AUTOMATIC SEGMENTATION FOR DENTAL OPERATION PLANNING

DIPLOMA THESIS

by Nguyen The Duy

Teacher in charge:
Prof. Dr. rer. nat. Stefan Gumhold
Professur für Computergraphik und Visualisierung
Technische Universität Dresden

Supervisor:
Dr. Stefan Zachow
Forschungsgruppe Medizinische Planung
Zuse Institut Berlin
Selbständigkeitserklärung

Ich, Nguyen the Duy, Matrikel-Nr. 3421516, versichere hiermit, dass ich meine Diplomarbeit mit dem Thema

*Automatic segmentation for dental operation planning*

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, wobei ich alle wörtlichen und sinngemäßen Zitate als solche gekennzeichnet habe.

Dresden, den ___________________________
Abstract

In this diploma thesis, we propose algorithms to automatically segment maxilla bones, isolate and number the upper teeth in head CT scans. Our maxilla segmentation method is based on statistical shape models (SSM). To drive the adaption process of the SSM, we have developed a method to analyse image data and compute the desired transformations and deformations.

Moreover, we propose a tooth detection algorithm which works on an adapted maxilla SSM and a head CT image. The algorithm divides the row of teeth into 16 regions using Dijkstra optimisation. Each of the regions either contains a tooth or a tooth gap. Afterwards, a classifier using histogram features is applied on the regions to identify the tooth. In our experiments, we achieved a maxilla segmentation accuracy of $\approx 0.5 \pm 0.5\text{mm}$ mean 3D surface distance. The tooth detection algorithm isolated teeth with a volume overlap of 39% – 96%. Finally, classification rate ranged from 51% – 96%.
Acknowledgements

First, I would like to thank my supervisor Dr. Stefan Zachow who introduced me into the topic of this diploma thesis and provided me with invaluable support and feedback.

Furthermore, my sincere thanks go to Prof. Dr. rer. nat. Stefan Gumhold who allowed me to write this diploma thesis at ZIB and gave me inspiration for solutions.

Moreover, I highly appreciate the support of my colleagues Dr. Hans Lamecker, Dagmar Kainmueller, Heiko Ramm, Carl-Martin Grewe, Max Kahnt, Robert Lieck and Moritz Ehlke who showed me interesting ideas for the improvement of my work.

Last but not least, I wish to thank my parents and my friends for their persistent support.
5.3.1 Experimental setup ................................................................. 50
5.3.2 Tooth separation parameters .................................................. 50
5.3.3 Tooth classification ................................................................. 54

6  Discussion .................................................................................. 56
  6.1 Maxilla segmentation .............................................................. 56
      6.1.1 Initialisation of the maxilla SSM ........................................ 56
      6.1.2 Automatic Threshold Finding .......................................... 56
      6.1.3 Rule-based maxilla cost function ...................................... 56
  6.2 Tooth separation ................................................................. 57
      6.2.1 Dimension of arc image .................................................. 57
      6.2.2 Search space ................................................................. 57
      6.2.3 Plane cost function ........................................................ 58
      6.2.4 Comparison to the Integral projection method .................... 58
  6.3 Tooth classification ............................................................. 58
  6.4 Performance on CBCT images ................................................ 60

7  Conclusions .............................................................................. 61
Chapter 1

Introduction

1.1 Motivation

Dental implantology is becoming more and more popular due to aesthetic and functional advantages. A successful operation requires that bone quantity and quality of maxilla (upper jaw) and mandible (lower jaw) is sufficient to ensure stability. Furthermore, it is essential that bone anchored implants are placed in such a way that they do not damage vulnerable structures e.g. dental roots or mandibular nerve.

As the planning step requires exact dimensioning of jaws and teeth, operation planning based on 3D computed tomography (CT) images has become popular. 3D images enable to exactly localise and measure relevant anatomic structures. However, tomography images tend to consist of a huge number of slices (over 100) which makes manual analysis time-intensive. An algorithm, which is able to segment jaws and teeth in head CT images without human interaction would significantly reduce effort.

1.2 Problem statement

The goal of this thesis is to design algorithms for automatic segmentation of the maxilla bone and numbering (tooth identification) of the upper row of teeth within tomographic data. Our work is based on the segmentation framework proposed by Seim et al. [SKH+08] and Kainmueller et al. [KLS+09].

In the first step, we modify the mandibular bone segmentation algorithm by Kainmueller et al. to segment the maxilla bone. We evaluate our algorithm on CT and cone beam CT (CBCT) images.

Afterwards, we use the results of the segmented maxilla to locate and number each upper tooth. We also measure tooth detection performance on CT images.

1.3 Image data

In this thesis, we work with tomographic images. Tomographic images describe the inner structure of objects by a series of slices. Whereas traditional X-Ray radiography projects the object volume on a 2D image, tomography provides more information as it virtually cuts the object into slices and creates an image out of each. Using tomographic images, a 3D model of the object can be computed which reveals more details. The intensity values on X-ray radiographies or CT images are measured in Hounsfield units (HU). This unit is named after the developer of the first CT scanner, Sir Godfrey Hounsfield. It is an arbitrary scale to describe the radiodensity of materials. As an example, water has the value of 0 HU whereas air is set to be -1000 HU. Table 1.1 shows typical hounsfield units for some materials.

Our algorithms work on CT and cone beam computed tomography (CBCT/CBVT) images. CBCT images provide a worse imaging quality than CT as they contain more noise and the field of view (fOv) is smaller. Additionally, intensity contrast on CBCT images is lower than on CT images making it harder to identify structures. However, radiation dose needed to create CBCT images is lower which makes this technology less harmful for the patient. For this reason, CBCT images have become an attractive alternative in dental implantology.
Table 1.1: Hounsfield units of some materials in CT. Source: [http://en.wikipedia.org/wiki/Hounsfield_scale](http://en.wikipedia.org/wiki/Hounsfield_scale)

<table>
<thead>
<tr>
<th>Material</th>
<th>Hounsfield units</th>
</tr>
</thead>
<tbody>
<tr>
<td>air</td>
<td>−1000 HU</td>
</tr>
<tr>
<td>soft tissue</td>
<td>−300 HU to −100 HU</td>
</tr>
<tr>
<td>water</td>
<td>0 HU</td>
</tr>
<tr>
<td>jaw bone</td>
<td>505 HU ± 274 HU</td>
</tr>
</tbody>
</table>

Figure 1.1: Differences between CT and CBCT images

1.4 Outline

This thesis is organised as follows:

In **Chapter 2**, we give an overview of existing methods for jaw segmentation and tooth numbering. Furthermore, we describe how statistical shape models (SSM) are used for segmentation.

In **Chapter 3**, we present our maxilla segmentation algorithm.

Using the results of the maxilla segmentation routine, we describe our tooth detection algorithm in **Chapter 4**.

In **Chapter 5**, we focus on the experiments conducted to validate the performance of our algorithms.

In **Chapter 6**, we discuss the results from the experiments, spot characteristics and point out potential for future work.

Finally, we draw a conclusion of our work in **Chapter 7**.
Chapter 2

Related work

2.1 Mathematical notation

In the following, we explain some mathematical notation we use throughout our thesis.

An image $I$ is a continuous mapping $I : X \rightarrow C$ where $X \subset \mathbb{R}^n$ is the set of spatial coordinates in the image domain and $C \subset \mathbb{R}^m$ the colour space. We work with 3D grey value images, such that $X \subset \mathbb{R}^3$ and $C \subset \mathbb{R}^m$. The colour space $C$ in our images stays within an interval $[I_{\text{min}}, I_{\text{max}}]$. We either use $I(x, y, z)$ or $I(\vec{x})$ to express the intensity of the image at a 3D point $\vec{x} = (x, y, z)$. Moreover, we call label fields $L(x, y, z)$ or $L(\vec{x})$ images with the colour space $C \subset \mathbb{N}$.

Furthermore, we work with triangulated surfaces. A triangulated surface $S = (V, E, F)$ is a compact, piecewise linear surface, which consists of finite sets of simplices (vertices $V$, edges $E$, and triangular faces $F$) whose connectivity is defined by a simplicial complex $K$. The triangulated surface is geometrically embedded in $\mathbb{R}^3$. Throughout our thesis, we sometimes refer to triangulated surfaces simply by surface.

2.2 Jaw segmentation

Our first task is to segment the maxilla bone. In literature, 2 approaches are used for segmentation of jaw bones:

Firstly, adaptive threshold segmentation techniques are applied. Barandiaran et al. [BMB+09] uses Otsu’s thresholding algorithm [Ots75] to determine optimal thresholds and region growing to achieve compact segmentations. Tognola et al. [TPP+06] on the other hand, proposes histogram equalisation in order to find an optimal threshold value. None of the 2 have stated quantitative results in their work.

CT images often contain noise from metal artefacts which distort the original image information. For this reason, methods purely based on image intensities can perform in this scenario. To overcome these shortcomings, model-based segmentation algorithms are used. The key assumption is that the structures to segment have typical appearance. Therefore, a model can be learned from reference segmentations which contain information about local image intensities and shape variation. After training the model, segmentation is done by transforming and deforming the model to achieve the best fit into image data. Mathematical optimisation techniques are used for this purpose.

Lilja et al. [LVA+07] proposes a method to segment the mandibular bone in limited angle CT images. In this kind of data, only a fraction of the data images the true geometry of the object whereas the other part is imaged in a distorted way. Thus, the other parts have to be discarded when segmenting the object of interest. The authors use an average shape model which was built from 40 aligned training shapes. To segment a new dataset, the authors crop the “unreliable” parts in the image and register the model using affine and elastic registration. The method is evaluated on 13 datasets with an average segmentation error of $0.32\text{mm} \pm 0.65\text{mm}$.

Rueda et al. [RGPAn06] proposes the usage of an active appearance model (AAM) for mandible segmentation in CT images. AAMs were first proposed by Cootes et al. [CET01]. Rueda et al. train an AAM for shape and intensity. Landmarks points represent the mean shape and intensity sampling over its surface gives an intensity model. To segment an unknown dataset, the model is used to generate synthetic images of the object. The one which best fits the original image, represents the solution of the problem. Performance evaluation is done for 215 datasets resulting in an average segmentation error of $1.63\text{mm} \pm 1.63\text{mm}$. 
Statistical shape models (SSM) \cite{CHTH94} are used by Lamecker et al. \cite{LZW06} and Kainmueller et al. \cite{KLS09} for mandible jaw segmentation. The SSM (see section 2.4) is generated from training shapes based on manual segmentations. The segmentation process involves image data driven deformation of the SSM using shape modes. Shape modes describe how regions of the object vary between datasets. Therefore, deformations using shape modes ensure that the shape of the deformed model remains “realistic”. In experiments, Lamecker et al. \cite{LZW06} achieve a mean segmentation error below 0.2cm in all datasets and Kainmueller et al. \cite{KLS09} a mean volume overlap to the gold standard segmentations (ground truth) of 88.4%.

As experimental results show, these routines are able to produce precise segmentations without human interaction. However, segmenting bones in regions where tooth breaks through the jaw remains problematic. The bone intensity value could vary significantly through high teeth intensity in the neighbourhood. Reconstruction kernels with smoothing effects and a low image resolution can amplify this effect. Finally, individual position of teeth makes it hard to integrate prior information into the model. As far as we know, past work has not focused on this problem, so far. A workaround is to build a model without considering the region where teeth break through the bone \cite{LZW06, KLS09}.

2.3 Tooth detection and numbering

The task of tooth numbering involves identifying teeth seen on an image and name them according to a standardised system. Recently, 2 numbering schemes are common (see fig. 2.1). A subtask of tooth numbering is tooth classification where only the kind of tooth matters. Teeth can be classified into molars and premolars or, to be more precise, molars, premolars, canines and incisors. In our work, we name teeth using the FDI numbering system (see fig. 2.1(b)).

![Tooth numbering systems](http://users.forthnet.gr/ath/abyss/dep1151_1.htm)

Numbering of the whole row of teeth is a problem which is often simplified by separately analysing each of its tooth. A popular preprocessing step is to isolate teeth by finding regions of interest (ROIs) containing only one tooth. Further classification is then done on each ROI.

Past work \cite{MA05, NAA08, LLH10} has preferred the integral projection method for tooth isolation. We explain this procedure on a 2D image $I(x, y)$ (see fig. 2.2(a)): The assumption is that it is possible to separate two adjacent teeth by a straight line $L^\Theta$. In the case of a “good” separation line, the sum of intensity values $I_{sum}$ along $L^\Theta$ will be smaller than $I_{sum}$ in a local environment. Due to different teeth alignment, the separation lines $L^\Theta$ do not have to be vertical, but can be rotated within a small range. The task of this method is to find lines with locally minimal $I_{sum}$.

Let $I = L^\Theta(x_{ref}, y)$ be a linear function which represents a separation line. $L^\Theta$ returns for each given $y$ coordinate the corresponding $x$ coordinate on $L^\Theta$, where $L^\Theta(x_{ref}, 0) = x_{ref}$. Figure 2.2(a) shows such a line (red). The angle $\Theta$ denotes how much the $L^\Theta$ is tilted. $\Theta = 0^\circ$ means that the line is vertical. The task is to find $\Theta^*$:

$$\Theta^* = \arg \min_{\Theta} I_{sum}(\Theta, x_{ref}) = \arg \min_{\Theta} \sum_{y} I(L^\Theta(x_{ref}, y), y)$$

(2.1)

Figure 2.2(b) shows the values of $I_{sum}(\Theta^*, x_{ref})$ for all $x_{ref}$. The $I_{sum}$ of red separation line in fig. 2.2(a) is marked by a red arrow.
Figure 2.2: Integral projection method

The method assumes lines with local minima of $I_{sum}(\Theta^*, x_{ref})$ the separation lines.

After this tooth isolation step, the actual classification is done for each tooth ROI. In literature, two classes of features are used: either the local area of the object is analysed (area based) [NAA08] or the shape (boundary-based) [MA05, LLH10]. Boundary based approaches require a segmentation containing the exact boundaries of the tooth whereas area based approaches work on rough tooth segmentations. For classification, a variety of machine learning techniques are applied.

Mahoor et al. [MA05] proposes a tooth numbering algorithm for 2D radiographs. First, the teeth ROIs are roughly classified into molars and premolars using a bayesian classifier. As features the authors use fourier descriptors computed from silhouettes of segmented teeth. After the rough classification step, a rule-based system corrects misclassifications. These rules consider the arrangement and fourier descriptors of the classified teeth. Finally, the corrected sequence is numberered using an other set of rules. Experimental results state a classification rate of 92.0% – 95.5%. However, only full rows of teeth are considered. No routine is described to handle the case of tooth gaps.

Nassar et al. [NAA08] propose a method which is able to deal with missing teeth on 2D radiographs. As preprocessing for classification, the dental image is divided into tooth ROIs and gap ROIs, respectively. The tooth ROIs are cropped and rotated in order to get a vertical orientation of the crown. Afterwards, the ROIs are classified as molars, premolars, canines or incisors. To learn a tooth appearance model, ROIs of each class are aligned and their mean images are computed. Eigenvectors from a principle component analysis (PCA) of each class’ training set (“Eigenteeth” images [TP91]) are used to describe intra-class variance. Using these 2 components, a new tooth image is approximated and the class with minimum least squares error between approximation and original tooth is chosen. Moreover, the gap
size between ROIs are considered and classified as missing tooth if they exceed certain manually chosen
threshold. Finally, the sequence of classified teeth and gaps is numbered by solving a string matching
problem. The gaps are considered as “don’t care” characters and based on the classified sequence, a
tooth sequence with minimum correction distance to a standard row of teeth is computed. Experimental
results state a classification precision of 91.67% for molars and 92.86% for premolars.

Lin et al. [LLH10] also use the combination of single tooth classification and string matching to solve
the numbering problem for 2D radiographs. After image preprocessing and ROI detection steps, the
algorithm requires tooth ROIs and gap region sizes as input. In order to perform a classification of the
ROIs into molars and premolars, a support vector machine (SVM) is trained. A feature vector consists of
the ROI’s relative width-height ratio and its relative crown size. After the classification step, a sequence
of molars and premolars is created. In order to detect missing teeth, the authors analyse the size of
the gaps between two teeth. Depending on the neighbouring crown sizes, the number of missing teeth
is computed by solving a simple optimisation problem. After adding missing teeth into the sequence,
the last step is aligning this sequence with the sequence of a full teeth row. The Smith-Waterman
algorithm [SW81] from the field of bioinformatics is modified for this purpose. The ROIs are numbered
based on the alignment result. Tests on datasets state a numbering accuracy of 98%.

To sum up, state-of-the-art approaches prefer a two-step numbering scheme. After isolating teeth
using integral projection method, each ROI is independently classified. In this step, boundary-based fea-
tures (shape, relative size) as well as region based features (intensity distribution) are used. Afterwards,
misclassifications are corrected by considering arrangement and neighbourhood. As far as we know, there
is recently no algorithm which is able to perform numbering in one step and thus avoid intermediate
misclassifications. This would avoid a strong dependence between single components.

Furthermore, 3D images have been rarely used, so far. A reason could be that artefacts caused by
dental fillings or metallic implants [SAY+98] distort the image information in a more severe way than in
2D radiographs.

Finally, the open question remains of how to number teeth which have completely moved to the place
of another one. This problem occurs in the case of braces. We believe that appearance and arrangement
have to be considered jointly in order to solve this scenario.
2.4 Statistical shape models (SSM) for medical image segmentation

2.4.1 Definition

Statistical shape models (SSM) are proposed by Cootes et al. [CHTH94] to analyse the geometry of a set of shapes using statistics. SSM have become popular for segmentation of medical images since the structures to be segmented have similar shape. Kendall [Ken84] defines this term as

"Shape is all the geometric information that remains when location, scale and rotational effects are filtered out from an object."

The SSM is created by applying statistical analysis on a set of training shapes and reveals object specific shape characteristics, especially variation around the average shape. To represent shape, a point distribution model is used. In segmentation using SSM, the degrees of freedom in shape variation are limited to a given shape space and thus only “object-characteristic” shapes are considered as legal segmentation.

2.4.2 Generation of SSMs

The SSM is learned from a set of manual segmentations or training shapes which contain characteristic shape variations. To answer the question where and how the shape varies, mathematical statistics is applied.

In order to be able to apply statistics, a representation for shape has to be introduced. Cootes et al. [CTC92] uses a set of points, called landmarks, to describe each training shape. Either these points are manually set or they are automatically generated e.g. through meshing. It is essential to maintain point correspondences between training shapes. For each point in a training shape, there is exactly one point in every other training shape which represents roughly the same part of the object. Furthermore, neighbourhood relations between points should be equivalent to the relations between their corresponding points. These conditions imply that each shape has to be represented by an equal number of points.

![Figure 2.3: Point correspondences between two shapes. For each point of the shape 1 a coloured arrow indicates the corresponding point in shape 2 and vice versa.](image)

Maintaining the correspondence condition can be challenging for automatically generated point sets e.g. through meshing. Lamecker et al. [LSHD04] proposes a method to establish correspondences between 2 topologically equivalent 3D shapes. The idea is to manually divide these shapes into regions, called patches. In order to get meaningful correspondences, the decomposition has to be topologically equivalent and the patches themselves should roughly represent the same anatomic regions. The actual correspondence problem is solved between pairs of corresponding patches by applying a common parametrisation that is mapped to each shape.
Input training shapes may have different pose and scale. However, we are only interested in shape characteristics. Individual transformation has to be eliminated by aligning the training shapes. Shape alignment is done by Cootes et al. [CTC92] using the point correspondences created before. The main idea is to minimise the weighted sum of squares between these corresponding points using procrustesanalysis.

Remaining geometric information can be captured by computing an SSM. The SSM captures shape variation in a mean shape and shape modes. The mean shape defines the average shape of the set of training shapes. Shape modes describe the decomposition of geometrical variation from the mean shape per vertex. Without loss of generality, we consider the generation of the SSM in 3D since our work is based on 3D data.

Using \( M \) aligned shapes, the SSM is computed as follows: Let \( S_i \in \mathbb{R}^{3N} \) be a shape vector containing the \( N \) landmarks of the \( i \)-th shape in the training set:

\[
S_i = (\vec{x}_i1, \ldots, \vec{x}_iN) = (x_{i1}, y_{i1}, z_{i1}, \ldots, x_{iN}, y_{iN}, z_{iN})^T \quad i = 1 \ldots M
\]

where \( \vec{x}_{ij} = (x_{ij}, y_{ij}, z_{ij}) \) is the \( j \)-th point in the \( i \)-th training shape. Using this set of shape vectors, the mean shape \( \overline{S} \) is computed as follows:

\[
\overline{S} = \frac{1}{M} \sum_{i=1}^{M} S_i
\]

Mathematically, shape modes are vectors \( p_i \in \mathbb{R}^{3N}, i = 1 \ldots M \) containing one direction vector for each vertex. Cootes et al. [CHTH94] proposes to apply a principle component analysis (PCA) on the shape vector set to compute \( p_i \). PCA is a popular method in data analysis to reveal the internal structure of a dataset. It is a transformation of the shape vector space by computing a set of orthonormal basis vectors which point to the directions of maximum variance. These computed basis vectors become the shape modes \( p_i \).

The first step is to compute a covariance matrix \( \Sigma \) from the mean shape and the training shapes:

\[
\Sigma = \frac{1}{M} \sum_{i=1}^{M} (S_i - \overline{S})(S_i - \overline{S})^T
\]

The shape modes \( p_i \) are represented by the unit eigenvectors which correspond to the positive non-zero eigenvalues \( \lambda_i \) of \( \Sigma \). The number of shape modes is given by the number of training shapes \( S_i \):

\[
\Sigma p_i = \lambda_i p_i \quad \lambda_i \neq 0, \quad p_i^T p_j = 0 \quad \forall i \neq j, \quad i = 1 \ldots M
\]
It can be shown that the shape mode corresponding to the biggest eigenvalue of $\Sigma$ represents the most significant shape variance. Shapes from shape space are created by adding a linear combination of the shape modes $p_i$ to the mean shape $S$:

$$S = \overline{S} + \sum_{i=1}^{M} b_i p_i$$  \hspace{1cm} (2.5)

where $b_i$ are called shape weights. Shape weights express the influence of each mode in a shape. In the following, we use the notation $S(b, T)$ to describe the shape vector of a SSM which has the set of shape weights of $b$ and is transformed by the transformation $T$:

$$S(b, T) = T(\overline{S} + \sum b_i p_i)$$  \hspace{1cm} (2.6)

### 2.4.3 Segmentation using SSMs

Having generated an SSM, it can be used to segment objects of similar shape in images. Seim et al. [SKH+08] proposes a framework for automatic segmentation using SSMs and evaluate it on pelvic bones in CT images. We use this framework for our segmentation algorithm and extend it according to our task.

The framework consists of 3 steps performed in sequence:

1. Pose Initialisation of the SSM using generalised Hough transform (GHT)
2. Adaption of the SSM (deformations within shape space)
3. Free-form deformation of the SSM (deformations leave shape space)

A detailed description of each step can be read in [SKH+08]. We explain in detail how the authors analyse image data to drive segmentation process in the steps 2 and 3, as this is the relevant part for our work. To sum up, the image analysis task in the steps 2 and 3 is to find out where each landmark or vertex of the SSM has to shift to be placed on the object boundary and on the anatomical region it represents. Having computed the desired shift for each vertex, optimisation techniques can be used to adapt the parameters $(b, T)$ of the SSM $S(b, T)$ to approximate the shifting movement as well as possible.

The desired shift of each vertex is encoded in a displacement field $\Delta R$. A displacement field $\Delta R \in \mathbb{R}^{3N}$ contains 3D vectors $\Delta \vec{r}_k$ for each vertex $\vec{x}_k$ in the SSM. This 3D displacement vector $\Delta \vec{r}_k$ denotes the desired shift of $\vec{x}_k$. To compute $\Delta \vec{r}_k$, local image data around $\vec{x}_k$ is analysed. To simplify computation and regularisation, Seim et al. [SKH+08] limit possible displacements $\Delta \vec{r}_k$ to directions perpendicular to the surface (Kainmueller et al. [KLS+10] propose a method to handle omni-directional displacements).

In the case of displacements perpendicular to the surface, the image data to be analysed is a 1D intensity profile (see fig. 2.5). Let $\vec{x}_k$ be a vertex of the SSM and $\vec{n}_k, \|\vec{n}_k\| = 1$ the surface normal vector at $\vec{x}_k$. An intensity profile $P_k$ on $\vec{x}_k$ is defined as a set of $P$ equidistantly sampled points $\vec{v}_i^k$ in the image data $I : \mathbb{R}^3 \rightarrow \mathbb{R}$ along $\vec{n}_k$ over a length $L$:

$$P_k := \{ \vec{v}_i^k := \vec{x}_k + ((i-1)/(P-1) - 1/2) \cdot L \cdot \vec{n}_k \} \quad \forall i = 1 \ldots P$$  \hspace{1cm} (2.7)

![Figure 2.5: Scheme of an intensity profile (red) and an example for a displacement vector (green).](image)

Based on the intensity profile features at $\vec{v}_i^k$, such as intensity value $I(\vec{v}_i^k)$ or derivative

$$dI(\vec{v}_i^k) = \frac{\partial I(\vec{v}_i^k)}{\partial \vec{n}_k} \approx \frac{I(\vec{v}_{i+1}^k) - I(\vec{v}_{i-1}^k)}{2}$$  \hspace{1cm} (2.8)
(see fig. 2.7), the $\vec{r}_k$ which suits best as object boundary is chosen as target point.

![Image](image_url)

(a) Slice of a head CT image containing the intersection of the maxilla surface (green line) and an intensity value profile (red line).

(b) Image intensity values along the profile in (a).

Figure 2.6: Intensity value profiles

A cost function $c_k(\vec{v}_i^k) : \mathbb{R} \rightarrow \mathbb{R}^+$ is used to encode this target point. On each profile point $\vec{v}_i^k$, $c_k(\vec{v}_i^k)$ assigns a cost value: high costs to unlikely target points and low costs to likely target points. The displacement $\Delta \vec{r}_k$ of $\vec{x}_k$ is then computed as follows:

$$\Delta \vec{r}_k = \vec{v}_k^* - \vec{x}_k$$

$$\vec{v}_k^* = \arg \min \vec{v}_i^k c_k(\vec{v}_i^k)$$

Cost functions can either be automatically generated from profiles on ground truth segmentations or manually designed to meet its special task.

2.4.4 Cost function for bone segmentation in CT images

There have been profile cost functions designed for segmenting bone structures in CT [SKH+08, KLS+09] and CBCT [LZW+06] images. In this section we give an overview of these approaches.
The task is to find the bone-tissue boundary given a profile $P_k$ with $P$ points (denoted by $\vec{v}_i^k, i = 1 \ldots P$). The left half (small point indices $i$) of $P_k$ contains points inside the SSM whereas the right half (big point indices $i$) contains points outside (see fig. 2.8).

Figure 2.8: Visualisation of ‘inside’ and ‘outside’ in a profile. The green line denotes the location of the SSM boundary.

Lamecker et al. [LZW+06] propose a strategy for segmenting the mandible in CBVT images in the shape adaption step (see [SKH+08] for explanation). The strategy does not assign costs to profile points but only computes the displacement $\Delta \vec{r}_k$ given a profile. The authors describe that mandibular bones typically have the highest intensity value near the boundary and lower intensity when going inside and outside (see fig. 2.9 (b)). Therefore, Lamecker et al. propose to detect the two major intensity peaks in smoothed profiles. If they differ significantly in intensity, the one with lower intensity is discarded. Finally, the point of inflection outside of the most outside peak (see fig. 2.9 (b)) is chosen as target point $\vec{v}_k^*$. 

Figure 2.9: Segmentation strategy in [LZW+06].
Seim et al. [SKH+08] designs a cost function for segmentation of the pelvic bone in CT data, which can be used in the shape adaption step as well as in the free-form step. The authors describe how to assign cost values to all profile points. Based on the image features \( I(\vec{v}_i) \) and \( dI(\vec{v}_i) \) (see fig. 2.7 (a) and (b)), Seim et al. assigns the following costs:

\[
c_k(\vec{v}_i) = \begin{cases} 
-\frac{g_{\min}}{dI(\vec{v}_i)} & \text{if } I(\vec{v}_i) \in [t_1, t_2] \land dI(\vec{v}_i) < -g_{\min} \\
1 & \text{else}
\end{cases} \tag{2.11}
\]

The cost function consists of two options. Which option is chosen for computation depends on 2 conditions:

- whether \( I(\vec{v}_i) \) has an intensity value lying in a “bone-typical” interval \([t_1, t_2]\)

- whether \( dI(\vec{v}_i) \) is negative and its magnitude exceeds the threshold \( g_{\min} \) \( \land dI(\vec{v}_i) < -g_{\min} \)

The assumption behind the second condition is that a high magnitude of \( dI(\vec{v}_i) \) indicates a point of material boundary. Since tissue with lower intensity lies outside the bone, \( dI(\vec{v}_i) \) on a boundary point should be negative.

If these 2 conditions are fulfilled, then the cost at this point is indirect proportional to \( dI(\vec{v}_i) \) and lower than 1. Otherwise, if at least one of the two conditions described before fail, maximum cost of 1 is assigned. The constants \( t_1, t_2, g_{\min} \) are chosen empirically.

Kainmueller et al. [KLS+09] propose a profile cost function for head bone segmentation in CT images which extends the idea described Eq. 2.11:

\[
c_k(\vec{v}_i) = \begin{cases} 
(2l + 1) \left( -\frac{g_{\min}}{dI(\vec{v}_i)} + C_{\text{Location}} \right) & \text{if } I(\vec{v}_i) \in [t - lw, t + lw] \land dI(\vec{v}_i) < -g_{\min} \land l = 0, 1, 2 \\
7 \left( -\frac{2g_{\min}}{dI(\vec{v}_i)} + C_{\text{Location}} \right) & \text{if } I(\vec{v}_i) \in [t - 3w, t + 3w] \land dI(\vec{v}_i) < -0.5g_{\min} \\
30 + C_{\text{Location}} & \text{else}
\end{cases} \tag{2.12}
\]

\[
C_{\text{Location}} = \frac{2|j - m|}{P} \tag{2.13}
\]

Firstly, cost structure is more smoothly. If \( I(\vec{v}_i) \) does not lie in the intensity value interval \([t - lw, t + lw]\), \( c(\vec{v}_i) \) is not assigned maximum cost, but is only penalised with higher cost dependent on how far \( I(\vec{v}_i) \) lies outside the interval (see first and second line of Eq. 2.12 first condition). The same concept is applied on \( dI(\vec{v}_i) \) and the threshold \( g_{\min} \) (see first and second line of Eq. 2.12 second condition).
Secondly, the location term $C_{\text{Location}}$ (Eq. 2.13) assigns low costs to points near to certain reference point $m$. With this term it is possible to express preference in displacement. $C_{\text{Location}}$ becomes important e.g. when all $\vec{v}_i$ would be assigned maximum cost (see fig. 2.11 (a)) or there are two local cost minima which have about the same cost (see fig. 2.11 (b)). In these cases, the prior information manually defined would solve ambiguities.

With choosing $m$, we can decide where to assign minimum cost when the desired displacement is ambiguous. The $\vec{v}_k$ with locally minimum cost (if multiple minima) or the $\vec{v}_k$ (if all profile points are assigned maximum cost) near to the most inside ($m = P$), most outside ($m = 0$) or central ($m = P/2 + 1$) profile point will be assigned globally minimum cost. In terms of displacements this means that the preferred movement direction of the vertex $x_k$ is inside ($m = P$), outside ($m = 0$) or no movement ($m = P/2 + 1$).
Chapter 3

Maxilla segmentation

This chapter describes our maxilla segmentation algorithm. First, we describe how we generate the maxilla SSM (see section 3.2). Afterwards, we describe how we extend the profile cost function proposed by Kainmueller et al. [KLS+09] to segment the maxilla, especially in the regions where teeth go through the bone (see section 3.5.1, 3.5.2, 3.5.3).

3.1 Medical background

The maxilla is a group of 2 bones which form the upper jaw. It is located adjacent to the cheek bones, the mouth cavity, the orbits and the sinus. The main function of the maxilla is to hold upper teeth.

In computed tomography images, this structure has high variance in shape and intensity among individuals. Shape varies dependent from the existence of teeth and from the shape of neighbouring sinus whereas intensity variations come from teeth, bone quality and the imaging modality (CT, CBCT). Figure 3.1 shows 2 individual maxilla surfaces.

A segmentation routine for the maxilla including teeth needs to robustly deal with these variations.
Figure 3.1: Shape and intensity variation among maxillae. The images show 2 individual maxillae with CT image intensity values of the corresponding dataset mapped on.

3.2 Generating an SSM of the maxilla

We use triangulated surfaces of manually segmented skulls to build our maxilla SSM (see §2.4.2 for details). The head CT scans are labelled by hand and the label field is transferred to a triangulation of boundaries of the segmented regions.

Using patch decompositions of the training surfaces, point correspondences are established between pairs of training surfaces. We choose a reference shape $S_R$ and compute correspondences $[LSHD04]$ between every other shape $S_i$ in the training set and $S_R$. Since all $S_i$ have the same parametrisation like $S_R$, the same correspondences apply between $S_i$ and $S_j$ with $i \neq j$ (see fig. 3.2).
Figure 3.2: Correspondences between training surfaces. Red arrows express that the training surfaces (1) \( S_1 \) and (2) \( S_2 \) have the same parametrisation like the reference surface (R) \( S_R \). Therefore, there are point correspondences between \( S_1 \) and \( S_2 \) (expressed by the black arrow).

To extract the maxilla from a skull, a cutting plane is defined from 3 landmarks on each skull surface. The landmarks are set on characteristic anatomic structures of the skulls. Using this cutting plane, the skull is divided into a lower part (maxilla bone, bottom side of the nose cavity, cheek bones) and an upper part which contains the cranial part of the skull (see fig. 3.3).

We concentrate on the subdivision of the lower part as this is relevant for our SSM. The maxilla
surface and the lower part of the sinus are divided into patches as preparation for the shape matching step [LSHD04].

The patches can be classified into 3 groups (see fig. 3.4):

1. sinus patches
2. tooth patches
3. bone patches

Figure 3.4: Patch groups in the maxilla SSM

(a) Patched maxilla SSM
(b) Sinus patches
(c) Tooth patches
(d) Bone patches

Sinus patches mark the border between the maxilla bone and the sinus. Tooth patches are located on the bottom side of the maxilla where teeth go through the jaw bone. There are 16 tooth patches, each covering the area of one tooth. If the row of teeth is complete, each patch represents a tooth. If teeth are missing, the patches cover the gap the missing tooth leaves and indicate which tooth is missing. To sum up, tooth patches encode size, location and tooth class (see fig. 3.5).
3.3 Overview maxilla segmentation

As basis for our maxilla segmentation algorithm, we use the segmentation framework provided by Seim et al. [SKH+08]. The algorithm requires a CT/CBCT scan as data input. After segmentation, the result is represented as maxilla surface which can also be converted into a label field. The scheme below (fig. 3.6) shows the steps of the segmentation framework (rounded boxes). Each step contains components (shown by rectangles) executed in series. We explain the components coloured in green in detail as they are our contribution. The other components are described in Seim et al [SKH+08].
3.4 SSM Initialisation

Seim et al. proposed in their framework\cite{SKH08} to use a Generalised Hough Transform (GHT) in the version described by Khoshelham\cite{Kho07} in 3D. GHT is a global method for detecting objects in images using a majority voting scheme to determine the optimal pose $T$ (translation, rotation and uniform scaling).

We have tested different approaches involving GHT to find an initial maxilla SSM pose:

1. GHT using the average maxilla template $T_{\text{Maxilla}}$

2. GHT using the average skull template including the maxilla

3. manually align $T_{\text{Maxilla}}$ to the mandible template $T_{\text{Mandible}}$, segment the mandible\cite{KLS09} and apply the transformation matrix of the segmented mandible to $T_{\text{Maxilla}}$

According to our tests, the first 2 configurations do not succeed. Often, initialisation is so bad that segmentation process is not able to drive the SSM to its correct place. In many cases, the shape of the optimised SSM is degenerated or is not located even near the imaged maxilla. We assume that the
problem lies in the high variance in shape and intensity of the object in contrast to the mandibular bone (see section 3.1).

We also tested GHT using the whole skull template. As the skull also contains other anatomic parts, there is more information to decide where the template should be placed. However, GHT still fails if the skull is not completely imaged. This is problematic in datasets where the skull does not completely lie within the field of view.

Finally, the last approach has proven to be the most robust one. We believe that that GHT works more robust on the mandible \cite{KLS09} than on the maxilla. However, with a fixed alignment of the maxilla to the mandible, we make an assumption that the angle between these two jaws does not vary significantly. We believe that datasets where an extremely open mouth is depicted could cause an initialisation which is not good enough to segment all datasets properly. However, in our test datasets, the angle between maxilla and mandible does not vary such that segmentation fails.

![Maxilla template](a)  ![Skull template](b)  ![Mandible template](c)

Figure 3.7: SSM initialisation templates (yellow) with aligned maxilla SSM (red)

### 3.5 Cost function for segmentation of the maxilla bone

The starting point of our research was applying the cost function proposed by Kainmueller et al. \cite{KLS09} on maxilla segmentation, as it has been proven to work well on the mandibular bone. After some evaluation, we adapt the cost function in 2 points:

First, in section 3.5.1, we extend the cost function by a routine to automatically compute the tissue-bone threshold parameter (variable $t$ in Eq. 2.12) for each image instead of manually setting it. We intend to increase robustness towards intensity variations.

Secondly, our experiments show that the cost function in its original version works poorly on regions where the profile $P_k$ contains tooth material. A high derivative value $dI(v_k)$ in regions where tooth material transitions to air, makes the SSM “wrap around” teeth (see fig. 3.9). The reason why this problem did not occur in the work of Kainmueller et al. was because they used an SSM with an open surface where tooth patches are excluded. In contrast, our maxilla SSM has a closed bone surface considering no tooth at all (see fig. 3.8). Therefore, we add a tooth patch specific segmentation strategy to segment the bone in these regions.
In section 3.5.2 and 3.5.3 we describe how to handle this challenge.

3.5.1 Automatic threshold finding

Our intention is to automatically find the parameter $t$ for the bone cost function (see Eq. 2.12). Therefore, we propose an automatic threshold finding (ATF) algorithm to compute $t$ from intensity histograms. As input, we need the image and the initialised SSM.
We analysed histograms of the region inside the bounding box (see fig. 3.10) from different datasets and spotted following common characteristics:

Firstly, in CT image histograms, there are intensity peaks at around $-1000\,\text{HU}$ (typical intensity for air) and immediately after $0\,\text{HU}$ (typical intensity for water) in all CT histograms. We believe that the peak at around $-1000\,\text{HU}$ corresponds to air whereas the peak right after $0\,\text{HU}$ corresponds to soft tissue. In fig. 3.11, these peaks are marked with (1) and (2). In CBCT images, intensity values $<-1000\,\text{HU}$ are clipped to $-1000\,\text{HU}$. We have spotted typical peaks at around $-115\,\text{HU}$ (1) and right after $0\,\text{HU}$ (2).

Secondly, after around $2000\,\text{HU}$, the CT histogram shape varies most. We consider intensity values $>2000\,\text{HU}$ as indicator for teeth. The appearance of this distribution depends on attenuation and amount of tooth material in the dataset. This characteristic is marked in fig. 3.11 with (3) as the region on the right hand side of the red vertical line. In CBCT images, we have not found a regularity which indicates the presence of teeth the region of high intensity values ($>800\,\text{HU}$).

To sum up, the features (1) and (2) are present in all the histograms we have examined. However, their location varies depending on individual image and imaging modality. In CT images, the distribution of intensity values $>2000\,\text{HU}$ even indicates the presence of teeth. For CBCT images, we have not found such a possible indicator.
Our approach is to approximate the distribution given by the histogram $H$ with a function $f(x)$ and use its parameters to heuristically determine material intensity thresholds.

$$f(x) = w_{\text{Air}} \cdot \exp \left( \frac{(x - \mu_{\text{Air}})^2}{\sigma_{\text{Air}}^2} \right) + w_{\text{Tissue}} \cdot \exp \left( \frac{(x - \mu_{\text{Tissue}})^2}{\sigma_{\text{Tissue}}^2} \right) + C \quad (3.1)$$

The function $f(x)$ is a sum of 2 gaussian functions ($G_1(x)$ and $G_2(x)$) and a constant $C$. The gaussian functions model the two material peaks and $C$ approximates the distribution of the remaining voxel intensities (see fig. 3.12). Our task is to find the parameter set $\Theta^* = (w_{\text{Air}}, \mu_{\text{Air}}, \sigma_{\text{Air}}, w_{\text{Tissue}}, \mu_{\text{Tissue}}, \sigma_{\text{Tissue}}, C^*)$ which best fits $f(x)$ to the histogram $H = (H_1, \ldots, H_B)$. $H$ is the histogram of voxel intensities in the region of the initialised SSM. $B$ denotes the number of bins of $H$. The optimisation task can be formulated as follows:

$$\Theta^* = \arg \min_{\Theta} \sum_{i=1}^{B} (f(val(i), \Theta) - H_i)^2 \quad (3.2)$$

where $val(i)$ is the voxel count in bin $i$.

We apply the Levenberg-Marquardt Algorithm (LMA) to compute $\Theta^*$. LMA is a popular non-linear least squares optimisation method. Since LMA is based on gradient descend, it can lead to local minimum solutions. Therefore, we need a good initialisation for $\Theta$.

To find good initial parameters for $f(x)$, we discretise the fitting problem by sampling parameter space and extensively searching for the combination of parameters with minimum least squares error

$$\sum_{i=1}^{B} (f(val(i), \Theta) - H_i)^2$$

This combination of parameters serves as initialisation for continuous LMA optimisation.
Using the parameters of our fitted function, we determine the material thresholds $t_{\text{material}}$ in the following way:

- $t_{\text{Tissue}} = -500HU$
- $t_{\text{Bone}} = \mu_{\text{Tissue}} + 2 \times \sigma_{\text{Tissue}}$
- $t_{\text{Tooth}} = \text{val}(i^*)$ where $\text{val}(i^*) > 800HU \land i^* = \text{arg min}_i (H_{i+1} - H_i)$

### 3.5.2 Profile interval classification

After computing the material thresholds $t_{\text{Tissue}}$, $t_{\text{Bone}}$, and $t_{\text{Tooth}}$, we can classify each vertex $\vec{v}_k$ of a gray value profile $P_k$ into one of the material classes $\text{Material}(\vec{v}_k) \in \{\text{air, tissue, bone, tooth}\}$ depending
on their intensity value:

\[
Material(v_k^i) = \begin{cases} 
\text{air} & \text{if } I(v_k^i) \leq t_{Tissue} \\
\text{tissue} & \text{if } t_{Tissue} < I(v_k^i) < t_{Bone} \\
\text{bone} & \text{if } t_{Bone} \leq I(v_k^i) < t_{Tooth} \\
\text{tooth} & \text{else}
\end{cases}
\]

However, there are regions on \( P_k \) where classification only based on intensity value would lead to wrong results. An example are transition regions from tooth to air (see fig. 3.13). In this case, some vertices would be classified as bone due to intensity interpolation. However, there is no bone. Consequently, the maxilla SSM surface would be driven to “wrap around” the tooth, like in fig. 3.9. The same situation happens when there are voxels with intensity value outside our predicted material intensity interval. These side effects result in small regions in the profile which are wrongly classified.

Our solution to tackle this problem is to enforce a minimum size for regions. Regions which do not satisfy this condition are split and the halves are assigned to the neighbouring regions if they exceed the size threshold. The only exception is when the small region is a tooth region, since we consider the intensity value of tooth regions as a reliable feature.

![Intensity profile in image](image1.png)

(a) Intensity profile in image

![Material regions without minimum size](image2.png)
(b) Material regions without minimum size

![Material regions with minimum size](image3.png)
(c) Material regions with minimum size

Figure 3.13: Minimum size of 1cm for material regions on an intensity profile
3.5.3 Rule-based cost strategy application

Using $Material(\vec{v}_i^k), i = 1 \ldots P$ determined in section 3.5.2 and the patch the vertex $\vec{x}_k$ belongs to, we apply an individual displacement computation strategy for each intensity value profile. After analysing intensity value profiles, we identified a limited number of standard material compositions on profiles. Thus, we propose a rule based approach using a set of manually designed rules to determine the strategy.

Our rules decide for each profile where to apply the bone cost function (Eq. 2.12) with $t = t_{Bone}$ and where to assign zero displacement to a profile. When applying zero-displacement, the displacement vector $\Delta \vec{r}_k$ has length 0 and the profile weight is set to 0. As result, the displacement of the vertex belonging to the profile is adjusted similar to neighbouring non-zero displacements. All in all, the purpose of the set of rules is to determine profiles where displacement computation using the CT bone cost function (Eq. 2.12) is reliable and compute the transformation and deformation of the SSM without considering the other profiles. Moreover, tooth regions on the profile are always assigned with maximum cost in order to completely avoid vertices moving into teeth.

Figure 3.15 shows the overall decision process.
Figure 3.15: Scheme of decision process
Chapter 4

Tooth detection

In this chapter, we describe our tooth detection algorithm. Given a CT dataset of a head and the adapted SSM, our algorithm isolates and numbers the teeth which are imaged. We use the FDI numbering scheme (see fig. 2.1).

4.1 Overview tooth detection

Our tooth detection algorithm consists of three steps: First, we create a tooth arc image (see section 4.2), a representation of the image data in the ROI containing the row of teeth. Afterwards, we divide the tooth arc image into sub regions (tooth ROI) which contain maximum one tooth (see section 4.3). Finally, we classify whether there is a tooth present in the tooth ROI or not (see section 4.4). The scheme below (fig. 4.1) shows the process.
4.2 Tooth arc image

We transform the image region containing the row of teeth into a representation we call *tooth arc image*. A tooth arc image can be seen as the curved 3D region the row of teeth “bent straight” (see fig. 4.2).

The advantage of this representation is that algorithms working on the tooth arc image deal with a conventional 3D image instead of a curved region. Tooth neighbourhood is implicitly encoded such that moving to the neighbouring tooth is equivalent to going along the x axis in tooth arc space. However, the disadvantage is that we have to accept distortion effects when transforming results from tooth arc image space back into image space.
We build the tooth arc image by computing a grid in the region containing the row of teeth (see fig. 4.3 (b)). To obtain the tooth arc image, we sample image intensity values at location of the grid points. Our grid is defined by two B-spline curves, $B_1(r)$ and $B_2(s)$, and one normal vector $\vec{n}$. $B_1(r)$ and $B_2(s)$ approximate the front and back arc of the jaw bone and interpolate corner points of the tooth patches (see fig. 4.3 (a)). As we know the indices of the vertices of the maxilla SSM which lie on the patch corners, we can use their locations to build the B-spline curves.

To compute the normal vector $\vec{n}$, we fit a plane $P_{fit}$ into the set of vertices belonging to a tooth patch. The normal of this fitted plane $P_{fit}$ is our normal vector $\vec{n}$. A 3D point $X_{Arc} = (x, y, z)$, $x, y, z \in [0, 1]$ in the tooth arc image domain is transformed into a location $X(x, y, z)$ in image space by

$$X(x, y, z) = yB_1(x) + (1 - y)B_2(x) + (z - 0.5)\vec{n} \quad (4.1)$$
4.3 Dijkstra-based tooth separation

4.3.1 Problem statement

As preprocessing for the actual tooth detection routine, our task is to divide the tooth arc image into 16 ROIs - each containing one tooth at maximum. If a tooth is missing, the ROI contains the gap the missing tooth leaves. Each of the 16 ROIs has a unique identifier (ID), such that we can distinguish it from the other ones. ROIs with the same ID computed on different images represent the same kind of tooth. We compute the ROIs on the tooth arc image and transform them to image space via formula Eq. 4.1. The tooth detection routine can then work independently on each ROI. If it has detected a tooth or a gap, we can identify the number using the ID of the tooth ROI. This identification only works if we assume teeth to stay on original location or are only slightly shifted.

Instead of computing the ROIs from the arc image, we compute the boundaries between 2 neighbouring ROIs. These boundaries are sufficient to reconstruct the ROIs themselves. We assume that the teeth imaged in the arc image are separated by planes (see fig. 4.4).

The task of finding the tooth ROIs is now reduced to the task of finding optimal position and pose of 15 planes ($P_1, \ldots, P_{i}, \ldots, P_{15}$) between the ROIs. We use a point $\vec{x}_P$ and 2 angles $\phi_1$ and $\phi_2$ to describe position and pose of a separation plane $P_i$.

The point $\vec{x}_P$ of the plane $P_i$ and has the tooth arc image coordinates $\vec{x}_P = (x_i, 0, 0)$ (see fig. 4.5 (a)). We refer to $\vec{x}_P$ by a single value $x_i$, since the other coordinates of $\vec{x}_P$ are always set to 0.
Furthermore, we integrate a-priori knowledge of the plane positions. Looking at several manually
segmented maxillae, we found out that the possible locations of a plane $P_i$ is limited to a small environ-
ment (see fig. 4.6 (a)). For all $M$ patched maxilla surfaces $S_j, j = 1 \ldots M$ with B-splines $B_{i1}(r_j^i)$ and
$B_{i2}(s_j^i)$, we compute the values of the parameter $r_j^i, s_j^i \in [0,1]$ at the patch corner vertices of the SSM.
Using these parameters $r_j^i$ and $s_j^i$, we compute a mean plane position $\pi_i$ for each plane (see fig. 4.6 (b)):
\[
\pi_i = \frac{1}{M} \sum_{j=1}^{M} \left( \frac{r_j^i + s_j^i}{2} \right)
\] (4.2)

In the following, we describe $x_i$ by $\pi_i$ and an offset $x_{off}$. The corresponding point $x_i$ on plane $P_i$ is
computed:
\[
x_i = \pi_i + x_{off}
\] (4.3)

(a) Position parameter of a plane

(b) Pose angles of a plane

Figure 4.5: Parametrisation of a separation plane in tooth arc image space

To describe the pose of a plane, we use 2 angles $\phi_1$ and $\phi_2$ (see fig. 4.5 (b)). If $\phi_1 = 0^\circ$ and $\phi_2 = 0^\circ$
apply then we define the pose as neutral (see fig. 4.7).

Moreover, we introduce a plane cost function to evaluate separation quality of each possible plane
pose and position. The cost function yields high costs if a plane badly separates teeth e.g. goes through
the middle of teeth and low costs if it separates them well. Our task is to find a combination of planes
which yield minimum sum of cost.
(a) Examples of maxilla SSMs. The red arrow shows the position of the plane which separates tooth 24 and tooth 25.

(b) $\mathbf{\tau}_i$ positions

Figure 4.6: Computing $\mathbf{\tau}_i$

If we do not include optimisation constraints, the solver would search for a position and pose which yield globally minimal cost. The solution would be all planes with the same pose at this position. To avoid such a configuration, we add hard constraints between neighbouring planes. The distance between neighbouring planes $d(P_i, P_{i+1})$ is only allowed to be in a fixed interval $[d_{\text{min}}(i), d_{\text{max}}(i)]$. We compute
Figure 4.7: A plane in neutral pose

\[ d(P_i, P_{i+1}) \] as the distance of the plane centres:

\[
d(P_i, P_{i+1}) = X_C(P_{i+1}) - X_C(P_i)
\]

\[ X_C(P) = x_{off}(P) + \frac{1}{2}(\tan \phi_1(P) + \tan \phi_2(P)) \]

(a) Distance between two planes is too small  
(b) Distance between two planes is too big  
(c) There should be about one tooth distance between two planes

Figure 4.9: Plane distance constraints

By discretising the number of possible plane positions and poses, we can formulate a discrete optimisation problem which can be solved globally.

### 4.3.2 Graph construction

Our method solves the problem described in section 4.3.1 by constructing a directed graph and transforming the task into a shortest-path problem. The shortest path problem is the problem of finding a path between two nodes in a directed or undirected graph such that the sum of weights of the edges lying on this path is minimal. It can be solved using Dijkstra’s Algorithm\[^{Dij59}\].

Our weighted graph \( G = (V, E) \) is constructed as follows: Every node \( V(i, x_{off}, \phi_1, \phi_2) \in V \) represents one and only one separation plane with the ID \( i = 1 \ldots 15 \) and the pose parameters

- \( x_{off} \in X_{off} = (0, \ldots, X_{off}^{max}) \)
- \( \phi_1 \in \Phi_1 = (0^\circ, \ldots, \phi_1^{max}) \) and \( \phi_2 \in \Phi_2 = (0^\circ, \ldots, \phi_2^{max}) \)
The possible positions are computed from equidistant sampling of the interval of possible $X_{off}$, $\Phi_1$ and $\Phi_2$. These intervals are manually set and represent our solution space.

The cost of the plane $C_{\text{Plane}}(v)$ corresponding to node $v = V(i, x_{off}, \phi_1, \phi_2)$ is encoded in the weight of the edge pointing to node $v$:

$$C_{\text{Plane}}(v) = w(e) \quad e = (\cdot, v)$$

(4.4)

![Figure 4.10: Representation of plane costs in a graph](image)

The set of edges represent the set of legal separation plane combinations. Firstly, there are only edges between neighbouring separation planes (between plane with ID $i$ and $i + 1$):

$$\forall e = (v_a, v_b) : \quad v_a = V(i, \cdot, \cdot, \cdot) \quad \land \quad v_b = V(i + 1, \cdot, \cdot, \cdot) \quad i = 1 \ldots 14$$

(4.5)

Secondly, there is only an edge between nodes if the distance between the planes they represent lie in a defined interval $[d_{\text{min}}(i), d_{\text{max}}(i)]$ (see fig. 4.11):

$$\forall e = (v_a, v_b) : \quad v_a = V(i, \cdot, \cdot, \cdot) \quad \land \quad v_b = V(i + 1, \cdot, \cdot, \cdot) \quad \land \quad d(v_a, v_b) \in [d_{\text{min}}(i), d_{\text{max}}(i)]$$

(4.6)
Furthermore, we add the nodes $v_{\text{start}}, v_{\text{end}} \in V$ which represent a start node and an end node (see fig. 4.12). These nodes are connected as follows:

\[
\forall e = (v_{\text{start}}, v) : \quad v = V(1, \cdot, \cdot) \quad \wedge \quad d(v_{\text{start}}, v) \in [d_{\text{min}}(1), d_{\text{max}}(1)] \quad (4.7)
\]

\[
\forall e = (v, v_{\text{end}}) : \quad v = V(15, \cdot, \cdot) \quad \wedge \quad d(v, v_{\text{end}}) \in [d_{\text{min}}(15), d_{\text{max}}(15)] \quad (4.8)
\]

There are no edges pointing to $v_{\text{start}}$ and no edges starting from $v_{\text{end}}$. All the edges $e = (\cdot, v_{\text{end}})$ have the same cost.

The conditions Eq. 4.6, Eq. 4.7 and Eq. 4.8 implement the distance constraints for neighbouring planes. The distance interval depends on the ID of the planes. It is computed from the tooth patch sizes of maxilla surfaces from manual segmentations.
For each maxilla surface $S_j, j = 1 \ldots M$, we compute the distance
\[ d_i^j = \frac{r_{i+1}^j + s_{i+1}^j - r_i^j + s_i^j}{2} \] (4.9)
between neighbouring planes, where $r_i^j, s_i^j \in [0,1]$ are the values of the B-spline arguments $B_j^1(r_i^j)$ and $B_j^2(s_i^j)$ at the patch corner points. We then calculate the mean distance $d_\mu(i) \in [0,1]$ and the standard deviation $d_\sigma(i) \in [0,1]$ by
\[ d_\mu(i) = \frac{1}{M} \sum_{j=1}^{M} d_i^j \] (4.10)
\[ d_\sigma(i) = \frac{1}{M} \sqrt{\sum_{j=1}^{M} (d_i^j - d_\mu)^2} \] (4.11)
To determine the distance interval $[d_{\text{min}}(i), d_{\text{max}}(i)]$ between neighbouring planes, we use $d_\mu(i)$ and $d_\sigma(i)$:
\[ d_{\text{min}}(i) = d_\mu(i) - 2d_\sigma(i) \] (4.12)
\[ d_{\text{max}}(i) = d_\mu(i) + 2d_\sigma(i) \] (4.13)

![Diagram](image)

(a) Position parameter strip

(b) Parameter strip bend straight with visualisation of $d_\mu(i)$ (1) and $d_\sigma(i)$ (2)

Figure 4.13: Computation of the plane distance interval

A path from $v_{\text{start}}$ to $v_{\text{end}}$ in $G$ represents a legal combination of planes. The shortest path in this graph has minimum sum of weights $\sum_e w(e)$ and thus minimum sum of plane costs $\sum_v C_{\text{Plane}}(v)$.

### 4.3.3 Plane cost function

The cost of a plane $P_i$ is determined by the image data in its local environment and the parameters of $P_i$. Let $I$ be the arc image data and $(x_{\text{off}}, \phi_1, \phi_2)$ be the parameters of a plane. We propose the plane cost function:
\[ C_{\text{Plane}}(I, x_{\text{off}}, \phi_1, \phi_2) = \alpha C_{\text{AVG}} + \beta C_{\text{STD}} - \gamma C_{\text{Gradient}} + \delta C_{\text{Pose}} \] (4.14)
The first term $C_{AVG}(I, x_{off}, \phi_1, \phi_2)$ represents the mean image intensity value of the sample points lying on the plane. To compute $C_{AVG}$, we equidistantly sample $N_P$ points $\vec{x}_n, n = 1 \ldots N_P$ on $P_i$, determine their intensity values $I(\vec{x}_n)$ and compute the mean intensity.

$$C_{AVG} = \frac{1}{N_P} \sum_{n=1}^{N_P} I(\vec{x}_n)$$  \hspace{1cm} (4.15)

Since teeth have high intensity values in CT images, costs rise if the plane is placed in the middle of a tooth. In contrast, planes placed in a tooth gap or between 2 teeth would result in low mean intensity and yield low cost.

The second term $C_{STD}(I, x_{off}, \phi_1, \phi_2)$ represents the standard deviation of the intensity values at all $\vec{x}_n, n = 1 \ldots N_P$.

$$C_{STD} = \frac{1}{N_P} \sqrt{\sum_{n=1}^{N_P} (I(\vec{x}_n) - C_{AVG})^2}$$  \hspace{1cm} (4.16)

We consider intensity values on a plane between teeth or in gaps to be homogeneous whereas a plane placed in the middle of a tooth contains very high intensity values in the tooth region and low intensity values around. This variance of intensities on the plane leads to a high standard deviation.

![Figure 4.14: Intensity values on the separation plane](image)

The third term, $C_{Gradient}(I, x_{off}, \phi_1, \phi_2)$, examines how intensity values on the plane change when we slightly translate the plane along the direction of its normal $\vec{n}_P$. Let $\vec{x}_n$ be a point on the plane and $\vec{x}_n^*$ be the same point slightly translated in direction of $\vec{n}_P$. We are then interested in the difference of intensity values $|I(\vec{x}_n) - I(\vec{x}_n^*)|$. To achieve a scalar cost value, we compute the mean of such differences.

$$C_{Gradient} = \frac{1}{N_P} \sum_{n=1}^{N_P} |I(\vec{x}_n) - I(\vec{x}_n^*)|$$  \hspace{1cm} (4.17)

We assume low image gradient inside teeth. The $C_{Gradient}$ of a plane placed in the middle of a tooth should result in a low difference, since both planes lie in the tooth. A plane placed on the edge of a tooth, however, should result in low cost, since the shifted plane lies in the tooth.
Finally, the pose cost $C_{\text{Pose}}(\phi_1, \phi_2)$ is computed independently from image data. Let $\phi_1 = 0$ and $\phi_2 = 0$ denote the neutral pose of a plane and $\phi_1^{\text{max}}$ and $\phi_2^{\text{max}}$ denote the maximum possible values for the angles. $C_{\text{Pose}}$ then penalises poses different from neutral pose:

$$C_{\text{Pose}} = \left( \frac{\phi_1}{\phi_1^{\text{max}}} \right)^2 + \left( \frac{\phi_2}{\phi_2^{\text{max}}} \right)^2$$

(4.18)

This term is supposed to be important if planes are set in tooth gaps. Image intensity values and gradient magnitude are low in this area, such that there is little information. The pose cost penalises strongly tilted planes. A strongly tilted plane would cause the other planes to tilt, as well. This effect comes up because the optimiser insists on preserving the distance constraints. The pose cost term is supposed to avoid plane being too far from neutral pose.

### 4.3.4 Central plane constraint

When testing our tooth separation algorithm, we found a weakness: in some datasets, a single plane is set “wrongly”, e.g. split a tooth. In order to preserve distance constraints, the other planes also shift and either split teeth or move between teeth they do not correspond to. The local error caused by a single plane resulted in a global error completely destroying correspondences between teeth and planes. We conclude that there were are “wrong” solutions in search space, which has minimum cost.

To tackle this problem, we limit search space by fixing the location of the central plane $P_8$ (plane which separates tooth 11 from tooth 21).

$$x_8 = \mathbf{x}_8$$

(4.19)

We assume that it is unlikely that a tooth is placed exactly in the centre of the row of teeth and would be split by this plane. Having this constraint, the algorithm only optimises the pose parameters of the central plane.
An other advantage of this constraint is that errors stay local. The central plane divides the row of teeth into 2 halves. If an error occurs in one half, the other planes can only be shift to the wrong position up to the central plane.

4.4 Tooth classification

4.4.1 Problem statement

Our tooth detection routine finds out which teeth in a tomography image are present. We have simplified this task by isolating them. Afterwards, we can independently decide for each ROI whether there is a tooth present or not. This is a classification problem where we can apply machine learning techniques. Having decided which ROIs contain teeth, we can use the ROI IDs to identify the tooth.
4.4.2 Training data generation

To train the classifier, we need histograms from ROIs which isolate teeth well. Since we cannot guarantee that our separation algorithm produces ROIs of such quality, we use manual tooth segmentations of the datasets to create such ROIs. The tooth segmentations are represented as label fields where each tooth is represented by a different label. In the first step, we build separation planes from segmented teeth. Afterwards, we used these planes to build the training ROIs.

To compute separation planes, we use an SVM. An SVM is a supervised learning algorithm to construct a plane that separates data points with different labels. In our case, the points are the voxels of two neighbouring teeth which are labelled with $i$ and $i+1$. Let $\vec{x}_i \in \mathbb{R}^3$ be the voxel locations in the tooth label field $L$ that are labelled by the label $y_i = L(\vec{x}_i) \in 1...16$. We train an SVM for each separation plane $P_i^{SVM}$ to separate 2 neighbouring teeth. The set of training points are:

$$T = \{(\vec{x}_1, y_1), \ldots, (\vec{x}_N, y_N)\} \quad \forall j = 1 \ldots N : \quad y_j = i \lor y_j = i + 1$$

The hyperplane computed by the SVM is our separation plane $P_i^{SVM}$ which separates the 2 teeth.

If we need to separate a tooth from a gap or divide a gap, we cannot create all data points from tooth voxels. In case a label is missing because no tooth is present, we use the patch information of the segmented maxilla surface.

If there is only one tooth with label $i$ to be separated from a gap, we create the data points from maxilla surface vertices. All vertices belong to the tooth patch containing the gap.

If the plane should divide a big gap into two parts, we extract 3 landmark points to create a plane. Two of them are vertex corner points (see Figure 4.2) $\vec{x}_1$ and $\vec{x}_2$ between the two patches corresponding to the gaps. To obtain the third point, we compute the middle point between $x_1$ and $x_2$ and add the normal vector $\vec{n}_{fit}$:

$$\vec{x}_3 = \vec{x}_1 + 0.5(\vec{x}_2 - \vec{x}_1) + \vec{n}_{fit}$$

$\vec{n}_{fit}$ is the normal vector of a plane $P_{fit}$ fitted into the set of all tooth patch vertices located on the maxilla surface (same as section 4.2).

Note that there are multiple planes which correctly separate 2 teeth. Our planes $P_i^{SVM}$ only represent one possible way. We use the planes to create training ROIs. Although not every training ROI perfectly isolates its tooth, our tests in section 5.3.3 show that their quality is good enough to train a strong classifier. To build training ROIs, we apply the tooth separation algorithm with the parameters which separate teeth well. After converting the ROI into a label field, we correct labels of voxels using our classifier. To build training ROIs, we apply the tooth separation algorithm with the parameters which separate teeth well. After converting the ROI into a label field, we correct labels of voxels using our classifier.

Let $\vec{x} \in X$ denote the location vector of a voxel in the domain of the label field. The neighbouring planes $P_i^{SVM}$ and $P_{i+1}^{SVM}$ are defined by one point on the plane $\vec{v}_i$ and $\vec{v}_{i+1}$ and their normals $\vec{n}_i$ and $\vec{n}_{i+1}$. The angle between $\vec{n}_i$ and $\vec{n}_{i+1}$ is smaller than 180°. If this does not apply we multiply one vector by $-1$ such that we get the normal pointing in the opposite direction. We check for every voxel with label $i$ whether it lies between $P_i$ and $P_{i+1}$ by comparing $d_1 = (\vec{x} - \vec{v}_i) \cdot \vec{n}_i$ and $d_2 = (\vec{x} - \vec{v}_{i+1}) \cdot \vec{n}_{i+1}$. If $d_1$ and $d_2$ have different signs, the voxel is correctly labelled. Otherwise we search the planes $P_j$ and $P_{j+1}$ where $\vec{s}$ lies in between.

**Algorithm 1** Compute training ROIs from tooth ROIs with good separation quality

1: **Input:** Tooth ROI as label field $L_{ROI}$, $P = (P_1^{SVM}, \ldots, P_{15}^{SVM})$
2: **for** $i \leftarrow 1$ **to** 15 **do**
3:     **for all** $\vec{x} \in X$ **do**
4:         **if** $L_{ROI}(\vec{x}) = i$ **then**
5:             $d_1 = (\vec{x} - \vec{v}_i) \cdot \vec{n}_i$
6:             $d_2 = (\vec{x} - \vec{v}_{i+1}) \cdot \vec{n}_{i+1}$
7:         **if** $\text{sign}(d_1) \neq \text{sign}(d_2)$ **then**
8:             **search** $j$ **such that:** $\text{sign}((\vec{x} - \vec{v}_j) \cdot \vec{n}_j) = \text{sign}((\vec{x} - \vec{v}_{j+1}) \cdot \vec{n}_{j+1})$
9:             $L_{ROI}(\vec{x}) \leftarrow j$
10:       **end if**
11:     **end if**
12: **end for**
13: **end for**
4.4.3 Features and Classifiers

We use intensity value histograms as features. Given a tooth ROI created by the separation algorithm and the image data, we extract 3 types of intensity histograms:

1. ROI histogram
2. root histogram
3. surface histogram

The ROI histogram considers the intensity values of all voxels in the ROI (fig. 4.18 (a) and (d)). Since tooth crowns can cause artefacts in CT images, we experimented with histograms, which omit the crown region. The root histogram (fig. 4.18 (b) and (e)) only considers intensity values of voxels located in the ROI and inside the segmented maxilla bone. The surface histogram (fig. 4.18 (c) and (f)) considers intensity values on SSM vertices belonging to the tooth patches and located in the ROI. Figure 4.19 shows examples of these histograms. We tried the AdaBoost.M1 classifier by Freund and Shapire [FS96] using decision trees as weak classifier and the support vector machine (SVM) [CV95] with an RBF-Kernel. We use histograms computed from training ROIs (see section 4.4.2) to train the classifiers.

![ROI types and image data lying inside](image)

Figure 4.18: ROI types and image data lying inside

![Feature space of histograms](image)

Figure 4.20: Feature space of histograms: PCA was applied on the set of feature vectors and the score of the principal components corresponding to the 2 highest values of the correlation matrix were plotted. Green points represent a feature vector of the “tooth” class whereas red points represent a feature vector of the “no tooth” class.
Figure 4.19: Histograms of the ROI types. The bins are normalised, such that the sum of their value is 1.
Chapter 5

Experimental results

5.1 Test data

Our test data consists of 43 head CT and 10 CBCT scans. All tested scans image the maxilla and the mandibular bone including teeth. Sometimes the whole skull is in the field of view, but sometimes only parts of the skull are visible. Each scan comes from a different patient. The number of upper teeth the patient possesses ranges from 0 to 16 (full row of teeth). Moreover, there are manually segmented maxilla surfaces for all CT datasets. In 39 cases, we also have tooth label fields. Unfortunately, we do not have ground truth data for the CBCT datasets, such that quantitative evaluation is not possible, yet.

5.2 Maxilla segmentation

5.2.1 Experimental setup

We experimentally investigated the performance of our maxilla segmentation algorithm (see chapter 3) and its components using 43 CT and 10 CBCT datasets. We applied our algorithm on the CT data in a leave one out (LOO) framework. This means we generated an SSM from 42 out of 43 maxilla surfaces and used the SSM to segment the remaining dataset. Having automatically segmented maxilla surface and ground-truth surface, we investigated the algorithm under different aspects:

In section 5.2.2, we measure how well the initialisation procedure works. In section 5.2.3, we analyse the impact of our ATF routine (see 3.5.1) and compare it to our maxilla cost function using fixed threshold values. Finally, we present segmentation results of the overall algorithm and evaluate the impact of our tooth patch segmentation strategy.

We measure segmentation quality with mean ($d_{AVG}$), root mean square ($d_{RMS}$) and maximum ($d_{max}$) surface distance between the automatically segmented maxilla surface and ground-truth maxilla surface. Let $v_i \in \mathbb{R}^3$ be a vertex of the automatically segmented surface containing $N$ vertices and $v'_j$ be a vertex of the ground-truth surface $R^*$ having $M$ vertices. The distance measures are then computed as follows:

\[
d_{AVG} = \frac{1}{N} \sum_{i=1}^{N} d(v_i, R^*)
\]

\[
d_{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} d(v_i, R^*)^2}
\]

\[
d_{max} = \arg \max_i d(v_i, R^*)
\]

\[
d(v_i, R^*) = \arg \min_j |v_i - v'_j|
\]

To visualise some of our results, we use boxplots. Boxplots are a convenient presentation of groups of numerical data which contain the following characteristic values: minimum, maximum, lower quartile, upper quartile, median, outliers (see fig. 5.1) and thus reveal more information about performance than single mean values or error bars (mean and standard deviation).
Initialisation of the maxilla SSM

To determine initialisation quality, we compared the bounding boxes of the ground-truth maxilla surfaces with the bounding boxes of our initialised SSM. Quantitatively, we use the Dice coefficient and the distance between the centroids of the surface bounding boxes to measure initialisation quality. The dice coefficient measures volume overlap. Let $V_a$ and $V_b$ be the volume of the two bounding boxes to compare and $V_{a\cap b}$ the volume of the common area of the bounding boxes. The Dice coefficient is then defined as

$$c_{Dice} = \frac{2V_{a\cap b}}{V_a + V_b}$$  \hspace{1cm} (5.5)

The advantage using Dice coefficient is that it does not consider shape, which can be very different between SSM and ground-truth maxilla surface. This bias would be present if we had used surface distance measures e.g. $d_{AVG}$.
5.2.3 Impact of ATF

We compared the segmentation results of our segmentation strategy including ATF to our cost function using constant threshold values for all datasets. We used the following parameters for the algorithm in section 3.5.2:

<table>
<thead>
<tr>
<th>Thresholds</th>
<th>Configuration: no ATF 1</th>
<th>Configuration: no ATF 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{\text{Tissue}}$</td>
<td>$-1000 \text{HU}$</td>
<td>$-1000 \text{HU}$</td>
</tr>
<tr>
<td>$t_{\text{Bone}}$</td>
<td>120 $\text{HU}$</td>
<td>300 $\text{HU}$</td>
</tr>
<tr>
<td>$t_{\text{Tooth}}$</td>
<td>800 $\text{HU}$</td>
<td>800 $\text{HU}$</td>
</tr>
</tbody>
</table>
5.2.4 Overall segmentation results

We evaluated segmentation performance of the overall algorithm. We compared our cost function including ATF and rule-based cost strategy to the bone CT cost function proposed by Kainmueller et al. [KLS+09] applied on all patches of the maxilla SSM.

The following boxplots show the results for our 43 datasets in all segmentation stages.
Furthermore, we computed the average surface distances $d_{AVG}$ per patch. We grouped the patches into tooth patches and non-tooth patches to investigate the impact of our tooth patch strategy. The following boxplots show $d_{AVG}$ per patch for tooth patches and non-tooth patches after the shape adaption step and the free-form step.
5.3 Tooth detection

5.3.1 Experimental setup

We tested our automatic tooth detection algorithm in a 10-fold cross validation framework. Cross validation is a technique to evaluate machine learning algorithms. In our case, we divided the test set $T$ into 10 parts $T = (F_1, \ldots, F_{10})$, called folds. In each round $i = 1..10$ we used 9 folds $T_{\text{train}} = (F_1, \ldots, F_{i-1}, F_{i+1}, F_{10})$ to train the system and tested the classifier on the remaining fold $T_{\text{test}} = F_i$. In order to tune the parameters of our tooth classifier, we applied a second cross validation loop only on the training data $T_{\text{train}}$. This evaluation framework is described in Hegenbart et al. [HU11].

Since we had manual tooth segmentations for 39 of the 43 datasets, we could quantitatively evaluate separation quality for the majority of test data. The manual segmentations are given as label field where each tooth is labelled differently. If teeth of two fields have the same label, they have the same tooth number. As evaluation measure, we computed the volume overlap $V_{\text{overlap}}$ of the tooth regions. Let $L_{\text{ROI}}(x,y,z) : \mathbb{R}^3 \rightarrow 0 \ldots 16$ be the teeth ROIs encoded in a label field and $L_{\text{Tooth}}(x,y,z) : \mathbb{R}^3 \rightarrow 0 \ldots 16$ the field containing the ground-truth teeth. The labelling system of both fields is equivalent which means if a tooth $T$ is represented by $l$ in $L_{\text{Tooth}}$, then the region which is supposed to contain $T$ should be labelled $l$ in $L_{\text{ROI}}$. Assuming this condition is fulfilled, we compute the volume overlap $V_{\text{overlap}}$ by

$$V_{\text{overlap}} = \frac{N_{\text{ROI} \cap \text{Tooth}}}{N_{\text{Total}}}$$

where

- $N_{\text{ROI} \cap \text{Tooth}}$ denotes the number of voxels $(x,y,z)$ with: $L_{\text{ROI}}(x,y,z) > 0 \land L_{\text{Tooth}}(x,y,z) > 0 \land L_{\text{ROI}}(x,y,z) = L_{\text{Tooth}}(x,y,z)$

- $N_{\text{ROI} \cap \text{Tooth}}$ denotes the number of voxels $(x,y,z)$ with: $L_{\text{ROI}}(x,y,z) > 0 \land L_{\text{Tooth}}(x,y,z) > 0$

5.3.2 Tooth separation parameters

We tested the tooth separation algorithm with different parameters (see table 5.1) to evaluate their influence. Since testing all combinations was computationally expensive, we concentrated on varying one parameter and fix the others. Table 5.1 shows the parameters of our tooth separation routine and the values they have if they are fixed.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Standard value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up</td>
<td>dimension of the arc image above the maxilla tooth patch surface in cm (see fig. 5.6)</td>
<td>1</td>
</tr>
<tr>
<td>Down</td>
<td>dimension of the arc image below the maxilla tooth patch surface in cm (see fig. 5.6)</td>
<td>1</td>
</tr>
<tr>
<td>Front</td>
<td>see fig. 5.6</td>
<td>1</td>
</tr>
<tr>
<td>Back</td>
<td>see fig. 5.6</td>
<td>1</td>
</tr>
<tr>
<td>$\Phi_{\text{max}}^1$</td>
<td>maximum value for $\phi_1$ in degrees</td>
<td>45</td>
</tr>
<tr>
<td>$\Phi_{\text{max}}^2$</td>
<td>maximum value for $\phi_2$ in degrees</td>
<td>45</td>
</tr>
<tr>
<td>$X_{\text{max}}^{\text{off}}$</td>
<td>maximum value of $x_{\text{off}}$</td>
<td>0.07</td>
</tr>
<tr>
<td>Slice distance</td>
<td>distances in voxel between the slices to compute $C_{\text{Gradient}}$</td>
<td>1</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>weighting of $C_{\text{AVG}}$</td>
<td>1</td>
</tr>
<tr>
<td>$\beta$</td>
<td>weighting of $C_{\text{Dev}}$</td>
<td>1</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>weighting of $C_{\text{Gradient}}$</td>
<td>1</td>
</tr>
<tr>
<td>$\delta$</td>
<td>weighting of $C_{\text{Pose}}$</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5.1: List of parameters of the tooth separation algorithm

Figure 5.6: Dimension of the tooth arc image
Figure 5.7: Arc image parameters
Figure 5.8: Transformation parameters
5.3.3 Tooth classification

Using the cross validation framework described in section 5.3.1, we evaluated our classification component. In each fold, we used the training ROIs (see section 4.4.2) belonging to the images of the training set $T_{train}$ to train the classifier. From the training ROIs, we determined the histograms. Afterwards, we applied the classifier on the ROI computed by our separation algorithm on the test set $T_{test}$. We experimented with the features:

1. ROI histogram (100 bins)
2. root histogram (100 bins)
3. surface histogram (50 bins)

As classifiers, we used AdaBoost.M1 [FS96] and SVM [CV95].
We assumed classification quality to be dependent on the separation quality of the ROIs. The classifiers are supposed to perform better the higher the volume overlap $V_{\text{overlap}}$ is. Therefore, we tested classification with ROIs of different separation quality. As measure for classification quality $Q_{\text{class}}$, we used the ratio between correctly classified ROIs and total number of ROIs:

$$Q_{\text{class}} = \frac{TP + TN}{N_{\text{Datasets}} \cdot 16}$$

where $TP$ denotes the number of true positives, $TN$ the number of true negatives and $N_{\text{Datasets}}$ the number images in our total test set.

Figure 5.10: Classification quality $Q_{\text{class}}$ dependent on separation quality $V_{\text{overlap}}$
Chapter 6

Discussion

6.1 Maxilla segmentation

6.1.1 Initialisation of the maxilla SSM

According to our tests, our initialisation procedure works robustly on our datasets. The shape segmentation step afterwards converges and drives the SSM to its place. As future work, we consider estimating mouth pose before shape adaption.

6.1.2 Automatic Threshold Finding

We could not prove that our ATF routine improves segmentation quality in the CT images. The figures indicate that the results of the algorithm with a fixed material threshold have less outliers than using ATF. Our conclusion is that our assumption of where the material peaks lie do not apply to all images. Additionally, the framework proposed by Seim et al.\cite{SKH08} seems to be robust towards threshold parameter setting making ATF obsolete.

6.1.3 Rule-based maxilla cost function

With our maxilla cost function, we are able to control segmentation flow even in regions where tooth goes through bone. However, in some datasets, too many vertices are assigned “zero-displacement“ such that the surface the stays at its place. There are no displacements in the neighbourhood, which could drive the SSM at the tooth patches to its place.

Figure 6.1: Profile which is assigned “zero-displacement“ because it contains tooth. The green contour shows the ground-truth segmentation whereas the red contour depicts the automatic segmentation.
Our future challenge is to analyse profiles containing tooth more detailly, such that at least some of them can still be used for shape adaption.

6.2 Tooth separation

6.2.1 Dimension of arc image

For the fixed set of term weights, angle pose parameters and translation pose parameters, we found out that the dimension of the tooth arc image has influence on the result. According to our experiments, tooth crowns seem to be the most reliable feature for separating teeth. This is the opposite of what we expected. We assumed that tooth crowns were less reliable than roots because metal artefacts are most present here.

In contrast, the more of the root is imaged in the tooth arc image, the worse separation quality becomes. Our explanation for this result is that roots of molars and premolars sometimes have a shape which is not separable by a plane. This region in the image may mislead the separation algorithm. We consider the shape of canine roots to be less problematic.

6.2.2 Search space

Parameters to restrict search space are the maximum separation plane angles $\Phi_1^\text{max}$ and $\Phi_2^\text{max}$ and the maximum separation plane distance $X_{\text{off}}^\text{max}$ from its mean position $\overline{x}_i$.

For the angles, our tests have shown that a maximum value of $30^\circ$ leads to best separation quality. Values higher than $30^\circ$ lead to significantly worse performance. We assume that there are solutions in search space which contain at least one extremely tilted plane, which has extremely low cost. As planes are coupled by distance constraints, the other planes are tilted, as well (see fig.6.2).

Concerning maximum translation from the mean position, our algorithm worked robustly. Changing the translation parameter $X_{\text{off}}^\text{max}$ did not change separation quality significantly. In our opinion, the optimal positions for separation planes in our test data do not vary significantly from their mean value $\overline{x}_i$. Therefore, even if $X_{\text{off}}^\text{max}$ is severely constrained, a good solution still exists in the remaining search space. If $X_{\text{off}}^\text{max}$ is high, the distance constraints avoid planes moving too far from each other or too near to each other. To conclude, we believe that limiting search space of translation is not necessary when applying the separation algorithm on our test data.
6.2.3 Plane cost function

The parameters of the cost function were the cost term weights ($\alpha, \beta, \gamma$ and $\delta$) and the distance of the slices for computation of $C_{\text{Gradient}}$. In all our experiments, we just varied one weight and fixed the others. The weights correlate to each other, such that simultaneously varying multiple may lead to different results. To obtain a complete impression of the impact these parameters, we would have to sample the whole parameter space, which is very computationally expensive. Therefore, our conclusions are based on only a subset of possible experiments.

According to our results, the cost terms $C_{\text{AVG}}$ and $C_{\text{Pose}}$ have the biggest impact on separation quality. The value of $C_{\text{AVG}}$ indicates whether the plane contains tooth voxels or not. Teeth have a significantly higher intensity value than other materials and thus raise the mean intensity value on the plane (see $C_{\text{AVG}}$ on table 6.1 (1) and (2)). $C_{\text{Pose}}$ is a regulariser for the plane pose and penalises solutions where plane become extremely tilted (see table 6.1 (4)). The cost term $C_{\text{STD}}$ also indicates the presence of tooth voxels, and plays a role when the plane cuts two teeth (table 6.1 (4)). In this case $C_{\text{AVG}}$ is low but $C_{\text{STD}}$ is high and penalises the “bad” plane pose. The cost term $C_{\text{Gradient}}$ had the least variation and leads in some cases to wrong solutions (see table 6.1: although we want (1) to be the better separation plane it has a higher $C_{\text{Gradient}}$ than (3) and (4)).

Concerning robustness, the figures 5.9 we cannot find a tendency. Although there are values for the weights of $C_{\text{AVG}}, C_{\text{STD}}$ and $C_{\text{Gradient}}$ where the algorithm perform best, but they are 0 (when the other weights are fixed to 1). We need to completely exploit search space to be able to come to a conclusion. There is only one weight parameter which shows a tendency: the algorithm performs robustly when the weighting of $C_{\text{Pose}}$ exceeds certain threshold. Therefore, regularisation has a big influence on separation quality. We assume for our test data that planes in neutral position or just slightly different from neutral position separate teeth well. Only in rare cases, they have to be extremely tilted.

For future work, we intend to design a model which needs less parameters in order to better explore its characteristics.

6.2.4 Comparison to the Integral projection method

We developed an approach to tooth separation which differs from the popular integral projection method [MA05, NAA08, LLH10] in the optimisation process.

In the integral projection method, separation planes are placed at relevant local minima in the projection image (see fig. 2.2). Each plane is independently computed from its neighbours. As the number of these local minima depends on gaps and their size, the number of separation planes needed is not known beforehand.

In contrast, our method assumes 15 planes to be placed as there are normally maximum 16 teeth in a human jaw. Our algorithm minimises the sum of cost of all planes subject to plane position and distance constraints between neighbouring planes. We used a global method whereas the integral projection method is local.

Solving the separation problem by simultaneously optimising all planes has advantages: Firstly, we are able to implement interaction constraints between planes and thus rule out plane combinations which make no sense. Secondly, we do not have to deal with the problem of finding relevant local minima in the cost function. Finally, we have information about which plane separates which teeth and can use it for numbering. We even have information about how many teeth are probably missing in a gap.

However, when planes are not optimised independently, we have to deal with side effects: in our tests we have found out that a plane which was badly placed can destroy the separation quality of other planes. Moreover, if a plane is shifted to the next tooth, and thus does not separate the teeth it is supposed to separate, the other ones will separate wrong teeth, as well. In this case correspondence between plane and teeth are destroyed and cannot be relied on in the classification step. To sum up, our routine only works on cases where teeth are not shifted too much or even shifted to the place of an other tooth. Our future challenge is to adapt optimisation such that cases where teeth are shifted far from expected positions are included in the model.

6.3 Tooth classification

According to our tests, classification performance was more dependent from separation quality of the ROIs than from the combination of feature and classifier. If the ROI isolated teeth well ($V_{\text{overlap}} > 60\%$), classification rate was almost equally high for all feature combinations and classifiers. We believe that
the key step in our tooth detection algorithm is the tooth separation step. If this performs well, even simple features like histograms are sufficient to achieve a reasonable classification rate.

If we assume good separation quality, histogram features have the following limitations: First, the intensity intervals of bone material and tooth interleave each other. In rare cases, bone material in the ROIs has high intensity such that the ROIs are classified as tooth. Similarly, there are teeth with low

<table>
<thead>
<tr>
<th>Case</th>
<th>Plane position and pose</th>
<th>Intensity values on plane</th>
<th>$C_{\text{AVG}}$</th>
<th>$C_{\text{STD}}$</th>
<th>$C_{\text{Gradient}}$</th>
<th>$C_{\text{Pose}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td>0.322799</td>
<td>0.385760</td>
<td>0.531924</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
<td>0.414319</td>
<td>0.462956</td>
<td>0.537266</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
<td>0.370868</td>
<td>0.442361</td>
<td>0.526788</td>
<td>0.454288</td>
</tr>
<tr>
<td>4</td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
<td>0.362000</td>
<td>0.438672</td>
<td>0.524397</td>
<td>0.804574</td>
</tr>
<tr>
<td>5</td>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
<td>0.290585</td>
<td>0.333610</td>
<td>0.511972</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
<td>0.303957</td>
<td>0.322087</td>
<td>0.514927</td>
<td>0.380148</td>
</tr>
</tbody>
</table>

Table 6.1: Examples of plane cost function values
intensity such that the classifier does not recognise the ROIs as tooth. Second, our histograms do not contain information about spatial distribution of the voxel intensities. Therefore, parts of lower teeth in a ROI can cause the classifier to classify the ROI as "tooth", although there is a gap. Future work could involve developing features which contain information about spatial distribution of voxel intensities, such as matching scores of templates.

Compared to related work, we have developed a tooth detection algorithm, which works on CT data and is able to automatically detect teeth. Moreover, the tooth detection process does not need exact boundary information of the teeth, but only a rough region.

6.4 Performance on CBCT images

We tested our algorithm on CBCT images. However, we do not have ground truth segmentations of maxilla and teeth for these datasets. Therefore, we cannot provide a quantitative evaluation but only assumptions based on looking at results.

Concerning maxilla segmentation, we assume that our routine performs worse on CBCT than on CT. CBCT images have lower contrast and therefore, it seems to be harder to find a correct tissue-bone threshold value. In our tests, we have not found a single tissue-bone threshold which leads to good segmentations for all datasets. Our algorithm using ATF created acceptable segmentations (see fig. 6.3). Thus, we assume that ATF has a higher impact on CBCT images making the algorithm significantly more robust. However, local variation of the bone-tissue threshold seem to be high, since we observed that our segmentation algorithm still has problems in sinus regions. According to our opinion, we have to extend the algorithm towards several local tissue-bone thresholds instead of a global one for the whole image.

![Maxilla segmentation with different tissue-bone thresholds on CBCT](image1)

Figure 6.3: Maxilla segmentation on a CBCT image with different configurations. The images show one slice of the CBCT image. The red line denotes the boundary of the maxilla on this slice.

Our tooth separation algorithm does not work as well on CBCT as it does on CT. We believe that the main problem lies in reduced intensity contrast (see fig. 6.4) making it harder to distinguish materials. We believe that image enhancement filters can solve this problem.

![Slice of tooth arc image on CT and CBCT](image2)

Figure 6.4: Slice of a tooth arc image on CT and CBCT
Chapter 7

Conclusions

In our thesis, we have developed algorithms which are able to segment the maxilla jaw and number upper teeth in CT images.

We extended the segmentation framework proposed by Seim et al. to cope with object intensity variance among individual scans and segmentation in regions where tooth goes through the bone. We have shown that that our segmentation algorithm robustly works on 43 CT scans with a mean distance of $\approx 0.5 \pm 0.5\text{mm}$. Moreover, we have proposed an algorithm for tooth numbering in CT datasets. Using histogram features and tooth isolation via Dijkstra’s algorithm, we exploit prior knowledge about position of teeth for localisation and numbering. The overall volume overlap of $\approx 95\%$ shows teeth can be well isolated when using good parameters and their possible location is reduced into a limited region.

The first tests on CBCT images reveal that our algorithm needs improvement to deal with low intensity contrast. For segmentation future work concentrates on developing an algorithm which chooses an individual segmentation strategy based on local intensity features. In tooth detection, we intend to include appearance information, such as using template matching and registration techniques. This idea could lead to an algorithm which does tooth localisation and classification in one step without intermediate results.
List of Figures

1.1 Differences between CT and CBCT images ........................................ 3
2.1 Tooth numbering systems (source: http://users.forthnet.gr/ath/abyss/dep1151_1.htm) ........................................ 5
2.2 Integral projection method ............................................................... 6
2.3 Point correspondences between two shapes. For each point of the shape 1 a coloured arrow indicates the corresponding point in shape 2 and vice versa. ........................................ 8
2.4 Correspondences between patches. The two surfaces are decomposed into patches. The three arrows show three examples of corresponding patches. Patches and their counterparts in the other surface are located approximately in the same region of the surface and are marked by the same colour. ........................................ 9
2.5 Scheme of an intensity profile (red) and an example for a displacement vector (green). ........................................ 10
2.6 Intensity value profiles ............................................................... 11
2.7 Profile features and cost function .................................................. 11
2.8 Visualisation of ‘inside’ and ‘outside’ in a profile. The green line denotes the location of the SSM boundary. ........................................ 12
2.9 Segmentation strategy in [LZW+06]. .................................................. 12
2.10 Cost function in [SKH+08]. .......................................................... 13
2.11 Cases where CLocation should solve ambiguities ................................ 14
3.1 Shape and intensity variation among maxillae. The images show 2 individual maxillae with CT image intensity values of the corresponding dataset mapped on. ........................................ 16
3.2 Correspondences between training surfaces. Red arrows express that the training surfaces (1) S1 and (2) S2 have the same parametrisation like the reference surface (R) SR. Therefore, there are point correspondences between S1 and S2 (expressed by the black arrow). ........................................ 17
3.3 Cutting plane (orange line) to divide the skull surface into 2 parts. ........................................ 17
3.4 Patch groups in the maxilla SSM .................................................... 18
3.5 Tooth patch decomposition of the maxilla SSM. The green numbers indicate existing teeth whereas the red numbers indicate missing teeth. ........................................ 19
3.6 Overview of the maxilla segmentation algorithm ................................ 20
3.7 SSM initialisation templates (yellow) with aligned maxilla SSM (red). ........................................ 21
3.8 Maxilla and mandible SSMs on regions where tooth goes through bone. ........................................ 22
3.9 One slice of a head CT dataset with segmented maxillae. The green contour shows the manually segmented maxilla. The red contour shows the automatically segmented maxilla using cost function Eq.2.12. The red contour “wraps around” teeth. ........................................ 22
3.10 Intensity histogram in the bounding box of an initialised SSM. ........................................ 23
3.11 Intensity histograms of images in the bounding box of their initialised SSM. ........................................ 24
3.12 Intensity Histogram and fitted function ........................................ 25
3.13 Minimum size of 1cm for material regions on an intensity profile ........................................ 26
3.14 Appearance of a “zero-displacement” profile cost function ........................................ 27
3.15 Minimum size of 0cm for material regions on a “zero-displacement” profile cost function ........................................ 28
4.1 Overall scheme of the detection algorithm ........................................ 30
4.2 Visualisation of the tooth arc image ........................................ 31
4.3 Grid building ............................................................... 32
4.4 Separation planes and separation surfaces ........................................ 32
4.5 Parametrisation of a separation plane in tooth arc image space ........................................ 33

62
List of Tables

1.1 Hounsfield units of some materials in CT. Source: http://en.wikipedia.org/wiki/Hounsfield_scale ......................................................... 3

5.1 List of parameters of the tooth separation algorithm ........................................... 51

6.1 Examples of plane cost function values ............................................................ 59
Bibliography


