades and multiscale feature computations. Naturally, such investigative work as this has some linear developmental stages as these.

In the event that someone would set out to do further work based on the present thesis work, the findings will give the future work a good head start by incorporating some of the learned principles, such as generating a large data set prior to commencing the work on building the feature descriptors. Once this is done, quite early on, initial tests of the classifier performance in relation to the number of trained trees is a good idea. Because this will give a good idea of how many trees are actually required. The advantage of identifying the number of required trees early is that this knowledge can

save much training and testing time in the subsequent experiments by lowering the number of included trees in the forests.

In the light of the cascading approach on building several randomized decision forests that complement each other, the following heuristics could lay the foundation for a new approach:

- Start by setting some initial parameters for the number of sampled shapes, number of samples per class, number of trees, tangent plane feature scale, mtry value and train a randomized decision forest.
- Make sure to save the model.
- Test the forest on a new set of shapes that were not included in the model training.
- Make the program automatically decide if some classes are hard to predict. As an example the program should pick the two paired classes, 23 and 24, that were manually chosen for further investigations in the experiments.
- Make the program make a split decision to automatically train a new forest containing a merged version of the classes 23 and 24 and also, a forest that only has the purpose of classifying into classes 23 and 24.
- Make the program test the cascade of the two models as was manually done in the experiments.

By making the program continuously break up the problems into smaller and train merged versions and dual-class forests - and saving the models along the way - a network of specialized randomized decision forests can be built up. It could be imagined that, when the network has become large, when presented with new data, by treating the data with the "sub-models", hard to predict classes could be predicted in this manner.

Conclusion

The analytical work of the present academic thesis has been founded upon the concept of training randomized decision forests for classification of anatomical regions of frontal human 3D scans based on weak classifiers. The main inspiration comes from Microsoft Kinect's body-part labelling system. A novel method has been invented for computing variables that are used in the training process of randomized decision forest. This "tangent plane feature" method uses simple computations to describe local differences between vertex tangent planes and surface parts.

The present analytical work comprises the concept of building up a large database of plausible frontal human surface shapes from an active shape model that is itself based upon over 600 actual facial scans. The use of such a model has allowed for investigations on tangent plane features and randomized decision forests that would otherwise not be possible due to the fact that randomized decision forests thrive on large amounts of training data.

Several experiments have been made in a linear, investigative practice. Some had the purpose of investigating the nature of randomized decision forests, namely the effect of the number of trees in a forest on the predictions. Other intrinsic parameters of randomized decision forests were investigated, such as the number of random features to sample from at each internal tree node. It was discovered that there was a slightly better performance when doing relatively few feature samples per node. This can be assigned to the fact that this increases the randomness between the trees in a forest, which again will lead to more randomized trees, and which, ultimately, is one of the main aspects of randomized decision forests: the randomization gives more different trees and thus, each tree's predictive abilities will cover a larger amount of the features that describe the anatomical regions.

Experiments on cascading classifiers with the purpose of raising the predictions of facial regions that were hard to classify have been carried out. However, no increase in the predictions were seen.

The framework of setting up both multi-scale feature based as well as multiple, single-scale feature based randomized decision forests led to interesting ways for determining what tangent plane feature scales could lead to the highest classification accuracy. It turned out that by scaling the tangent plane features by 10% of the diagonal of the bounding box of the shape and keeping a number of feature samples per node at 10 out of 36 possible features, the highest average prediction of 85% was reached.

The results from the present analysis makes the options of further investigating

the tangent plane features in a randomized decision forest setting, a sensible choice.

Future Work

The variation in the diagonal length of the bounding box for all shapes has been computed, but the variation between the shapes has not been further investigated. Since the active shape models and active appearance models were developed by Tim Cootes and his colleagues, many have investigated the variational nature of this principal component based method. The active shape model method is based on perturbations in the total number of vertices and as such, models based on for example sparse principal components could open up for an analysis between the relation between tangent plane features and local facial features that are unique for single persons.

In my bachelor thesis I investigated the correlation between sparse components of two dimensional mid-sagittal brain scans of the corpus callosum and parameters that were indicators of neurodegenerative diseases [Ots11]. This concept could be applied to a three dimensional setting in which local deformations of facial abnormalities are detected by the use of tangent plane features and randomized decision forests.

A natural expansion of investigations along the same road would be to incorporate a new data set based on an active shape model of the full human head.

The tangent plane features were inspired by Shotton et al. [SSK⁺13] who computed two dimensional depth images and base their weak classifiers on these intensity images. The data set that has been used in the present thesis work could be subject to a new analytical approach that involves the use of depth image features. The performance of that work would then be directly comparable to the present work.

The concept of cascading classifiers by the use of multiple randomized decision forests has been conducted in the present work. However, cascades of classifiers by a combination of tangent plane features and depth image features could also be a viable path of research.

Bibliography

- [AG97] Yali Amit and Donald Geman. Shape quantization and recognition with randomized trees. *Neural computation*, 9(7):1545–1588, 1997.
- [BC04] L. Breiman and A. Cutler. Random forests (fortran implementation). <http://stat-www.berkeley.edu/users/breiman/RandomForests/>, 2004.
- [BGAA12] Jakob Andreas Bærentzen, Jens Gravesen, François Anton, and Henrik Aanæs. *Guide to Computational Geometry Processing*. Springer, 2012.
- [Bre99] Leo Breiman. Random forests-random features. Technical report, Technical Report 567, Department of Statistics, UC Berkeley, 1999. 31, 1999.
- [BV03] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(9):1063–1074, Sept 2003.
- [CET⁺01] Timothy F Cootes, Gareth J Edwards, Christopher J Taylor, et al. Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence*, 23(6):681–685, 2001.
- [CILS12] Tim F Cootes, Mircea C Ionita, Claudia Lindner, and Patrick Sauer. Robust and accurate shape model fitting using random forest regression voting. In *Computer Vision–ECCV 2012*, pages 278–291. Springer, 2012.
- [CSK11] A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning. Technical Report MSR-TR-2011-114, Microsoft Research, Oct 2011.
- [CTCG95] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38 – 59, 1995.
- [CTF13] Hakan Cevikalp, Bill Triggs, and Vojtech Franc. Face and landmark detection by using cascade of classifiers. In *Title: 10th IEEE International Conference on Automatic Face and Gesture Recognition*, 2013.

- [DGFVG12] Matthias Dantone, Juergen Gall, Gabriele Fanelli, and Luc Van Gool. Real-time facial feature detection using conditional regression forests. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2578–2585. IEEE, 2012.
- [DRT11] Houtao Deng, George Runger, and Eugene Tuv. Bias of importance measures for multi-valued attributes and solutions. In *Artificial Neural Networks and Machine Learning–ICANN 2011*, pages 293–300. Springer, 2011.
- [FAHP13] Jens Fagertun, Tobias Andersen, Thomas Hansen, and Rasmus Reinhold Paulsen. *3D gender recognition using cognitive modeling*. IEEE, 2013.
- [FDG⁺13] Gabriele Fanelli, Matthias Dantone, Juergen Gall, Andrea Fossati, and Luc Van Gool. Random forests for real time 3d face analysis. *International Journal of Computer Vision*, 101(3):437–458, 2013.
- [FDVG13] Gabriele Fanelli, Matthias Dantone, and Luc Van Gool. Real time 3d face alignment with random forests-based active appearance models. *status: accepted*, 2013.
- [FHR⁺13] J. Fagertun, S. Harder, A. Rosengren, C.K. Moeller, T. Werge, and R.R. Paulsen. Variability of 3d facial landmarks. *submitted to Medical Image Analysis*, 2013.
- [HPL⁺13] Stine Harder, Rasmus R Paulsen, Martin Larsen, et al. A three dimensional children head database for acoustical research and development. In *Proceedings of Meetings on Acoustics*, volume 19, page 050013. Acoustical Society of America, 2013.
- [Jai13] Abhishek Jaiahtilal. "randomforest-matlab". random forest (regression, classification and clustering) implementation for matlab (and standalone). <http://code.google.com/p/randomforest-matlab/>, 2013.
- [LW02] Andy Liaw and Matthew Wiener. Classification and regression by randomforest <http://CRAN.R-project.org/doc/Rnews/>. *R News*, 2(3):18–22, 2002.
- [Ots11] N. T. Otsu. Shape analysis of brain structures, bachelor thesis, 2011. Supervised by Professor Rasmus Larsen, rl@imm.dtu.dk, DTU Informatics.
- [Pau] Rasmus Reinhold Paulsen. Sumatra (the surface manipulation and transformation toolkit). <http://www2.imm.dtu.dk/image/MRFSurface/download.html>.

- [PEK⁺12] OB Pedersen, Christian Erikstrup, Sebastian Ranzi Kotze, E Sørensen, MS Petersen, Katrine Grau, and Henrik Ullum. The danish blood donor study: a large, prospective cohort and biobank for medical research. *Vox sanguinis*, 102(3):271–271, 2012.
- [PKA⁺09] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, pages 296–301. IEEE, 2009.
- [SG02] Mikkel B Stegmann and David Delgado Gomez. A brief introduction to statistical shape analysis. *Informatics and Mathematical Modelling, Technical University of Denmark, DTU*, 15, 2002.
- [SSK⁺13] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.
- [YP13] Heng Yang and Ioannis Patras. Face parts localization using structured-output regression forests. In *Computer Vision—ACCV 2012*, pages 667–679. Springer, 2013.
- [ZR12] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2879–2886. IEEE, 2012.

