SEGMENTATION OF THE MANDIBULAR CANAL IN CONE-BEAM CT DATA

Dirk-Jan Kroon

De promotiecommissie :

voorzitter en secretaris:	
Prof.dr.ir. A.J. Mouthaan	Universiteit Twente
promotor:	
Prof.dr.ir. C.H. Slump	Universiteit Twente
leden:	
Prof.dr.ir H.F.J.M. Koopman	Universiteit Twente
Prof.dr.ir. A. Stein	Universiteit Twente
Prof.dr. S.J. Bergé	Radboud University Nijmegen Medical Centre
Prof.dr.ir. P.H.N. de With	Technische Universiteit Eindhoven

This research is financially supported by Virtual Patient Project, provincie Overijssel

Signals & Systems group,
EEMCS Faculty, University of Twente
P.O. Box 217, 7500 AE Enschede, the Netherlands

Print: Gildeprint B.V
Typesetting: LaTeX2e

ISBN 978-90-365-3280-8
DOI 10.3990/1.9789036532808

SEGMENTATION OF THE MANDIBULAR CANAL IN CONE-BEAM CT DATA

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof. dr. H. Brinksma,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op 1 december 2011 om 14.45

door

Dirk-Jan Kroon
geboren op 29 maart 1982
te Amstelveen

Dit proefschrift is goedgekeurd door de promotor:

Prof.dr.ir. C.H. Slump

Contents

Nomenclature

Abbreviations

AAM	Active appearance model
ASM	Active shape model
CBCT	Cone-beam computed tomography
CED	Coherence-enhancing diffusion
CT	Computed tomography
EED	Edge-enhancing diffusion
FA	Fractional anisotropy
FDK	Feldkamp-Davis-Kreiss
FFD	Free form deformation
FLAIR	Fluid attenuated inversion recovery
GVF	Gradient vector field
HDCS	Hybrid diffusion filter with a continuous switch
IAN	Inferior Alveolar Nerve
LK	Lucas Kanade
MD	Mean diffusivity
MDL	Minimum description length
MI	Mutual information
MIP	Maximum intensity projection
MRI	Magnetic resonance imaging
MSCT	Multi-slice helical computed tomagraphy
OPG	Orthopantomogram

PCA	Principal component analysis
RMS	Root mean square
ROI	Region of interest
RPM	Regularized non-linear diffusion
SSD	Sum of squared differences
STD	Standard deviation
SVD	Singular value decomposition
VR	Volume rendering

1
Introduction

Oral and maxillofacial surgery, is often planned using 2D and 3D diagnostic images of the teeth, mandible and other facial structures [1]. The common 2D imaging modality in dental surgery are X-ray photos[2]. Three dimensional imaging such as computed tomography (CT) is often only available for advanced surgery in hospitals. Cone beam computed tomography (CBCT) is a high resolution, low cost alternative to CT [3]. CBCT is based on 2D X-ray projections, and has a lower dose than conventional CT [4], and therefore may become the imaging technique for oral and maxillofacial surgery.

The mandible is the lower-jaw bone, which is the largest and strongest bone of the face. It is a complex bone, which serves for the reception of the lower teeth, and also contains channels with blood vessels and nerves. The nerves give sensation to the lower-lip, tongue and teeth. It is important in dental implantology and wisdom teeth removal to plan a safety margin around the facial nerves [5]. Because impairment of the channels containing the nerves can traumatize, crush or completely cut the nerve. Damaged mandibular nerves often recover in about three to six months, because axons slowly regenerate [6]. If after a year recovery has not occurred the nerve damage is likely to be permanent. The incidence of damage during the removal of wisdom teeth has been reported in the range of 5% [6] [7].

The inferior alveolar nerve running through the mandibular canal is the third branch of the trigeminal nerve. The three parts of the trigeminal nucleus in the brain receive different types of sensory information, pain, temperature and touch. Therefore damage to the mandibular nerve can cause pain, or permanent loss of tactile sensation of the lower lip and chin [5].

In this thesis we focus on extracting the mandibular canal from cone beam CT, which will allow the planning of safety margins around the nerve. Exact localization of the mandibular nerve canal in CBCT data is highly challenging, because CBCT has a lower dose and thus a higher noise to signal ratio than conventional CT [8].

This introduction continues with background information to support the topics of the thesis. In the next section, Section 1.1 we present an short overview about the mandible and its surroundings. Followed by section 1.2, about the change in shape of the mandible due to age. Then we focus in section 1.3, on the background of the Mandibular canal, which is the main topic of our research. The data used in this thesis are Cone-Beam CT scans, thus we give a short overview in Section 1.4. Section 1.5 and Section 1.6 present the research objectives and outline of this thesis.

1.1 Mandible

The mandible[1] is the lower-jaw bone of the face. It consist of three parts, the body, which is the central curved horizontal part, and the rami which are located at both ends of the body and are vertical quadrilateral shaped parts of the bone. See figure 1.1. The tooth bearing part of the body is called alveolar process, and the corner between ramus and body, is called angle.

During the third and fourth week of pregnancy the first pharyngeal arch develops [9]. It grows outwards from the two sides left and right and as two processes, a lower (mandibular) part and an upper (maxillary) process, figure 1.2. The place where the two pieces of mandibular bone fuse together is called the symphysis menti. The symphysis menti which is shaped as a ridge, which divides near the bottom of the mandible, enclosing a small triangle shaped area, the mental protuberance.

At the top of the ramus there is a triangular shaped protuberance, called the coronoid process where the temporalis muscle attaches. The ramus also has a round prominence called the condyle, which makes the temporomandibular joint with the temporal bone.

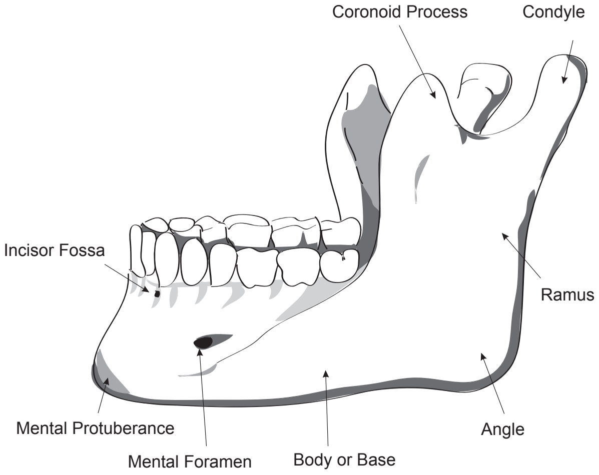
The mandible contains cancellous bone which has a structure as in a sponge [10], making it light but strong.

The mental foramina are two tear shaped holes at both sides of the mandible close to the mental protuberance. The holes permits the passage of blood vessels and the mental nerve, a trunk of inferior alveolar nerve. The inferior alveolar nerve runs through the mandibular canal and enters through the mandibular foramen which is a hole at the inner aspect of the mandible at the middle of the ramus.

The mandible connects to the temporal and zygomatic bone (or zygomatic process) by ligaments: the temporomandibular ligament, the sphenomandibular ligament, the stylomandibular ligament and the articular capsule containing the articular disc, figure 1.3. The movements permitted by the ligaments are large, the mandible may be depressed or elevated, moved forward or backward, and even a small amount of side to side movement is possible.

The mandible contains many muscle attachments 1.4 [11]. The two major muscles are the masseter and temporalis muscle. The masseter is a thick quadrilateral shaped

[1] We use the Terminologia Anatomica, which is the international standard for anatomical nomenclature.



(a)

Figure 1.1 The mandible, outer surface side view.

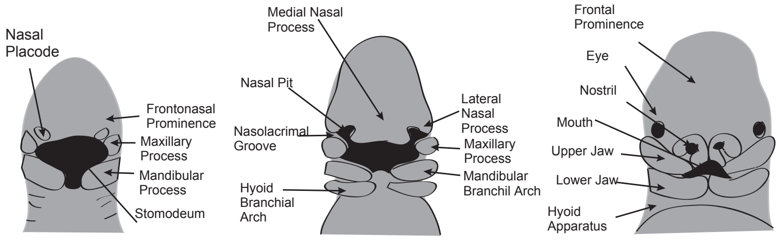


Figure 1.2 Development of human embryo head (ventral view) in the third week [12].

muscle, consisting of two parts, the superficial and the deep part. The superficial part is attached to the zygomatic process of the maxilla, and to a part of the lower body of the zygomatic arch, and connects to the lower half of the lateral surface of the ramus of the mandible. The deep portion is smaller and attached to the zygomatich arch and connects into the upper half of the ramus and the lateral surface of the coronoid process. The temporalis muscle attaches to the temporal bone of the skull and connects to the coronoid process of the mandible.

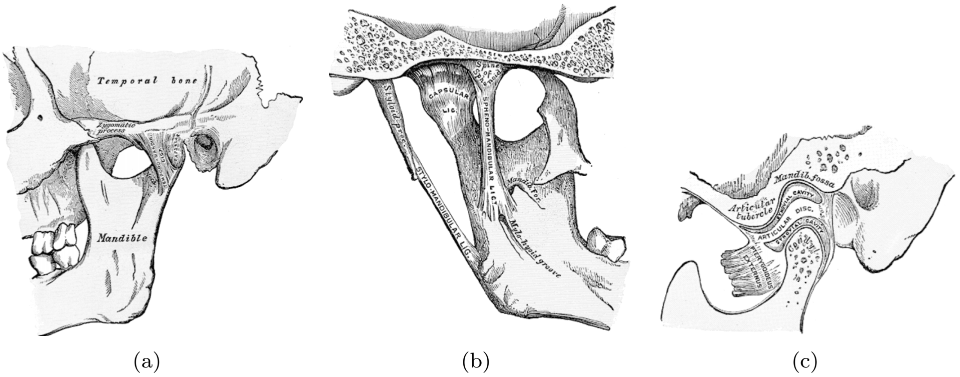


Figure 1.3 Ligaments of the mandible . The Images are from "Gray's Anatomy of the Human Body".

1.2 Ageing

At the symphysis menti the two segments of the mandibular bone fuse, upwards from below. In the beginning of the second year of a newborn there may be still a trace of separation visible.

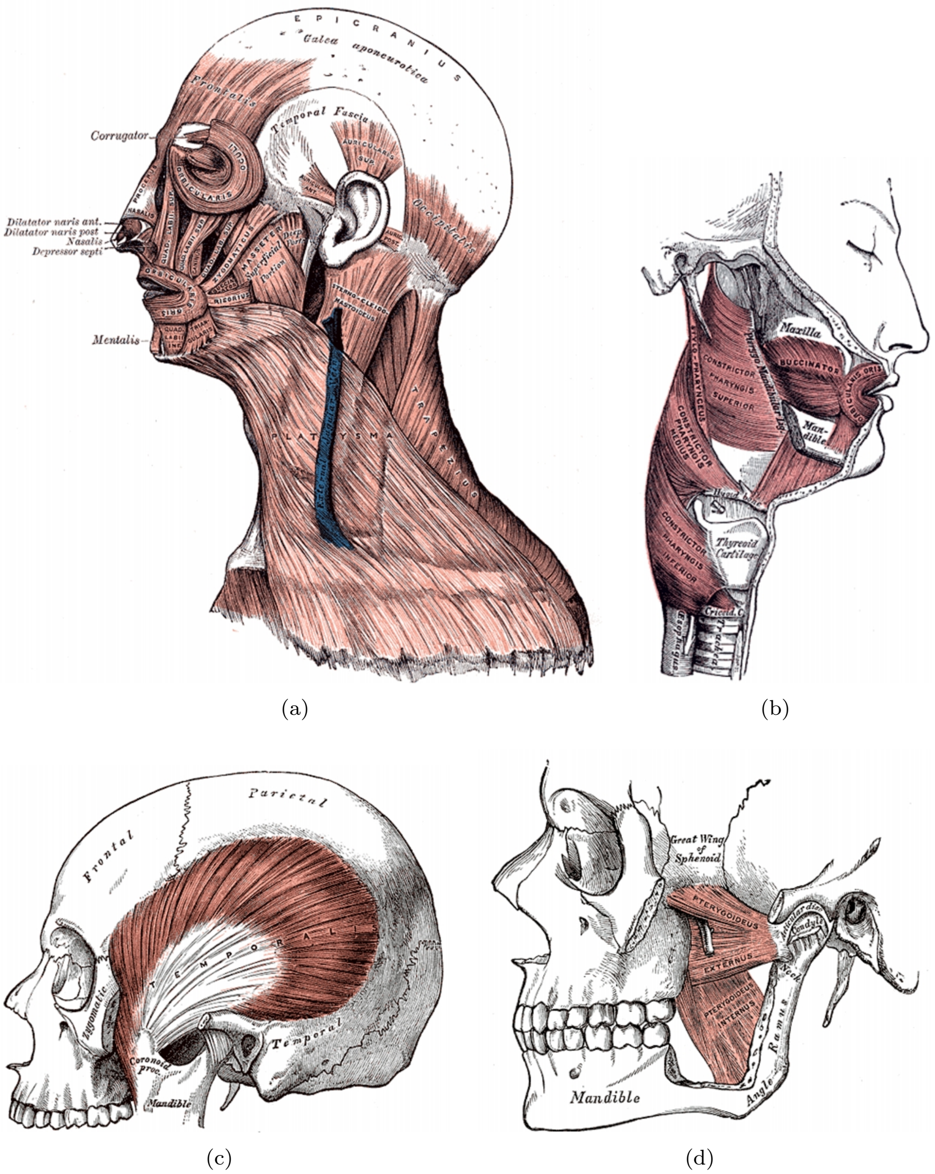
In the first years the body of the mandible becomes elongated in length, to provide the space for teeth to develop, see figure 1.5. Also the depth of the body increases due to grow of the alveolar part, which will contain the root of the teeth to be developed in the future [13].

In an adult the alveolar and subdental part of the body are almost always equal in depth. The ramus is almost vertical in most adults. With the mental foramen at the height of the depth center line of the body.

In old age, the mandible changes due to teeth loss and subsequent absorbtion of the alveolar process [14]. As a result, the position of the mental foramen is close to the alveolar border.

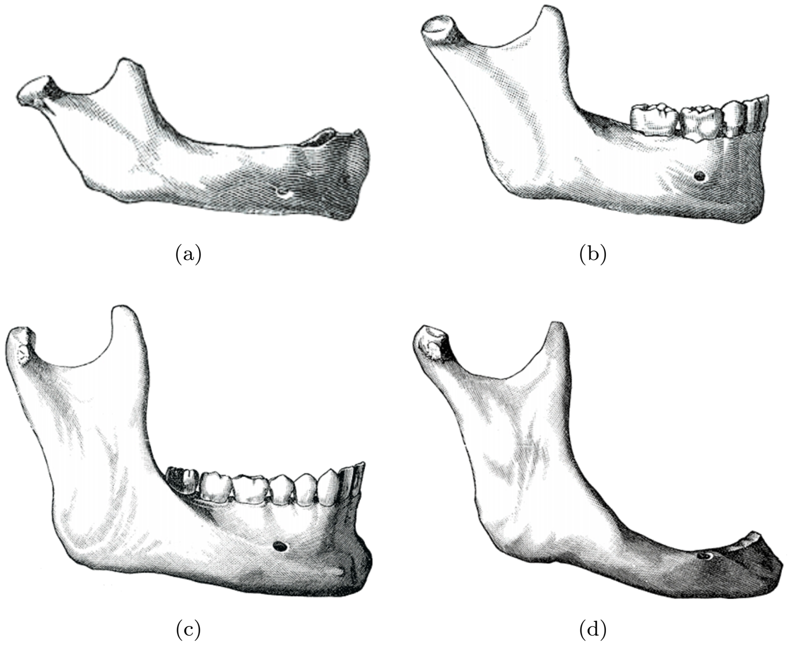
1.3 The Mandibular Canal

Inside the mandible there is a small canal called the mandibular canal, containing the inferior alveolar bundle (nerve, artery and vein) [15]. The canal starts at the mandibular foramen, a small hole in the ramus, and runs to the mental foramen, a small hole close to the mental protuberance, see figure 1.6. The average canal diameter found literature, range from $2.0mm - 2.4mm$ in one study, up to $5.0mm$ in another study[16]. The average length of the mandibular canal in our 13 CBCT scans is $70mm$, see chapter 10. The mandibular canal is normally located using panoramic radiographs, which show for 99% [17] of all patients a single canal in each side of the mandible. But new studies done with cone beam CT, show bifid canals for almost all



(a) (b) (c) (d)

Figure 1.4 The mandibular muscles. The Images are from ”Gray’s Anatomy of the Human Body”.



(a)

(b)

(c)

(d)

Figure 1.5 Aging of the mandible. Mandible of a newborn (a), of a child (b), of a adult (c), and of elderly person (d). The Images are from "Gray's Anatomy of the Human Body".

patients, for instance 28 findings in the mandible sides of 18 patients [15]. Bifid canals often branch from the main mandibular canal, and can contain branches of the inferior alveolar nerves, artery and vein. The diameter of the bifid canals are almost always smaller than the mandibular canal, and can range in length from 0.14 to 2.45cm [18]. There are four types of bifid mandibular canals: forward canal, retromolar canal, dental canal, and buccolingual canal [18]. A retromolar bifid canal runs to an area close to the ramus before the molars, a dental canal runs to the roots of the second and third molars. The forward canal arising from the superior wall of the mandibular running to the second molar region. The buccolingual canal is a bifid canal which arises from the buccal or lingual wall of the mandibular canal.

1.4 Cone Beam CT Data

Cone beam computed tomography (CBCT) is a type of three dimensional computed tomography (CT).

Today multi-slice helical CT (MSCT) scanners are commonly used in hospitals. The outside of a MSCT scanner contains a tube through which the patient who is positioned on a table, is moved. Around the tube there is a wheel spinning with X-ray

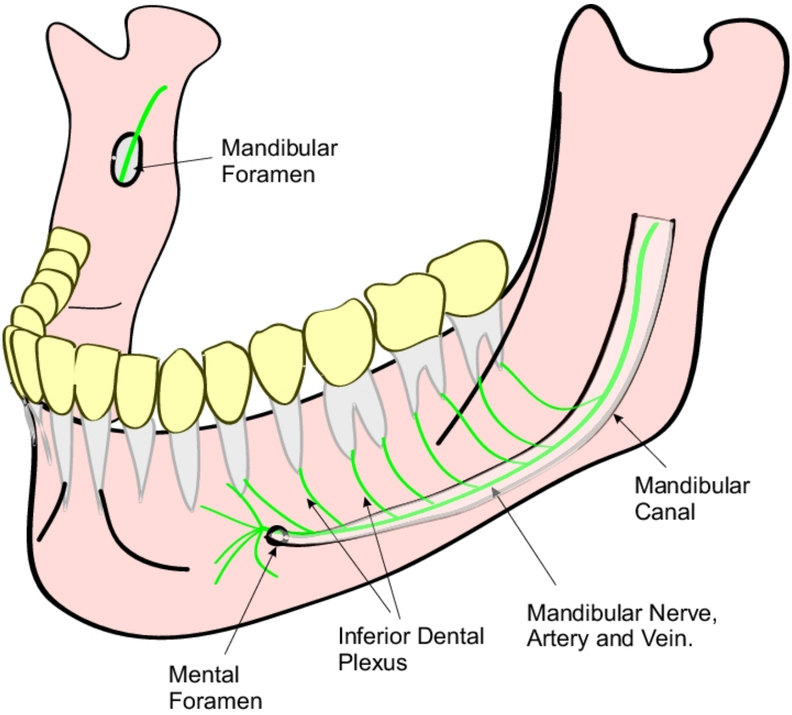


Figure 1.6 The mandibular canal inside the mandible.

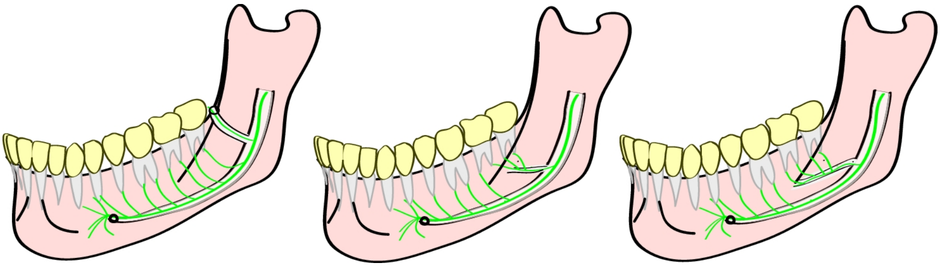


Figure 1.7 Bifid canals: a retro molar canal, a dental canal and a forward canal.

detectors and opposite to the detectors and X-ray source. In an multi-slice system up to 320 rows of X-ray detectors are used [19], with up to 912 detectors in one row. X-rays are produced in a flat fan-shaped pattern, which projects through the patient onto the detectors. To reconstruct a CT slice we need projection data from a full rotation of the source and detectors in the plane of the slice. But during scanning the table moves, resulting in helical projection data. Thus interpolation and rebinning [20] is needed before an image volume can be reconstructed.

CBCT is closely related to normal radiography. Instead of using rows of X-ray detectors, it uses a flat panel X-ray imaging device [8] , and not a flat fan-shaped pattern but a cone shaped X-ray beam. This is the same setup as used for standard 2D X-ray photos. To reconstruct a 3D volume we need X-ray images from multiple angles. For this reason, the source and detector-plate are mounted at opposite sides of an curved shaped metal bar. This bar rotates while, the head or other body part of a patient is positioned in between the detector and X-ray source.

CBCT has several advantages in comparison to helical CT scanners. Efficient sampling without the need for rebinning is the main advantage. All the data in one 2D projection of a CBCT scan has a single time instance, thus allowing efficient gated imaging and less respiratory motion artifacts than in multi-slice CT. The cost of a CBCT scanner is much lower than a CT scanner and also the device is smaller thus more suitable for dentists. Main disadvantage of CBCT is that areas with the same bone tissue do no result in the same Hounsfield unit, due to non-uniformity of X-ray illumination and scattering. Because CBCT has a high scatter-to-primary (SPR) ratio, which is caused by the use of a wide cone-beam [8]. A CBCT scan has a much lower total X-ray dose (13 to 82 μSv) than multi-slice CT (474 to 1160 μSv) [21]. The low dose is good for the patient but is also a disadvantage, because a lower dose results in more noise, lower dynamic range and lower image quality in the reconstruction.

The standard method for 3D reconstruction from cone-beam projections is the Feldkamp-Davis-Kreiss (FDK) algorithm [22]. This is a 3D generalization of 2D fan beam filtered back projection. This is a fast and robust reconstruction algorithm, but requires a large number of projections. After reconstruction with FDK, the image often contains streak artifacts. These streak artifacts are caused by materials which totally block X-ray such as metal implants, beam hardening and scatter. There are more advanced algorithms appearing which rely on optimizers to reconstruct the image volumes with well-defined boundary conditions and fewer artifacts. But those iterative algorithms are slow because a 3D volume contains a large number of voxels thus a large number of unknowns values. Therefore also a large number of approaches have appeared which are extensions of FDK [23] aiming to reduce artifacts and noise.

1.5 Research Objective

The aim of our research is to develop an automatic system for the extraction of the mandibular canal from CBCT data. This system allows the creation of an safety margin around the canal in case of dental implant surgery. To achieve this objective, we address the following research questions

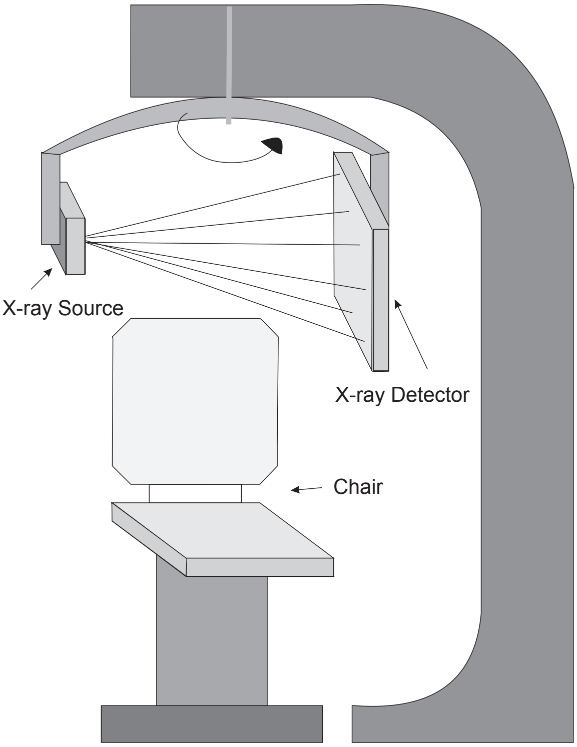


Figure 1.8 Cone-beam CT for dental applications.

- Is it possible to get high enough accuracy and robustness to replace human annotation of the mandibular canal in surgery planning?

- Is it possible to accurately extract the mandibular canal from CBCT only based on intensities, or is shape information from a training set needed?

- Which of the three approaches is more suitable for extraction of the mandibular canal, an active shape model, active appearance model or an atlas based registration method?

- What is a suitable method to get corresponding points between patient CBCT data sets of the mandible?

- Does adding edge enhancing filtering of the CBCT data as pre-processing step, increase the accuracy in mandibular canal localization?

Our conclusions and recommendations with respect to these questions are discussed in Chapter 11

1.6 Outline

Chapter 2, "Overview of mandibular canal segmentation methods", gives an overview of existing automated and automatic methods for mandibular canal segmentation in literature. In Chapter 3, "Diffusion Filtering", we describe a method for anisotropic diffusion filtering with optimized rotational invariance, to reduce the noise in the CBCT data. In Chapter 4, "Nerve Tracking", a method based on Lukas Kanade[24] template tracking is introduced, for automatic extraction of the mandibular canal based on intensities. In Chapter 5, "Active Shape Model", the active shape model and active appearance model approach of Cootes et al. [25][26] based on principal component analysis of corresponding points is introduced, and implementation details are explained. In Chapter 6, "Demon Registration", and Chapter 7, "B-spline Registration", image registration methods are introduced which can be used for atlas-registration based extraction of the mandibular canal, and as a way to find corresponding points between mandibles of several patients. In Chapter 8 "Shape Context Registration", we describe a robust method for finding corresponding points between object surfaces. Chapter 9 "Minimum Description Length", gives a method to improve the quality of an active shape, model. In Chapter 10 "Results", we evaluate the segmentation performance of the introduced methods. The performance is tested on segmentation of the mandible and mandibular canals from CBCT. Finally in Chapter 11, "Conclusion", we give our conclusions, recommendations, and answers on the research questions.

2 Mandibular Canal Segmentation Literature

The first step after formulating research goals to solve a certain problem, is always searching for existing solutions to the problem. In this chapter we show an overview of existing methods for mandibular canal segmentation. The first section describes the literature search The second section gives an overview of methods found in literature. Followed by a section discussing the current methods in use.

2.1 Literature Search

We will use a number of public databases to construct a list of methods in literature to segment the mandibular canal in CBCT data. The first, PubMed, comprises more than 20 million citations for biomedical literature from Medline, life science journals, and online books *(http://www.ncbi.nlm.nih.gov/pubmed).* We also used web of knowledge from Thomson Reuters *(apps.isiknowledge.com).* The third database we use is Elsevier's Science Direct *(http://www.sciencedirect.com/).* To get a more comprehensive list, we also use the cross-references of the retrieved literature, and use "cited by" search engines, *(www.scopus.com)* from ScienceDirect and CiteSeerx *(http://citeseerx.ist.psu.edu)* We have used the following key-words: mandibular canal, segmentation, extraction, automatic, cone-beam ct, computed tomography, inferior alveolar nerve. (Last updated 27-06-2011)

2.2 Literature Overview

1998 "*Tracing of thin tubular structures in computer tomographic data*", by Stein et al. [27]. They propose a mandibular canal detection algorithm based on

Dijkstra's algorithm, which is used on CT scans. Followed by a balloon snake (deformable model) to subtract boundaries of the mandibular canal. They limit the Dijkstra's search, by a series of erosions and dilations to only trace inside the bone. User interaction is limited to identification of a start and end-point.

2004a "*Nerves - level sets for interactive 3d segmentation of nerve channels*", by Hanssen et al. [28] . This method improves on the method of Stein. Instead of using the basic Dijkstra's algorithm, fast marching is used which gives more accurate distance results for image volumes. The balloon snake to extract the boundaries of the mandibular canal is replaced by a geodesic active surface computed with level sets. User interaction is needed to identify the start and end-point of the mandibular canal.

2004b "*Computer-based extraction of the inferior alveolar nerve canal in 3D space.*" Kondo et al. [29]. This method first computes panoramic CT images from the CT data. Therefore first the mandible is segmented from the volume data by a threshold, followed by hole filling, and user based selection of the mandible from the remaining image objects. The mandibular canal is then roughly segmented using image gradients, into a binary volume. The canal is then extracted using some binary mask based line-tracking method. The method is tested on two samples. Results give an approximately mean distance to expert labeling of the canal of around 0.7 voxels, with a STD of 0.7 voxels. The distance errors are in the range of half a millimeter.

2006 "*Automatic segmentation of jaw tissues in CT using active appearance models and semi-automatic landmarking*", by Rueda et al. [30]. This method is based on a 2D active appearance model. The model is constructed from manual annotation of the boundaries of the mandibular canal, bone and nerve. The segmentation accuracy is a mean distance of $4.7mm$ for the dental nerve and $3.4mm$ to expert labeling of the mandibular canal.

2006 "*Automatic detection of inferior alveolar nerve canals on CT images*", Sotthivirat et al. [31] propose a canals segmentation method, based on 2-D panoramic images. By using morphological opening, closing and edge detection operations, the canal is found in these images. The 2-D panoramic images are constructed from manual selected points on the centerline of the mandible in a slice of the CT data. The method is tested on one data set and only successfully extracted a part of the mandibular canal.

2008 "*An adaptive region growing method to segment inferior alveolar nerve canal from 3D medical images for dental implant surgery*", by Yau et al. [32], first reslices the image volume, to a kind of panoramic image volume. Then an initial point is selected by the user inside the canal, followed by local region growing and automatic selection of a new initial point for region growing. The whole mandibular canal is iteratively extracted.

2009 "*Automatic Extraction of Mandibular Nerve and Bone from Cone-Beam CT Data*", by Kainmueller et al. [33]. The method is based on segmenting the mandible using an active shape model (ASM), which is constructed, from 106 clinical data sets. A training data set is first manually decomposed into 8 patches, and then an automatic method is used to find the surface correspondences needed to build an ASM, see [34]. The mandibular canal is included in the PCA model, but only to get an initial position of the canal after ASM based bone segmentation. Thus the canal is ASM segmentation is not driven by image features but only by the shape constraints. The actual mandibular canal segmentation is done by a Dijkstra's algorithm based optimization method. This method tries to find the darkest tunnel close to the initial position of the canal found, which was obtained with the PCA model. The method gives accurate result but has difficulty in precisely detecting the canal path near the canal ends. The right nerve can be detected with an average error of $1.0mm$ and standard deviation of $0.6mm$, the left nerve $1.20mm$ and deviation of $0.90mm$

2011"*Automatic extraction of inferior alveolar nerve canal using feature-enhancing panoramic volume rendering*", Kim et al. [35]. First the CT data is rendered using a 3-D panoramic volume rendering algorithm. This rendering algorithm uses texture features to enhance the mental foramens. The path of the mandibular canal is then computed using a line-tracking algorithm. Fast Marching is used with a new speed function to extract the whole region of the mandibular canal. The new speed term is introduced for stopping the front from evolving outward at both openings at the side of the canal and near the foramens. Average time required to process one data set is $13s$. The proposed method is tested on 10 CBCT data sets and has a mean distance to expert segmentation of $0.73mm$ and STD of $0.69mm$

2.3 Discussion

In the previous section we introduce eight papers on semi-automatic segmentation of the mandibular canal. Commonly used methods often use a pre-processing step with shortest path methods such as fast marching or Dijkstra's algorithm. To detect the mandibular canal, the assumption is made that the canal contains low intensities surrounded by a boundary of voxels with high Hounsfield values. A speed map can then be constructed with the speed inversely related to intensity. Thus the shortest path is through the black tunnel of the mandibular canal.

We have tested some of the common methods from literature to see which show potential for our CBCT data. The first fast marching on our CBCT data, took often shortcuts outside our mandibular canal. This is because our canals have not a distinct contrast with the surroundings, probably due to a low X-ray dose. Also the boundary of the bone is not always present or visible in our CBCT data sets. We have also tested the region-growing segmentation method, but missing edge segments resulted into leakage outside the canal.

Our CBCT data contains mandibular canals with missing edges. Also the canals have no lower intensities than the surrounding cancellous bone. Thus we probably have to include a priori shape-information into our segmentation. This can be done by using an ASM or AAM method. These models are trained using manually segmented data sets, as done by Kainmueller et al. [33]. Another way to introduce a priori information about shape and appearance is atlas-registration.

Both atlas registration, ASM and AAM based mandibular canal extraction algorithms will be developed and evaluated in this research.

3
Diffusion Filtering[1]

3.1 Introduction

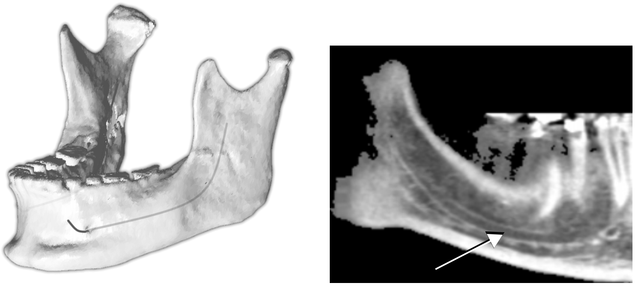
Cone-beam computed tomography (CBCT) is an increasingly utilized imaging modality for dental surgery planning [36], due to the low hardware cost and high resolution images at a relative low radiation dose. For the surgical planning of implants, the mandibular nerve canals have to be segmented. In these scans the mandibular nerve canals are hardly visible. In implant placement, the segmentation is used to guard the safety margin around the canals during surgery. CBCT scanners have a relatively low radiation dose [36] thus the small mandibular canal is characterized by low contrast in a noisy image, see figure 3.1. Which gives us the problem to find a method to improve image contrast in CBCT scans for small structures.

Currently the best way to improve contrast in a CT image is to apply iterative reconstruction methods with regularization to suppress streak-artifacts and to improve smoothness in uniform regions [37]. In practice, standard CBCT systems do not provide the required raw-scanner data for this approach. Therefore post reconstruction noise filtering is the practical method to improve image quality. A medical image is often assumed to have piecewise smooth regions with noise, separated by sharp edges. There are many methods available in the literature to denoise such an image [38], in this chapter we focus on edge enhancing diffusion filtering.

[1] This chapter is based on the following publications:

i) D.J. Kroon et al. ”Optimized Anisotropic Rotational Invariant Diffusion Scheme on Cone-Beam CT”, MICCAI, 2010

ii) D.J. Kroon et al. ”Coherence Filtering to Enhance the Mandibular Canal in Cone-Beam CT Data”, IEEE-EMBS, 2009



(a) (b)

Figure 3.1 Isosurface rendering of the jaw, with overlay of the mandibular canals (a). Slice of panoramic transformed high quality CBCT scan showing the mandibular canal (b).

Linear diffusion equals Gaussian filtering in which the diffusion time controls the smoothing scale. To preserve the edges, Perona-Malik introduced regularized nonlinear diffusion (RPM) [39]. Edge preservation is achieved by lowering the scalar diffusion constant in the neighborhood of steep edges. This method results in piecewise smooth regions, however, image edges remain noisy. Instead of using a scalar diffusion constant, a tensor can be used to adapt the diffusion to the underlying image structure. So we smooth with small elongated kernels along edges, and Gaussian like kernels in uniform regions. The tensor can be constructed in two ways, as a coherence-enhancing diffusion (CED) [40] or as an edge-enhancing diffusion (EED). Recently the CED and EED algorithms are combined in an hybrid diffusion filter with a continuous switch (HDCS) [41]. If the local image structure is tubular HDCS switches to CED and if it is planar it switches to EED.

This chapter will focus on the discretization schemes of the anisotropic diffusion tensor. We will evaluate the performance of the standard discretization scheme and the rotational invariant scheme of Weickert [42], and introduce a new scheme in which optimal filtering kernels are constructed using numerical optimization.

This chapter is organized as following. In the section two we introduce the diffusion filtering algorithm and discretization schemes. The new optimized scheme is introduced in the third section. Followed by, an evaluation of the diffusion schemes on synthetic and real images. The final section gives the discussion and conclusions.

3.2 Diffusion Filtering

Anisotropic diffusion filtering is an iterative edge preserving smoothing method. It describes the local image structure using a structure tensor also referred to as the ”second-moment matrix”, for details see [40]. This descriptor is transformed into a diffusion tensor $\mathbf{D}$. The diffusion equation is commonly written in an iterative forward difference approximation [42]:

$$\frac{\partial u}{\partial t} = \nabla \cdot (\mathbf{D}\nabla u) \quad \Rightarrow \quad u_{k+1} \cong u_k + \tau\left(\nabla \cdot (\mathbf{D}\nabla u)\right) \tag{3.1}$$

Where u ($u = u(t, x, y, z)$) is the image, x, y, z the pixel coordinates and t the diffusion time. In the discrete function, τ the time step-size and k the number of the iteration. The eigenvectors of the diffusion tensor $\mathbf{D}$ are set equal to the eigenvectors $\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}$ with $\mathbf{v_1} = [v_{11}, v_{12}, v_{13}]$ of the structure tensor (note the symmetry):

$$\mathbf{D} = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{12} & D_{22} & D_{23} \\ D_{13} & D_{23} & D_{33} \end{bmatrix} \quad \mathit{with} \quad D_{i\,j} = \sum_{n=1..3} \lambda_n \, v_{n\,i} \, v_{n\,j} \tag{3.2}$$

The eigenvalues of the diffusion tensor are $\lambda_1, \lambda_2, \lambda_3$. Because our CBCT scans contain planar and tubular structures as well, we choose to use HDCS [41], with switches between CED and EED eigenvalues depending on the local image structure . When using the diffusion tensor for plane enhancing diffusion (EED), the eigenvalues are set to:

$$\lambda_{e_1} = \begin{cases} 1, & (|\nabla u_\sigma|^2 = 0) \\ 1 - \exp\left(\frac{-C}{(|\nabla u_\sigma|^2/\lambda_e^2)^4}\right) & (|\nabla u_\sigma|^2 > 0) \end{cases} \quad , \ \lambda_{e_2} = 1 \ , \ \ \lambda_{e_3} = 1 \tag{3.3}$$

With u_σ the image smoothed with a Gaussian kernel of σ, and λ_e the planar structure contrast parameter. In case of tube enhancing diffusion (CED) the eigenvalues are set to:

$$\lambda_{c_1} = \alpha \ , \ \ \lambda_{c_2} = \alpha \ , \ \ \lambda_{c_3} = \begin{cases} 1, & (|\nabla u_\sigma|^2 = 0) \\ \alpha - (1-\alpha)\exp\left(\frac{-\log(2)\lambda_c^2}{(\mu_2/(\alpha+\mu_3))^4}\right) & (|\nabla u_\sigma|^2 > 0) \end{cases} \tag{3.4}$$

With λ_c the tube-like structure contrast parameter, and $\alpha = 0.001$. The Hybrid diffusion with continuous switch, combines the CED and EED eigenvalues depending on the image structure. The weight between CED and EED eigenvalues is determined by the value ϵ :

$$\lambda_{h_i} = (1-\epsilon)\lambda_{c_i} + \epsilon\lambda_{e_i} \tag{3.5}$$

When the structure tensor has two large eigenvalues, the image structure is planar, and can be best filtered with EED. When there is only one large eigenvalue, the image structure is tube like, and can be best filtered with CED. This can be measured with the ξ equation below:

$$\xi = \left(\frac{\mu_1}{\alpha + \mu_2} - \frac{\mu_2}{\alpha + \mu_3}\right) \tag{3.6}$$

With μ the eigenvalues of the structure tensor, and $\alpha = 0.001$. In case of a plate like structure $\xi >> 0$, with a tubular structure $\xi << 0$, and if the eigenvalues are equal as in a homogeneous or blob-like region $\xi \ = 0$. The epsilon value which switches between CED and EED is written as:

$$\epsilon = \exp \frac{\mu_2(\lambda_h^2(\xi \ - |\xi|) - 2\mu_3}{2\lambda_h^4} \tag{3.7}$$

Which contains a contrast parameter λ_h, setting a soft threshold between the eigenvalues of a small structure and noise.

3.3 Discretization Schemes

We can write the divergence operator equation 3.1 in 3D as:

$$\nabla \cdot (\mathbf{D}\nabla u) = \partial_x j_1 + \partial_y j_2 + \partial_z j_3 \tag{3.8}$$

With j_1, j_2, j_3 the flux components which are described by:

$$\begin{aligned} j_1 &= D_{11}\left(\partial_x u\right) + D_{12}\left(\partial_y u\right) + D_{13}\left(\partial_z u\right) \\ j_2 &= D_{12}\left(\partial_x u\right) + D_{22}\left(\partial_y u\right) + D_{23}\left(\partial_z u\right) \\ j_3 &= D_{13}\left(\partial_x u\right) + D_{23}\left(\partial_y u\right) + D_{33}\left(\partial_z u\right) \end{aligned} \tag{3.9}$$

For the standard discretization of the divergence operator central differences are used:

$$\begin{aligned} \partial_x\left(D_{11}\left(\partial_x u\right)\right) = \frac{1}{2}\Bigg(& D_{11(i+1,j,k)}\frac{u_{(i+1,j,k)} - u_{(i,j,k)}}{2} \\ & - D_{11(i-1,j,k)}\frac{u_{(i+1,j,k)} - u_{(i,j,k)}}{2}\Bigg) \end{aligned} \tag{3.10}$$

$$\begin{aligned} \partial_y\left(D_{12}\left(\partial_x u\right)\right) = \frac{1}{2}\Bigg(& D_{12(i,j+1,k)}\frac{u_{(i+1,j+1,k)} - u_{(i-1,j+1,k)}}{2} \\ & - D_{12(i,j-1,k)}\frac{u_{(i+1,j-1,k)} - u_{(i-1,j-1,k)}}{2}\Bigg) \end{aligned} \tag{3.11}$$

The other terms are written in the same way [43], and are combined to a pixel-location dependent 3×3 or $3 \times 3 \times 3$ convolution stencil. Non-negative discretization makes the modification that stencil elements remain positive for various gray values. Rotation invariant anisotropic diffusion is important with curved like structures such as the mandibular canal. Weickert [42] showed that larger stencils than 3×3 (2D) are needed to fix the number of degrees of freedom to allow rotation invariance. This is achieved by implementing the equations 3.8 and 3.9, with Scharr's rotational invariant 3×3 filters for the image derivatives ∂_x and ∂_y, resulting in an rotational invariant implicit 5×5 stencil.

3.4 Optimized Scheme

Another way to write the divergence operator using the product rule [44] is:

$$\nabla \cdot (\mathbf{D}\nabla u) = div(\mathbf{D})\nabla u + trace(\mathbf{D}(\nabla\nabla^T u)) \tag{3.12}$$

We obtain for the divergence part of the equation:

$$\begin{aligned} div(\mathbf{D})\nabla u =& (\partial_x u)(\partial_x D_{11} + \partial_y D_{12} + \partial_z D_{13}) \\ &+ (\partial_y u)(\partial_x D_{12} + \partial_y D_{22} + \partial_z D_{23}) \\ &+ (\partial_z u)(\partial_x D_{13} + \partial_y D_{23} + \partial_z D_{33}) \end{aligned} \tag{3.13}$$

We write the Hessian part of the equation as:

$$\begin{aligned} trace(\mathbf{D}(\nabla\nabla^T u)) = (\partial_{xx} u)D_{11} + (\partial_{yy} u)D_{22} + (\partial_{zz} u)D_{33} \\ +2(\partial_{xy} u)D_{12} + 2(\partial_{xz} u)D_{13} + 2(\partial_{yz} u)D_{23} \end{aligned} \tag{3.14}$$

Equation 3.13 is discretized using 3 × 3 × 3 derivative kernels, and the Hessian of equation 3.14 with a 5 × 5 × 5 second derivative kernel. In 2D the spatial kernels can be written as:

$$M_{xx} = \begin{bmatrix} p_1 & p_2 & p_3 & p_2 & p_1 \\ p_4 & p_5 & p_6 & p_5 & p_4 \\ -p_7 & -p_8 & -p_9 & -p_8 & -p_7 \\ p_4 & p_5 & p_6 & p_5 & p_4 \\ p_1 & p_2 & p_3 & p_2 & p_1 \end{bmatrix} \tag{3.15}$$

$$M_{xy} = \begin{bmatrix} p_{10} & p_{11} & 0 & -p_{11} & -p_{10} \\ p_{11} & p_{12} & 0 & -p_{12} & -p_{11} \\ 0 & 0 & 0 & 0 & 0 \\ -p_{11} & -p_{12} & 0 & p_{12} & p_{11} \\ -p_{10} & -p_{11} & 0 & p_{11} & p_{10} \end{bmatrix} \tag{3.16}$$

$$M_x = \begin{bmatrix} p_{13} & p_{14} & p_{13} \\ 0 & 0 & 0 \\ -p_{13} & -p_{14} & -p_{13} \end{bmatrix} \tag{3.17}$$

The kernel values $\mathbf{p} = [p_1, p_2 ..., p_{14}]$ can be found analytically or by numerical optimization. We choose numerical optimization, because it can optimize the whole process, while analytical derivation is only feasible for separate parts of the process, with simplifications such as ignoring numerical round of effects. We optimize the diffusion kernel using the following cost function:

$$e = \arg\min_{\mathbf{p}} (e_f(\mathbf{p}) + \beta e_g(\mathbf{p})) \tag{3.18}$$

This function finds a balance between the edge orientation invariant filtering performance e_f, and isotropic diffusion performance e_g, with weight constant β. With the first term e_f we want to find the best edge enhancement for edges with several orientations and spatial frequencies. Therefore we use the difference between an image with circles of varying spatial frequencies without noise I, and an image with Gaussian noise added I_{noise}, which is diffusion filtered (see figure 3.3). With $F(I_{noise}, \mathbf{p})$ the diffusion filtering of the image with noise using kernel values $\mathbf{p}$:

$$e_f(\mathbf{p}) = \sum_{\mathbf{x}} |F(I_{noise}, \mathbf{p}) - I|, \;\; with \;\; I = sin(x^2 + y^2) \tag{3.19}$$

With the second term e_g we want to achieve Gaussian like diffusion in uniform regions. We use an image I_{point} which is zero except the center pixel equal to one. The term e_g is set to the difference between the isotropic noise filtered image I_{point} and a least squares fitted Gaussian kernel with sigma a. We set both diffusion tensor eigenvalues to one, corresponding to a uniform region.

$$e_g(\mathbf{p}) = \arg\min_a \sum_{\mathbf{x}} \left(F(I_{point}, \mathbf{p}) - \frac{1}{\pi\sqrt{a}} \exp\left(-|x|^2/a\right) \right)^2 \tag{3.20}$$

We use the Matlab Nelder-Mead Simplex minimizer [45] because it is robust against local minima. Also a quasi-Newton minimizer is used [46], because the minimizer has a high convergence speed. We use 10 iterations of the Simplex Method followed by minimizing until convergence with the quasi Newton optimizer. This is done iteratively until the simplex method also converges. Parameters used for the circle image are, size 255×255, $\tau = 0.1$, iterations 5, $\sigma = 1$, $\rho = 10$, CED eigenvalues, Gaussian noise variance 0.1, and x and y coordinates in the range $[-10, 10]$. The parameters of I_{point} are image size 51×51 and 5 iterations, constant $\beta = 200$. The computed kernel values $\mathbf{p}$ are:

0.008	0.049	0.032	0.038	0.111	0.448	0.081
0.334	0.937	0.001	0.028	0.194	0.006	0.948

It is important to note that the scheme is optimized for rotational invariance, but that the derivative kernels are not rotational invariant, for instance M_x approximates a central difference instead of a Scharr like kernel.

In 3D the approach is the same; a spherical function in an image volume is used, with 33 instead of 14 unknown kernel variables. The optimized kernels are available in our open source diffusion toolbox[2]

3.5 Evaluation

We evaluate the properties of the standard, rotation and optimized diffusion scheme with respect to three image based criteria. The first is noise removal in uniform regions, the second preservation and enhancement of image edges independent of rotation and size. The final test is the combined filtering performance on a real CBCT data set.

In this first test we look at noise smoothing in uniform regions. To do this we use the image I_{point} introduced in the optimization section, with the same filtering parameters and 100 iterations.

[2] The open source diffusion code, http://www.mathworks.com/matlabcentral/fileexchange/25449

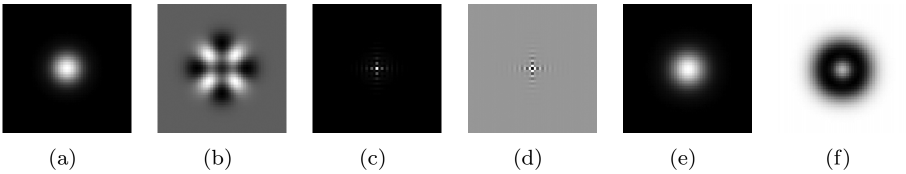


Figure 3.2 Uniform diffusion of a pixel with standard discretization (a), rotation invariant (c), optimized scheme (e). Sub figures (b), (d) and (f) show the difference between the image result and least squares fitted 2D Gaussian function. The values in (b) are in the order of $1 \cdot 10^{-5}$, (d) in the order of $1 \cdot 10^{-2}$ and (f) in the order of $1 \cdot 10^{-4}$.

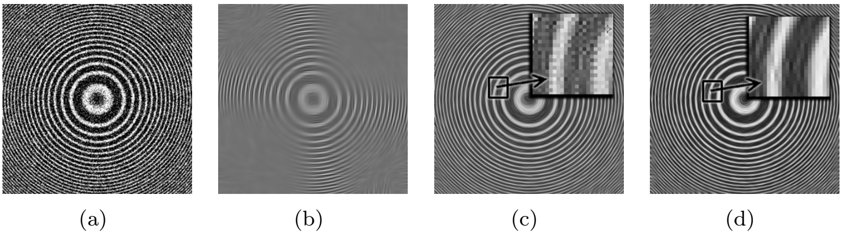


Figure 3.3 The sub figures show the test image (a), after diffusion with the standard scheme (b), with the rotational invariant scheme (c), and the optimized scheme (d).

Figure 3.2 shows the image results and difference between a least squares fitted Gaussian 2D function and the diffusion result. Ideal uniform diffusion is equal to Gaussian filtering, thus the standard diffusion and the optimized scheme perform well. The rotation invariant result does not look like a Gaussian. This is because the scheme is based on Sobel like derivative kernels, which do not use the local pixel value but only the neighboring values.

In the second test we look at rotation invariant edge enhancement, using the circle image with Gaussian noise I_{noise}, the same parameters as in the optimization section and 100 iterations. Figure 3.3 shows that only the rotational invariant and optimized scheme are edge orientation independent. The rotational invariant scheme suffers from checkerboard artifacts due to the Scharr derivative kernels which only use neighbor pixels and not the current pixel.

The final test is performed on 8 CBCT preprocessed human-head data sets of $400 \times 400 \times 551$ voxels. The preprocessing consist of clustering the data sets in to three intensity classes background, tissue and bone, using bias field corrected fuzzy clustering [47], which is robust to streak artifacts. The resulting image data serves as ground truth for the edges. The edges are detected by applying a threshold on the gradient magnitude. Uniform regions are defined as the voxels which are at least six

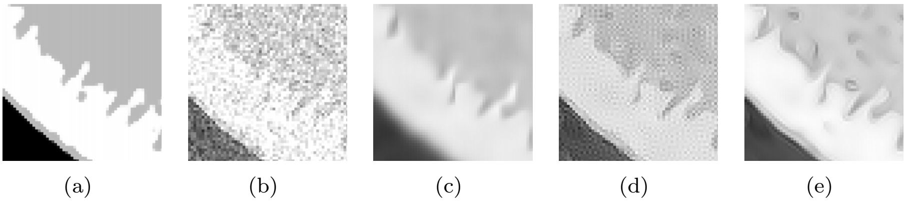


Figure 3.4 Small part of HDCS filtered bone structure, ground truth (a), Gaussian noise added (b), standard scheme (c), rotation invariant (d) and optimized scheme (e).

Table 3.1 This table shows the amount of high spatial frequencies after diffusion filtering. Therefore the diffusion filtering results are Gaussian smoothed, and sum of squared differences between Gaussian smoothed original filtering results are shown. Results of standard and optimized diffusion scheme.

	Edge			Uniform region		
data set	unfiltered	optimized	standard	unfiltered	optimized	standard
1	$6.2 \cdot 10^4$	$2.0 \cdot 10^4$	$1.1 \cdot 10^3$	$4.6 \cdot 10^4$	$1.6 \cdot 10^3$	7.6
2	$6.6 \cdot 10^4$	$2.1 \cdot 10^4$	$1.5 \cdot 10^3$	$4.7 \cdot 10^4$	$1.2 \cdot 10^3$	6.7
3	$6.4 \cdot 10^4$	$2.1 \cdot 10^4$	$1.2 \cdot 10^3$	$4.6 \cdot 10^4$	$1.4 \cdot 10^3$	6.5
4	$6.5 \cdot 10^4$	$2.0 \cdot 10^4$	$1.5 \cdot 10^3$	$4.7 \cdot 10^4$	$1.2 \cdot 10^3$	6.2
5	$6.9 \cdot 10^4$	$2.3 \cdot 10^4$	$1.3 \cdot 10^3$	$4.6 \cdot 10^4$	$1.5 \cdot 10^3$	6.9
6	$6.7 \cdot 10^4$	$2.6 \cdot 10^4$	$2.0 \cdot 10^3$	$4.5 \cdot 10^4$	$2.2 \cdot 10^3$	9.9
7	$7.0 \cdot 10^4$	$2.2 \cdot 10^4$	$1.5 \cdot 10^3$	$4.7 \cdot 10^4$	$1.3 \cdot 10^3$	6.8
8	$6.4 \cdot 10^4$	$2.2 \cdot 10^4$	$1.3 \cdot 10^3$	$4.6 \cdot 10^4$	$1.4 \cdot 10^3$	6.7

voxels away from an edge. Finally Gaussian noise of variance 0.01 is added to the image data. The image data is filtered with the standard and the optimized scheme using HDCS eigenvalues, with parameters $\sigma = 0.5$, $\rho = 2$, $\tau = 0.15$, HDCS parameters $\lambda_e = 30$, $\lambda_h = 30$, $\lambda_c = 15$ and 26 iterations, see figure 3.4.

We compare the performance between the standard and the optimized diffusion scheme. The summed squared pixel distance between Gaussian low pass filtered and original diffusion results, is used as a performance value. A steep edge contains high frequencies which will be removed by the low pass filter, resulting in a large pixel distance. In uniform regions, high frequency noise will also be removed, thus a large pixel distance is a sign of noise which is not removed by the diffusion filtering. We calculate the amount of high frequencies for the edge pixels and for the uniform regions after filtering. The results are shown in table 3.1. The standard scheme gives the best smoothing performance for uniform regions, with a 200 times smaller distance compared to the optimized scheme. This is because the optimized scheme preserved the edges of some random noise structures. The same noise structures are also visible in the rotation invariant scheme in image 3.4. In the HDCS eigenvalues there is a

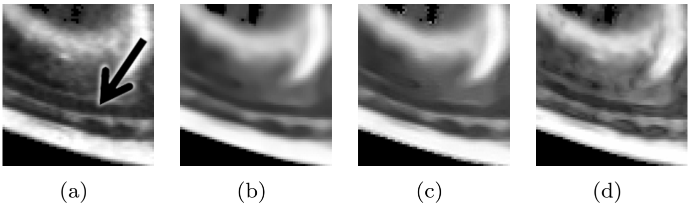


Figure 3.5 Small part of HDCS filtered scan (*a*), mandibular canal (*arrow*) , standard scheme (*b*), rotation invariant (*c*) and optimized scheme (*d*). The optimized scheme better preserves the original edges and image structure.

threshold value λ_h to separate between noise and a image structures. But in this case the signal to noise ratio is too low to allow a good separation between noise and real image structures. Also on the real object edges the optimized scheme gives the highest pixel distance. This can be due to remaining noise on the edges or due to a steeper image edge than with the standard scheme. Figure 3.4 shows that this high value is not caused by noise, but due to the more pronounced image edges.

Finally we show the filtering results of all schemes on an CBCT scan which is transformed to make the jaw flat, see figure 3.5. The optimized scheme gives the best enhancement and preservation of the mandibular canal.

3.6 Conclusion

The introduced 2D/3D anisotropic diffusion scheme shows better edge enhancement in our synthetic and CBCT data, compared to the standard and the rotation invariant scheme. Filtering is Gaussian in uniform image regions without checkerboard artifacts. The results show that the better edge preservation also causes noise structures above a certain threshold to be preserved; this will be a problem with CBCT scans containing pronounced streak artifacts. The cause of the problem is not the optimal scheme, but has to be solved by a better separation between noise edges and real edges in the diffusion tensor construction part.

4
Nerve Tracking

4.1 Introduction

Information about the location of the mandibular canal is essential in case of dental surgery [48], because entering the canal space during implant placement can damage the inferior alveolar nerve or blood vessels. Cone beam CT (CBCT) is becoming an increasingly utilized imaging modality in dental examinations, with a factor of ten lower dose than conventional CT [36]. The goal of our research is to find an automatic method which can segment the mandibular canal in CBCT, to allow an accurate safety margin during surgery planning.

Our data set of CBCT scans are from the department of oral and maxillofacial surgery, Radboud university Nijmegen medical centre, the Netherlands. The data set contains images of patients which have lost teeth or/and have metal implants, see figure 4.2. After loss of natural teeth, rapid bone deterioration starts because of the lack of physical stimulation of the jawbone in that area.

The literature describes several methods to segment, automatically or user assisted, the mandibular canal in CT. Methods based on fast marching shortest path tracing [28], region growing [32] and segmentation methods based on image gradients [29]. We have tested these methods on our CBCT data, however, all fail because of low contrast between canal and surrounding tissue, high noise and missing edges (figure 4.1).

Better suitable for CBCT are the texture and shape based models, such as the active appearance model (AAM) of Rueda et al. [30]. AAM models are constructed from corresponding points between data sets. The AAM requires that the location of a single corresponding point must Gaussian distribution between the subjects. Our data consist of males and females, dentate and edentate people, thus there is no

Gaussian distribution of surface shapes.

In 1981 Lucas-Kanade (LK) introduced an image alignment technique based on minimizing the sum of squared errors between a template and an image. This method was used by others to do template tracking of cars in traffic movies [24].

In this thesis we use a modified LK template tracking method to track the mandibular canal in our CBCT scan. The main idea is to sample the volume scan in slices perpendicular to the mandibular canal. The user then selects a window around the canal in one slice and this is used as a template to find the canal in the neighboring frames, until the whole length of the mandibular canal is traced.

The first section describes the tracking algorithm, the second section the re-slicing to movie frames of the CBCT scan. Followed by the results and finally the conclusions.

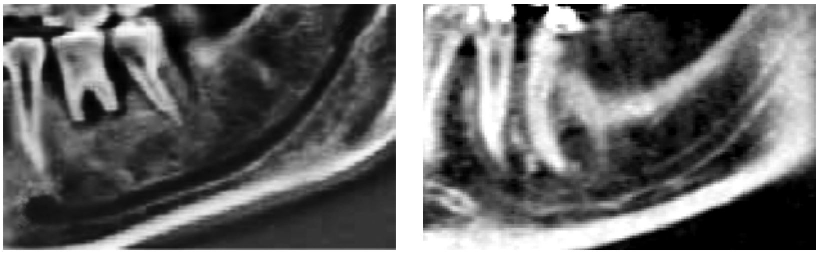


Figure 4.1 Mandibular canal visibility in a scan-slice, comparison of CT (left)and CBCT (right).

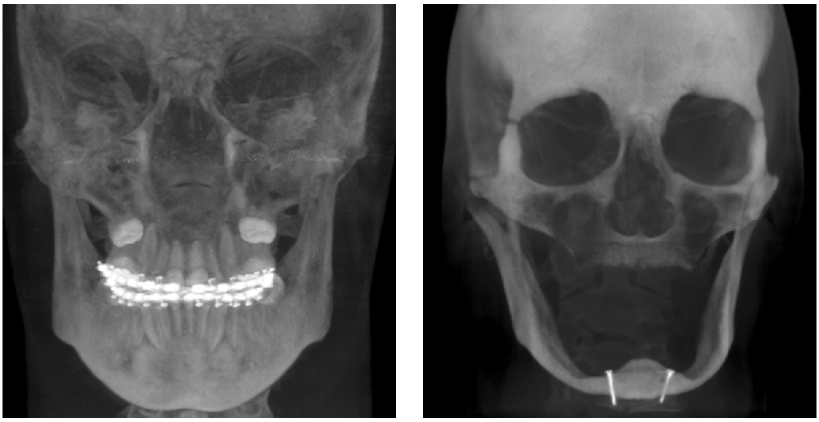


Figure 4.2 Volume rendering of patients in the data set. The figure shows a young patient with braces and an older patient which had surgery because of severe bone erosion.

4.2 LK Template Tracking

The LK algorithm minimizes the sum of squared differences (SSD) between the template image T and the image I, minimizing:

$$\sum_{\mathbf{x}}[I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x})]^2 \tag{4.1}$$

With $\mathbf{W}(\mathbf{x};\mathbf{p})$ the warping equation of the pixel coordinates $\mathbf{x} = (x, y)$ in the template. With warp parameters $\mathbf{p} = [p_1, p_2, .., p_n]$ with $n = 6$ incase of an affine warp. The LK algorithm assumes that a current estimate of $\mathbf{p}$ is known and iteratively solves for increments $\Delta\mathbf{p}$ of the parameters.

We will use the inverse additive LK algorithm of Hager and Belhumeur [49], in which the role of template and image are switched allowing more pre-calculations and less iterative calculations. The SSD equation (4.1) is linearized with a first order Taylor expansion:

$$\sum_{\mathbf{x}}\left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) + \nabla I\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\Delta\mathbf{p} - T(\mathbf{x})\right]^2 \tag{4.2}$$

The inverse algorithm assumes that the initial warp parameters $\mathbf{p}$ are close to the optimal warp parameters. Therefore $I(\mathbf{W}(\mathbf{x};\mathbf{p})) \approx T(\mathbf{x})$, with $\nabla I\frac{\partial\mathbf{W}}{\partial\mathbf{p}} \approx \nabla T$. Inverting $\frac{\partial\mathbf{W}}{\partial\mathbf{p}}$ and changing $\Delta\mathbf{p}$ into $-\Delta\mathbf{p}$ completes the change of role of template and image. This gives the goal of the additive inverse LK algorithm:

$$\sum_{\mathbf{x}}\left[T(\mathbf{x}) + \nabla T\left(\frac{\partial\mathbf{W}}{\partial\mathbf{x}}\right)^{-1}\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\Delta\mathbf{p} - I(\mathbf{W}(\mathbf{x};\mathbf{p})))\right]^2 \tag{4.3}$$

In case of affine warping we can write the product of the two Jacobian's as:

$$\left(\frac{\partial\mathbf{W}}{\partial\mathbf{x}}\right)^{-1}\frac{\partial\mathbf{W}}{\partial\mathbf{p}} = \Gamma(\mathbf{x})\Sigma(\mathbf{p}) \tag{4.4}$$

$$\Gamma(\mathbf{x}) = \begin{pmatrix} x & 0 & y & 0 & 1 & 0 \\ 0 & x & 0 & y & 0 & 1 \end{pmatrix} \tag{4.5}$$

With $\Sigma(\mathbf{p})$ an affine warp matrix controlled by parameters $\mathbf{p}$.

The equations needed to calculate the inverse LK tracking are given below, their derivation can be found in Baker et al. [24].

The modified Hessian:

$$H_* = \sum_{\mathbf{x}}\left[\nabla T\Gamma(\mathbf{x})\right]^T\left[\nabla T\Gamma(\mathbf{x}\right] \tag{4.6}$$

The iterative parameter update $\Delta\mathbf{p}$:

$$\begin{gathered}\Delta\mathbf{p} = \Sigma(\mathbf{p})^{-1}\Delta\mathbf{p}_* \\ \Delta\mathbf{p}_* = H_*^{-1}\textstyle\sum_{\mathbf{x}}\left[\nabla T\Gamma(\mathbf{x})\right]^T\left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x})\right]\end{gathered} \tag{4.7}$$

The inverse of the affine warping $\Sigma(\mathbf{p})$:

$$\Sigma(\mathbf{p})^{-1} = \begin{pmatrix} 1{+}p_1 & p_3 & 0 & 0 & 0 & 0 \\ p_2 & 1{+}p_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1{+}p_1 & p_3 & 0 & 0 \\ 0 & 0 & p_2 & 1{+}p_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1{+}p_1 & p_3 \\ 0 & 0 & 0 & 0 & p_2 & 1{+}p_4 \end{pmatrix} \tag{4.8}$$

The inverse LK method is implemented as [24]:

```
Pre-compute
(1)  Evaluate the gradient $\nabla T$ of the template $T(\mathbf{x})$
(2)  Evaluate $\Gamma(\mathbf{x})$
(3)  Compute the six steepest descent images $\nabla T\Gamma(\mathbf{x})$
(4)  Compute the modified Hessian $H_*$
Iterate
(5)  Warp $I$ with $W(\mathbf{x};\mathbf{p})$ to compute $I(W(\mathbf{x};\mathbf{p}))$
(6)  Compute the error image $I(W(\mathbf{x};\mathbf{p})) - T(\mathbf{x})$
(7)  compute $\sum_{\mathbf{x}}[\nabla T\Gamma(\mathbf{x})]^T[I(W(\mathbf{x};\mathbf{p})) - T(\mathbf{x})]$
(8)  Compute $\Delta\mathbf{p}_*$
(9)  Compute $\Sigma(\mathbf{p})^{-1}$ and update
     $W(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{p} - \Sigma(\mathbf{p})^{-1}\Delta\mathbf{p}_*$
until $\|\Delta\mathbf{p}\| \leq \epsilon$
```

The gradient of the template ∇T is calculated by convolving a large template with the derivatives of a Gaussian kernel, and crop to the wanted template size afterwards. For noise robustness we use a large sigma σ_l to calculated the template gradients for the first few iterations n_s, and then a smaller sigma σ_s.

Schreiber [50] added weights to the equations to have control over the influence of every pixel in the template, giving template pixels which contain background less influence. He calculates the error between template and pixels of LK tracking result, and uses the median of the error distribution to measure reliability. Adding the weights $\omega(\mathbf{x})$ changes equation 4.6 and 4.7 to:

$$H_* = \sum_{\mathbf{x}} \omega(\mathbf{x}) \left[\nabla T\Gamma(\mathbf{x})\right]^T \left[\nabla T\Gamma(\mathbf{x})\right] \tag{4.9}$$

$$\Delta\mathbf{p}_* = H_*^{-1} \sum_{\mathbf{x}} \omega(\mathbf{x}) \left[\nabla T\Gamma(\mathbf{x})\right]^T \left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x})\right] \tag{4.10}$$

Because the robust estimator of Schreiber did not classify our pixels correctly as background or foreground, we set the weight matrix $\omega(\mathbf{x}) \in [0,1]$ to,

$$\omega(\mathbf{x}) = 1 - \sqrt{\frac{\|\mathbf{x}\|}{d}} \tag{4.11}$$

assuming a circular mandibular channel, and d the distance from center to the farthest border location of our template.

The appearance of an object changes in time. Thus the template needs to be updated during tracking with the found object images R_n to keep representing the followed object. A disadvantage of template updating is that small errors in the found position parameters $\mathbf{p}$ will cause the template to drift away from the original object. Matthews et al. [51] solution was to align the updated template every tracking iteration with the original template, in order to stay on track. We have investigated this approach but the appearance of the mandibular changes too much to allow alinement of the first template onto the updated templates. Instead we assume that the template center intensities vary less than edge pixels, and use this information in the template update:

$$T_{n+1}(\mathbf{x}) = T_n(\mathbf{x}) \left(1 - \frac{\|\mathbf{x}\|^2}{d^2} \right) + \frac{\|\mathbf{x}\|^2}{d^2} R_n(\mathbf{x}) \tag{4.12}$$

4.3 Sampling of the Mandible

Experts in dentistry are used to diagnose panoramic X-rays of the full mouth, also called orthopantomograms (OPG). For this reason most modern CT dental software support flattening of the jaw in the CT-data through re-sampling, creating a volume with a maximum intensity projection (MIP) which looks like a panoramic X-ray. We created an automatic re-sampling method, consisting of the following steps (figure 4.4):

(1) Apply an intensity threshold to keep only the bone
(2) Make a depth image, with the distances from the bottom to the nearest bone voxels above the bottom.
(3) Calculate the median (like) distance of the front part of the depth image
(4) Gaussian filter the depth image. Apply an threshold to get the depth values above the median depth value of step 3. This will create an image with only the jawline
(5) Find the center of the jawline for every image row
(6) Smooth, make the sampling uniform and extrapolate the center line using a spline function
(7) Define the outer and inner jaw line using a fixed distance from the center line.
(8) Use the outer and inner centerline as x, y re-sampling coordinates.

Recently a paper was published by Akhoondali et al. [52] which uses a similar approach for automatic panoramic re-slicing.

The flat jaw volume allows the user to select a region of the mandibular canal as template, see 4.3. The appearance of the canal changes from a circle to an ellipsoid in the slices because the canal is still curved in the z-direction. Therefore we do not use the panoramic slices as movie frames, but sample slices perpendicular to the canal in the original volume during tracking. Every iteration, a fixed step is done in the direction of the channel.

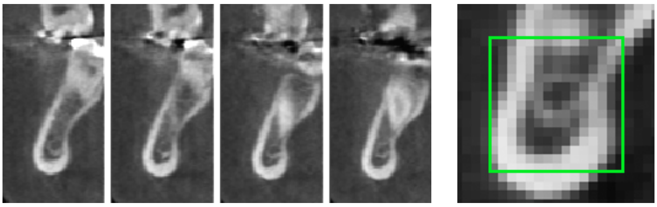


Figure 4.3 Sampled slices through the mandible in CBCT data. Also shown is the LK template. All pixels outside the green box are not part of the template, but used to calculate the image derivatives.

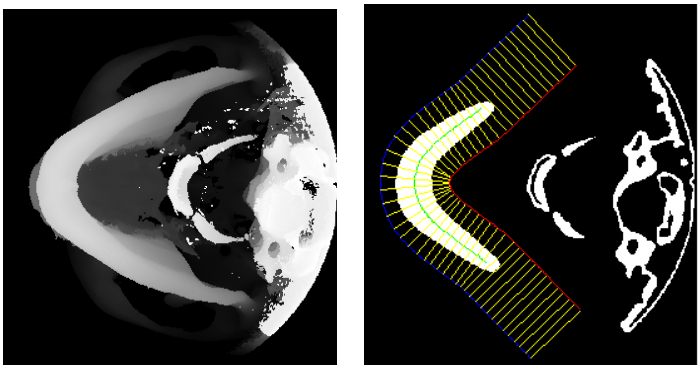


Figure 4.4 Jaw depth map and re-sampling of the scan.

The needed normal and direction of the canal is calculated using current and a number n_p of previous canal positions . It is also possible to find the normal by rotating the sampled slice until the difference between template and sampled slice is minimal. But that approach is less robust to image noise. The canal normal calculated from the previous positions still allows in plane rotation of the frame. Thus we penalize the distance between new and old corner points of the sampled slice to fix the in-plane rotation.

4.4 Results

Our data set consist of seven CBCT scans from the department of oral and maxillofacial surgery, Radboud university Nijmegen medical centre, with a voxel resolution of $0.4mm$. We re-slice the data sets to a panoramic volume and manually segmented the narrow part of the mandibular canal using pixel selection in a slice view.

Secondly we selected starting points in the center of the mandibular canals for every scan, using the panoramic slice view. We choose a fixed template size with a size of 14×14, and add a boundary of five pixels for reliable template derivatives.

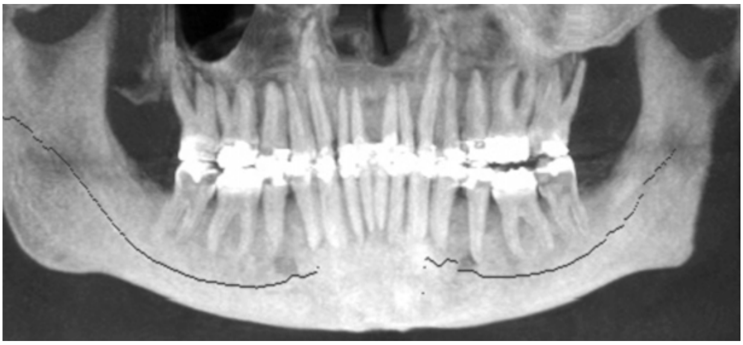


Figure 4.5 MIP with tracking result overlay.

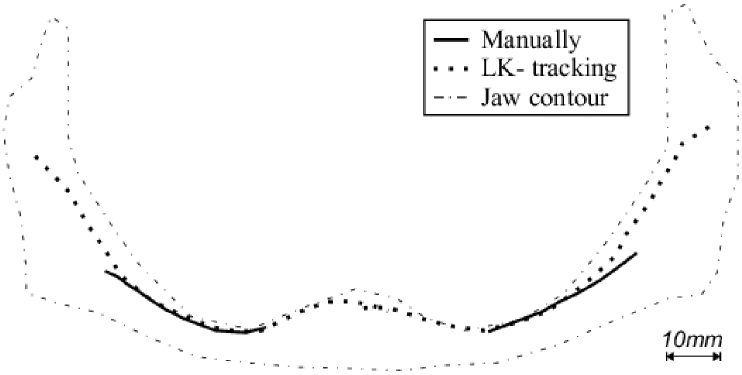


Figure 4.6 Difference between manual segmentation and tracking results. In this case only small implants can be used because the distance between contour and canal is small.

The tracing was done in the original CBCT scan volumes, using the image derivative parameters, $\sigma_l = 1.5$, $\sigma_s = 0.7$ and $n_s = 3$, for calculating the normal we use $n_p = 7$. These parameters where found experimentally. The noise and the small template size caused the affine tracking to be unreliable, thus we switch for the final results to only translation transformation.

To measure the performance, we use the distance between the points in the manual segmentation and the closest points in the tracking results. The distance between automatic and manual segmented canal segments of all seven scans is in the order of two pixels, 1.47 pixels $0.6mm$ for the data set shown in figure 4.6 . The maximum distance between manual segmentation and tracking result is often $3mm$ or more at the ends due to drift. The drift is caused by template updating, missing edges and molar teeth touching the channel.

4.5 Conclusions and Recommendations

We propose a method for automatic mandibular canal segmentation for Cone Beam CT scans, based on Lucas Kanade template tracking. The pixel based methods in the literature are not usable on these CBCT scans because of the high noise and low contrast in the scan data.

Template tracking sampling slices perpendicular to the mandibular canal, give accurate localization results for the center part of the canal. The method fails at the end points of the mandibular canal, because it suffers from drift. This drift in the template update is caused by invisible edges, touching structures and lag in position and normal because of the use of previous points to calculate them.

Possible solutions for these drift problems are, edge enhancement and denoising by using coherence enhancing filtering, see chapter 3 , using more user-selected points to guide the tracking, or use some sort of template backtracking.

But because of missing edges, there is a need to introduce a priori information, such as a probability atlas of the mandibular canal position. Therefore there is a need to establish some reference framework, based on stable features of the mandible, such as the bottom part which suffers less from bone erosion.

5
Active Shape Model

5.1 Introduction

This chapter describes the active shape models (ASM) [25] and active appearance models (AAM) [26], of Cootes and Taylor. We extend these models to 3D, and introduce enhancements for more accurate segmentation results. The next section describes some background on segmentation, followed by a section about learning shapes with an ASM. The fourth section adds intensity information to the ASM. The fifth section describes the segmentation process of an ASM model. The sixth and seventh sections describe the AAM which not only models the object shape as in an ASM but also models the texture. The eighth section describes the extension to 3D, and also other extensions which improve the accuracy and robustness of the models. The ninth section gives test results of the extended models on a set of 2D hand photos and 3D mandibles. With finally the conclusions.

5.2 Background

An image consists of a matrix of values which represent appearance properties of the captured objects. Most medical images consist of piecewise uniform gray level regions representing the captured objects. The objects in the images are separated by image-edges, and the images are distorted with image noise. An object is easy to segment if there is a high contrast between object and background, or when the object has sharp image edges. In that case basic segmentation tools as a gray value threshold, clustering or region-growing like algorithms can be used.

But in most realistic cases an object consist of multiple sub-objects made of several

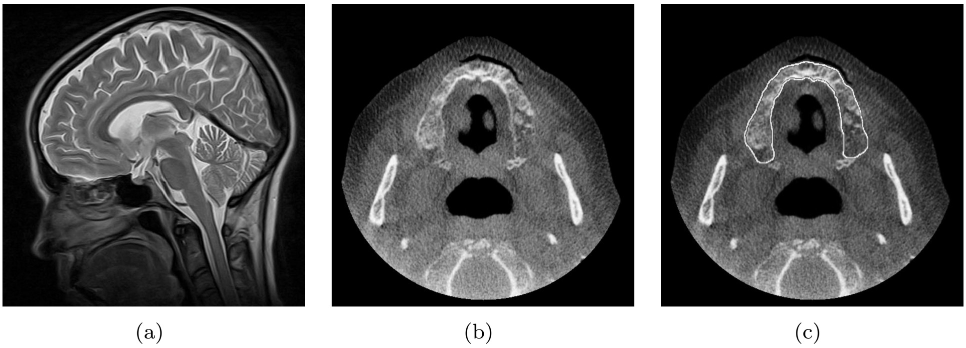


Figure 5.1 MRI image (a). The gray matter in this image is easy to segment because it has sharp edges and uniform regions. CBCT image of the maxilla (b). The maxilla has missing edges and no uniform gray value. Manual annotation of the maxilla, based on gray values but also on a priori knowledge of the expert about the usual shape of a maxilla (c).

materials, each showing their own gray value. For instance a human bone contains spongy bone, periosteum and compact bone. All those materials have a different Hounsfield unit [53] and texture thus different gray values in CT data. It is also possible that there is no edge between two objects because they contain the same material or different materials but showing the same gray value. Then there are also problems like illumination differences, partial volume effect and reconstruction artifacts.

Priori knowledge has to be included in the segmentation algorithm, to segment difficult image objects such as the maxilla in CBCT data (figure 5.1). Most biomedical objects are smooth, thus many algorithms for general segmentation purposes use a smoothness constraint on the boundary of the segmentation. Commonly used methods are active contours (snakes) [54] [55] and level-set methods [56]. These methods give good result in case of missing edges and noisy regions. It is also possible to include appearance information such as gray level and texture to the a priori knowledge. Examples are template-matching or atlas registration algorithms [57]. A disadvantage of these methods is that a simple smooth constraint is only a small part of the known shape knowledge, and a registration atlas only contains one example of the possible shape. Therefore we will focus in this chapter on the active shape models (ASM) [25] and active appearance models (AAM) [26], of Cootes and Taylor. These segmentation algorithms are trained using example images with manually drawn contours. They learn the mean object shape and variances of the object shape using principal component analysis (PCA) [58]. This shape model is used during segmentation, to constrain the segmentation. Only segmentation shapes are allowed which are possible with a Gaussian distribution assumption of the training data. Thus if the height of a contour varies between training photos, it will be allowed to take a value between two or three times the standard deviation of the training data during segmentation.

5.3 Shape Model Construction

The first step to build an ASM model of a simple object is to draw a contour line around the object in al training images. The next step is to create n corresponding landmark points on the object boundaries, this is commonly done by manual annotation. Then there is often a shape alignment step, which removes translation, rotation and size differences between the training images. The aligned (x, y) positions of the landmark points in one data set are then grouped into a column vector:

$$\mathbf{x}_1 = (x_1, x_2, .., x_n, y_1, y_2, .., y_n)^T \tag{5.1}$$

The position vectors of all s training contours are then grouped into one matrix:

$$X = (\mathbf{x}_1, \mathbf{x}_2, .., \mathbf{x}_s) \tag{5.2}$$

PCA is then applied to the shape vectors in X. This can be done in two ways. The first method. Compute the mean shape:

$$\bar{\mathbf{x}} = \frac{1}{s}\sum_{i=1}^{s}\mathbf{x}_i \tag{5.3}$$

Construct the covariance matrix:

$$\mathbf{S} = \frac{1}{s-1}\sum_{i=1}^{s}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \tag{5.4}$$

Calculate the eigenvalues: $\lambda = (\lambda_1, \lambda_2, ..\lambda_n)$ and corresponding eigenvectors $\mathbf{\Phi} = (\mathbf{v}_1, \mathbf{v}_2, ..\mathbf{v}_n)$. This equation applies:

$$\mathbf{S}\mathbf{v}_i = \lambda_i \mathbf{v}_i \tag{5.5}$$

The second method is to use singular value decomposition. First normalize the vectors:

$$\dot{\mathbf{x}}_i = \frac{1}{\sqrt{n-1}}(\mathbf{x}_i - \bar{\mathbf{x}}) \tag{5.6}$$

Combine the vectors in a matrix:

$$\dot{X} = (\dot{\mathbf{x}}_1, \dot{\mathbf{x}}_2, .., \dot{\mathbf{x}}_n) \tag{5.7}$$

Singular value decomposition of the matrix $\dot{X}$:

$$\dot{X} = \mathbf{\Phi}_s \Sigma V^T \tag{5.8}$$

Σ is an $m \times n$ diagonal matrix with nonnegative real numbers on the diagonal and V is an $n \times n$ unitary matrix. The eigenvalues are $\lambda = \left(\Sigma^2_{(1,1)}, \Sigma^2_{(2,2)}, .., \Sigma^2_{(n,n)}\right)$ and Φ_s a unitary matrix containing the eigenvectors. The number of training data sets is often (very) small in comparison to the number of landmarks. Leading to a singular correlation matrix, and over fitting of the training data. To reduce over fitting effects the next step is cropping the number of eigenvalues to t, keeping 90% or 99.5% of the variance in the training data. This is done by removing the lowest eigenvalues and corresponding eigenvectors.

Now we have obtained a shape model, which can convert any example contour to model parameters. Reducing a large number of contour landmarks $\mathbf{x}$ to a small number of parameters $\mathbf{b}_s$:

$$\mathbf{b}_s = \Phi_s^T(\mathbf{x} - \bar{\mathbf{x}}) \tag{5.9}$$

These model parameters can be converted back to contour coordinates by:

$$\tilde{\mathbf{x}} = \bar{\mathbf{x}} + \Phi_s \mathbf{b}_s \tag{5.10}$$

The new contour coordinates $\tilde{\mathbf{x}}$ is an estimate of the original contour $\mathbf{x}$. We constrain b to the range $\pm m\sqrt{\lambda_i}$ with m between 2 and 3. This constraint will allow only shapes estimates which are possible within 2 or 3 standard deviations of the distribution of shapes in the training data.

5.4 Appearance Model Construction

The above described shape model does not generate segmentation contours, but only serves as a shape-constraint on an existing contour. To segment an image we also need a method to produce estimations of the object boundary in the image. Therefore we need to include an appearance part into our ASM.

The method of Cootes and Taylor [25], starts with calculation of the normal vectors of all the contour landmarks. Then k evenly spaced points are sampled along the normal both in negative and positive direction. This results in a, gray profile $\mathbf{s}$ with length $2k + 1$ for every landmark point in a data set. Then the first order derivative is calculated for the gray profile:

$$\mathbf{g}(i) = \mathbf{s}(i + 2) - \mathbf{s}(i) \text{ with } i = 1, 2.., 2k - 1 \tag{5.11}$$

This is normalized by:

$$\mathbf{g}(i) = \frac{1}{l}\mathbf{g}(i) \text{ with } l = \sum_{i=1}^{2k-1} \mathbf{g}(i) \tag{5.12}$$

A landmark has a gray profile $\mathbf{g}_1, \mathbf{g}_2, .., \mathbf{g}_n$ in all n data sets. The mean $\bar{\mathbf{g}}$ and covariance matrix S_g can be calculate for the gray profiles. This covariance matrix can be used to make a cost function based on the Mahalanobis distance:

$$f(\mathbf{g}_i) = (\mathbf{g}_i - \bar{\mathbf{g}})^T S_g^{-1}(\mathbf{g}_i - \bar{\mathbf{g}}) \tag{5.13}$$

This cost function represents the distance of a new profile $\mathbf{g}_i$ to the Gaussian distribution of the sampled profiles in the training data.

5.5 Active Shape Model Search

The ASM search process can be described by the following steps (see figure 5.2):

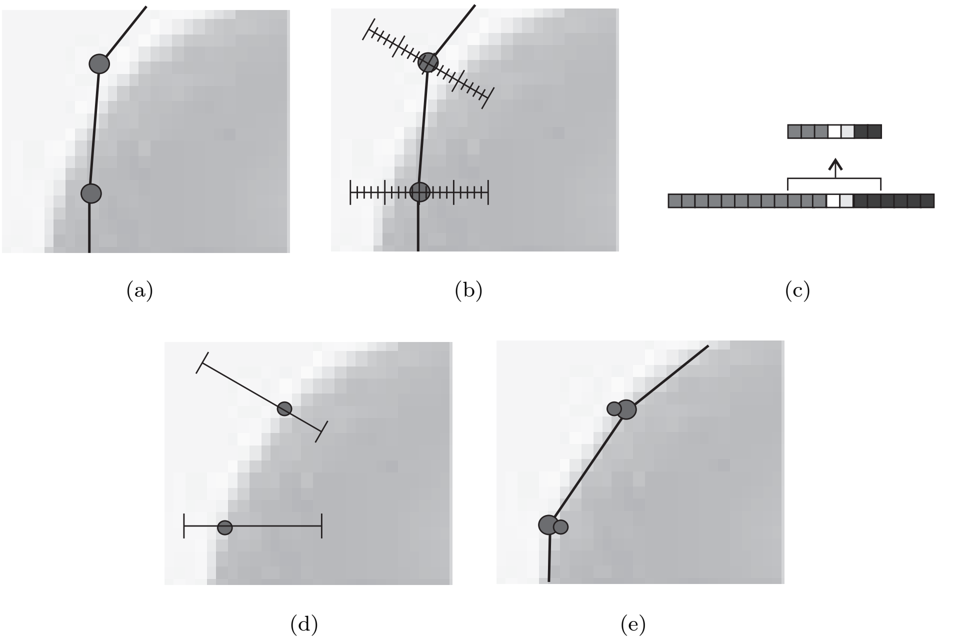
1. Position the mean object shape close to the object in the image
2. Sample long gray profiles on the contour normals
3. Search on the long gray profile for a line piece with length $2k + 1$, with the lowest Mahalanobis distance.
4. Move every point to the center of the line piece with the lowest Mahalanobis distance.
5. Convert the x, y positions into normalized coordinates $\mathbf{x}$, and convert them into model parameters $\mathbf{b}_s$, and limit the model parameters to the range $\pm m\sqrt{\lambda_i}$
6. Convert the model parameters $\mathbf{b}_s$ back to normalized contour positions $\mathbf{x}$, and convert them back to x, y positions.
7. Go to step 2, until the movement is smaller than a certain threshold or until the maximum number of iterations is reached.

This search method only works if the initial contour is close enough to the object contour to be segmented. Also it can get stuck on other structures due to local minima. To solve these problems, a multi-scale approach is used, in which two or more appearance-models are constructed for low to high image resolutions. First the ASM search is run on a low-resolution image until convergence, then the resolution is doubled and the next appearance model is used, and finally the original resolution is used. This will make the ASM search more robust and will speed up the convergence.

5.6 Active Appearance Model

An active appearance model (AAM) [26] is an extension to an active shape model. Instead of only learning the shape variations also appearance is included in the PCA model. This is done with the following steps:

1. Construct an ASM of the training data
2. Warp the image data of all training sets to the mean object shape.
3. Re-sample the warped training data into column vectors.
4. Perform PCA on the warped image data.
5. Combine the shape and appearance models into one PCA model


(a) (b) (c)

(d) (e)

Figure 5.2 ASM search process. Initial landmark positions (a). Long search profiles sampled on contour normals (b). Search for line piece with smallest Mahalanobis distance to training profiles (c). Move landmark points to center of line piece with lowest Mahalanobis distance (d). Map the x,y points to model coordinates, constrain them to be within 3 standard deviations from the training shapes, and transform them back to x,y positions (e)

When the ASM of the training data is constructed also the mean contour shape $\bar{\mathbf{x}}$ is obtained. Warping the data sets to the mean shape $\bar{\mathbf{x}}$ is usually done by creating a Delaunay triangulation of the mean shape $\bar{\mathbf{x}}$, resulting in a list of vertices $n \times 2$ and a list of faces $m \times 3$ with vertex indices. The warping of a data set can then be done by using, image I_i as texture, the contour points $\mathbf{x}_i$ as texture vertices, and the mean shape contour points $\bar{\mathbf{x}}$ as position vertices. Then rasterisation algorithms such as available in openGL or directX can warp the image. Most rasterisation algorithms use barycentric coordinates to calculate the texture coordinates for the pixels inside the triangulated object. The texture coordinates can be used with linear or cubic interpolation to sample the texture image.

The first step before resampling is discretization of the mean-contour lines to pixels in an binary image, followed by a flood-fill. The pixel coordinates $L_x = (x_1, x_2, ..x_n)$ and $L_y = (y_1, y_2, ..y_n)$ of the object pixels are then shaped in to a location vector $L = (L_x^T, L_y^T)$. This location vector can be used to create column vectors $\mathbf{g}_{im}$ of training images which are warped to the mean shape. Because training images often have illumination differences, we normalize $\mathbf{g}_{im}$ by applying a scaling α, and offset β.

$$\mathbf{g} = \frac{(\mathbf{g}_{im} - \beta)}{\alpha} \tag{5.14}$$

We set α equal to the standard deviation and β to the mean of $\mathbf{g}_{im}$.

By applying PCA on the appearance data, we obtain:

$$\mathbf{b}_g = \Phi_a^T(\mathbf{g} - \bar{\mathbf{g}}) \tag{5.15}$$

with $\bar{\mathbf{g}}$ the mean appearance data and Φ_g the reduced eigenvectors. These model parameters $\mathbf{b}_g$ can be converted back to contour coordinates by:

$$\tilde{\mathbf{g}} = \bar{\mathbf{g}} + \Phi_a \mathbf{b}_g \tag{5.16}$$

Now we have obtained both model parameters $\mathbf{b}_s$ for the shape and $\mathbf{b}_g$ for the appearance. Shape and appearance are often correlated, thus we can combine them, and reduce the combined models with PCA. Combining is done by:

$$\mathbf{b} = \begin{pmatrix} W_s \mathbf{b}_s \\ \mathbf{b}_g \end{pmatrix} \tag{5.17}$$

With W_s a diagonal matrix which changes the weight of the shape model parameters. This is needed to make the effect on the RMS change of appearance of a small change in a shape parameter $\mathbf{b}_s(i)$ approximately equal to the effect of a small change in an appearance parameter $\mathbf{b}_g(i)$. To do this we displace each element of $\mathbf{b}_s$ from its optimum value in each training example and sample the image give the displaced shape. The RMS change in $\mathbf{g}$ per unit change in shape parameter $\mathbf{b}_s$ give the weight W_s.

After applying PCA on the model vector $\mathbf{b}$ as done on the shape and appearance data, we obtain:

$$\mathbf{c} = \Phi_{sa}^T(\mathbf{b} - \bar{\mathbf{b}}) \tag{5.18}$$

with $\bar{\mathbf{b}}$ the combined shape appearance parameters and Φ_{sa} the reduced eigenvectors. These model parameters $\mathbf{c}$ can be converted back to the combined shape appearance parameters $\mathbf{b}$ by:

$$\tilde{\mathbf{b}} = \bar{\mathbf{b}} + \Phi_{sa}\mathbf{c} \tag{5.19}$$

5.7 Active Appearance Model Search

The AAM model contains a PCA model of the appearance of the object. This means that if we have some initial model parameters $\mathbf{c}$, we can use them to synthesize a model image I_m using the learned information. If we now warp the image data of a test image with initial contour to the mean shape, we obtain an image I_i. The difference δI between the model image I_m and segmented image I_i gives a measure of correctness of the current contour. We can vary the model parameters to minimize the magnitude of the difference $\Delta = |\delta I|^2$. A good contour produces a segmentation image similar to the model image of that contour, resulting in a low intensity difference.

Thus we try to minimize the intensity difference by changing the contour shape and position. The shape is described by model parameters $\mathbf{c}$ and pose parameters $\mathbf{t}$ and $\mathbf{s}$. With $\mathbf{t} = t_x, t_y$ the translations in x and y direction, and $\mathbf{s} = s_x, s_y$ a combination of scaling and rotation. The parameters can be combined in AAM model parameters $\mathbf{p}$ with $\mathbf{p} = [\mathbf{c}||\mathbf{t}||\mathbf{s}]$. The scaling rotation parameters s_x, s_y are related to the contour scale s sand rotation r by:

$$s = \sqrt{s_x^2 + s_y^2} \quad \text{and} \quad r = 2\arctan\frac{\sqrt{s_x^2 + s_y^2} - s_x}{s_y} \tag{5.20}$$

Using finite differences to minimize the intensity difference Δ for a test image, will require a large number of image warps and matrix calculations. Instead we try to find a direct linear relationship between the difference vector δI and required model parameter update δp during training.

Let us say we have a model appearance vector $\mathbf{g}_m$, and an appearance vector $\mathbf{g}_o$ from an almost aligned segmentation contour. Then we can define a difference vector $\mathbf{r}$:

$$\mathbf{r}(\mathbf{p}) = (\mathbf{g}_m - \mathbf{g}_o) \tag{5.21}$$

We can describe the local neighborhood of $\mathbf{r}$ on the current parameter setting $\mathbf{p}$, by a first order Taylor expansion:

$$\mathbf{r}(\mathbf{p} + \delta\mathbf{p}) = \mathbf{r}(\mathbf{p}) + \frac{\partial\mathbf{r}}{\partial\mathbf{p}}\delta\mathbf{p} \tag{5.22}$$

The goal is to minimize the difference vector $\mathbf{r}$, thus after updating the parameter $\mathbf{p}$, we want $\mathbf{r}$ to be zero:

$$0 = \mathbf{r}(\mathbf{p}) + \frac{\partial\mathbf{r}\delta\mathbf{p}}{\partial\mathbf{p}} \tag{5.23}$$

A normal matrix inverse can only be used for square matrices, thus instead we use the Moore-Penrose pseudo-inverse:

$$\delta\mathbf{p} = -R\mathbf{r}(\mathbf{p}) \text{ with } R = \left(\frac{\partial\mathbf{r}}{\partial\mathbf{p}}^T \frac{\partial\mathbf{r}}{\partial\mathbf{p}}\right)^{-1} \frac{\partial\mathbf{r}}{\partial\mathbf{p}}^T \tag{5.24}$$

To find the values of the matrix R the idea is to perturb the known contour parameters of the training data sets by a small amount. Re-warp the training images with the permuted contour to the mean shape. The result can be used to describe a linear relationship between intensity difference between model image and sampled image, and change in model parameters.

We describe the process for one training set with image I. First warp the image I with the correct parameters $\mathbf{p}$ to the mean shape and convert to vector $\mathbf{g}_m$. Then convert $\mathbf{g}_m$ to PCA model parameters $\mathbf{c}$ and back to gray values $\tilde{\mathbf{g}}_m$:

$$\mathbf{c} = \Phi_{sa}^T \left(\begin{bmatrix} Ws\Phi_s^T(\mathbf{x} - \bar{\mathbf{x}}) \\ \Phi_a^T(\mathbf{g}_m - \bar{\mathbf{g}}) \end{bmatrix} - \bar{\mathbf{b}} \right) \tag{5.25}$$

$$\begin{aligned}[\mathbf{b}_s \| \mathbf{b}_a] &= \bar{\mathbf{b}} + \Phi_{sa}\mathbf{c} \\ \tilde{\mathbf{g}}_m &= \bar{\mathbf{g}} + \Phi_a \mathbf{b}_a\end{aligned} \tag{5.26}$$

Offset parameter $\mathbf{p}_i$ by a small value α, times the standard deviation σ_i of the parameter:

$$\dot{\mathbf{p}}_i = \mathbf{p}_i + \alpha\sigma_i \text{ with } \sigma_i = \sqrt{\lambda(i)} \tag{5.27}$$

Warp the image I with the permuted parameters $\dot{\mathbf{p}}$ to the mean shape and convert to vector $\mathbf{g}_o$. Then convert $\mathbf{g}_o$ to PCA model parameters $\mathbf{c}$ and back to gray values $\tilde{\mathbf{g}}_o$.

The weighted sum of difference between the original model an gray values of the model created from the permuted parameters give the "intensity offset" ratio

$$\frac{\partial \mathbf{r}}{\partial \mathbf{p}_i} = (\tilde{\mathbf{g}}_m - \tilde{\mathbf{g}}_o) w_i; \tag{5.28}$$

$$w_i = \frac{1}{\alpha} exp\left(\frac{-\alpha^2}{2\sigma_i^2}\right) \tag{5.29}$$

We perform this procedure for a number of offset values α, and use the mean of all $\frac{\partial \mathbf{r}_i}{\partial \mathbf{p}_i}$ approximations, to make the estimate more accurate. We stay within 0.5 standard deviations of the training data, with offsets values:

$$[-0.5, -0.3, -0.1, 0.0, 0.1, 0.3, 0.5] \tag{5.30}$$

For the pose parameter variances we use $\sigma_i = 1$.

Above procedure is for one data set, we combine the results of multiple data sets by using the mean of all R matrices.

$$R = \frac{1}{n}\sum_{i=1}^{n} R_i \tag{5.31}$$

Note, in case of only a few images, a weighted mean depending on the likelihood of the training set is expected to work better then this normal mean. For instance if we know that our patient age is normally above 50, and we have data sets from patients with age 55, 52 and 21, we lower the weight of the latest.

The constructed linear model is only valid for the local neighborhood of the object. Thus the segmentation contour must be close to the real contour. To allow larger movement a multi-scale model can be used. This multi-scale model is created by iteratively down scaling the training images by a factor of two. Training a separate AAM for every image scale. We start an image segmentation with the lowest resolution image, and after convergence or the maximum number of iterations, we move to a higher resolution image.

5.8 Extensions

5.8.1 3D ASM and AAM

The extension to 3D of the ASM and AAM is straightforward. Instead of x, y positions our position data now contains x, y, z vertices. A contour is a set of points connected by lines, in 3D the points (vertices) are connected by triangular faces. If alignment of landmark points, to remove rotation, translation and/or size is used, it has to be changed to allow 3D alignment. Alignment is often done in the quaternion domain [59], because quaternions are not susceptible to gimbal lock in case of object rotation.

We replace the line normals of the search step of ASM, by vertex normals. These vertex normals can be calculated from surrounding face normals. The $2D$ image interpolation of line samples is replaced by $3D$ interpolation.

In the AAM, instead of a triangulation of the data we use a tetrahedral mesh, to warp the image-volumes.

5.8.2 ASM, Search Distance

The ASM uses the Mahalanobis distance, to find the best new landmark position. First the local pixels are sampled in the direction of the normal. The derivative of this profile is compared to the distribution of the profile derivatives in the training data. This type of distance measure is appropriate for edges with large gray value contrast. But is not suitable for low contrast edges or if there is a large variation in gray values between data sets. For example, if half of the data sets have a shadow at the object edge and the other half a specular high light.

Therefore we introduce a new distance measure. We perform PCA on the edge gray value profiles corresponding to a certain point in the training data. We use the resulting eigenvectors to calculate model parameter values for new profiles in the test data. During the search step we divide the model-values of a new profile by the corresponding standard deviations found in the training profiles. We define the distance as the quadratic sum of these normalized model values.

5.8.3 AAM, B-spline Warp

In an AAM triangulation or a tetrahedral mesh is used to warp the image to the mean shape. This piecewise linear warp is only C^0 continuous at the triangle/tetrahedron boundaries.

Cubic splines can be used to create a smooth C^2 continuous warp of the images, with continuous first and second order derivatives . We use the free form deformation (FFD) grid of B-splines as introduced by Rueckert [60]. We use the method of Lee et al. [61] to get a least-squares fit of a B-spline deformation grid to the transformation described by the model vertices. A diffeomorphic constraint can be used to prevent folding of the transformation fields, see chapter 7.

This FFD method has also disadvantage, such as being more CPU expensive to calculate. But the main disadvantage is the least-square fit, of the transformation field to the vertices. If now one vertices moves it influences the whole transformation field, this increases the condition number with respect to the inversion of the model/gray value matrix. This can lead to a less accurate AAM search, and segmentation quality.

5.8.4 AAM, Simplex Minimizer

During the search step, the AAM uses the pre-calculated inverse model parameters/gray value matrix, to find the update of the model parameters. This inverse Jacobian matrix is only valid if the segmentation model is close to the real object. Also it is only optimal for step-sizes close to the ones used during training.

We can update the model parameters by computing the error between modeled image and real values, and use error gradients to find the optimal parameter update. To calculate the error gradient, we have to warp the image for every model parameter. This requires many image-warps for every search iteration, making it very CPU expensive.

Instead, we propose to use a boundary constrained Nelder-Mead simplex optimizer [45]. This optimizer requires a lot of iterations before convergence in comparison to a Newton optimizer. But does not require numerical or analytic gradients, and is robust against local minima. Stegmann et al. [62] states that simulated annealing gives higher accuracy than a quasi-Newton optimization in AAM segmentation. This stresses the importance of robustness against local minima. To speed-up, we do not use the simplex optimizer from the start, but first optimize with the pre-calculated inverse Jacobian matrix.

The pre-calculated Jacobian matrix minimizes the quadratic error between modeled image and sampled image. This quadratic cost function is very sensitive to outliers such as teeth fillings in the mandible. Instead we can also use

$$I_e = \log\left(1 + \frac{(I_m - I_s)^2}{\sigma_s}\right) \tag{5.32}$$

Which is more robust against outliers, with σ_s a soft outlier threshold value.

5.8.5 AAM, Initialize Model Parameters

To get a more robust segmentation result a multi-scale AAM model is used. When the AAM has finished on the lowest resolution image, we end up with model-parameters, pose-parameters and vertex positions.

The model parameters are an combination of shape and appearance parameters. With a higher resolution the appearance-model changes thus these model parameters cannot be used to initialize the AAM in a higher resolution image.

The most straightforward method is to use the shape information of the current search and sample intensities from the higher resolution image, to obtain initial model parameters. But because shape and appearance are linked, this will directly cause the shape to change from the previous found "optimal" shape.

Therefore we see the model-parameters of the higher resolution image as unknowns, and use an constrained non-linear optimizer [63] [64] to find the model-parameters which give a shape which is close to the shape of the lower resolution segmentation result.

5.8.6 AAM, Border

An AAM constructs an appearance model of the gray values inside an object. For some bones in CT data gray values at the boundary are approximately equal to gray values more to the center of the object. This situation can cause shrinkage [62] of the object contour or a less accurate fit of the bone shape.

To deal with this problem we can include ”the layer” of closest boundary background pixels to the appearance model. Then the appearance model also models the image edge, leading to a more accurate segmentation at the object boundaries. This border AAM is only suitable for objects with steady or Gaussian distributed background intensities. This makes the method suitable for CT segmentation but less for tasks such as face detection. Stegmann et al. [62] suggest to model only the pixels close to the object boundaries and dismiss pixels in the object center. This increases the speed and allows large intensity variations in the object center. Because the mandible is relative thin we do not dismiss any pixels.

To include the layer of boundary pixels, we first calculate the vertex-normals of the mesh. The normals are used during sampling to create extra boundary vertices a few pixels away from the real boundary vertices. In this way we can include a small layer of background pixels in the appearance model. These extra boundary vertices are created during sampling and are not include in the shape model.

5.8.7 AAM, Start Position

The AAM uses a pre-computed matrix to convert intensity differences into model parameter updates. This approach is only valid if the segmentation contour is close enough to the object in the image, and is not robust to local minima.

The model is usually initialized by a user which positions the mean shape model on top of the object in the image. If we instead initialize the AAM by using the common location of for instance the Mandible in a scan, we will have large error in the initial position. To allow a large amount of inaccuracy we use the following approach:

1. Obtain initial pose parameters like translation, scaling and rotation. For instance the mean values of the training data sets.

2. Do k_1 iterations of the lowest scale of the AAM model, for example $k_1 = 5$.

3. Calculate the error I_e between model and image intensities.

4. Perturb the translation by adding random values from a uniform distribution, for example with values between $[-7, 7]$ pixels of the lowest resolution.

5. Now we repeat steps 2 and 3. If the new error I_e is lower we use the perturbed values as new initial pose parameters otherwise we keep the old pose parameters.

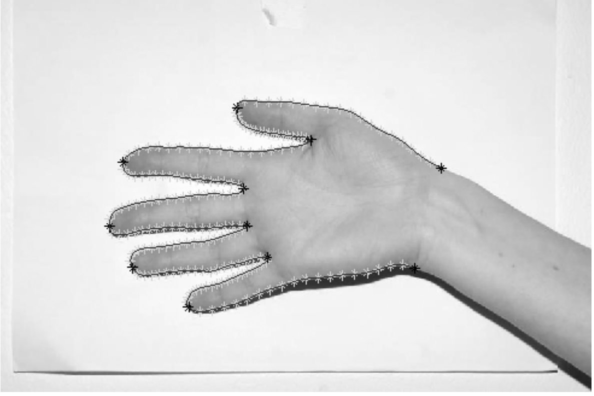


Figure 5.3 Hand photo, with manual contour, with in red the major landmarks and in green the interpolated line points.

6. We repeat this process for k_2 iterations, for example $k_2 = 20$.

The ASM model is more robust against small errors in the initial model position because it does not use a pre-computed update matrix. For large initial position errors we can adapt this method to the ASM. To obtain a measure of alignment we can use the Mahalanobis distance or PCA profile distance from the ASM search, After k_1 ASM iterations we calculate the distance errors for the profiles of the current landmark positions, and use the sum of distance errors as alignment error in the method above.

5.9 Results

5.9.1 Hand Photos

We have acquired 10 photos of hands with a resolution of 750 × 500. We manually draw a contour of around every hand, with landmark points on major point such as the finger tips. We then interpolated 20 evenly distributed points between every two landmark points. See figure 5.3.

We use these hand data sets to evaluate the performance of our improved ASM search method. We train the ASM on 9 data sets and test on the scan which was left out. This is repeated 10 times, with another scan left out each time. The ASM parameters which we use are shown in table 5.1. The first time we test with the original method to find the contour point update, based on the Mahalanobis distance of profile derivatives. The second time we test with our proposed methods which uses PCA on the profiles and tries to minimize the model values. For the segmentation results on all data sets see figure 5.4. The original method has a mean dice coefficient of 0.80. Our new method outperforms the old method with a Dice coefficient of 0.86.

We test the performance of the AAM with B-spline warp extension also on the

Table 5.1 The ASM segmentation parameters.

parameter	value	information
k	10	Length of landmark profile
n_s	10	Search length in pixels
n_{scales}	2	resolution scales
m	3	limit shape to $m\sqrt{\lambda}$
n_{search}	40	Number of iterations

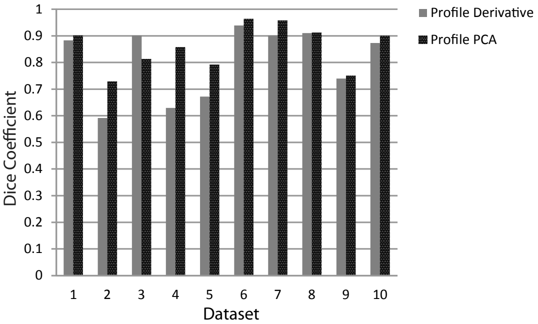


Figure 5.4 Comparison of original ASM profile search method, versus PCA profile search method. Shown are the Dice coefficients, of training a ASM model with a leave one out scheme.

hand data sets. The parameters which we use are shown in table 5.2, and the results in figure 5.5.

5.9.2 Mandible

We manually segmented the mandible in 13 patient CBCT data sets. These data sets are from dentate and edentate patients. After manual voxel based segmentation we converted the mandible segmentations to triangulated surface descriptions with marching cubes. Then the vertices are used as input for the shape context registration

Table 5.2 The AAM training and segmentation parameters.

parameter	value	information
n_b	5	Border added (pixels)
s	[100 100]	Texture size used (pixels)
n_{scales}	4	Number of AAM image resolutions.
n_{search}	15	Number of iterations
m	3	limit shape to $m\sqrt{\lambda}$

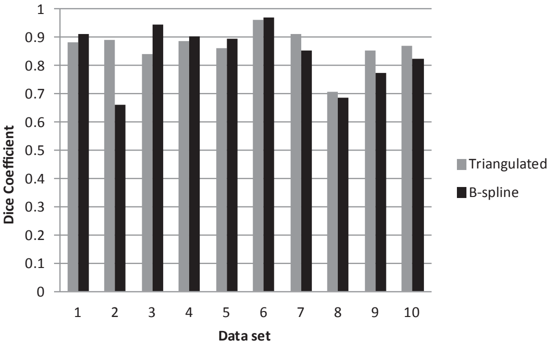


Figure 5.5 Comparison of segmentation performance of the original AAM, versus an AAM with warp based on free form deformation instead of triangulation. We use a leave one out scheme. Shown is the dice coefficient between the hand area of the AAM segmentation and manual segmentation of the hand.

Table 5.3 The AAM training and segmentation parameters.

parameter	value	information
n_b	5	Border added (pixels)
s	[150 150 150]	Texture size used (pixels)
n_{scales}	4	Number of AAM image resolutions.
n_{search}	15	Number of iterations

described in chapter 8. This resulted in dense surface descriptions of the data sets, by corresponding points (11202). We have combined the resulting surface descriptions, with expert annotations of the mandibular canals. These expert annotations were re-sampled to have 80 uniformly distributed points along each line.

We want to test the influence of the number of training data sets. Therefore we split our data in two parts, we use data set $1 - 11$ as training data, and 12 and 13 as test data sets. Now we first compose an AAM only using the first three sets, then an AAM using the four sets, until we use all 11 data sets. For the AAM we use the border AAM extension and also simplex optimization of the segmentation result. The other AAM parameters can be found in table 5.3. The Dice coefficient of the segmentation results is shown in figure 5.6, the mean distance to expert segmentation in figure 5.6. The Dice coefficient slowly increases when adding more training data sets. The mean distance to expert segmentation of the nerve canals fluctuates, but does not decrease in our results. Probably this is due to the small number of points describing the canals in comparison to the surface description of the bone (less than 1%). Therefore the model fit to the test set will mainly depend on the bone shape. Probably only after training on a large amount of data there will be freedom in the AAM model to move the nerve more independently from the bone shape.

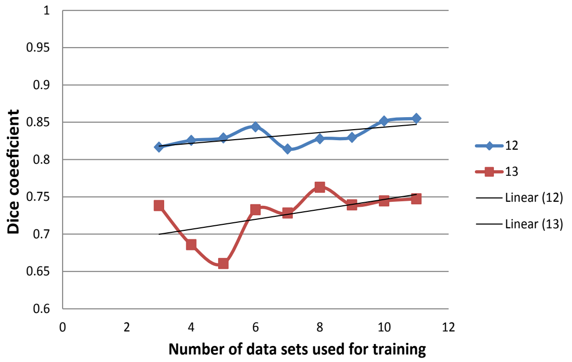


Figure 5.6 Dice coefficient versus number of training datasets. Shown are the results with dataset 12 and 13 as test sets.

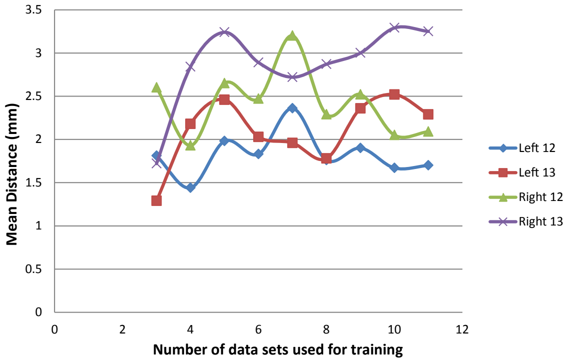
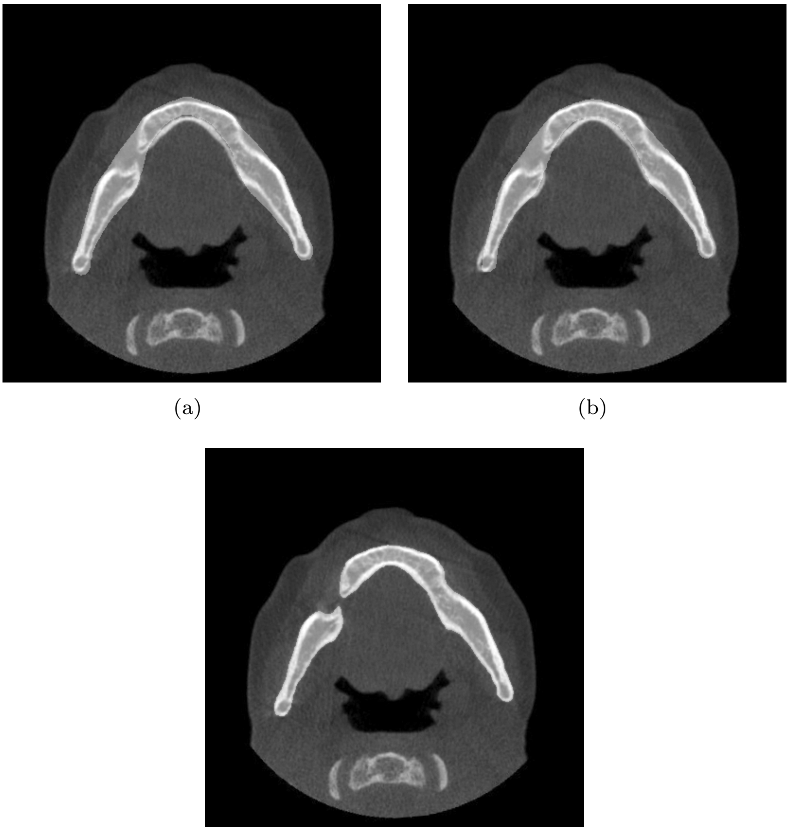


Figure 5.7 Mean distance to expert segmentation of the mandibular canal versus number of training data sets. Shown are the results with data set 12 and 13 as test sets.

5.10 Conclusion

This chapter describes the ASM and AAM model of Cootes and Taylor [25] [26]. We extend the models and evaluate the performance on 2D hand photos and 3D mandibles. The first extension is a new ASM search method based on PCA instead of the original Mahalanobis cost function. We evaluate the performance on 10 photos of hands. The overlap between manual segmentation and ASM segmentation of the hand is improved. The original method has a mean Dice coefficient of 0.80, and the new method a mean coefficient of 0.86.

The second extension replaces the original triangle based image warp by a smooth free form deformation (FFD) warp. The new warp increases the Dice coefficient for some hand data sets, but the other half has a lower Dice coefficient. Thus the smooth



(a)

(b)

(c)

Figure 5.8 AAM segmentation results when using 3 training sets (a), and the results when 11 data sets are used (b). Data set 12 is used as test image volume. Manual segmentation of a cross-section of the data set is shown in sub figure c.

deformation not always increases the dice coefficient. Probably, this is because the warp is less local. The FFD is fitted least squares to the vertices. Thus if one vertex moves it influence the whole transformation grid. This global effect makes the inverse Jacobian matrix less well defined.

We have implemented the AAM in 3D including the border AAM extension and simplex optimization. We tested the influence of the number of training data sets on the accuracy of mandible segmentation from CBCT. The Dice coefficients slowly increase if we add more data sets to the training data. We also measured the distance between manual annotation of the mandibular canals and AAM annotation of the canal. The mean distance error between automatic and manual annotation does fluctuate, but does not increase when training data sets are added. This is probably because the number of model points describing the canals is less than 1% of the points on the bone surface. Also the percentage of pixels describing the canal is relatively small. The PCA model is based on correlations. With a small amount of data sets, variations between bone and nerve look correlated, which are not correlated in real live. More training data sets will result in a model in which the nerve can move more independent from the bone shape. The bone contains far more points, which gives the bone segmentation error relative more weight. If more training data sets are added the bone segmentation error will become small. In that case the points describing the nerve will get more influence on the nerve segmentation result. In future work it will be important to train the model on more scans, such as used by Kainmueller et al. [33] (106 scans). Also increasing the influence of the points in the mandibular nerve can probably decrease the segmentation error of the canals. This can be achieved by using more points to describe the mandibular canals or using a weighted PCA model.

6

Demon Registration[1]

6.1 Introduction

Quantitative analysis of Multiple Sclerosis brain lesions, e.g. analysis of progressions requires accurate segmentation. We have developed an automatic lesions segmentation system [65], which uses multiple MRI modalities of a patient FLAIR, T1, T2, MD and FA. Pixel accurate lesion segmentation is only possible with pixel accurate registration between the patient scans, thus we need an accurate multiple MRI modality registration method.

Thirion introduced a registration algorithm called 'demons algorithm' [66]. This method is based on pixel velocities caused by edge based forces. The resulting pixel velocity / transformation field is filtered by a Gaussian kernel for global registration. It has a high registration precision [67], but is only able to register images of the same modality. A solution to allow demon registration of multiple modalities is a representation transformation in which for example a T1 scan is changed to appear similar to a T2 scan. In this chapter, we introduce and evaluate a joint histogram based MRI representation transformation method.

[1] This chapter is based on the following publication:

i) D.J. Kroon et al. "MRI Modality Transformation in Demon Registration", IEEE ISBI, 2009

6.2 Demon Registration Model

6.2.1 Classic Demon Registration

The optical flow equation for finding small deformations in temporal image sequences is used as basis of the demon registration forces. For a given point p in a static image F, let f be intensity and m the intensity in a moving image M. The estimated displacement (velocity) $\mathbf{u}$ required for point p to match the corresponding point in M is given by Thirion [66]

$$\mathbf{u} = \frac{(m-f)\nabla f}{|\nabla f|^2 + (m-f)^2} \tag{6.1}$$

where $\mathbf{u} = (u_x, u_y)$ in 2D, and ∇f is the gradient of the static image. There are two forces an internal edge based force ∇f and the external force $(m-f)$. The term $(m-f)^2$ is added by Thirion to make the velocity equation more stable, so to use it in image registration. Since this displacement $\mathbf{u}$ is based on local information, Gaussian smoothing of the velocity field is included as regularization. The demon equation is a local approximation, thus needs to be solved iteratively to register two images. Bro-Nielsen and Gramkow [68] demonstrated that the demon algorithm approximates the CPU expensive viscous fluid model registration.

The original equation only uses the edges in the static image as passive internal force, He Wang et al. [69] add an equation with the image edge forces of the moving image that improves the registration convergence speed and stability.

$$\mathbf{u} = \frac{(m-f)\nabla f}{|\nabla f|^2 + \alpha^2(m-f)^2} + \frac{(m-f)\nabla m}{|\nabla m|^2 + \alpha^2(m-f)^2} \tag{6.2}$$

The normalization factor α is proposed by Cachier et al. [70] to adjust the force strength.

6.2.2 Image Registration Model

Vercauteren et al. [71] describe a standard registration model; with a registration energy consisting of a similarity function, a transformation error function and smoothness regularization. They use as similarity measure the squared pixel distance, and the squared gradient of the transformation field as smoothness regularization. The resulting iterative registration algorithm can be written as follows:

- Given the transformation field S compute a correspondence update field U by minimizing E,

  $$E(U) = \|F - M \circ (S+U)\|^2 + \frac{\sigma_i^2}{\sigma_x^2}\|U\|^2 \tag{6.3}$$

  With F the static image, M the moving image, transformation field S describing the translation in x,y of every pixel from its original position, with U the (iteration) update of S, $\circ$ denotes image transformation, and σ_i and σ_x a constant for intensity uncertainty (image noise) and transformation uncertainty.

- If a fluid-like regularization is used, let $U \leftarrow K_{fluid} * U$. The convolution kernel K_{fluid} is typically a Gaussian kernel.
- Update the transformation field $S \leftarrow S + U$
- A diffusion-like regularization can be included, with $S \leftarrow K_{diff} * S$ (Not used in demon registration).

The update step for minimizing the energy $E(U)$ can be calculated using classic Taylor expansion. We rewrite E for the pixel p, with f the pixel intensity from static image F and m the intensity from the transformed image $M \circ S$, $\mathbf{u}$ the x,y update in translation of the pixel from U, and with ∇m the image gradient at pixel p.

$$E(\mathbf{u}) = \|f - m + \mathbf{u}\nabla m\|^2 + \frac{\sigma_i^2}{\sigma_x^2}\|\mathbf{u}\|^2 \tag{6.4}$$

Then we can calculate the error gradient:

$$\nabla E(\mathbf{u}) = 2(\nabla m)^T (f - m + \mathbf{u}\nabla m) + 2\frac{\sigma_i^2}{\sigma_x^2}\mathbf{u} \tag{6.5}$$

Assuming that E is minimum at $\nabla E(\mathbf{u}) = 0$, we can calculate the needed update:

$$\mathbf{u} = \frac{f - m}{\|\nabla m\|^2 + \frac{\sigma_i^2}{\sigma_x^2}}\nabla m \tag{6.6}$$

We see that if we use the local estimation $\sigma_i(p) = |f - m|$ as the image noise and $\sigma_x = \frac{1}{\alpha}$ we end up with the expression of the demons algorithm in equation 1.

6.2.3 Minimizing

Gradient descent is a basic solver for $argmin_{\mathbf{x}}(E(\mathbf{x}))$, it converges not as fast as higher order minimizers. With a large number of variables as is the case with image transformation fields, it is more memory efficient.

$$\mathbf{x} \leftarrow \mathbf{x} - \mu\frac{\nabla E}{\|\nabla E\|}\ ,\ \nabla E = \left[\frac{\partial E}{\partial x_1}, \frac{\partial E}{\partial x_2}, .., \frac{\partial E}{\partial x_n}\right] \tag{6.7}$$

in which μ is the step size which is found through line search using the error equation. We can also write the extended demon Registration in gradient descent format using equation 6.4 as E and 6.6 as ∇E, also an active edge force can be added as in equation 6.2. Equation 6.6 is derived from $\nabla E(\mathbf{u}) = 0$, thus it also provides a start value μ for the line search.

6.3 Modality Transformation

6.3.1 Mutual Information

Mutual information is commonly used as a similarity measure in multiple modality registration. The mutual information of moving image M and static image S is defined as:

$$I(M,F) = \sum_{M,F} p(m,f) log \left(\frac{p(m,f)}{p(m)p(f)} \right) \tag{6.8}$$

In this equation $p(m)$ and $p(f)$ are the probabilities of the gray values in resp. image M and F, $p(m,s)$ is the joint probability of the images gray values which can be derived from the joint histogram. Mutual information is global and gives only one similarity value for the whole image area, which is a disadvantage when using finite difference methods for local registration.

The idea behind mutual information registration is that every image has certain uniform intensity regions separated by edges. These regions correspond with regions in another image but with different intensity and texture. In an iterative registration process, corresponding regions will overlap more increasing the peaks in the joint histogram.

6.3.2 Proposed Method

We propose to use the joint histogram peaks to transform one image representation in to the other, allowing fast intensity based local image registration such as demon registration.

The joint histogram $H(I,J)$ of image I and J can be written as 6.9, looping through all pixel locations

$$H(\lfloor I(\mathbf{x})N \rfloor, \lfloor J(\mathbf{x})N \rfloor) = H(\lfloor I(\mathbf{x})N \rfloor, \lfloor J(\mathbf{x})N \rfloor) + 1 \tag{6.9}$$

With floor function $\lfloor\rfloor$, N the number of bins and with $I,J \in [0,1]$ and $\mathbf{x}$ is the pixel location.

We transform the image I into I_T with the same representation as J. This is done by finding for every pixel the gray value in image J which overlaps most often with the pixel gray value in image I.

$$I_T(\mathbf{x}) = argmax_j(H(\lfloor I(\mathbf{x})N \rfloor, \lfloor jN \rfloor)) \tag{6.10}$$

In medical images two regions can have the same gray value in one modality, but in another both regions can have totally different gray values. Also medical images suffer from slowly varying intensity non uniformities called the bias field in MRI. This implies that we have to use a more local modality transformation. We solve this problem by calculating a separate local mutual information histogram for every pixel by using Gaussian windows.

6.3.3 Combined with Demon Registration

When we transform an image from one MRI representation to another, the transformation is poorly defined on edges of the image, and the new image can contain some false edges. Thus a modality transformed image is not very useful to serve as edge forces. Thus our final demon registration algorithm with representation transformation [2] is:

$$\begin{aligned} E = \frac{1}{2}\left\|F_T - M \circ (S+U)\right\|^2 \\ +\frac{1}{2}\left\|F - M_T \circ (S+U)\right\|^2 + \frac{\sigma_i^2}{\sigma_x^2}\left\|U\right\|^2 \end{aligned} \tag{6.11}$$

$$\begin{aligned} \nabla E = (M_T \circ S - F)\left(\frac{\nabla F}{\left|\nabla F\right|^2 + \alpha^2 (M_T \circ S - F)^2}\right) \\ +(M \circ S - F_T)\left(\frac{\nabla M}{\left|\nabla M\right|^2 + \alpha^2 (M \circ S - F_T)^2}\right) \end{aligned} \tag{6.12}$$

With E the registration error, M_T and F_T the modality transformed static and moving image, S the transformation field, U the update of the transformation field (used by the line search).

To avoid local minima and to speed up registration, a scale space approach is used. We first resize the original images to 8×8 pixels and register these small images. Next we resize the found transformation fields and original images to 16×16, and so on, until the original resolution is reached.

6.4 Results

6.4.1 Setup

To test the performance of the demon registration algorithm we need perfect aligned ground truth data from multiple modalities. For this reason we use the BrainWeb MRI Simulated Normal Brain Database [72]. This database can provide T1 and T2 images with several noise and bias configurations. The noise in the simulated images has Rayleigh statistics in the background and Rician statistics in the signal regions. The 'percent noise' number represents the percent ratio of the standard deviation of the white Gaussian noise versus the signal for a reference tissue. Bias fields are varying fields estimated from real MRI scans; for a 20% level, the multiplicative biasfield has a range of values of 0.90 to 1.10 over the brain area [73].

Because demon registration is developed for local non-rigid transformation, we test the our algorithm using a spherical distortion (spherize filter) on an part of a T1 brain slice, location center slice z-plane (90), x,y coordinate 115,60 radius 34. The spherical distortion is described by [74]:

$$R = R_0/(\sqrt{2}sin(\frac{\gamma}{2})) \tag{6.13}$$

$$X_c = \frac{1}{2}(1 + cot(\frac{\gamma}{2}))R_0, Y_c = \frac{1}{2}(1 - cot(\frac{\gamma}{2}))R_0 \tag{6.14}$$

[2] Matlab implementation available on MathWorks.com 'File Exchange'

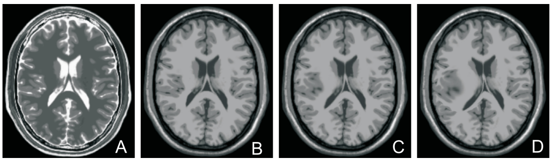


Figure 6.1 Figure A and B shows a T2 and T1 slice without noise or bias field. Figure C and D shows the local spherize transformed T1 slice with a γ value of 30 and 60 degrees.

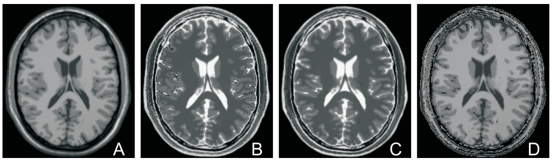


Figure 6.2 Figure A and C shows a T1 and T2 slice without noise or bias field, which are modality transformed into image, B and D, respectively.

$$\begin{bmatrix}\acute{x}\\ \acute{y}\end{bmatrix} = \frac{Y_c + \sqrt{R^2 - (\sqrt{x^2+y^2} - X_c)^2}}{\sqrt{x^2+y^2}} \begin{bmatrix}x\\ y\end{bmatrix} \tag{6.15}$$

With R_0 the 2D radius of the distortion, R the radius of the virtual 3D sphere, γ the amount of distortion range 0 to 90 degrees, pixel coordinates x, y with $\acute{x}, \acute{y}$ the spherical transformed coordinates

An example of the transformed images are shown in figure 6.1. Before the demon registration can be done both the T1 and T2 are transformed to their opposite MRI representations, using local joint histogram peaks between the T1 and T2 image, see figure 6.2.

The parameters in the demon registration are chosen to suppress noise but still allow local transformations, σ of the transformation Gaussian smoothing is 8, α is chosen 2.5. The Gaussian window used for modality transformation is chosen 100×100 with $\sigma = 33$, and after registration, a Gaussian window is used for modality transformation with size 70×70 and $\sigma = 23$ followed by a second registration pass.

6.4.2 Methods Used as Comparison

We compare the registration performance with the free form deformation (FFD) registration method grid existing of 1D B-splines which is introduced by Rueckert et al. [60]. The control points of the grid are moved to transform an image, and a similarity measure between a target image and transformed image is used to determine if the registration improves. We have implemented the algorithm including multi-scales refinement, and for fast and stable mutual information registration, we calculate the mutual information measure separately for each control point, from his neighborhood.

Edges of the regions in T1 and T2 can be registered onto each other, for comparison to our representation transformation method. In [75] the normalized gradient field

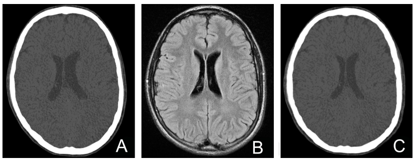


Figure 6.3 A CT slice (A) is registered with demon registration on a T1 (B) slice of the same patient,figure C shows the result.

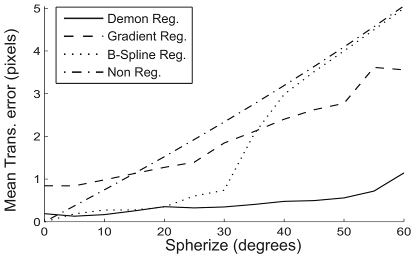


Figure 6.4 Registration performance with increasing distortion. A spherical transformed T1 slice is registered on a T2 slice, with modality transformed demon registration, gradient images registration and B-spline registration.

is used. We have tested the normalized gradient field and a canny edge detector transforming both T1 an T2 to the same ”edge representation’. These approaches give large registration errors, because some region edges are detected in T1 but not in T2. Finally we decided to high pass filter the images and normalize the images with $|I|/(|I|+\beta)$, with constant $\beta = 0.1$, this gave more reliable results.

6.4.3 Simulations

The first simulation is by spherical transforming a part of a bias and noise free T1 slice as in figure 6.1. We show the effect of the amount of distortion between a T1 and T2 slice versus registration result. The mean transformation error is calculated on the area of the spherical distortion 70×70, and is the distance in position between the correct pixel location and location after registration. The transformation error after registration is shown for the B-spline, demon and gradient registration, see figure 6.4. Modality transform with demon registration clearly outperforms the other registration methods. The spherical distortion has non smooth transformation edges, thus the Rueckert B-spline registration which produces curvature smooth transformation fields performs less for a spherical distortion of more than 30 degrees. The Gradient Registration has an error due to edges which are not present simultaneously in both modalities.

The second simulation is to test the influence from a bias field on the registration. A bias field will broaden the histogram peaks used for detection, thus tissues cannot be classified by one gray value. This problem is partly solved in our representation

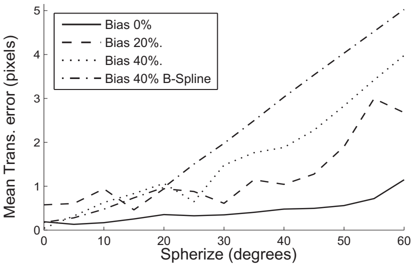


Figure 6.5 Bias field registration performance. With spherical transformed T1 slice registered on T2 slice.

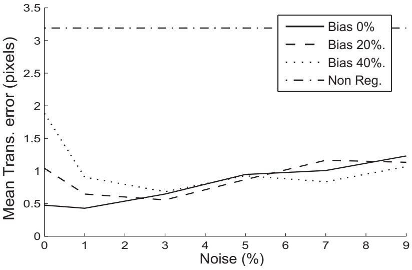


Figure 6.6 Noise registration performance. Spherical transformed T1 slice registered on T2 slice, both with bias fields.

transformation method by using a local Gaussian window for the joint histogram. The results can be found in figure 6.5.

The final simulation is to test the influence of noise on the registration result, see figure 6.6. With increasing noise the transformation error increases slightly. From zero to one percent noise the registration error becomes better, this is due to local minima during registration. A simulated annealing optimizer is most likely better than the current gradient decent.

6.5 Conclusions

Modality transformation using the intensity peaks in a joint histogram seems to work well for deformed MRI images, and probably also with CT see figure 6.3. The bias field has small effect on the registration error with small deformations, but with large deformations doubling the bias fields will also double the transformation error. Gradient / Edge based registration suffers from the fact that region edges do not show in all modalities, which results in incorrect transformations during registration with aligned image data. This problem can be solved by using a wider Gaussian to smooth the registration velocity field, but in that case registration is no longer local. Rueckerts B-spline registration is not capable to deal with large spherical deformations, probably because the B-spline grid can only represent really smooth transformations.

In conclusion demon registration with image transformation gives the best results while dealing with large spherical distortions and good T1 and T2 MRI images. The registration method used can probably be improved by new modality transformations during the demon registration iterations and using an simulated annealing optimizer.

7

B-spline Registration[1]

This chapter is about registration, with as main focus the nonrigid registration introduced by Rueckert et al. [60]. The first section explains the usefulness of image registration. The second section describes a rigid registration method, and basics of rigid registration. The third section describes the non-rigid registration of Rueckert, followed by a section about image warp smoothness penalties. The last section discusses the introduced free form deformation (FFD) registration, and compares it to Demon registration.

7.1 Image Registration

In image registration we have usually two images: a static/reference image, and a moving/deforming image. The goal is to align the moving image as good as possible with the static image. This is useful for image segmentation and detection of differences between images [76]. For example, intensity differences between a picture of a patient to an aligned picture of a healthy person can be used to detect a disease [77]. Or local transformations between medical images from multiple time steps can say something about tumor growth [78]. In segmentation the idea is that one of the data sets is accompanied by a segmented-image with object labels. The image without labels is then registered to an image with object labels. The labels can then be warped to the other image, creating a segmentation of that image.

[1] This chapter is partly based on the following publication:

i) D.J. Kroon et al. "Multiple Sclerosis Detection in Multispectral Magnetic Resonance Images with Principal Components Analysis", MICCAI, A Grand Challenge II, 2008

7.2 Rigid Registration

We have two 3D images which we want to align. The first is the moving image I_m and the second isthe static image I_s. Both have the same image dimensions X, Y, Z. We denote the domain for the image coordinates of the images as $\mathbf{\Omega} = (x, y, z)|0 \leq x < X, 0 \leq y < Y, 0 \leq z < Z$

7.2.1 Transformation

The first step is to define the transformation between the two images. In case of rigid alignment we have rotation and translation. We can describe these transformations by a rotation matrix $M_\mathbf{r}$ and translation matrix $M_\mathbf{t}$:

$$M_\mathbf{r} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(r_x) & -\sin(r_x) & 0 \\ 0 & \sin(r_x) & \cos(r_x) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(r_y) & 0 & \sin(r_y) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(r_y) & 0 & \cos(r_y) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(r_z) & -\sin(r_z) & 0 & 0 \\ \sin(r_z) & \cos(r_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.1}$$

$$M_\mathbf{t} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.2}$$

The warp can then be described by:

$$\begin{bmatrix} x_n \\ y_n \\ z_n \\ 1 \end{bmatrix} = M_\mathbf{r} M_\mathbf{t} \begin{bmatrix} x_m \\ y_m \\ z_m \\ 1 \end{bmatrix} \tag{7.3}$$

With (x_n, y_n, z_n) a pixel coordinate in the warped image I_n, and (x_m, y_m, z_m) a pixel coordinate in the original moving image I_m. But we can also invert the transformation, and use backwards interpolation of the original moving image:

$$\begin{bmatrix} x_m \\ y_m \\ z_m \\ 1 \end{bmatrix} = (M_\mathbf{r} M_\mathbf{t})^{-1} \begin{bmatrix} x_n \\ y_n \\ z_n \\ 1 \end{bmatrix} \tag{7.4}$$

The advantage of backwards warping is that the interpolation between pixels is done in the original image, allowing fast methods such as tri-linear and cubic interpolation. In case of forward mapping often piecewise linear transformations are used, but they require CPU expensive triangulation of the warped points.

7.2.2 Image Similarity

To determine the quality of alignment between the warped image I_n and static image I_m a similarity measure is needed. One of the most used similarity measure is the sum of squared differences (SSD):

$$E = \sum_{\mathbf{x} \in \Omega} \left(I_n(\mathbf{x}) - I_m(\mathbf{x}) \right)^2 \tag{7.5}$$

With E a measure of the alignment error between the images. This error is inversely related to the quality of alignment. Other often used similarity measures are, mutual information (MI), normalized mutual information (NMI) and local mutual information (LMI) [79]. The advantage of mutual information is that it allows registration between different image modalities and is robust against illumination differences. Other similarity measures are [80], gradient differences, gradient correlation, normalized cross correlation, pattern intensity, and log of absolute difference.

7.2.3 Optimization

In previous sub-sections, we introduced a method to warp the image and a method to measure the quality of the warp. But how do we determine the unknown parameters for the rotation and translation matrices? The most common way is to first construct an error-gradient $\frac{dE}{d\mathbf{p}} = \left[\frac{dE}{dp_1}, \frac{dE}{dp_2}, .., \frac{dE}{dp_6} \right]$ for the unknown warp parameters $\mathbf{p} = [r_x, r_y, r_z, t_x, t_y, t_z]$. This error gradient can calculate using finite differences, with h a small value.

$$\frac{dE}{dp_i} = E(p_i + h) - E(p_i) \tag{7.6}$$

Calculating the six unknowns requires six finite difference image warps, and one warp for the initial error.

Now we know the error gradient, we can move the parameters in the opposite direction to minimize the error and maximize the alignment. To determine the amount of movement in that direction we need to calculate the step length.

The step length can be calculated by warping the image for a certain step length and calculate the error between the images. After a few tries we can fit a polynomial through the step length versus error and use the derivative of the polynomial to find the optimal step length. The finite differences are only locally valid, thus we have to calculate the error gradient and step size iteratively to align the images. The described method above is called steepest-decent minimization. An method which converges faster but requires second order derivatives of the error is Newton optimization. Second order derivatives require many finite difference warps. Therefore we do not use a Newton but a Quasi Newton optimizer [81]. This optimizer estimates the second-order information from gradient information of previous optimization steps.

7.3 Non-Rigid Registration

The non-rigid alignment is based on the method introduced by Rueckert et al. [60]. This method is based on free form deformation (FFD) with a B-spline grid. To define a spline based FFD, we denote the domain of the image volume as $\mathbf{\Omega} = (x, y, z)|0 \leq x < X, 0 \leq y < Y, 0 \leq z < Z$. Let $\mathbf{\Phi}$ denote a n_x x n_y x n_z mesh of control points $\phi_{i,j,k}$ with uniform spacing d_x,d_y and d_z. Then, the FFD with transformation T can be written as the 3-D tensor product of 1D cubic B-splines

$$\mathbf{T} = \sum_{l=0}^{3}\sum_{m=0}^{3}\sum_{n=0}^{3} \mathbf{B}_l(v)\mathbf{B}_m(u)\mathbf{B}_n(w)\mathbf{\Phi}_(i+l, j+m, k+n) \tag{7.7}$$

where

$$i = \lfloor x/n_x \rfloor - 1 \ , \ j = \lfloor y/n_y \rfloor - 1 \ , \ k = \lfloor z/n_z \rfloor - 1 \tag{7.8}$$

$$u = x/d_x - \lfloor x/d_x \rfloor \ , \ v = y/d_y - \lfloor y/d_y \rfloor \ , \ w = z/d_z - \lfloor z/d_z \rfloor \tag{7.9}$$

Note: equation 7.9 is our corrected version of the equation in the paper of Rueckert et al. [60] which stated that $u = x/n_x - \lfloor x/n_x \rfloor$.

$\mathbf{B}_l$ represents the lth basis function of the B-spline

$$\begin{array}{l} \mathbf{B}_0(u) = (1-u)^3/6 \\ \mathbf{B}_1(u) = (3u^3 - 6u^2 + 4)/6 \\ \mathbf{B}_2(u) = (-3u^3 - 3u^2 + 3u + 1)/6 \\ \mathbf{B}_3(u) = u^3/6 \end{array} \tag{7.10}$$

The quality of alignment of the moving image I_m with the static image I_s with a certain control grid $\mathbf{\Phi}$, can be described by a similarity measure such as SSD. The iterative alignment is done with steepest gradient decent optimization of the error function $\mathbf{E}$. The gradient of a certain control point $\mathbf{\Phi}(i)$ is calculated by moving a control point a Δ step up, down, right, left, forwards and backwards. We then compute the similarity error ϵ in the affected area for all those six transformations, after which the central gradient $\nabla\mathbf{E}(\mathbf{\Phi}(i))$ at the point is known.

$$\mathbf{\Phi}(t+1) = \mathbf{\Phi}(t) + \mu\frac{\nabla\mathbf{E}}{||\nabla\mathbf{E}||} \tag{7.11}$$

$$\nabla\mathbf{E} = \frac{\partial\mathbf{E}(\mathbf{\Phi})}{\partial\mathbf{\Phi}} \tag{7.12}$$

The FFD registration is often done at multiple image scales. Starting with a control grid of a few knots and low image resolution. After the registration converges, the control grid is refined and image resolution increased. This multi-scale approach increases the registration speed and decreases the change of local minima.

7.4 Penalties

In case of tissue registration often a certain smoothness of the deformation is assumed. These kind of smoothness constraints can be added by introducing a smoothness penalty to the registration cost function. The penalty which Rueckert et al. [60] added is the 3-D counterpart of the 2-D bending energy of a thin-plate of metal:

$$\begin{aligned} C_{smooth} = \tfrac{1}{V} \textstyle\int_0^X \int_0^Y \int_0^Z \Big[\left(\tfrac{\partial^2 \mathbf{T}}{\partial x^2}\right)^2 + \left(\tfrac{\partial^2 \mathbf{T}}{\partial y^2}\right)^2 + \left(\tfrac{\partial^2 \mathbf{T}}{\partial z^2}\right)^2 \\ + \left(\tfrac{\partial^2 \mathbf{T}}{\partial x \partial y}\right)^2 + \left(\tfrac{\partial^2 \mathbf{T}}{\partial x \partial z}\right)^2 + \left(\tfrac{\partial^2 \mathbf{T}}{\partial y \partial z}\right)^2 \Big] dxdydz \end{aligned} \tag{7.13}$$

Another often used constraint is a volume preserving cost function. The local relative volume change of the transformation field is measured by the local Jacobian determinant of the transformation field. The Jacobian matrix is a first order affine approximation of the free form deformation:

$$J = \begin{bmatrix} \frac{\partial \mathbf{T}_x}{\partial x} & \frac{\partial \mathbf{T}_x}{\partial y} & \frac{\partial \mathbf{T}_x}{\partial z} \\ \frac{\partial \mathbf{T}_y}{\partial x} & \frac{\partial \mathbf{T}_y}{\partial y} & \frac{\partial \mathbf{T}_y}{\partial z} \\ \frac{\partial \mathbf{T}_z}{\partial x} & \frac{\partial \mathbf{T}_z}{\partial y} & \frac{\partial \mathbf{T}_z}{\partial z} \end{bmatrix} \tag{7.14}$$

The Jacobian matrix can be calculated for every pixel coordinate of the volume. The transformation is called diffeomorphic, when the determinants of all Jacobian matrices are above zero. This means that the deformation field is smooth and invertible. In that case for every pixel in the static image consist an unique pixel in the moving image and the other way around. Thus there is no folding of the deformation fields. During registration the following cost function C_{smooth} can be added to keep the transformation diffeomorphic and volume preserving:

$$C_{smooth} = \frac{1}{V} \int_0^X \int_0^Y \int_0^Z \left| \log \left(\det \left(J(x,y,z) \right) \right) \right| dxdydz \tag{7.15}$$

The absolute log inside this equation cannot be analytical intergraded over the image volume. Therefore we need to calculate the cost function separately for all the pixel coordinates in the image. This makes the regularization CPU expensive.

Therefore other methods are developed to keep the transformation diffeomorphic. One method which can be used with B-splines is the composition method [82]. The idea is that if we warp an image with a diffeomorphic transformation field, and then warp the result with another diffeomorphic field, the combined effect is also diffeomorphic. A FFD is diffeomorphic if the nodes which describe the transformation only move a small distance. Choi and Lee [83] determined that if the movement of the FFD nodes is smaller than $\frac{1}{2.48}$ of the node distance d_x, d_y, d_z, the warp is diffeomorphic. For larger diffeomorphic deformations we can use a composition of FFD warps

7.5 Discussion

The FFD registration using a B-spline grid is one of the most used registration methods in Medical Imaging with more than 3300 publications on Elsevier's Science Direct *(http://www.sciencedirect.com/)*. FFD registration has many advantages. It is robust in case of noise, especially when used in a multi-scale registration approach. It allows several similarity measures for alignment to be used. Volume preserving and smoothness penalties can be easily integrated in the registration. The registration almost always produces physical possible smooth transformation fields. It is also relative fast because the number of control nodes which must be optimized is lower than the number of pixels in the image. In a basic FFD the movement of a node is only influenced by the pixels in the local $4 \times 4 \times 4$ neighborhood of nodes.

The major disadvantage to the FFD registration is the inverse FFD. The FFD cannot be inverted and described by the same control point grid without loss of precision. To get an accurate inverse, the number of control points must be increased or the inverse of the discrete transformation fields must be used. This makes symmetric and group wise image registration difficult.

In the previous chapter we introduced demon registration. Demon registration cannot easily incorporate other similarity measures such as mutual information. The FFD registration uses a grid of B-splines, which is not suitable for deformations such as rotations and twists. Demon registration often has better registration results than FFD registration in terms of intensity differences. This is because demon registration is a kind of fluid registration, allowing complex and large deformations. This is also a disadvantage, because in demon registration pixels can flow long distances, creating deformations which are physically not possible. A FFD allows pixel based volume preserving penalties for example to make bone pixels rigid. In fluid registration there is no static reference such as the uniform-grid in FFD registration, making pixel based penalties more difficult to use.

The above comparison changes, in case of diffeomorphic registration through composition. Because in case of composition complex deformations can be created by using small warps, changing the FFD registration to a more fluid like registration.

In chapter 10, we compare the performance of demon registration and B-spline registration on Mandible segmentation, and mandibular canal localization.

8 Shape Context Registration

8.1 Introduction

Obtaining the location and shape of the mandible in CT data is an important preprocessing step for localization of the nerve-channels inside the mandible. A robust method already used for mandible segmentation [33] is an Active-Shape Model (ASM). This is a deformable model with shape constraints, learned from training sets. This a priori shape information makes the (ASM) robust against intensity variations and implicitly provides us with a reference coordinate systems for localization of objects inside the mandible [84].

ASM uses Principal Component Analysis (PCA) to learn shape variations and correlations between corresponding point-sets in training data sets. In this report we focus on building a dense Point Correspondence Model (PCM) for mandible surfaces. Constructing a dense PCM by manual annotating CT data is a difficult and time consuming task. Literature already describes methods for PCM building. For example, description by spherical harmonic basis functions (SPHARM), Minimum Description Length (MDL) and subdivision surface (MSS) [85]. SPHARM is mainly suitable for spherical closed surface objects. MSS has to be manually initialized by anatomical landmarks. MDL maps the object first to a spherical surface, and then manipulates the correspondences to obtain an ASM model with minimum description length [86]. A major disadvantage is that the absolute minimum is obtained when all points collapse into one single point. Keeping one set of points fixed as master, does not solve this problem entirely [87].

In this chapter we introduce an alternative method for dense PCM construction. It does not have a spherical or closed surface constraint and is fully automatic. We use the method to construct a PCM for the surface of the mandible in cone-beam CT. The method can also be used on other types of objects or as preprocessing step for the existing MDL method. Our method is based on finding correspondences using the Shape Context (SC) model [88], followed by a novel iterative closes point (ICP) method. The surface mapping is constrained by a B-spline based Free Form Deformation (FFD) grid [61], with positive Jacobian to produce diffeomorphic transformations.

The next section describes the 2D shape context method, and the extension to 3D followed by a sub-section of further improvements. The third section describes the symmetric ICP method. This is Followed by the results of building a PCM of the mandible. The last section describes the conclusions and discussion.

8.2 Corresponding Points using Shape Contexts

8.2.1 Shape Contexts

The first step in ASM building is describing the contour or surface of the object in each training data set by a set of points $P = p_1, ...p_n$ with $p_i \in \Re^2$ or $p_i \in \Re^3$. Every point in a training data set must have a corresponding point in every other data set. These point correspondences allow PCA to find all major shape variations of the object. A way to construct a point correspondence model is to non-rigidly register the point-set of one data set, to all other point-sets. A point must always have the same-relative position. For instance, always be on the tip of the nose in a face training set. An unique descriptor can be used to capture the relative-location of a point, and to find corresponding points.

Shape Context (SC) by Belongie et al. [88] is a method which can find one-to-one correspondences between two 2D point clouds P with n points, and Q with m points. The method consists of two steps: the first is constructing a feature vector for each point in both data sets. The second step is matching the feature vectors of the point clouds, by minimizing the total matching cost using the Hungarian [89] algorithm.

The feature vector of a point in the SC algorithm is a log-polar histogram, represented as vector. This histogram collects the location of all other points relative to the selected point, described by log-distance and angle. We can calculate the matching cost C_{pq} of two points p and q, with feature vectors g and h with length K, using χ^2 statistics:

$$C_{pq} = \frac{1}{2} \sum_{k=1}^{K} \frac{[g(k) - h(k)]^2}{g(k) + h(k)} \tag{8.1}$$

When we calculate the cost between all points in a data set, we end up with a $m \times n$ matrix. To achieve the one-to-one correspondences we define the best matching, as the matching with the lowest total connecting cost between points. This can be done in $O(N^3)$ time using the Hungarian method [89]. Rows and columns with fixed cost are added to the cost matrix to make it square, and to compensate for the difference in the number of points and to compensate for outliers.

The points which have found a corresponding point are used to construct a spline based warping-field. This field warps the point based surface description from one data set to the shape of the object in the other data set. This creates a new point description of the data set with one-to-one point correspondence in the other data set.

8.2.2 Extension to 3D

The extension of 2D shape context to 3D can be done in a straightforward way, by replacing the log-polar histogram by a 3D log-spherical histogram. We describe the points by the log of the radius, inclination (polar angle) and azimuth. But uniform sampling of inclination and azimuth on a sphere, will not result in an uniform distributed histogram, due to changing surface-area. We solve this by replacing the inclination by the cosine of the inclination. The following equation gives our mapping from Cartesian coordinates (x, y, z) with radius r to histogram coordinates (o_r, o_θ, o_ϕ):

$$\begin{aligned} o_r &= \frac{n_r}{\log(r_{max}) - \log(r_{min})} \log\left(\frac{r}{r_{min}}\right) \\ o_\theta &= n_\theta \frac{z}{r} \\ o_\phi &= \frac{n_\phi}{2\pi} \tan^{-1}(\frac{y}{x}) \end{aligned} \tag{8.2}$$

With n_r, n_θ, n_ϕ the number of histogram bins, and r_{max}, r_{min} are constants describing the maximum and minimum distance in the point cloud.

8.2.3 Enhancements

Histogram

The log-spherical histogram is often sparse, which leads to discretization effects and inaccurate matching. A solution is to transform not only the points but also the triangles of the surface geometry to histogram space, and use them to create extra points. In our case the mapping is nonlinear, thus instead we transform the triangulated surface to a voxel volume in image space. This is done by iteratively splitting the polygons into smaller polygons. If a polygon is smaller than a voxel, it is drawn in the volume. The boundary voxels are then used as points instead of the vertices. This creates a much denser histogram. The resolution of the image volume, can be used for the trade of between memory usage, computing speed and histogram density.

Approximating B-splines

The shape context results will contain a certain percentage of wrong point correspondences. Locating the areas with wrongly matched points is possible, but we do not know which of the points is wrongly matched, or if a matching point exist. Interpolating thin plate splines (TPS) can be used to construct the transformation field between a list of matched points, but this field will be non-smooth due to the wrongly matched points. Therefore the approach is to approximate the transformation between points sets with regularized thin plate splines, allowing a certain amount of correspondence error. After calculating the transformation field the original point cloud is warped.

Repeating the shape context process on the now roughly aligned data sets will result in a more accurate and robust registration.

Instead of regularized TPS, we fit a multi-level Free Form Deformation (FFD) grid. We denote the domain of the image volume as $\mathbf{\Omega} = (x, y, z)|0 \leq x < X, 0 \leq y < Y, 0 \leq z < Z$. Let $\mathbf{\Phi}$ denote a $n_x \times n_y \times n_z$ mesh of control points with uniform spacing d_x,d_y and d_z. Then, the FFD with transformation $\mathbf{T}$ can be written as the 3-D tensor product of the familiar 1D cubic B-splines $\mathbf{B}_l$ with basis l [60] :

$$\mathbf{T} = \sum_{l=0}^{3} \sum_{m=0}^{3} \sum_{n=0}^{3} \mathbf{B}_l(v)\mathbf{B}_m(u)\mathbf{B}_n(w)\mathbf{\Phi}(i+l, j+m, k+n) \tag{8.3}$$

where

$$i = \lfloor x/n_x \rfloor - 1\ ,\ j = \lfloor y/n_y \rfloor - 1\ ,\ k = \lfloor z/n_z \rfloor - 1 \tag{8.4}$$

$$u = x/d_x - \lfloor x/d_x \rfloor\ ,\ v = y/d_y - \lfloor y/d_y \rfloor\ ,\ w = z/d_z - \lfloor z/d_z \rfloor \tag{8.5}$$

We approximate the transform field between data sets by a FFD grid $\mathbf{\Phi}_g(i, j, k)$. Which is constructed from the point correspondences with the method of Lee et al. [61]. The method starts with a rough FFD grid with is iteratively refined to get a better approximation of the transformation. Thus we can control the regularization of the transformation field by the number of FFD refinements. The method of Lee does not guaranty a diffeomorphic transformation. Smooth and folding free deformation grids have a Jacobian $\mathbf{J}_{\mathbf{T}}(\mathbf{x})$ which is always larger than zero. For this reasons we calculated the discrete Jacobian for all voxel coordinates $\mathbf{x}$ in the data set volume [90].

$$\mathbf{J}_{\mathbf{T}}(\mathbf{x}) = \det \begin{bmatrix} \frac{\partial \mathbf{T}_x(\mathbf{x})}{\partial x} & \frac{\partial \mathbf{T}_x(\mathbf{x})}{\partial y} & \frac{\partial \mathbf{T}_x(\mathbf{x})}{\partial z} \\ \frac{\partial \mathbf{T}_y(\mathbf{x})}{\partial x} & \frac{\partial \mathbf{T}_y(\mathbf{x})}{\partial y} & \frac{\partial \mathbf{T}_y(\mathbf{x})}{\partial z} \\ \frac{\partial \mathbf{T}_z(\mathbf{x})}{\partial x} & \frac{\partial \mathbf{T}_z(\mathbf{x})}{\partial y} & \frac{\partial \mathbf{T}_z(\mathbf{x})}{\partial z} \end{bmatrix} \tag{8.6}$$

We minimize the distance between a uniform initialized FFD grid $\mathbf{\Phi}_t(i, j, k)$ and the fitted FFD grid $\mathbf{\Phi}_g(i, j, k)$. Obtaining a folding free transformation field by using an quasi-Newton optimizer, and the sum of the following cost functions:

$$O_{cost} = \frac{1}{n_x n_y n_z} \sum_{i,j,k} \|\mathbf{\Phi}_t(i, j, k) - \mathbf{\Phi}_g(i, j, k)\| \tag{8.7}$$

$$J_{cost} = \frac{1}{XYZ} \sum_{\mathbf{x}} \frac{|\log(\max(\mathbf{J}(\mathbf{x})), \epsilon)|}{\max(\mathbf{J}(\mathbf{x}), \epsilon)} \tag{8.8}$$

With ϵ the floating point precision near zero. The derivatives of the cost function are obtained through finite differences.

Spatial Normalization

The size, translation and orientation of the mandible varies between data sets. We normalize the data sets by subtracting the mean and dividing the point coordinates by their distance to the center. We also perform Principal component analysis (PCA) on the points, resulting in a 3×3 eigenspace matrix in 3D [91]. We use the inverse of this matrix to align the points with the x, y, z axis. This normalization can change the axis order and axis signs. Thereforee we calculate the Euclidean distance between the unrotated points and rotate points, and change the order and sign of the axis, until the distance is minimal.

Hungarian Algorithm

The original shape context uses the Hungarian algorithm to get a list of one-to-one matched point. This algorithm matches 1000 points in about 8 seconds, but it takes 2 hours to match 10000 points, making it unsuitable for large point sets. Wrong connections will be created in case of surface sampling differences, due to the one-to-one restriction of the Hungarian algorithm. Therefore we replace the Hungarian algorithm and allow multiple-to-one connections. Connecting each point in P to the point in Q with the lowest matching cost C_{pq}. We also add the connections from the other data set Q to P. We regularize the obtained point transformations with the FFD approximation, and solve the whole problem iteratively.

In practice, points often correspond to points at approximately the same position in another data set. Thus we can split the correspondence problem into sub-volumes for faster processing.

8.3 Corresponding Points using Iterative Closest Point

Finding corresponding points between roughly aligned objects can be done with Iterative Closest Point (ICP) matching. The algorithm matches closest points between two point data sets, for example using kd-trees [92]. The matched points are used to construct a transformation field, which warps the points to better align the point clouds. The closest point matching followed by warping is repeated for a few iterations to get accurate alignment.

8.3.1 Distance Field

Local minimums can prevent surfaces to slide along each other, in point to point matching. Therefore we perform point to plane matching, based on distance fields. There exist fast algorithms to calculate the Euclidean distance transform for a binary volume [93]. The polygon rasterisation algorithm introduced in the Histogram subsection, can be used to discretize our surface to a binary volume V_B, which is then converted to the distance volume D. Matching a point $\mathbf{x} = (x, y, z)$ to the closest interpolated point $\mathbf{y} = (x, y, z)$ on the other surface :

$$\mathbf{y} = \mathbf{x} - D(\mathbf{x}) \frac{G(\mathbf{x})}{\|G(\mathbf{x})\|} \quad with \quad G(\mathbf{x}) = \left[\frac{\partial D(\mathbf{x})}{dx}, \frac{\partial D(\mathbf{x})}{dy}, \frac{\partial D(\mathbf{x})}{dz} \right] \tag{8.9}$$

Table 8.1 Surface registration using SC and ICP iterations. Regularized by a R times refined B-spline grid ranging from $[7, 7, 8]$ to $[204, 204, 279]$. With Dice coefficient, and mean and variance between manually and registered landmarks.

#	Type	R	Mean	Var	Dice
0	start	0	18.5	5.6	0.18
1	SC	1	8.1	3.0	0.44
2	SC	2	6.2	2.5	0.57
3	SC	3	4.1	1.8	0.74
4	SC	3	3.6	1.8	0.78
5	SC	4	3.2	1.6	0.87
6	SC	4	3.1	1.6	0.89
7	SC	5	2.8	1.5	0.92
8	SC	5	2.7	1.5	0.92
9	ICP	5	2.5	1.4	0.94
10	ICP	5	2.4	1.4	0.95
11	ICP	6	2.4	1.4	0.95
12	ICP	6	2.4	1.4	0.95
13	ICP	7	2.4	1.5	0.95
14	ICP	7	2.5	1.4	0.95

In practice we only want to match points with surfaces with approximately the same surface-normal. Therefore we uniformly sample a sphere with radius one with twelve vertices $V_i = [v_x, v_y, v_z]$ $i = 1, 2, ..12$. We also calculate the surface normal $N_B(i, j, k)$ for the pixels in our binary volume V_B and calculate the vertex normals N_P of all the points. Now we match all points for which $V_1 \cdot N_P(\mathbf{x}) \geq \alpha$ to a binary volume with only the boundary pixels for which $V_1 \cdot N_B(i, j, k) \geq \frac{1}{2}\alpha$. With α a constant set to $\alpha = V_i \cdot V_{i+1}$. In this way we can do the point surface matching separately for all 12 orientations.

We first perform point to plane matching from all points in P to the boundary of Q, and then we match Q to P. The results are combined into one FFD which warps P. The symmetric matching is done to get faster convergence and a more robust registration.

8.4 Results

We use a database containing 10 Cone-Beam CT scans, The scans contain $400 \times 400 \times 550$ voxels with a resolution of $0.4 \times 0.4 \times 0.4mm$. Some of the scans are from patients without teeth, causing severe erosion of the mandible. We segment the mandible from the data sets in to binary volumes, with manual initialized 2D active contours. Followed by Marching-Cubes to get the triangulated surfaces geometry. For our correspondence point matching we select the first data set containing $47,858$ vertices, and register it to all other data sets using the following steps. First eight iterative shape context steps, followed by six ICP iterations. During the iterations the alignment improves and number of false matched points decreases, thus we start with a small number of B-spline knots and refine the grid with each iteration, lowering the regularization. We use the Dice coefficient $s = 2|V_1 \cap V_2|/(|V_1| + |V_2|)$ to measure the similarity between the volume defined by the original triangulated surface, and the volume of the fitted triangulation. We also manually labeled all data sets with 12 points, at major landmark positions, to measure the quality of the correspondences.

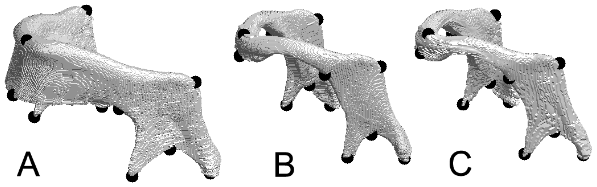


Figure 8.1 A: Surface of data set 1 with landmarks, B: data set 2, C: Surface and Landmarks of data set 1 deformed into data set 2.

Table 8.2 Landmark distance (mm) and Dice coefficient after registration.

ID/L	1	2	3	4	5	6	7	8	9	10	11	12	Mean	Std	Dice
1	2.3	4.0	2.2	1.5	2.6	2.6	1.6	5.1	4.4	0.8	0.6	1.6	2.46	1.40	0.95
2	0.9	3.4	0.7	6.0	1.9	6.9	5.9	0.9	5.9	3.5	1.3	3.2	3.38	2.28	0.97
3	2.2	3.0	2.2	2.9	1.3	0.2	5.4	2.7	3.3	1.9	3.1	3.8	2.67	1.29	0.98
4	1.2	1.2	2.5	6.1	3.9	3.2	5.1	6.2	5.4	2.0	3.3	6.4	3.88	1.39	0.95
5	3.0	0.3	0.5	5.0	3.5	1.6	4.9	5.7	5.7	2.0	1.3	4.2	2.90	1.81	0.97
6	4.5	2.7	3.0	3.6	4.4	3.9	4.6	4.5	2.0	4.5	1.8	1.1	3.38	1.23	0.94
7	3.2	2.5	1.6	1.4	7.6	6.9	6.1	9.1	2.7	4.6	3.4	1.1	4.20	2.64	0.95
8	1.3	2.1	1.7	5.4	10.0	7.4	2.1	2.4	4.4	6.0	0.6	2.9	3.86	2.85	0.97
9	1.7	4.0	2.5	5.4	7.8	6.2	3.4	2.1	7.5	1.4	2.0	1.7	3.82	2.35	0.97

The size of the FFD grid, Dice coefficient, and landmark distance for the first data set through all the iterations, are shown in Table 8.1. Table 8.2 shows the Dice coefficient, and landmark distance for the all data sets after the combined shape context and ICP registration. See figure 8.1 for an example result.

8.5 Discussion and Conclusion

We introduced a method to obtain a dense PCM, by combining 3D shape context with a symmetric ICP method, regularized by a diffeomorphic FFD. The method is not restricted to spherical or closed shape objects.

Robust surface registration with Dice coefficients above 0.94 is obtained, on our mandibular data. We annotated the data set for a second time to find the intra-observer variability and found a distance mean of $2.4mm$ with a standard deviation of 1.88. Distances between registered and user applied landmarks are in the order of $3.4mm$ which is $1mm$ larger than the intra-observer distance.

Future research on a larger set of data with accurate ground-truth landmarks is needed. Allowing comparison between the performance of the introduced method and the existing MDL method and other PCM methods.

9 Minimum description Length

9.1 Introduction

In chapter 5 we introduce an active shape model (ASM), which is a popular method for robust segmentation using the learned shape a priori knowledge. An ASM is based on principal component analysis (PCA) of corresponding points on object contours in training images. The model learns the variations in shape between objects in annotated data sets of patients. Manual annotation of 2D images to obtain corresponding points is possible, but time consuming. Manual annotation of corresponding points in 3D images is difficult, requires volume or surface render techniques, and require thousands of points to accurately describe for instance the surface of the mandible. This introduces the need for automated methods for constructing corresponding models, such as the shape context registration in chapter 8. Several other automated methods to find the correspondences in 3D have been proposed. Brett et al. [94] introduced points matching based on Polyhedral-Based Correspondences, Yongmei et al [95] uses a combination of geodesic distance and surface curvature. These methods in literature are all based on geometric properties of the shape. But there are also another class of methods, which try to minimize the description length of an active shape model. If corresponding points are on exact the same relative location on every object contour, there will be less variance in the active-shape model then if points have no accurate correspondence. The ASM description length gives a measure of the variance in the training data. Thus minimum description length (MDL) can be used as cost function, for moving contour points to optimize the correspondences. The first MDL algorithm was introduced by Davies et al. [86]. In this chapter, we present an robust MDL method for 1D, 2D and 3D corresponding point optimization, which extends on the work of Heimann et al. [96].

9.2 Minimizing Description Length

First a short description of principal component analysis for active shape models (ASM), for more information see chapter 5. An ASM models learns from training images in which the object contour or surface is described by line-pieces or a surface mesh. The contour or surface is then automatically or manually annotated by points which all have corresponding points through all training images. Then there is often a shape alignment step, which removes translation, rotation and size differences between the training images. The aligned (x, y) or (x, y, z) positions of the landmark points in one data set are then grouped into a column vector:

$$\mathbf{x}_1 = (x_1, x_2, .., x_n, y_1, y_2, .., y_n, z_1, z_2, .., z_n)^T \tag{9.1}$$

The position vectors of all training contours are then grouped into one matrix:

$$X = (\mathbf{x}_1, \mathbf{x}_2, .., \mathbf{x}_n) \tag{9.2}$$

PCA can be applied on the contour using singular value decomposition (SVD), but first we need to normalize the area. This is done by subtracting a matrix $\bar{X}$with all the columns set to the mean of X and scaling by the number of samples s.

$$\dot{\mathbf{X}} = \frac{1}{\sqrt{s-1}}\left(\mathbf{X} - \bar{\mathbf{X}}\right) \tag{9.3}$$

Singular value decomposition of the matrix $\dot{X}$,

$$\dot{X} = \mathbf{\Phi}_s \Sigma V^T \tag{9.4}$$

Σ is an $m \times n$ diagonal matrix with nonnegative real numbers on the diagonal and V is an $n \times n$ unitary matrix. The eigenvalues are $\lambda = \left(\Sigma^2_{(1,1)}, \Sigma^2_{(2,2)}, .., \Sigma^2_{(n,n)}\right)$ and Φ_s a unitary matrix containing the eigenvectors.

Now we have obtained a shape model, which can convert any example contour to model parameters. Reducing a large number of contour landmarks $\mathbf{x}$ to a small number of parameters b:

$$b = \Phi_s^T(\mathbf{x} - \bar{\mathbf{x}}) \tag{9.5}$$

These model parameters can be converted back to contour coordinates by:

$$\tilde{\mathbf{x}} = \bar{\mathbf{x}} + \Phi_s b \tag{9.6}$$

The new contour coordinates $\tilde{\mathbf{x}}$ is an estimate of the original contour $\mathbf{x}$, constrained by the variances in shapes learned from the training data.

The PCA model analyzes the shapes using the corresponding points in the images. The quality of correspondence influences the amount of variance, if points are located on exact the same relative location the variance in the model will be low, and if inaccurate located the variance will be high. The amount of variance can be described by the description length. There are several equations which give a cost function for the description length. We define our cost function for the description length F as:

$$F = \sum_m \mathcal{L}_m \text{ with } \mathcal{L}_m = \begin{cases} 1 + log(\lambda_m/\lambda_{cut}) & \text{for} \quad \lambda_m \geq \lambda_{cut} \\ \lambda_m/\lambda_{cut} & \text{for} \quad \lambda_m < \lambda_{cut} \end{cases} \tag{9.7}$$

This equation contains λ_{cut} which describes a threshold between contour-signal and contour-noise. The contour-noise is often caused by discretization effects or/and a inaccurate description of the object contour

$$\lambda_{cut} = \left(\frac{\sigma}{\bar{r}}\right)^2 \tag{9.8}$$

With σ the standard deviation of the contour noise and $\bar{r}$ the average radius of the training shapes. In our experiments we use a constant value $\lambda_{cut} = 10^{-5}$.

To find the minimum description length, we need to find the minimum of our cost function F. Cost minimization algorithms which are memory efficient and relatively fast with a large amount of unknowns are steepest-decent and quasi-Newton optimization. They both depend on the gradient of the cost function.

Therefore we want to calculate the derivative of the singular values of the SVD using the approach of Heimann et al. [96]:

$$\frac{\partial d_m}{\partial \dot{x}_{ij}} = \phi_{im} v_{jm} \tag{9.9}$$

With d_m is the m-th singular value which is related to the eigenvalue by $\lambda_m = d_m^2$. In this equation $\dot{x}_{ij}$ is a value in the normalized contour coordinates matrix $\dot{X}$, and ϕ_{im} and v_{jm} matrix values from SVD matrices $\mathbf{\Phi}_s$ and V. We can use this to calculate the derivative of our cost function:

$$\frac{\partial F}{\partial \dot{x}_{ij}} = \sum_m \frac{\mathcal{L}_m}{\partial \dot{x}_{ij}} \text{ with } \frac{\mathcal{L}_m}{\partial \dot{x}_{ij}} = \begin{cases} 2\phi_{im} v_{jm}/d_m & \text{for} \quad \lambda_m \geq \lambda_{cut} \\ 2\phi_{im} v_{jm} d_m/\lambda_{cut} & \text{for} \quad \lambda_m < \lambda_{cut} \end{cases} \tag{9.10}$$

Obtained using the following analytical derivatives:

$$\begin{aligned} \tfrac{1+log(\lambda_m/\lambda_{cut})}{\partial \dot{x}_{ij}} &= \tfrac{log(\lambda_m)}{\partial \dot{x}_{ij}} = \tfrac{log(d_m^2)}{\partial \dot{x}_{ij}} = 2\tfrac{\partial d_m}{\partial \dot{x}_{ij}} \tfrac{1}{d_m} \\ \tfrac{\lambda_m/\lambda_{cut}}{\partial \dot{x}_{ij}} &= \tfrac{d_m^2/\lambda_{cut}}{\partial \dot{x}_{ij}} = \tfrac{2}{\lambda_{cut}} \tfrac{\partial d_m}{\partial \dot{x}_{ij}} d_m \end{aligned} \tag{9.11}$$

The SVD gradient $\frac{\partial F}{\partial \dot{X}}$ is linearly related to the gradient of individual contour points$\left[\frac{\partial F}{\partial x_i}, \frac{\partial F}{\partial y_i}, \frac{\partial F}{\partial z_i}\right]$. We cannot use the gradient directly to optimize individual points, for two reasons. The first is that landmark points may freely move, but only on the object surface or contour. Secondly, the absolute minimum description length is found when all points move to one location, collapsing into a single point. But then the landmarks points do not describe the shape of the object any longer.

9.3 Mapping

As said in the previous section, we need to regularize the movement of points when minimizing description length. This regularization is needed to stop points collapsing together and to keep the points on the object surfaces. One way to achieve is this goal, is creating an in between mapping. We first convert MDL gradients into point movements on a line, circle or sphere, and then map those point movements back to movements of points on the original shapes. A simple geometric object like a line, allows movement smoothing, to keep points from collapsing, more global MDL optimization, and multi-scale approaches.

9.3.1 Mapping to a Line

Suppose we have s data sets each containing a contour line $L = [L_1, L_2, .., L_s]$. With a contour line L_i consisting of n_i connect points $L_i = [\mathbf{p}_1, \mathbf{p}_2..., \mathbf{p}_{n_i}]$, with $\mathbf{p}_j = [x_j, y_j, z_j]$. We want to find corresponding landmark points on each line.

The first step is to map the points onto the same geometric structure, in this case a 1D line with length one. With point $\mathbf{p}_j$ mapping to line position $\boldsymbol{\alpha}_j$ and range $\boldsymbol{\alpha} \in [0, 1]$. For the initial $\boldsymbol{\alpha}$ values we normalize the line distance $\mathbf{d}$ of a contour line.

$$\mathbf{d}(j) = \mathbf{d}(j-1) + ||\mathbf{p}_j - \mathbf{p}_{j-1}|| \text{ with } j = [2, 3, .., n_i] \text{ and } \mathbf{d}(1) = 0 \tag{9.12}$$

$$\boldsymbol{\alpha}(j) = \frac{1}{\mathbf{d}(n)}\mathbf{d}(j) \text{ with } j = [1, 2, .., n_i] \tag{9.13}$$

After we mapped the contour line of every data set to the 1D line, we create m uniform sampled landmark points $\boldsymbol{\alpha}_{lm}$ on the 1D line.

$$\boldsymbol{\alpha}_{lm}(j) = \frac{j-1}{m-1} \text{ with } j = [1, 2, .., m] \tag{9.14}$$

The uniform landmark points of all the data sets can then be grouped into one matrix $\mathcal{A} = [\alpha_{lm(1)}, \alpha_{lm(2)}..., \alpha_{lm(s)}]$.

A contour line consist not only of points $\mathbf{p}$ but also of line pieces connecting the points. These can be described by an array F, in which each row gives the indices of two connected points .

$$F = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ .. & .. \\ n-1 & n \end{bmatrix} \tag{9.15}$$

Now if we want to map a certain landmark point $\boldsymbol{\alpha}_{lm}(j)$ to x, y, z coordinates $\mathbf{p}_{lm}(j)$, the first thing is to find the closest point on the 1D line $\boldsymbol{\alpha}(j)$. This point is part of one or two line pieces. We check which of the two line pieces is the one which encloses $\boldsymbol{\alpha}_{lm}(j)$. Now we can calculate an interpolation value r on this line piece F_j:

$$r = \frac{|\boldsymbol{\alpha}(F_{j1}) - \boldsymbol{\alpha}_{lm}(j)|}{|\boldsymbol{\alpha}(F_{j1}) - \boldsymbol{\alpha}_{lm}(j)| + |\boldsymbol{\alpha}(F_{j2}) - \boldsymbol{\alpha}_{lm}(j)|} \tag{9.16}$$

The $\mathbf{p}_{lm}(j)$ coordinate is:

$$\mathbf{p}_{lm}(j) = \mathbf{p}(F_{j1})(1-r) + \mathbf{p}(F_{j2})r \tag{9.17}$$

When we warp all landmark points of one data sets to x, y, z coordinates, we can re-arange them in to a vector $\mathbf{b}_i$. If we stack the vectors of all data sets together we get matrix B:

$$B = (\mathbf{b}_1, \mathbf{b}_2, .., \mathbf{b}_s) \tag{9.18}$$

If we calculate the MDL gradient of B we get $\frac{\partial F}{\partial B_{ij}}$, which we want to translate in landmark movements $\frac{\partial F}{\partial \mathcal{A}_{ij}}$:

$$\frac{\partial F}{\partial \mathcal{A}_{ij}} = \frac{\partial F}{\partial B_{ij}} \frac{\partial B_{ij}}{\partial \mathcal{A}_{ij}} \tag{9.19}$$

Finite differences can be used to obtain $\frac{\partial B_{ij}}{\partial \mathcal{A}_{ij}}$, by mapping both $\alpha_{lm}(j)$ and $\alpha_{lm}(j)+\Delta h$ to xyz coordinates, with $\Delta h > 0$ set to a small value.

Now we have obtained the gradient $\frac{\partial F}{\partial \mathcal{A}_{ij}}$ we can do a step into the direction of this gradient to minimize the description length. The size of the step can be determined using line-search.

Instead of updating the landmark-positions on the 1D line, we can also move the contour-points on the 1D line describing the mapping to the contours. See figure, 9.1.The main advantage is that the mapping stays valid if we change the number of landmarks. We use a cropped Gaussian kernel to map the movement from the landmark points to the movement of contour points:

$$c(d) = \begin{cases} e^{\frac{-d^2}{2\sigma^2}} - e^{\frac{-(3\sigma)^2}{2\sigma^2}} & \text{for } d < 3\sigma \\ 0 & \text{for } d \geq 3\sigma \end{cases} \tag{9.20}$$

The variable d gives the distance between the landmark-position and the contour-point position on the 1D line. The update of a contour point $\boldsymbol{\alpha}(k) = \boldsymbol{\alpha}(k) + \Delta\boldsymbol{\alpha}(k)$ of data set i with step length w is:

$$\Delta\alpha_{lm}(j) = \frac{\partial F}{\partial \mathcal{A}_{ij}} w \tag{9.21}$$

$$\Delta\bar{\alpha}_{lm} = \frac{1}{m} \sum_{j=1}^{m} \Delta\alpha_{lm}(j) \tag{9.22}$$

$$\Delta\boldsymbol{\alpha}(k) = -\frac{\sum_{j=1}^{m} c(|\boldsymbol{\alpha}(k) - \boldsymbol{\alpha}_{lm}(j)|)(\Delta\alpha_{lm} - \Delta\bar{\alpha}_{lm})}{\sum_{j=1}^{m} c(|\boldsymbol{\alpha}(k) - \boldsymbol{\alpha}_{lm}(j)|)} - \Delta\bar{\alpha}_{lm} \tag{9.23}$$

After updating the contourpoints $\boldsymbol{\alpha}$ on the 1D line we have to normalize them again to the range $[0, 1]$:

$$\boldsymbol{\alpha} = \frac{\boldsymbol{\alpha} - \min(\boldsymbol{\alpha})}{\max(\boldsymbol{\alpha}) - \min(\boldsymbol{\alpha})} \tag{9.24}$$

The MDL process can be described by the follow steps:

1. Align the contourlines of the data sets, for example by using the eigenvectors of the x, y, z data.
2. Map the points which make up the aligned contourlines to 1D lines with range $[0, 1]$
3. Put evenly spaced landmark points on the 1D lines
4. Warp the landmark points back to the x,y,z coordinates (of the aligned data sets)
5. Calculate the description-length value and x,y,z gradient
6. Map the x,y,z gradient to the 1D line gradients of the data sets
7. Instead of moving the landmark points, use a Gaussian kernel to convert the landmark movement to inverse movement of the contour points on the 1D line, thus updating the mapping from 1D lines to contour lines. Do a line-search for the optimal update step. This involves changing the mapping and re-warping of the landmark points for several step lengths, to fit a polynomial which gives an approximation of the step size with lowest description length.
8. Goto step 3, until the decrease in description length is smaller than a certain value
9. Decrease the sigma used for the Gaussian interpolation kernel
10. Goto step 3, until the sigma is smaller than a certain value.

During optimization it is possible that the contour folds. Folding can be detected by calculating the forward difference between contour points, if the difference is zero or negative folding occurs. This can be prevented by using a smaller update step, or by locally increasing the sigma of the Gaussian interpolation kernel. By keeping every update of the map smooth and fold free also the total transformation will contain no folding. Because the composition of diffeomorphic transformations is by definition also diffeomorphic.

9.3.2 Mapping to a Circle

Minimizing the description-length using a circle mapping follows the same steps as MDL on a line, but then for closed contour lines.

- When we map points to a circle we normalize the range to $\boldsymbol{\alpha} \in [0, 2\pi]$. For the contour points we calculate $\boldsymbol{\beta} = 2 \arctan \frac{(y-\bar{y})}{\sqrt{(x-\bar{x})^2+(y-\bar{y})^2}+x-\bar{x}}$ Then rotate α (update with a constant), to make the difference between α and β as small is possible.
- Before updating the points we normalize the update $\Delta\boldsymbol{\alpha}$ to the range $[-\pi, \pi]$ by adding or subtracting factors of 2π.

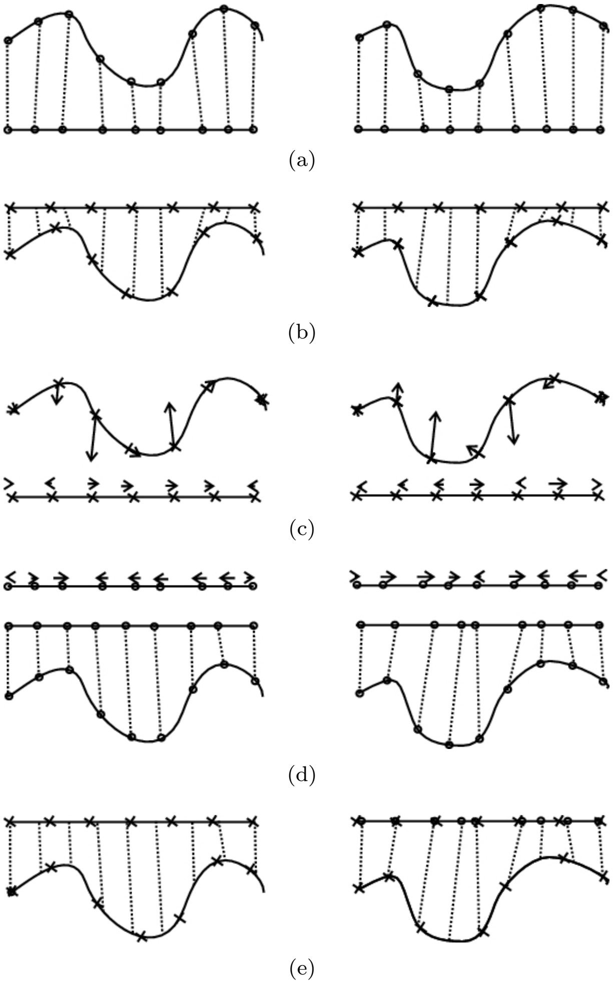


Figure 9.1 Line MDL. Two data sets of a curved line both described by a few points (x,y) are mapped to a 1D line (a). Uniform sampled landmarks are placed on the 1D line, and mapped back to (x,y) coordinates (b). The MDL gradient is calculated for the landmark points (c). Instead of moving the landmarks the mappings described by the curved-line points are updated (d). Uniform sampled landmarks are placed and mapped. Now the description-length of the landmark-model is lowered (e).

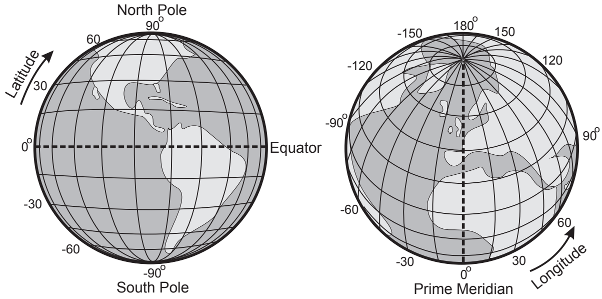


Figure 9.2 Geographic coordinate system of the earth. Every coordinate is defined by a latitude and longitude value.

9.3.3 Mapping to a Sphere

We want to map a triangulated surface of an closed surface object to a sphere. The earth has a spherical shape. The commonly used geographic coordinate system, describes locations on the earth by longitude and latitude value pairs, see figure 9.2. The triangulated surface consist of vertices V in X, Y, Z coordinates, and faces F in which connect the vertices. The vertices mapped to a sphere can be described by U in latitude and longitude coordinates. Thus essentially this is a mapping from $V \in \mathbb{R}^3$ to $U \in \mathbb{R}^2$. Thus mapping to a sphere is approximately equal to mapping an $3D$ object to a cylinder or a plane.

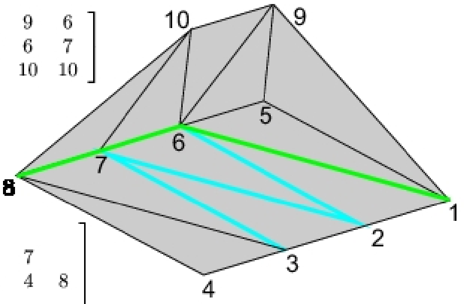
The latitude on the North pole is -90 degrees and on the South Pole 90 degrees. To define the longitude a reference line is chosen, the meridian, running from the north to the South Pole. At the meridian the longitude is zero, if we go to westward the Longitude decreases till -180 degrees, if we go to the east the longitude increases till 180 degrees. At the poles the longitude is undefined, and the latitude makes a jump from -90 to 90 degrees.

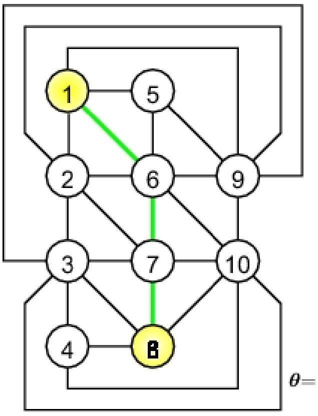
The first step to map an triangulated object surface to a sphere is defining a north and South Pole [97]. For this reason we calculate the Euclidean distance between all points in V, and define the points which are farthest apart as north V_n and south V_s pole. These two points can also be chosen in another way, but they must be far apart, to roughly preserve surface area. Instead of using the direct maximum Euclidian distance in the mesh as in [97], we use the approximately maximum distance through the mesh. Because that will result in an better spherical warp in the end. To do this we select a random vertex of the mesh, and use Dijkstra's algorithm [98] to find the point which is furthest away from the vertex. We use the Euclidian length of the mesh edges as cost function. After finding that point we repeat the method four times. The poles are then selected as the points found in the last two Dijkstra iterations.

The next step is to calculate a latitude value for all vertices, starting from the South Pole which we set to a latitude of 0, till the North pole which we set to π. This is done by formulating the corresponding continues problem as Laplace's equation $\nabla^2\theta = 0$. We solve this equation numerically by setting the North pole to $\theta_n = \pi$ and

$$F = \begin{bmatrix} 5 & 6 & 6 & 7 & 7 & 8 & 1 & 1 & 5 & 3 & 10 & 10 & 2 & 9 & 9 & 6 \\ 1 & 1 & 2 & 2 & 3 & 3 & 5 & 9 & 6 & 10 & 7 & 8 & 9 & 10 & 6 & 7 \\ 6 & 2 & 7 & 3 & 8 & 4 & 9 & 2 & 9 & 4 & 8 & 4 & 3 & 3 & 10 & 10 \end{bmatrix}$$

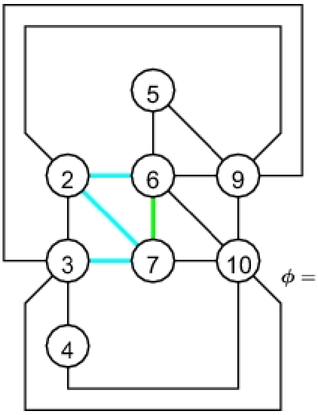
$$V = \begin{bmatrix} 0 & 2 & 4 & 6 & 0 & 2 & 4 & 6 & 2 & 4 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 \end{bmatrix}$$





$$N = \begin{bmatrix} 6 & 5 & 9 & 2 & & \\ 6 & 1 & 9 & 3 & 7 & \\ 7 & 2 & 9 & 10 & 4 & 8 \\ 8 & 3 & 10 & & & \\ 1 & 6 & 9 & & & \\ 5 & 1 & 2 & 7 & 10 & 9 \\ 6 & 2 & 3 & 8 & 10 & \\ 7 & 3 & 4 & 10 & & \\ 1 & 5 & 6 & 10 & 3 & 2 \\ 4 & 3 & 9 & 6 & 7 & 8 \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} 0 \\ 1.25 \\ 1.96 \\ 2.35 \\ 0.79 \\ 1.18 \\ 1.90 \\ \pi \\ 1.19 \\ 1.95 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 5 & -1 & 0 & 0 & -1 & -1 & 0 & -1 & 0 \\ 0 & -1 & 6 & -1 & 0 & 0 & -1 & -1 & -1 & -1 \\ 0 & 0 & -1 & 3 & 0 & 0 & 0 & -1 & 0 & -1 \\ -1 & 0 & 0 & 0 & 3 & -1 & 0 & 0 & -1 & 0 \\ -1 & -1 & 0 & 0 & -1 & 6 & -1 & 0 & -1 & -1 \\ 0 & -1 & -1 & 0 & 0 & -1 & 5 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & -1 & -1 & 0 & -1 & -1 & 0 & 0 & 6 & -1 \\ 0 & 0 & -1 & -1 & 0 & -1 & -1 & -1 & -1 & 6 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \pi \\ 0 \\ 0 \end{bmatrix}$$



$$\phi = \begin{bmatrix} . \\ 5.17 \\ 3.94 \\ 3.09 \\ 2.08 \\ 1.22 \\ 0 \\ . \\ 2.93 \\ 2.24 \end{bmatrix} = \begin{bmatrix} . & . & . & . & . & . & . & . & . & . \\ . & 4 & -1 & 0 & 0 & -1 & -1 & . & 0 & 0 \\ . & -1 & 5 & -1 & 0 & 0 & -1 & . & -1 & -1 \\ . & 0 & -1 & 2 & 0 & 0 & 0 & . & 0 & -1 \\ . & 0 & 0 & 0 & 2 & -1 & 0 & . & -1 & 0 \\ . & -1 & 0 & 0 & -1 & 5 & -1 & . & -1 & -1 \\ . & -1 & -1 & 0 & 0 & -1 & 4 & . & 0 & -1 \\ . & . & . & . & . & . & . & . & . & . \\ . & -1 & -1 & 0 & -1 & -1 & 0 & . & 5 & -1 \\ . & 0 & -1 & -1 & 0 & -1 & -1 & . & -1 & 5 \end{bmatrix}^{-1} \begin{bmatrix} . \\ 4\pi \\ 2\pi \\ 0 \\ 0 \\ -2\pi \\ -4\pi \\ . \\ 0 \\ 0 \end{bmatrix}$$

Figure 9.3 Example of triangulated mesh to sphere mapping (latitude, longitude). The pole vertices are shown in yellow (vertex 1 and 8). Steepest latitude route between poles in green, and western neighbors connections in blue.

$\theta_s = 0$, and then assign the latitude θ of each vertex, to the average of its neighbour vertices. We can write this as a linear systems of equations, see [97] and figure 9.3.

The third step is to define a meridian line, before we can define the longitude. This is done by starting at the North pole vertex, and to find the neighbor vertex with the lowest latitude value. Then we move to that neighbor and step again to the vertex with the lowest value. In this way we walk from the north to South Pole, creating a line of vertices, which describes a shortest path along the mesh. The meridian line is set to this shortest path.

The fourth step is to define for every neighbor vertex of the meridian line if it is a east or western neighbor. A face consist of three vertices, if two of the vertices are part of the meridian line and one is neighbor, then we can flip the face to make the order of vertices in the face equal to order of vertices on the line. If the face needs to be flipped it is an eastern neighbor otherwise a western neighbor.

The last step to calculate the longitude starts with removing the poles from the connected vertices, because longitude is undefined for the poles. We use the same Laplace's equation as by the Latitude. Because the longitude wraps around at 2π to 0 we do not just use the mean of the neighbors longitudes ϕ add the vertices belonging to the meridian line but add $+2\pi$ for every western neighbor connection. For western neighbor vertices we add -2π for every connection with meridian line vertices. Before solving the linear system we have to determine set a arbitrary vertex value to zero (defines the orientation/meridian) in the linear system. We set the value of the first vertex from the meridian line to zero. This can be done by adding a nonzero value to the position of the vertex on the first row of the linear system.

Now the triangular mesh is mapped to the sphere, but it does not preserve angles or surface. It is not possible to preserve surface, but we can optimize the mesh to make it more surface preserving, by scaling the latitude values. Therefore we start with an uniform mapping where $\mathbf{x} = [0, \pi/12, \pi/6, .., \pi]$ maps to $\mathbf{y} = [0, \pi/n, 2\pi/n, .., \pi]$, but then we change the values in y to minimize a surface preserving cost function. The original latitudes are warped using linear interpolation using the mapping $x \rightarrow y$. Our volume preserving cost function is defined as:

$$\begin{array}{c} \mathbf{d}(i) = \max(\mathbf{y}(i+1) - \mathbf{y}(i), 0) \text{ with } i = 0...n-1 \\ E = \sum_{i=0}^{m} \left(\frac{\mathbf{a}_1(i)}{\sum_{j=0}^{m} \mathbf{a}_1(j)} - \frac{\mathbf{a}_0(i)}{\sum_{j=0}^{m} \mathbf{a}_0(j)} \right)^2 + \frac{1}{n} \sum_{i=0}^{n} \mathbf{d}(i) \end{array} \tag{9.25}$$

With $\mathbf{a}_1$ the area of the faces on the sphere and $\mathbf{a}_0$ the area of the faces in the mesh, and m the number of faces. It contains a distance constraint to keep the mapping from folding over

9.4 Optimizing on a Sphere

For $3D$ triangulated objects such as a mandible, we optimize the points not directly with MDL, because then points will move away from the surface. We use a spherical mapping to constrain the movement of points to the surface. The spherical mapping maps x,y,z to latitude, longitude thus $R^3 \rightarrow R^2$. This Spherical mapping also allows smoothing and interpolation of points during MDL optimization. It consist of the following steps

1. Map the triangulation of the object in all data sets to spheres.

2. Define a number of approximately uniform distributed landmarks on each sphere

3. Map the uniform landmarks back to object coordinates. This can be done by starting a ray from the landmark on the sphere to the center of the sphere. Calculate the intersection with the spherical deformed object triangulation. Convert the x,y,z location of the intersection to barycentric coordinates of the intersected triangle. Use the barycentric coordinates to find the x,y,z location on the same face in the original object.

4. Calculate the MDL gradient of the spherical landmarks which where mapped back to the objects coordinates.

5. Convert the x,y,z MDL gradients on the object back to sphere x,y,z gradients

6. Convert the x,y,z sphere gradients to longitude, latitude gradients on the sphere

7. Instead of moving the uniform distributed landmark points with the found longitude and latitude, we warp the vertices of the triangulation in the opposite direction. This is done by interpolation of the gradients of the landmark points to the vertices using a Gaussian kernel. To determine the distance for the Gaussian kernel, we use the circle distance between a vertex and the landmark points. To find the optimal step-size we use a line-search method on the MDL error.

8. We repeat above steps, until convergence. Then we increase the number of uniform distributed landmark points and lower the sigma of the interpolation kernel. In this way we obtain a course to fine MDL registration of corresponding landmark points

If we update the latitude or longitude of a vertex with a certain step value as described above, the resulting Euclidian movement distance will vary depending on the initial place of the vertex. Also there are update problems of the longitude at the poles and meridian. Because the longitude is undefined on the poles, and wraps around at the meridian. Heimann et al. [96], uses truncated Gaussian kernels which do not cover the poles of the sphere. Random rotation matrices are used to frequently change the location of the poles, to still optimize the whole object during the MDL optimization. Disadvantage of this approach is that MDL result is highly dependable on the generated random rotation matrices.

Therefore we use for every landmark point a private coordinate system, in which the landmarks longitude and latitude are chosen $\phi = \pi$ $\theta = 0$, to keep them as far away from the poles and meridian in their own coordinate system as possible. The private coordinate system is defined by a rotation matrix constructed from the position vector of the landmark point pointing to the center of the sphere. Going

Table 9.1 The MDL parameters of the hand photos.

parameter	value	information
σ	$[2.3, 2.0, 1.7, 1.5,$ $1.2, 1.0, 0.7, 0.5, 0.4]$	Sigma of Gaussian interpolation kernel
$n_{Landmarks}$	120	Number of landmarks used for optimization
$step$	10^{-3}	finite difference step
f_{tol}	10^{-2}	Minimum update value, otherwise next scale

from a vector to a rotation matrix is ill defined, thus we use a random vector to fix the rotation around the landmark position vector.

Now if we obtain the x,y,z sphere update of a landmark point, we rotate the object sphere vertices and landmark point with the obtained rotation matrix, convert the update to latitude and longitude update. Then update the object vertex sphere positions by interpolation of the latitude and longitude update from the landmark point using a Gaussian kernel. These steps are repeated for all the landmark points, before going to the next MDL iteration.

9.5 Results

9.5.1 Hand Photos

We have acquired 10 photos of hands with a resolution of 750 × 500. We segment the hand from the photos using manual pixel based segmentation. We have obtained the contour described by line-pieces with marching squares. The number of points in the contours varies from 558 till 754. We place 10 annotation landmarks on position such as the finger tip, in every data set. Then we run the MDL optimization as described before, see the parameters in 9.1.

The mapping of a hand contour to the circle is shown in figure 9.4. We warped the landmarks of one data set to all other data sets using the obtained correspondences, see figure 9.5. The pixel distance between the manual landmarks during the process, and description length value, are shown in 9.6.

9.5.2 Mandible

We manually segmented the mandible in 13 patient CBCT data sets. The data set both contains dentate and edentate mandibles. After manual voxel based segmentation we converted the mandible segmentations to triangulated surfaces descriptions with marching cubes. Then the vertices where used as input for the shape context registration described in the previous chapter 8. The resulting surface descriptions in which all the data sets has the same amount of vertices, and in which the vertices have approximately correspondence between the data sets are used as input of our 3D MDL optimization.

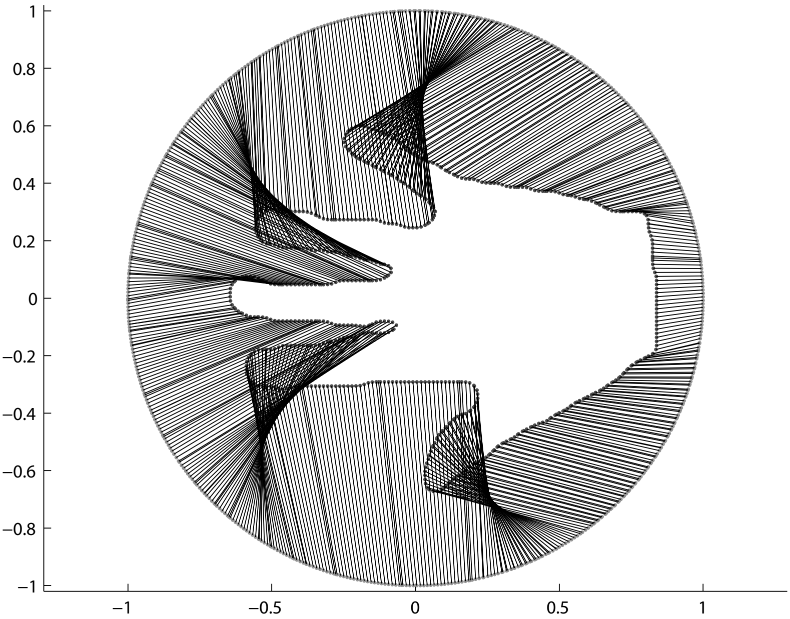


Figure 9.4 Mapping of contour to circle for MDL optimization.

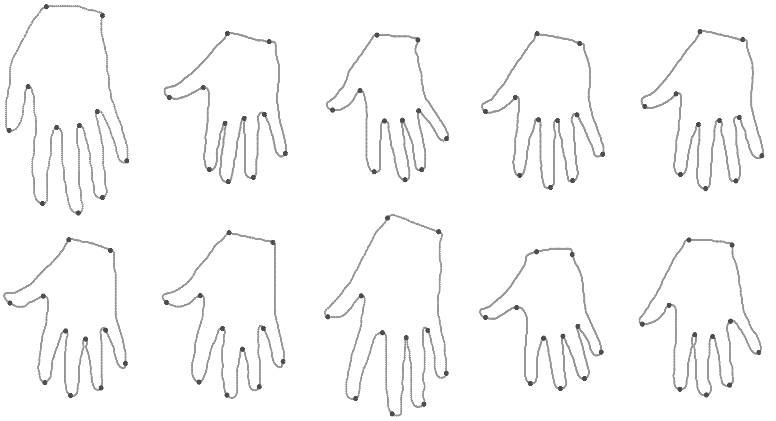


Figure 9.5 Hand Contours after MDL optimization. We warped the manual landmarks of one data set to the other data sets, these points are shown in red.

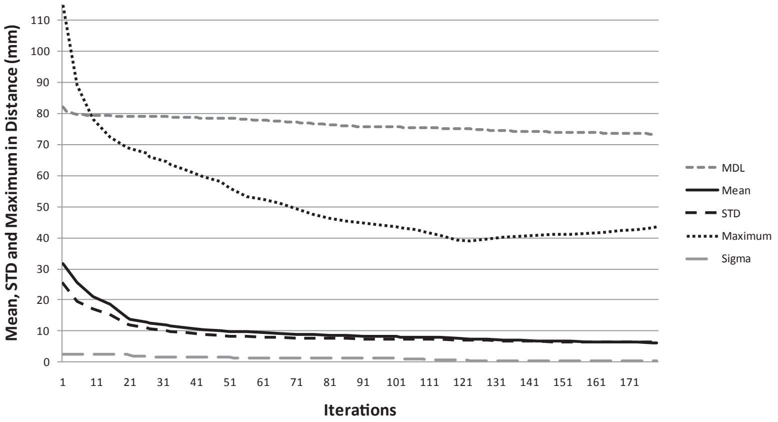


Figure 9.6 Distance between manual landmarks during MDL optimization of the hand contours. Also shown sigma of the interpolation kernel, and description length.

Table 9.2 The MDL parameters of the mandible.

parameter	value	information
σ	$[0.5, 0.4, 0.3]$	Sigma of Gaussian interpolation kernel
$n_{Landmarks}$	2562	Number of landmarks used for MDL optimization
$step$	10^{-3}	finite difference step
f_{tol}	10^{-2}	Miminum MDL value update, otherwise new scale

The shape context registration step for initial correspondences was used, for two reasons. The MDL is slow with a large amount of optimization landmarks, so we optimize with a small amount of landmarks. All data set has the same amount of vertices (11022), thus we can use them to still obtain a dense surface description, with a small amount of landmarks (2562). The second reason, is that the minimum description length is low if points locally cluster into the same coordinate. This decreases the amount of points in other regions, effectively lowering the quality of the object description. This happened when we started from unaligned surface descriptions.

The parameters which where use to obtain the MDL result are shown in table 9.2. Two data sets with the intial MDL landmark positions and final positions are shown in figure 9.7, and description-length in figure 9.8. The figure shows also shows that the Coronoid Process does not contain MDL landmarks. This is due to the fact that the MDL landmarks where approximately uniformly distributed on the sphere. But the mapping was not area-preserving, and the coronoid process was mapped to a very small spherical area, because it is a sharp structure.

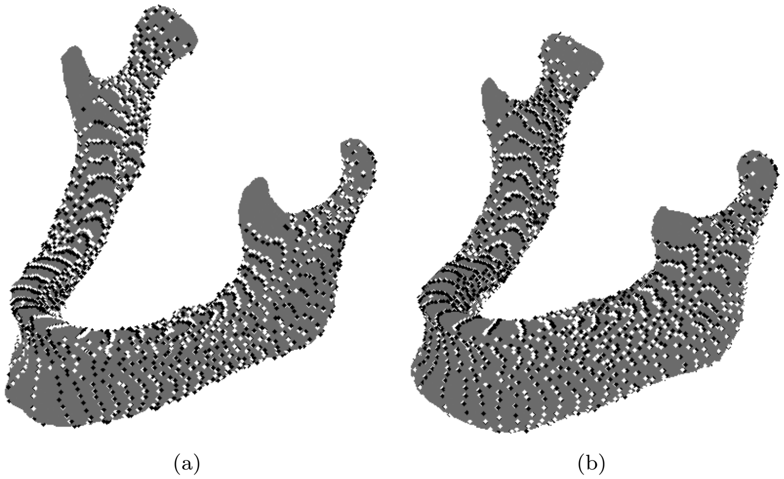


Figure 9.7 Results of the MDL optimization . Showing two of the mandibles, with the black dots the initial MDL landmarks and white dots the landmarks after running the MDL optimization.

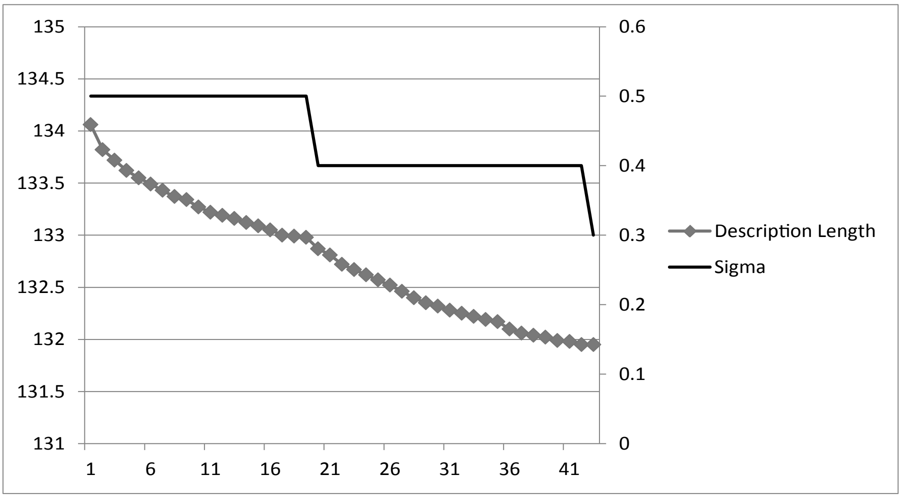
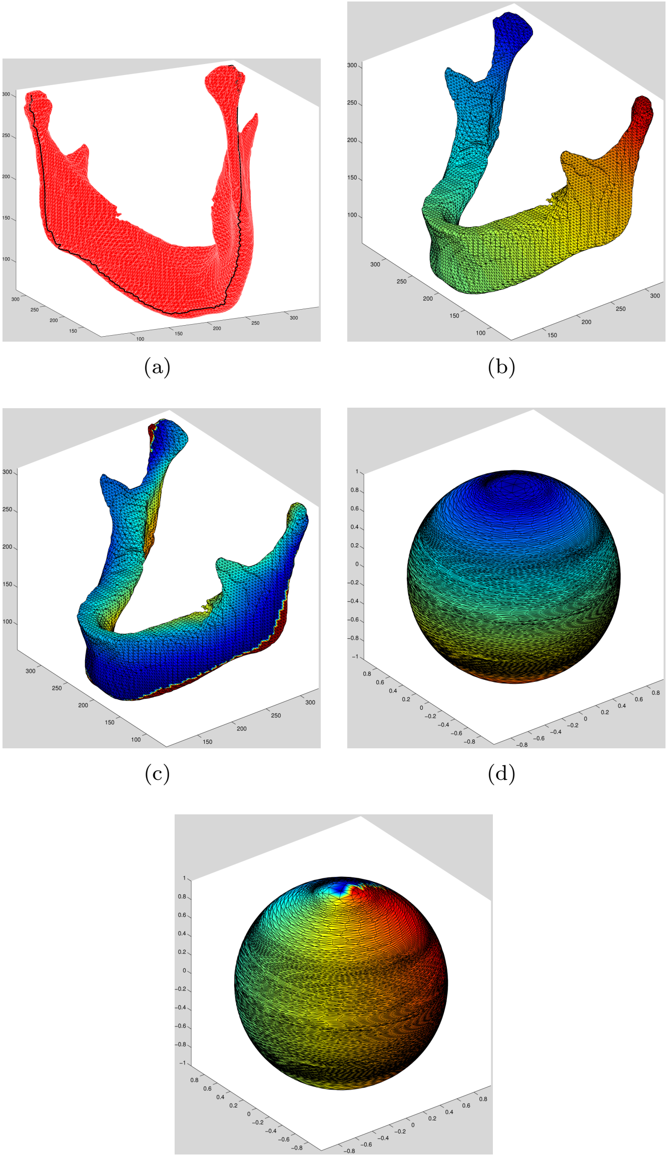


Figure 9.8 MDL optimalization of the mandible surfaces. We show the decrease of description length and sigma of the interpolation kernel, during the MDL iterations.



(a) (b)

(c) (d)

(e)

Figure 9.9 Mapping of the triangulated surface of the mandible to a sphere. Sub-figure a shows the mandible with the Meridian. Sub-figure b and c show respectively the latitude and longitude calculate for the vertices of the mandible, sub-figure d and e the mandible latitude and longitude shown as sphere .

9.6 Discussion

The 3D MDL introduced by Heimann et al. [96] was implemented and extended with some small improvements, as described in this chapter. If we look at the short result section, we see that the algorithm decreases the description length of the model, but has some problems:

- The spherical mapping is not area preserving. This means that the landmarks are approximately uniform distributed on the sphere but not on the mandible itself. Especially small sharp structures will map to small spherical areas, thus will probably not be optimized. The sigma of the Gaussian interpolation kernel is constant but the effect on the object itself will vary duo to the area change after mapping.

- The MDL method not only aligns corresponding points but also causes landmark points to cluster if the smoothing regularization becomes smaller. Because if local points collapse to a single point the description length also becomes smaller, resulting a poor description of the original object.

- The MDL method is based on PCA, thus assumes a Gaussian distribution of the landmarks. From some biomedical-shapes exist no negative variant, for example a human cannot have a negative length. Thus if there is a small elevation in one mandible the method can wrongly align the elevation with a small dip in another mandible.

Our recommendations to solve these disadvantages are:

- Distribute the MDL landmarks uniform on the original objects instead of on the sphere. Then all so spiky objects will contain landmarks.

- Don't use a Gaussian kernel to interpolate the MDL landmark moment to the vertices, but adapt the sigma and shape to the amount of object area described in that part of the sphere.

- Include an intensity or Jacobian based cost function, to the MDL process. To stop landmark points to locally collapse into a single coordinate.

10 Results

We evaluate in this chapter the introduced mandibular canal localization methods. First we use the LK tracking method of chapter 4 and segment the mandibular canal from 13 CBCT data sets. Then we evaluate the performance of the registration methods, B-spline registration from Rueckert et al. [60] and demon registration [66]. We also test the performance of a trained active shape model (ASM) [25] and active appearance model (AAM) [26] on the CBCT data. The performance of the methods is compared on accuracy of mandibular bone segmentation, and on distance to expert segmentation of the mandibular canals.

In the first section we introduce the CBCT training data, and describe the provided expert annotations. In the second section we describe two basic data pre-processing steps. In the third section we use the LK tracking method to segment the mandibular canals in the data sets. In the fourth section we describe the results of the registration methods. Followed by a section with the results of the active shape model, and a section about the active appearance model. Then a section comparing the ASM and AAM method with the registration methods. Followed by a section with methods to improve the ASM result, and a section about improving the AAM result. Finally, a section with the conclusions.

10.1 Data

For our experiments we obtained 13 CBCT data sets from 13 patient heads. These data sets where provided by the department of oral and maxillofacial surgery, Radboud university Nijmegen medical centre in the Netherlands. The mandibular canals of all patients are annotated by medical experts, and data segmented in two classes,

Table 10.1 Description of the CBCT data sets of the patients based on the DICOM information tags.

id	edentate/dentate	age	male/female	resolution
1	e	26	m	0.4
2	d	25	m	0.4
3	d	25	m	0.4
4	e	.	f	0.4
5	e	58	f	0.4
6	d	30	m	0.4
7	e	39	m	0.3
8	d	.	f	0.4
9	d	.	f	0.4
10	d	53	m	0.4
11	e	69	f	0.3
12	e	.	.	0.4
13	d	41	m	0.3

mandible and non-mandible voxels. The annotation of the mandibular canals consist of a list of manual selected connected points in the center of the mandibular canals.

The data was acquired on an i-CAT CBCT scanner, with peak voltage $120kV$, X-ray tube current 48 mA, frame time 67 ms. The Data consist of 6 people without teeth edentate and 7 people with teeth dentate. Sex of the patients, 7 male, 6 female and 1 not specified in the Dicom tags. Age of the patients with known age varies between 26 and 69. For more information see table 10.1. In figure 10.1 we show illustrating images of two patients data sets.

10.2 Pre-processing

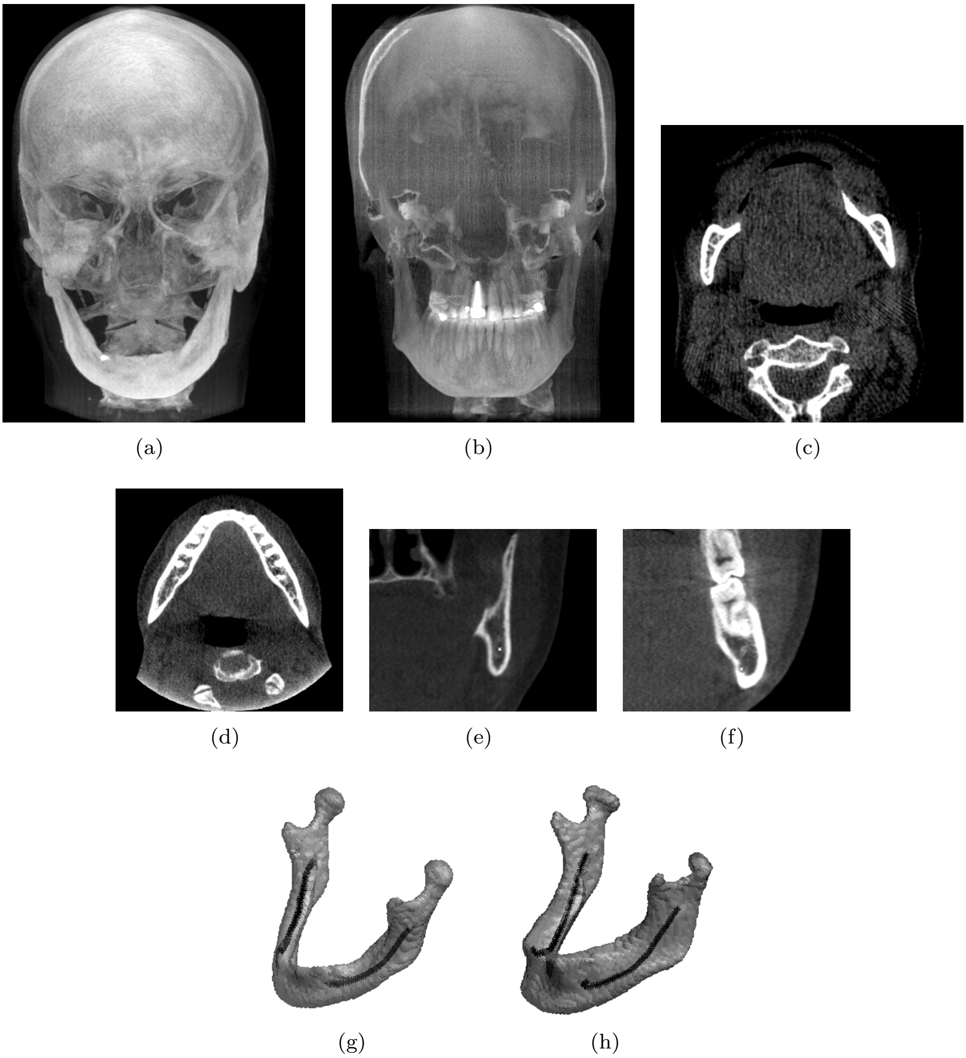
We can describe the distance on a line of connect points by **d**

$$\mathbf{d}(i) = \sum_{i=1}^{n} ||\mathbf{x_i} - \mathbf{x}_{i-1}|| \tag{10.1}$$

To obtain a dense description we re-interpolated the connected line to make it more uniform. We use spline interpolation with as basis variable **d** to create 80 uniform sampled points along every annotated canal. The Hounsfield values for bone and tissue varies highly between data sets in our CBCT data. Therefore we scale the original values to decrease the variance, between data sets:

$$I_h = I_o - \tilde{I}_o \tag{10.2}$$

With I_o the original intensity values in a data set, I_h the corrected Hounsfield units. The variable $\tilde{I}_o$ is the median of all I_o values above -700, thus without air voxels.


(a) (b) (c)

(d) (e) (f)

(g) (h)

Figure 10.1 Example data sets of two patients. Sub figure a and b show MIP (Maximum Intensity Projection) images of both patients, Sub figure c and d a CBCT slice. Sub figure e and f the mandibular canal annotated by a point in a slice. Sub figure g, and h, show the segmented mandible including the annotated mandibular canals of the two patients.

Table 10.2 The LK nerve tracking parameters.

parameter	value	information
s_t	14×14	Template size
s_b	5	border for image derivatives
σ_s	0.7	Image derivative sigma
σ_l	1.5	Image derivative sigma
n_s	3	LK iterations with σ_s, followed by σ_l
n_p	7	samples used to calculate normal

The default parameters in our registration algorithms are experimentally optimized for a gray value range of $0 - 1$. Thus data sets must be first converted to that range, to use the defaults. For our CBCT data we scale from Hounsfield units to gray values with the following equation:

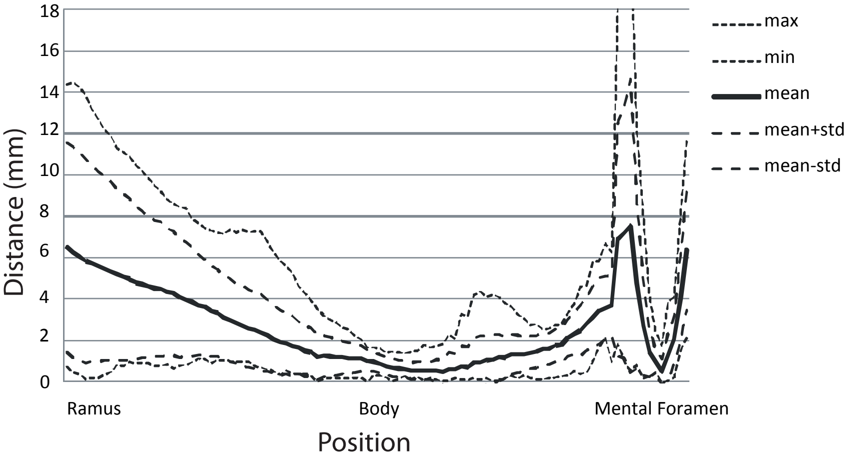
$$I_n = \begin{cases} 0 & \text{if } I_h < -200 \\ 1 & \text{if } I_h > 1500 \\ \frac{I_h+200}{1700} & \text{otherwise} \end{cases} \tag{10.3}$$

The equation has two boundaries the first -200 is chosen to prevent the registration algorithms to focus too much at the skin air boundary during optimization. The second 1500 is chosen to make the registration algorithm more robust against dental implants and other restorative material. By mapping high metal based values closer to the range of bone values.

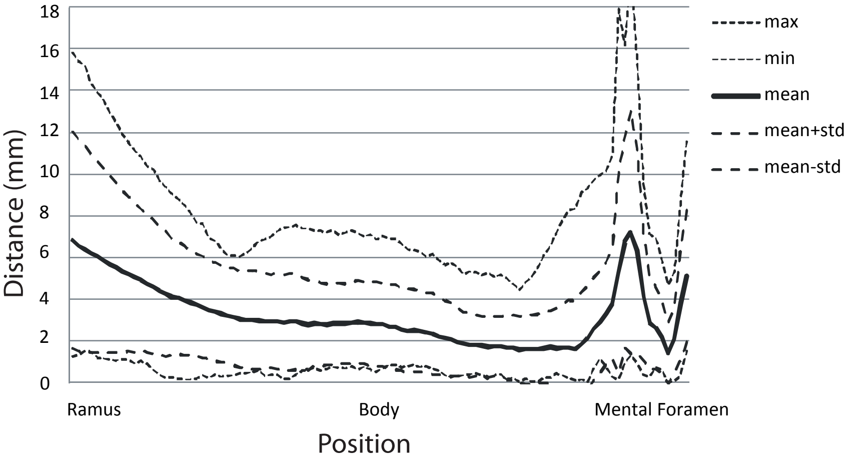
10.3 Experiment Tracking

In this experiment we evaluate the performance of the LK nerve tracking introduced in chapter 4. We use the same tracking parameters as in chapter 4, see table 10.2. The tracking method starts from a user selected point in the mandibular canal. We use the expert annotation of the mandibular canals to pick for every canal a starting point, which is located about $20mm$ from the mental foramen.

The tracking is performed on all the 13 CBCT scans. The tracking does not automatically stop at the ends of the mandibular canal, therefore we manually remove the parts outside the canal. The distance between the tracking results and expert annotations are shown in table 10.3. The average mean euclidian distance to expert annotation is $3.01mm$, with a standard deviation to the mean of $1.34mm$. The average standard deviation of the distance error is $2.42mm$. The high mandibular canal localization error is mainly caused by template drift at the ends of the mandibular canal, see figure 10.2.



(a)



(b)

Figure 10.2 LK mandibular canal tracking accuracy. The table shows the euclidian distance between the mandibular canal annotation from the LK tracking and expert annotation. Subfigure a shows the distance errors of the right mandibular canal, and subfigure b of the left canal.

Table 10.3 LK tracking, mandibular canal localization accuracy. The table shows the euclidian distance between the annotations of the LK tracking method and the expert mandibular canal annotations. Mean, standard deviation and maximum distance are shown for the right and left mandibular canal.

	Right			Left		
Id	mean	STD	max	mean	STD	max
1	2.09	1.22	5.26	2.43	1.70	6.01
2	4.76	4.05	19.62	5.09	2.27	15.42
3	4.59	4.14	20.99	5.19	4.33	20.28
4	4.01	3.44	9.33	3.48	2.06	7.23
5	2.04	0.94	5.28	3.52	1.56	6.43
6	3.49	3.93	21.20	4.01	4.03	21.22
7	1.22	0.96	6.24	1.17	0.96	8.13
8	2.25	2.46	13.75	2.94	3.40	15.65
9	2.48	2.08	8.10	1.72	1.30	6.06
10	1.93	1.87	11.69	4.85	2.02	7.58
11	0.98	0.78	7.16	0.96	0.83	4.58
12	4.24	4.11	16.52	3.98	3.79	16.32
13	2.08	3.23	28.94	2.86	1.42	16.68

10.4 Experiment Registration

In this experiment we use the two introduced registration methods, demon registration and B-spline registration (see chapters 6, 7) to find the mandibular nerve location. These methods are based on aligning two images, in which one of the images is deformed to align to a static image. Most of our data sets consist of a whole head instead of only the mandible, thus our first step is to crop the data sets to only include the mandible. Instead of constructing an mandible atlas from multiple training images, we use a single image as atlas for mandibular canal segmentation.

To select a good atlas image, we first register all the training images to the test image using an affine registration (see chapter 7). Then we swap the roles, and register the test-image to all training images. Then we determine the squared differences (SSD) between registered test image and every atlas image, and every registered atlas image and test image. To get a robust measure of similarity we take the mean of the SSD's of both registration results. The training-image with the lowest SSD is then select as atlas image for the test image, see table 10.4. As you can see in the table, the best match for a dentate data set is usually a dentate data set (77%). The best match for a female is sometimes a male mandible or the other way around.

We test the performance of non-rigid registration on mandible segmentation and mandibular canal segmentation by non-rigid registration of every CBCT data set to their best match. Then we use the B-spline grid or transformation field to also warp the manual mandible segmentation of the segmentation. Performance of the mandible

Table 10.4 Similarity from affine SSD, showing the id's of the six best matches for every data set. Followed by the DICOM properties of each data set, first letter indicates dentate(d) or edentate(e), second letter indicate female(f) or male(m) or unknown(.).

1	2	3	4	5	6	7	8	9	10	11	12	13
12	12	9	12	1	3	13	2	3	12	12	1	10
5	5	13	5	4	2	5	3	2	5	4	4	5
4	1	12	1	12	12	11	5	4	1	5	5	1
10	10	5	11	10	11	1	11	12	4	1	10	12
11	4	1	10	11	9	12	1	10	13	10	11	3
13	11	2	2	13	8	10	12	1	11	2	2	2
em	dm	dm	ef	ef	dm	em	df	df	dm	ef	e.	dm
e.	e.	df	e.	em	dm	dm	dm	dm	e.	e.	em	dm
ef	ef	dm	ef	ef	dm	ef	dm	dm	ef	ef	ef	ef
ef	em	e.	em	e.	e.	ef	ef	ef	em	ef	ef	em
dm	dm	ef	ef	dm	ef	em	ef	e.	ef	em	dm	e.
ef	ef	em	dm	ef	df	e.	em	dm	dm	dm	ef	dm
dm	ef	dm	dm	dm	df	dm	e.	em	ef	dm	dm	dm

Table 10.5 The B-spline registration parameters.

parameter	value	information
Similarity	SSD	Sum of squared differences
MaxRef	2	Refinements of Initial Spacing 32x32x32
Registration	Both	Affine and nonrigid
Interpolation	Linear	Image interpolation

Table 10.6 The Demon registration parameters.

parameter	value	information
Similarity	p	Intensity difference
MaxRef	2	Smallest registration scale 0.25
Registration	Both	Affine and nonrigid
σ_{Fluid}	4	Gaussian smoothing of update field
α	4	Noise constant
σ_{Diff}	1	Gaussian smooth of transformation field
Interpolation	Linear	Image interpolation

Table 10.7 B-spline atlas registration bone segmentation accuracy. Dice coefficient describing the overlap, of the mandible described by the registered atlas, and the manual segmented mandibular bone.

Id test	Id atlas	Dice Coefficient
1	12	0.78
2	12	0.80
3	9	0.78
4	12	0.77
5	1	0.77
6	3	0.71
7	13	0.55
8	2	0.70
9	3	0.83
10	12	0.83
11	12	0.83
12	1	0.79
13	10	0.68

segmentation can then be calculated using the dice coefficient:

$$s = \frac{2|X \cap Y|}{|X| + |Y|} \tag{10.4}$$

With $X \cap Y$ the number of overlapping voxels, and $|X|$ the number of voxels in the static data set and $|Y|$ the number of pixels in the warped manual segmented data set. We can describe the performance of mandibular canal segmentation. By the point to line distance from all mandibular canal points of the registered scan to line of mandibular canal points of the static scan, and the other way around.

The parameters used for the B-spline registration can be found in table 10.5 and the parameters for the demon registration can be found in table 10.6. The Dice coefficients of the registered atlas images to the data sets can be found in table 10.7 and table 10.8. The distance between the manual selected and mandibular canal from the registered atlas image can be found in table and 10.9 table 10.10.

10.5 Experiment Active Shape Model

An active shape model (ASM) in 3D uses surface-normals to search for the ideal landmark position during ASM segmentation. The mandibular canals are described by lines of connect points. Line normals are not fixed but can rotate around the line. To fix this degree of freedom we have to add extra information. Because our surface model is based on a triangulation, we choose to convert the line description of the mandibular canal to a triangulated hollow tube, with a radius of one pixel ($0.4mm$). Therefore we first have to define a normal plane on every point of the line. We do

Table 10.8 Demon atlas registration bone segmentation accuracy. Dice coefficient describing the overlap, of the mandible described by the registered atlas, and the manual segmented mandibular bone.

Id test	Id atlas	Dice Coefficient
1	12	0.83
2	12	0.75
3	9	0.83
4	12	0.71
5	1	0.74
6	3	0.86
7	13	0.54
8	2	0.89
9	3	0.87
10	12	0.72
11	12	0.86
12	1	0.78
13	10	0.54

Table 10.9 B-spline registration, mandibular canal localization accuracy. The table shows the euclidian distance between the warped mandibular canal of the registered atlas and the expert mandibular canal segmentation. Mean, standard deviation and maximum distance are shown for the right and left mandibular canal.

	Right			Left		
Id	mean	STD	max	mean	STD	max
1	3.54	1.76	7.83	4.75	2.88	12.33
2	2.08	1.04	4.20	1.66	2.05	8.34
3	3.65	1.48	11.48	3.32	1.72	10.46
4	1.67	0.70	4.39	2.40	1.52	7.18
5	2.85	1.95	6.22	3.14	1.44	6.51
6	3.88	2.02	6.52	4.32	1.69	7.07
7	8.65	3.79	13.40	12.8	4.32	17.50
8	3.22	1.32	5.18	2.28	2.11	13.04
9	3.28	1.03	8.99	3.10	1.15	8.34
10	2.08	1.03	4.87	2.61	1.23	5.76
11	1.90	0.86	4.93	2.67	1.64	7.29
12	3.43	1.59	8.01	5.97	3.40	15.27
13	5.89	2.37	12.93	6.52	2.89	14.24

Table 10.10 Demon registration, mandibular canal localization accuracy. The table shows the euclidian distance between the warped mandibular canal of the registered atlas and the expert mandibular canal segmentation. Mean, standard deviation and maximum distance are shown for the right and left mandibular canal.

	Right			Left		
Id	mean	std	max	mean	std	max
1	3.63	1.56	9.01	6.31	3.96	19.75
2	2.51	2.02	9.60	2.39	3.46	13.87
3	6.82	2.88	13.48	6.03	2.56	10.55
4	1.90	1.32	5.38	2.48	1.57	6.91
5	3.26	2.27	7.60	2.33	2.00	7.85
6	5.38	3.36	13.83	5.52	3.28	15.64
7	19.50	8.72	32.04	15.48	20.73	19.66
8	3.34	1.13	6.00	3.49	2.97	16.50
9	3.58	1.83	7.73	3.00	1.61	7.66
10	3.98	2.18	8.15	5.26	2.53	9.43
11	2.29	1.37	7.54	2.55	1.99	10.37
12	4.38	1.89	9.08	6.20	3.42	12.50
13	4.05	1.45	6.34	5.06	1.36	8.16

this for the first line piece by:

$$\begin{aligned} \mathbf{v_l} &= \frac{\mathbf{x}_1 - \mathbf{x}_2}{||\mathbf{x}_1 - \mathbf{x}_2||} \\ \mathbf{n}_a &= \frac{\mathbf{v_l} - \frac{\mathbf{v_r}}{\mathbf{v_r}\mathbf{v_l}^T}}{||\mathbf{v_l} - \frac{\mathbf{v_r}}{\mathbf{v_r}\mathbf{v_l}^T}||} \\ \mathbf{n}_b &= \frac{\mathbf{v_l} \times \mathbf{n}_a}{||\mathbf{v_l} \times \mathbf{n}_a||} \end{aligned} \tag{10.5}$$

With $\mathbf{v_r}$ a random normalized vector which is non-collinear with $\mathbf{v_1}$. Now we can construct a circle of vertices $\mathbf{v}_{c1}$ around the line piece by:

$$\begin{aligned} \boldsymbol{\alpha} &= [0, \tfrac{1}{4}\pi, .., 1\tfrac{3}{4}\pi] + \alpha_o \\ \mathbf{v}_{c1} &= \cos(\boldsymbol{\alpha})\mathbf{n}_a + \sin(\boldsymbol{\alpha})\mathbf{n}_b \end{aligned} \tag{10.6}$$

In our case we want the circle vertices $\mathbf{v}_{c1}$ in one data set to be oriented in the same way as in another data set thus we use the constant vector $\mathbf{v_r} = [0, 0, 1]$. We calculate the circle vertices of all other line points in the same way. We choose the rotation offset α_o to minimize the euclidian distance between vertices of the current circle and the previous circle of the mandibular canal. After calculating all circle vertices we connect the neighbor circles by faces, now we have obtain a triangulated description of our mandibular canal. By taking the mean of vertices belonging to the same circle we can obtain a line description by points after ASM segmentation.

Table 10.11 shows the setting we usefor the training our ASM. To test the our ASM we use a leave-one-out scheme, where we use one trainingset as test image and use the other trainingsets to train an ASM. To test mandible segmentation performance we calculate the dice coefficient between the manual mandible classification

Table 10.11 The ASM segmentation parameters.

parameter	value	information
k	8	Length of landmark profile
n_s	6	Search length in pixels
n_{scales}	3	resolution scales
m	3	limit shape to $m\sqrt{\lambda}$
n_{search}	15	Number of iterations

Table 10.12 ASM bone segmentation accuracy. Dice coefficient describing the overlap, of the mandible described by the registered ASM, and the manual segmented mandibular bone.

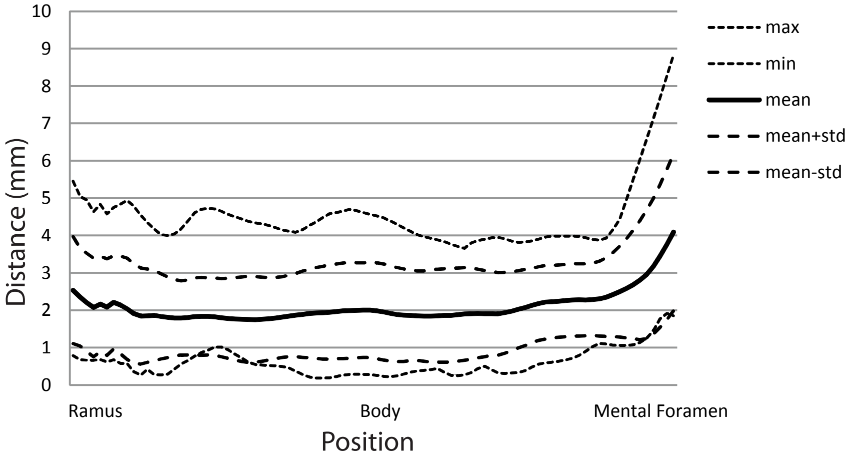
Id test	Dice Coefficient
1	0.72
2	0.86
3	0.82
4	0.67
5	0.69
6	0.77
7	0.68
8	0.72
9	0.76
10	0.76
11	0.77
12	0.76
13	0.77

and discretized registration surface, see table 10.12. We use for the mandibular canal accuracy the line to point distance, from expert annotation to registration result and the other way around, see table 10.13 and figure 10.3.

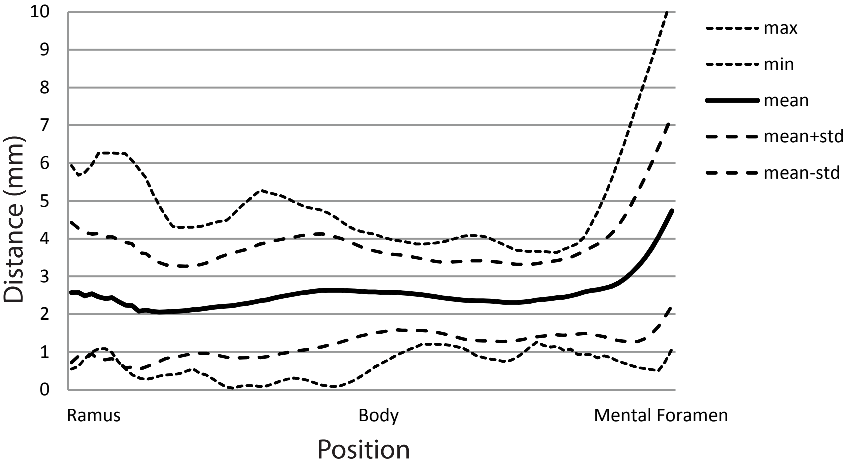
10.6 Experiment Active Appearance Model

We have also tested an active appearance model (AAM). Our landmark points include the object surface vertices and the expert annotation points of both nerve canals (90 points each).

Table 10.14 shows the setting we use For the training our AAM. To test the our AAM we use a leave-one-out scheme, where we use one training set as test image and use the other trainingsets to train an AAM. To test mandible segmentation performance we calculate the dice coefficient between the manual mandible classification and discretized registration surface, see table 10.15. We use for the mandibular canal accuracy the line to point distance, from manual annotation to registration result and the other way around, see table 10.16.



(a)



(b)

Figure 10.3 ASM mandibular canal localization accuracy. The table shows the euclidian distance between the mandibular canal annotation from the ASM model and expert annotation. Subfigure a shows the distance errors of the right mandibular canal, and subfigure b of the left canal.

Table 10.13 ASM mandibular canal localization accuracy. The table shows the euclidian distance between the annotations from the ASM model and expert annotations of the mandibular canal. Mean, standard deviation and maximum distance are shown for the right and left mandibular canal.

	Right			Left		
Id	mean	std	max	mean	std	max
1	1.79	1.02	4.60	3.68	1.65	8.98
2	1.51	0.83	4.70	1.79	0.51	3.24
3	3.35	1.64	8.74	2.75	1.17	6.54
4	2.14	1.31	6.88	3.02	1.27	5.65
5	2.42	1.39	5.28	3.19	2.71	9.94
6	1.55	2.06	16.12	3.21	1.48	9.71
7	1.48	0.58	5.87	2.27	1.59	9.22
8	2.69	1.85	7.66	2.93	1.93	9.57
9	2.95	1.97	9.90	2.46	1.56	8.68
10	2.38	1.14	6.16	2.01	1.15	5.79
11	1.52	0.67	5.64	1.64	1.31	6.08
12	2.24	1.05	4.97	2.32	1.60	6.57
13	1.82	0.93	3.87	1.22	0.85	5.40

Table 10.14 The AAM training and segmentation parameters.

parameter	value	information
n_b	5	Border added (pixels)
s	[150 150 150]	Texture size used (pixels)
n_{scales}	4	Number of AAM image resolutions.
n_{search}	15	Number of iterations

10.7 Methods Comparison

The mandibular bone segmentation accuracy is described by the Dice coefficient, which gives the overlap between the manual and automatic segmentations. The mean and standard deviation of the Dice coefficient from the results of the B-spline, demon, ASM, and AAM segmentation are:

- B-spline registration, mean 0.76, STD 0.08
- Demon registration, mean 0.76, STD 0.12
- ASM, mean 0.75, STD 0.05
- AAM, mean 0.76, STD 0.18

If we look at segmentation of the mandible the methods perform approximately equally well. The ASM method has a slightly lower dice coefficient but also produces slightly more constant results.

Table 10.15 AAM bone segmentation accuracy. Dice coefficient describing the overlap, of the mandible described by the registered AAM, and the manual segmented mandibular bone.

Id test	Dice Coefficient
1	0.71
2	0.88
3	0.84
4	0.73
5	0.80
6	0.85
7	0.17
8	0.78
9	0.81
10	0.83
11	0.81
12	0.87
13	0.82

Table 10.16 AAM mandibular canal localization accuracy. The table shows the euclidian distance between the mandibular canal annotations of the AAM model and expert segmented annotations. Mean, standard deviation and maximum distance are shown for the right and left mandibular canal.

	Right			Left		
Id	mean	std	max	mean	std	max
1	1.68	0.98	4.26	3.87	2.05	8.47
2	1.38	0.70	3.91	1.57	0.59	4.14
3	3.20	0.91	5.94	2.76	1.06	6.10
4	1.79	0.64	4.22	1.48	0.65	4.08
5	1.48	0.84	2.88	1.62	0.89	3.29
6	1.68	1.27	8.32	3.15	1.33	7.95
7	1.65	0.77	3.09	2.61	1.44	10.70
8	1.48	0.91	3.21	1.58	1.10	6.81
9	2.91	1.38	9.44	2.64	1.12	5.12
10	1.66	0.95	6.35	1.55	0.75	3.80
11	1.45	0.85	5.29	1.33	0.92	4.10
12	1.88	1.01	4.60	2.22	1.80	7.54
13	1.29	0.45	3.31	1.77	0.43	3.04

We describe the mandibular canal segmentation accuracy, by the euclidian distance from expert annotation to automatic annotation. The average mean distance and standard deviation of the mean distance, for LK tracking, B-spline, demon, ASM and AAM segmentation are:

- LK tracking, mean $3.01mm$, STD $1.34mm$
- B-spline registration, mean $3.91mm$, STD $2.45mm$
- Demon registration, mean $5.02mm$, STD $3.98mm$
- ASM, mean $2.27mm$, STD $0.69mm$
- AAM, mean $1.99mm$, STD $0.70mm$

The average standard deviation of automatic annotation to expert annotation of the mandibular canal, for the LK tracking, B-spline, demon, ASM and AAM segmentation are:

- LK tracking, mean $2.42mm$
- B-spline registration, mean $1.88mm$
- Demon registration, mean $3.21mm$
- ASM, mean $1.34mm$
- AAM, mean $0.99mm$

As can be seen the demon registration produces the worst result, with an average distance to the annotated mandibular canal of $5mm$, combined with a large standard deviation. The LK tracking and B-spline registration have an average distance to the annotated mandibular canal of respectively $3mm$ and $4mm$. The AAM and ASM methods have low average distances. With AAM the lowest average distance $1.99mm$.

10.8 Improving the ASM Results

The number of training scans is 12 for the ASM and AAM in a leave one out experiment. These scans only describe a small part of all possible variations in the mandible. PCA uses the correlation matrix, and with such a small data set surface deformations can look correlated while they are not correlated at all. Kainmueller et al. [33] used more than 100 data sets to build there ASM model.

Therefore we use three methods of improving our ASM results, these are introduced in the next subsections.

Table 10.17 The snake parameters.

parameter	value	information
γ	1	Time step
Itt	15	Iterations
σ_1	2	Sigma used for image derivatives
w_{line}	0.04	Attraction to white lines
w_{edge}	2	Attraction to edges
σ_2	2	Sigma for gradient of edge energy
μ	0.2	GVF noise constant
Itt_g	5	GVF iterations
α	0.2	Membrane energy
β	0.2	Thin plate energy
δ	0.1	Balloon force
κ	2	Weight of image force
λ	0.8	Surface normal alignment

10.8.1 Snake

An active contour or snake [54], is a contour of connected lines as in a ASM. The main difference is that in a snake the segmentation is regularized by internal forces from physics such as a bending force, and an ASM by learned shape variations. The search in a snake is based on edge attractions through image-gradients and pixel intensities. We use an $3D$ implementation of a snake [55], including gradient vector flow (GVF) [99] for to extend the attraction of edges to a further distance.

We use the surface of the bone produced by the ASM as initial shape for our snake, and then run the snake, to get an better alignment of the moving surface with the bone in the data set. Parameters used are available in table 10.17

After snake registration we merge our new bone-surface with the existing ASM surfaces of the mandibular canal. Then we convert the combined surface to ASM model parameters and back to x, y, z coordinates. Now we have obtained new x, y, z coordinates for our nerves, which correspond to the new snake aligned bone-surface. The resulting dice coefficients of the data sets can be found in table 10.18, and new nerve distances in table 10.19. The snake surface optimization increases our Dice coefficient from 0.75 to 0.80. The mean distance to expert segmentation increases from $2.27mm$ to $2.33mm$.

10.8.2 Anisotropic Diffusion Filtering

In chapter 3, we introduced edge enhancing anisotropic diffusion filtering. We filtered the data sets with this method using the parameters in table 10.20. Then we did the ASM segmentation again, but now on the data sets with the diffusion filtered results, followed by snake segmentation. After running the segmentation we didn't not notice any differences in the Dice coefficients. See table 10.20, for the distance between automatic localization and manual annotations. As you can see the distance

Table 10.18 Snake optimization of the surface from the ASM, bone segmentation performance. Dice coefficient between registered mandibular bone surface and manual segmented bone.

Id test	Dice Coefficient
1	0.81
2	0.92
3	0.87
4	0.76
5	0.77
6	0.83
7	0.59
8	0.78
9	0.82
10	0.81
11	0.84
12	0.77
13	0.79

Table 10.19 Distance between automatic and manual segmented mandibular canal after ASM and snake optimization.

	Right			Left		
Id	mean	std	max	mean	std	max
1	1.82	1.01	4.71	3.66	1.66	8.91
2	1.50	0.83	4.69	1.81	0.52	3.23
3	3.43	1.69	9.06	2.82	1.16	6.55
4	2.24	1.27	6.87	2.97	1.24	5.56
5	2.40	1.37	5.26	3.22	2.70	9.96
6	1.57	2.05	16.1	3.21	1.52	9.83
7	1.49	0.59	5.92	2.28	1.59	9.24
8	2.66	1.87	7.64	2.98	1.90	9.63
9	2.97	1.97	9.95	2.44	1.53	8.56
10	2.40	1.16	6.19	2.03	1.15	5.79
11	1.52	0.67	5.63	1.64	1.3	6.04
12	2.23	1.05	4.95	2.32	1.61	6.61
13	1.84	0.94	3.85	1.25	0.85	5.40

Table 10.20 Distance between automatic and manual segmented mandibular canal. Automatic results after anisotropic filtering and snake optimization.

	Right			Left		
Id	mean	std	max	mean	std	max
1	1.52	0.97	4.16	3.71	1.27	8.81
2	1.48	0.84	4.76	1.81	0.54	3.49
3	3.28	1.74	8.98	2.76	1.16	6.36
4	2.68	1.47	7.69	3.29	1.38	6.11
5	4.00	1.57	6.74	3.68	3.29	11.9
6	1.80	1.83	15.1	3.38	1.45	9.12
7	2.46	0.85	4.73	2.77	1.90	7.34
8	2.74	1.88	7.78	2.87	1.92	9.06
9	2.99	2.01	10.3	2.47	1.54	8.73
10	3.09	1.51	5.46	2.64	1.42	7.60
11	1.47	0.59	5.07	1.92	1.45	6.19
12	2.75	1.16	4.91	2.57	1.87	7.17
13	2.03	0.96	4.22	1.11	0.74	5.51

values sometimes improve a few tenths of a millimeter, but the mean distance value increases to $2.59mm$.

10.8.3 Fast Marching

Almost all recent mandibular canal segmentation papers use Dijkstra's algorithm [98], or Fast Marching [100] to improve the location of the nerve. This are distance algorithms which are useful to calculate the shortest path between two points. In those papers they assume that the mandibular canal has a lower intensity than the surroundings. But this is not true for our data, probably because of the low imaging dose. Most parts of the mandibular canal have a circular bony wall with a high Hounsfield value. We will use this information later on to make a filter to lower the intensities inside the canal.

But first we assume that the canal has a low intensity already. Thus we can make a speed map in which speed inversely related to intensity. In this way the shortest path between end-points of the mandibular canal, will be the canal it self. Tissue outside the bone also has a low intensity, thus to prevent the shortest path from running outside the bone, we use the AAM snake optimized bone-surface to select the background region and set all those pixels to a high pixel value.

As next step we use the vertices of the canal surface from previous steps, and calculate the curved pixel length l_{canal} along the canal. Then we construct uniform distributed normal planes along the channel. Now we can warp the local neighborhood of the canal to a small volume I_l of dimensions $4mm \times 4mm \times l_{canal}$. The curved canal in the CBCT data is a straight line in the new volume I_l . To make the channel black, we use the following filter, which is the inverse of a slice through the bony

Table 10.21 Distance between automatic and manual segmented mandibular canal. Automatic results after snake optimization and fast marching shortest path optimization.

	Right			Left		
Id	mean	std	max	mean	std	max
1	2.18	0.99	4.36	2.78	1.16	5.32
2	0.93	0.81	4.09	0.94	0.71	3.11
3	3.78	1.70	8.75	2.83	1.09	6.54
4	2.26	1.30	6.42	3.11	1.25	5.49
5	2.45	1.67	5.38	3.51	2.90	9.94
6	1.55	2.02	16.1	3.26	1.61	9.70
7	1.57	0.66	5.87	2.40	1.43	9.29
8	2.92	1.55	7.47	3.03	1.89	9.55
9	3.39	2.10	9.90	2.58	1.79	8.68
10	1.86	1.01	6.55	2.27	1.11	5.79
11	1.48	0.69	5.50	1.98	1.22	5.96
12	2.14	0.89	4.71	2.45	1.54	6.53
13	1.71	0.94	3.58	1.45	0.78	5.40

canal:

$$H_c = \frac{1}{30}\begin{bmatrix} -5 & -6 & -3 & -1 & -3 & -6 & -5 \\ -6 & 0 & 3 & 4 & 3 & 0 & -6 \\ -3 & 3 & 5 & 6 & 5 & 3 & -3 \\ -1 & 4 & 6 & 6 & 6 & 4 & -1 \\ -3 & 3 & 5 & 6 & 5 & 3 & -3 \\ -6 & 0 & 3 & 4 & 3 & 0 & -6 \\ -5 & -6 & -3 & -1 & -3 & -6 & -5 \end{bmatrix} + \frac{6}{1470} \tag{10.7}$$

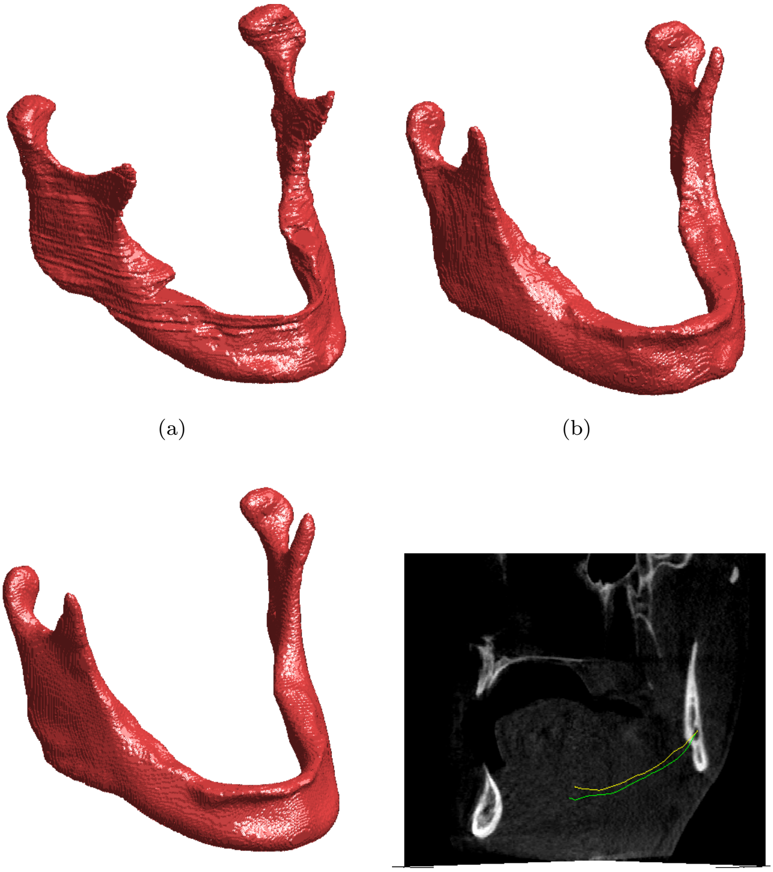
This filter is use as convolution filter on every slice of I_l corresponding to a normal plane of the curved canal, creating the volume I_f. Then we set our final volume $I_n$to the average of the unfiltered volume I_l and filtered volume I_f.

To convert the intensities of I_n to a speed map S we use:

$$S = e^{(I_n * H_g)} + e^{-||\nabla I_n||} \tag{10.8}$$

With H_g a Gaussian kernel with sigma 1.5. We define the source and sink as the end points of our mandibular canal estimation. And after running fast marching with speed map S we detect the shortest route using Runge-Kutta type RK4. Then we warp the shortest route back to the original volume.

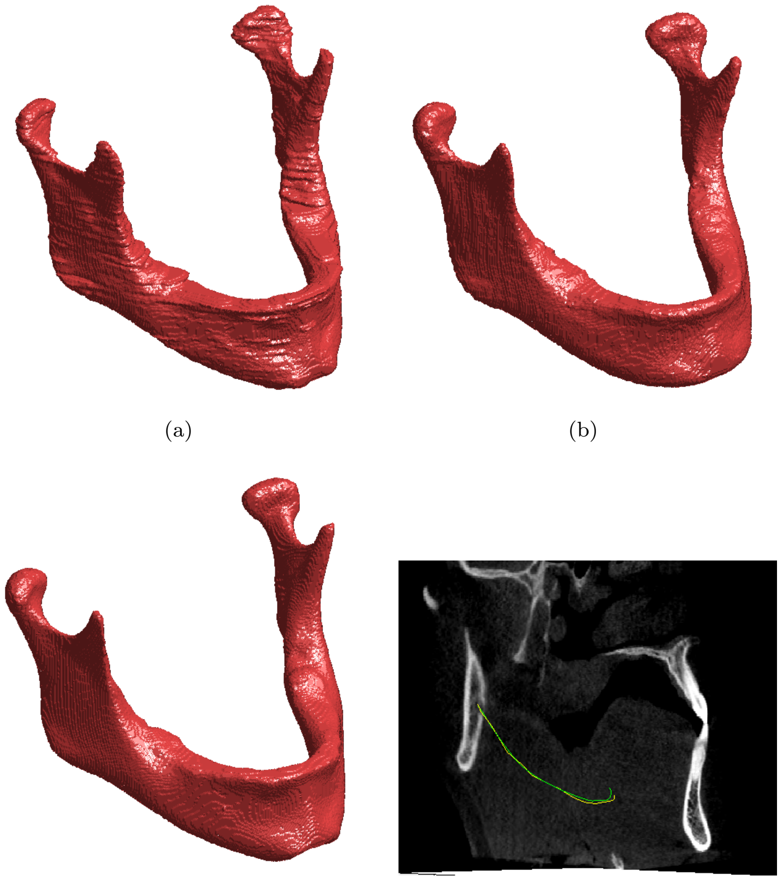
The resulting distances to expert segmentation can be found in table 10.21. Some canal localization results improve, but the average mean distance error is slightly increased to 2.29mm.



(a) (b)

(c) (d)

Figure 10.4 Bone segmentation and mandibular canal localization results, for data set 1. Manual segmentation (a). ASM segmentation (b), and surface of ASM segmentation optimized by snake (c). Comparison between expert localisation (yellow) and ASM localisation (green) of the mandibular canal.



(a)

(b)

(c)

(d)

Figure 10.5 Bone segmentation and mandibular canal localization results, for data set 2. Manual segmentation (a). ASM segmentation (b), and surface of ASM segmentation optimized by snake (c). Comparison between expert localisation (yellow) and ASM localisation (green) of the mandibular canal.

Table 10.22 The AAM training and segmentation parameters, used for the mandibular canal AAM.

parameter	value	information
n_b	0	Border added (pixels)
s	[100 100 100]	Texture size used (pixels)
n_{scales}	1	Number of AAM image resolutions.
n_{search}	20 Number of iterations	

10.9 Improving the AAM results

10.9.1 Mandibular Canal AAM

The AAM model used before is trained on a combination of the bone surface and mandibular canal points. This model gives a rough but robust estimate of the mandibular canal position. PCA is global linear model the movement of a single bone vertex will also impact the position of the points of the mandibular canal. These correlations are not always physically justified.

Therefore it is a logical step to build also an AAM of only a single mandibular canal, and use the result of the combined model as initialization. Also an advantage is that we have twice as much mandibular canals as mandibles, resulting in twice as much data sets by simply flipping the left or right mandibular canal.

We only have lines of connected points, to include the local neighborhood of a canal to the appearance model, we need tubes. We use the method described in the section "Experiment Active Shape Model", to generate such a triangulated tube. We use a diameter of 16 pixels (6.4 mm). We construct this tube every time the AAM requires appearance information, but train the shape model only on the line points.

Because we already have a rough initialization of the canal we don't use lower resolution images, but use a single scale model. We also do not optimize the result with the simplex optimizer because it can move the canal model to positions which also look like a canal but are far away form the real mandibular canal position. For all parameters see table 10.22.

We have tested the model on five of the thirteen data sets. The distances between automatic and expert annotation of the canal are shown in table 10.23.

10.9.2 Weighted Intensities

The AAM uses the difference between model intensities and image intensities as alignment error and to optimize the model parameters. The influence on the alignment error of a pixel at the location of the mandibular canal is the same as everywhere else in the bone. In our case we want to fit the model as accurate as possible to the mandibular canal, and do not care about bone segmentation accuracy. Therefore the idea is to increase the influence of the pixels corresponding to the mandibular canal, and decrease the influence of the other pixels.

Table 10.23 Distance between automatic and manual segmented mandibular canal. Automatic results. With 'before' the distance result of the combined AAM model, and 'after' the distance after optimizing with the AAM model of only the canal.

		Right			Left		
Id	b/a	mean	std	max	mean	std	max
1	before	1.68	0.98	4.26	3.87	2.05	8.47
1	after	1.51	0.92	3.36	4.26	2.31	8.22
2	before	1.38	0.70	3.91	1.57	0.59	4.14
2	after	3.90	1.38	8.29	0.89	0.59	2.22
3	before	3.20	0.91	5.94	2.76	1.06	6.10
3	after	4.78	0.91	7.60	3.27	1.11	6.38
4	before	1.79	0.64	4.22	1.48	0.65	4.08
4	after	1.95	0.86	5.38	1.65	0.90	3.30

The AAM is a linear model thus if there are large differences between model intensities and image intensities at a certain model location they will have a large influence on the model parameter update, Therefore we can simply increase the influence of the pixels corresponding to the mandibular canal, by multiplying these intensities by a large value.

In an AAM the image intensities are warped to a mean shape image I_{ms}. We sample the gray value vector $\mathbf{g}$ from the image coordinates $L = (L_x^T, L_y^T)$ in the mean shape image. This vector $\mathbf{g}$ is used for the appearance part of the AAM model. To increase the influence of the mandibular canal pixels we multiply the mean shape image I_{ms} with array A_{ms}. In this array values far from the canal have a small value and pixels close to the canal a high value. We construct A_{ms}, by discretizing the canal coordinates to an image with all background pixels set to zero and canal pixels set to one. We smooth the image with a Gaussian kernel with sigma 5, then multiply the array by $1/\max(A_{ms})$ to scale the values between $[0, 1]$ and add 0.1 to all pixels, see figure 10.6.

We use the weighted intensities only for the highest scale of the AAM search. In this scale the model is already close to the mandibular canal in the image. In the lower resolution image scales we use the same AAM as in section 10.6. The distances between expert annotation and AAM segmentation of the canal are show in table 10.24. The mean of the mean distance error is $1.88mm$ with the STD of the mean distance error $0.69mm$, the average standard deviation is $1.05mm$.

10.10 Conclusion

We tested five methods to obtain an approximation of the mandibular canals. LK tracking, B-spline registration, demon registration, ASM and AAM segmentation. The average Dice coefficient of the bone segmentation of the 13 data sets where for both registration methods and ASM and AAM segmentation around 0.76.

Both registration methods produced approximations of the mandibular canal

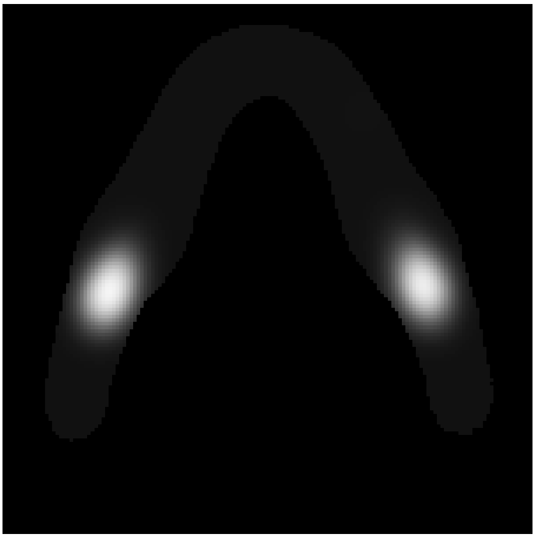


Figure 10.6 Slice of matrix A_{ms}, which is multiplied with the warped image intensities I_{ms}, to increase the influence of the pixels close to mandibular canal.

Table 10.24 Distance between AAM and manual segmented mandibular canal, using location based intensity weights.

	Right			Left		
Id	mean	std	max	mean	std	max
1	1.70	0.80	3.93	3.67	1.87	8.92
2	1.36	0.61	3.31	1.02	0.51	3.49
3	2.92	0.91	5.49	2.82	1.05	6.63
4	1.30	0.68	4.08	1.11	0.69	4.41
5	1.51	0.89	4.17	1.38	0.83	5.47
6	1.99	1.36	9.39	3	0.96	6.95
7	1.45	0.76	3.02	2.34	1.27	9.19
8	1.72	1.08	4.33	1.33	1.24	6.87
9	2.72	1.52	10.08	2.36	1.41	9.14
10	1.94	1.31	8.40	1.56	1.35	6.25
11	1.86	1.07	6.09	1.86	1.05	5.05
12	1.70	1.08	4.39	1.98	1.89	7.35
13	1.03	0.59	2.46	1.15	0.59	3.25

which where far away from the medical expert annotation around $4mm$ and $5mm$. With LK tracking a mean distance to expert annotation of around $3mm$. The AAM and ASM method have the lowest average distance respectively $1.99mm$ and $2.27mm$ to the medical expert annotation. But the resulting distance between ground truth from the medical expert and AAM/ASM is higher than in recent literature average distance to ground truth $1.1mm$ [33], and around $0.7mm$ [35]. One of the main causes is the low number of training data sets with only cover a small amount of variation between mandibles.

We increased Dice coefficient of the ASM bone segmentation to 0.80, by optimizing the surface description with a 3D snake. We test the same produce, the ASM segmentation followed by snake optimization but now after filtering the data with anisotropic diffusion. The filtering step did not increase the Dice coefficient of the bone segmentation and has a negligible effect on the distance between expert label and automatic detection. Finally we optimized ASM the mandibular canal approximation by using fast-marching. In literature the canal always has a lower intensity than the surroundings, but in our case the intensities are not lower. Therefore we use a circular convolution filter, to lower the intensities in the mandibular canal. This is done after warping the approximated mandibular canal to a straight line.

Our average mean distance error of the ASM is $2.29mm$ after shortest path tracing. In comparison to literature we have a high distance error. This is probably due to lower-dose scans, resulting in less contrast between mandibular canal and surrounding, thus not a clear dark tunnel for our fast marching optimization.

In a study of Liang et al. [101] they compare the image quality of several CBCT machines of several vendors. The image quality varies widely between the machines. In some scans the mandibular canal is hardly visible due to noise or smoothing. In scans of other brands it is a conspicuous black noise free canal. In this paper there seems no direct relation between dose and image quality, the brand of the scanner is more important.

Another reason for our large error is the small amount of training data sets (twelve) in comparison to more than hundred in [33]. Post-optimization with the AAM of only the mandibular canals (twenty four) was therefore also expected to increase the accuracy. But the results afterwards were not better. Probably because the initial canal model is still not close enough to the real mandibular canal.

The last method increasing the influence of pixels close to the mandibular canal in the highest AAM image scale improves our results. We obtain an average mean distance error of $1.88mm$ and average STD of $1.05mm$. In this method the shape and appearance from the mandible still have some influence on the mandibular canal shape. Making it more robust than the mandibular canal AAM which does not have this regularization.

11 Conclusion

The aim of our research is to develop towards an automatic system for the extraction of the mandibular canal from CBCT data. In the introduction chapter we have formulated research questions, involved in reaching this goal. We will discuss the answers to our research questions in the subsections below. Followed by a general conclusion and discussion.

11.1 Research Questions

Research Question 1

Is it possible to get high enough accuracy and robustness to replace human annotation of the mandibular canal in surgery planning?

To replace human annotation of the mandibular canal we need to be as accurate and robust as a human observer. In Gerlach et al. [102], the reproducibility of 3 mandibular canal tracing methods in CBCT data are tested. The best performing method, combining panoramic reconstructions with cross-sections had a mean inter observer variability of 1.3mm with a STD of 0.384mm. With largest deviations in the anterior loop region around 2.0mm due to "the incomplete bony wall in combination with the unpredictable recurrent course" [102]. Our implementation of a mandible AAM segmentation model is fully automatic. The AAM with weighted intensities extension has a mean distance to expert segmentation of the mandibular canal of 1.88mm and average STD of 1.05mm. Thus our mean and STD results have a 0.6mm larger error than differences between expert segmentations.

This results in a safety margin around the AAM annotations of approximately 5mm (see figure 10.3). We measured the distance between the mandibular canal and the the top of the bone in the body of the 13 mandibles. The mean distance is 13.3mm, the STD 6.1mm and minimum distance 3mm. The smallest implant of Bicon Dental Implants [103] has a length of 6mm. This dental implant is small enough to allow a safety margin of 5mm in 8 of our 13 patients.

Our ASM and AAM method heavily depends on the number of data sets for learning shape variations. We have used 12 training data sets but in literature more than 100 data sets are used [33] obtaining a mean distance error of around 1.1mm. Thus to improve towards human accuracy, we have to use more training data.

Research Question 2

Is it possible to accurately extract the mandibular canal from CBCT only based on intensities, or is shape information from a training set needed?

In chapter 4 we introduce template matching, to track the mandibular canal. This method was initialized manually by selecting a point inside the mandibular canal. The method has a mean distance from manual segmentation in the range of the inter-observer distance at the center of the canal but the mean distance of the whole mandibular canal is around 3.0mm. This is due to template drift, which causes large segmentation errors of often 7mm or more at the ends of the mandibular canal. Also the method does not stop tracking after exiting the canal. Our AAM is fully automatic has a higher mean accuracy, and stops at the end of the mandibular canal. Thus our method with a priori information performs better at the ends of the mandibular canal.

In our literature overview, chapter 2 we have included some older papers which do not use a priori information to segment the mandibular canal and have a larger distance error than the ASM based method in [33]. But the recent publication (2011) of Kim et al. [35], does not use training or atlas information, and outperforms all other methods.

Summarizing, a priori information makes the mandibular canal segmentation more robust, but does not necessarily give the most accurate segmentation results. Note, the contrast between mandibular canal and surroundings in CBCT data depend on radiation dose, X-ray spectrum and volume reconstruction method. This can be the reason that that some literature without a priori shape-information outperforms methods with shape information.

Research Question 3

Which of the three approaches is more suitable for extraction of the mandibular canal, an active shape model, active appearance model or an atlas based registration method?

In our results (chapter 10), we compare the performance of the three methods. If we look at bone segmentation there is no large difference between accuracy of the segmentation of an ASM (mean Dice coefficient 0.75) and AAM, B-spline or demon registration (0.76). But if we look at the AAM and ASM results on mandibular canal localization (without post-processing) and the results of the B-spline and demon registration, we notice a large difference. For AAM and ASM segmentation the distance to the manual annotated canal is respectively 1.99mm and 2.27mm versus 4mm and 5mm for the B-spline and demon registration.

This difference is probably caused by the large variation in shapes and texture between mandibles of patients, the high noise level, low contrast and small dimensions of the mandibular canal. The ASM and AAM will try to produce results which are possible regarding the variances in the training data. This makes them more robust to noise and illumination differences compared with only using one scan and a smoothing constraint as in the registration methods. The AAM method outperforms the ASM model if we look at mandibular canal localization. This is expected because the AAM model includes appearance information explicitly in the PCA model making the model more robust. But also not expected because there is a high correlation between shape and intensities in the model. Therefore more training data is needed for the AAM to incorporate enough freedom in the shape model. Another problem are the teeth and teeth-fillings, which cannot be described by a simple linear appearance model as used in an AAM.

Research Question 4

What is a suitable method to get corresponding points between patient CBCT data sets of the mandible?

When we look at the pertinent literature there are three major methods, used in shape model building. The first method is to use point features such as local curvature to find correspondences between surfaces. The second method describes the shape by spherical harmonics, and finds surface correspondences through spherical harmonics. The third method is minimum description length (MDL), in this method a PCA model is constructed from a rough set of corresponding points. These corresponding points are moved over the surface to minimize the variance in the PCA model. Because, when points between datasets have a good correspondence, a compact PCA model is obtained. The main disadvantage of MDL and spherical harmonics is that objects have to be mapped to a sphere. The mandible is not resembling a sphere and area-preserving mapping to a sphere is for this complex shape not possible. Thus some small areas of the sphere will contain a large number of object vertices and some large areas of the sphere a low number of object vertices, see chapter 9. Optimizing with a MDL is therefore difficult, and will not result in a good uniform corresponding point description of the original objects. MDL is about minimizing the description length of the ASM, but the global minimum is when all object points collapse into a single point. In practice points will locally cluster, degrading the quality of the description of the original object.

Therefore we introduce a new point correspondence optimization method. This method does not need the spherical mapping and uses surface features. Our method is based on shape context surface registration, see chapter 8. The shape context descriptor gives a description of a point based on not only local points (as in curvature methods), but includes information of all points of the surface. We use this shape context in combination with a diffeomorphic B-spline grid, to register iteratively and smooth a set of points from one data set to another data set. The B-spline grid gives us a multi-resolution framework, and allows global to very local matching of the point descriptions of the surfaces. Note, if we look at the results of ASM segmentation with an ASM constructed with shape context, we see that test data set 2 has very good segmentation results. This is probably because we use the surface vertices of this data set as base points. We have registered this set to all other data sets to obtain corresponding points. Thus the disadvantage is that our method is not optimal for all data sets as in MDL, but favors the basis data set.

Research Question 5

Does adding edge enhancing filtering of the CBCT data as pre-processing step, increase the accuracy in mandibular canal localization?

We have tested edge enhancing anisotropic diffusion filtering of the CBCT data before applying the ASM, see chapter 10. The mandibular canal is better visible after filtering and noise in the canal is decreased. But the Dice coefficients of the ASM do not increase after filtering, and the distances between ASM and expert segmentations of the mandibular canals do not decrease.

The anisotropic diffusion filtering connects missing edge segments thus can be used to improve the result of fast marching. But in our data the mandibular canal has to be pre-filtered by for example a circular filter, to make the intensities lower than the surroundings. The circular pre-filter removes the need for anisotropic filtering.

Thus to conclude, edge-enhancing filtering of the data does not improve the accuracy of mandibular canal localization in our data.

11.2 Final Conclusion

We have evaluated LK tracking, demon and B-spline registration, ASM and AAM segmentation. The mandibular canal localization performance of the methods are tested, and accuracy of mandible segmentation for all methods expect LK tracking.

The methods have approximately the same accuracy if we look at mandible segmentation. But if we look at nerve localization the AAM and ASM are far more accurate. The AAM with weighted intensities extension has a mean Euclidian distance to a human observer of around $1.88mm$. Our automatic mandibular canal localization, is not as accurate as human annotation which has an inter observer distance of around $1.3mm$. To improve the accuracy of our ASM and AAM results we need more training sets to cover all variability between mandibles. The higher accuracy of methods in literature is probably due to better image quality. Higher image quality can be obtained with a higher dose or better reconstruction algorithms, resulting in more contrast between canal and surroundings. In a study of Liang et al. [101] the image quality of several CBCT machines of several brands is compared. The image quality varies widely between the machines. In some scans the mandibular canal is hardly visible due to noise or smoothing. In scans of other vendors it is a high contrast noise free canal.

11.3 Contributions

Our 3D ASM mandibular canal segmentation method is based on the method of Kainmueller et al. [33]. The main improvement to the ASM model is including the appearance of the mandibular canal into the model. In the original method the shape of the canal is completely determined by the shape of the mandible surface. Our method allows the mandibular canal to move more independent from the mandible surface, resulting in a better localization of the canal.

Rueda et al. [30] introduced a 2D AAM based mandibular canal segmentation method. In this thesis we introduce a 3D AAM with some extensions to improve the accuracy. The main advantage of a 3D model is that it explicitly includes prior knowledge about the curved shape of the canal, allowing segmentation of scans with severe image noise and occlusions.

Chapter 3, Diffusion Filtering

We introduce a diffusion scheme which does rotation invariant edge preserving filtering as in the method of Weickert [42], but without the checkerboard artifacts. Gaussian based image derivatives are often assumed to give the most accurate image derivatives. The results in this chapter show that numerical optimized image derivatives show better rotation invariance than Gaussian based derivatives. This result is the main contribution of this chapter. We also wrote a short paper about optimizing the image derivatives for Sobel like filtering [1].

Chapter 4, Nerve Tracking

In this chapter we track the mandibular canal through an image volume, while locally sampling slices orthogonal to the object creating a movie stream. The movie frames are used for Lucas Kanade tracking, which returns the position of the mandibular canal needed to sample the next movie frame. To our knowledge this method has not been described in literature, and the tracking is useable for other elongated objects such as blood vessels.

Chapter 5, Active Shape Model

This chapter describes some new extentions to the ASM and AAM model.

We introduce an illumination robust method to find optimal contour positions, during ASM search based on PCA. The results show more accurate results than the old Mahanoblish distance based method.

The original AAM uses as triangulation based warp, which is not a smooth warp. We introduce a B-spline based warp method based on the method of Lee et al.[61].

Also a method is introduced which randomly permutes the initial position parameters of the ASM or AAM a few times. The position parameters with the lowest

[1] Kroon D.J, "Numerical Optimization of Kernel Based Image Derivatives", University of Twente, Enschede. *http://www.k-zone.nl/Kroon_DerivativePaper.pdf*

model error after a few ASM or AAM itterations are used as initial parameters. This is is straightforward, but was not expected to increase the AAM segmentation results with ten percent for already accurate initial positions. This is probably because the AAM is not robust against local search minima.

Chapter 6, Demon Registration

In this chapter we rewrite the demon registration equation to make it useable with line search and a gradient based optimizer (partly already done by Thirion[66]), instead of the implicitly steepest decent of the original method.

The main contribution of this chapter is the mutual histogram based method to convert the appearance of a scan from one modality into another modality. This allows the usage of fast intensity based registration methods such as demon registration, to be used with scans from multiple modalities

Chapter 8, Shape context registration

Finding corresponding points with shape context is straightforward, but using it iteratively with a point based B-spline warp is novel. Our B-spline warp allows multi scale corresponding point optimization for accurate correspondences. Our B-spline warp followed by the iterative closes point (ICP) method keeps corresponding points on the object surfaces. Commonly points are kept on the object surface by a spherical or plane based mapping, during corresponding point optimization. This mapping is not needed with our method, allowing non-convex and complex objects with holes to be fitted with corresponding points.

Our ICP method contains also some improvement such as symmetric forces, and point to surface correspondences using distance fields.

Chapter 9, Minimum description length

In this chapter we re-write the MDL optimization to be used with a quasi-Newton optimizer instead of steepest decent minimization.

The original MDL method uses random matrices to deal with the poles and Meridian in the spherical mapping. We use a separate coordinate system for every point, allowing multi-scale regularization without pole and meridian problems.

Chapter 10, Results

In literature a composite AAM first segments the image with an AAM of the main object, and then uses the result as initialization of an AAM of the sub object. We introduce a more robust approach, by changing the weight of pixels corresponding to the sub-structures instead of using a main and a sub model.

Disseminations

All the image processing methods used for this thesis including, registration, visualization and corresponding points methods are written from scratch and made available as open source code. The source codes including comments and examples are available on *http://www.Mathworks.com*.

For many of the methods there existed no open source implementations. For other methods, such as 3D MDL there existed no Matlab or other easy to read open source code. Especially our B-spline based registration tool is useful for image processing. It contains many of the available extensions, including diffeomorphic and volume preserving penalties, image based registration, point based registration, affine registration and many image similarity measures. Many other codes are also made available, which are not part of this thesis but used in other publications and research.

References

[1] F. Quereshy, T. Savell, and J. Palomo, “Applications of Cone Beam Computed Tomography in the Practice of Oral and Maxillofacial Surgery,” *Journal of oral and maxillofacial surgery* , vol. 66, pp. 791–6, 2008.

[2] M. Wakoh and K. Kuroyanagi, “Digital Imaging Modalities for Dental Practice,” *The Bulletin of Tokyo Dental College*, vol. 42, pp. 1–14, 2001.

[3] P. Sukovic, “Cone Beam Computed Tomography in Craniofacial Imaging,” *Orthod Craniofac Res*, vol. 6 Suppl 1, pp. 31–6, 2003.

[4] K. Tsiklakis, C. Donta, S. Gavala, K. Karayianni, V. Kamenopoulou, and C. Hourdakis, “Dose Reduction in Maxillofacial Imaging Using Low Dose Cone Beam CT,” *European Journal of Radiology*, vol. 56, pp. 413 – 417, 2005.

[5] M. Sandulescu, M. Traistaru, M. Nitescu, and I. Sirbu, “A Morphological Study of the Mandibular Canal in Partially Edentulous Patients,” *Therapeutics, Pharmacology and Clinical Toxicology*, vol. 14, pp. 42–52, 2010.

[6] P. Robinson, “Observations on the Recovery of Sensation Following Inferior Alveolar Nerve Injuries,” *British Journal of Oral and Maxillofacial Surgery*, vol. 26, pp. 177 – 189, 1988.

[7] R. Robert, P. Bacchetti, and M. Pogrel, “Frequency of Trigeminal Nerve Injuries Following Third Molar Removal,” *Journal of oral and maxillofacial surgery*, vol. 63, pp. 732–5, 205.

[8] R. Orth, M. Wallace, and M. Kuo, “C-arm Cone-beam CT: General Principles and Technical Considerations for Use in Interventional Radiology,” *Journal of Vascular and Interventional Radiology*, vol. 19, no. 6, pp. 814 – 820, 2008.

[9] J. Avery, P. Steele, and N. Avery, *Oral Development and Histology.* Thieme Medical Publishers Inc, 2001.

[10] A. O’Mahony, J. Williams, J. Katz, and P. Spencer, “Anisotropic Elastic Properties of Cancellous Bone from a Human Edentulous Mandible,” *Clinical Oral Implants Research*, vol. 11, no. 5, pp. 415–421, 2000.

[11] M. Schuenke, E. Schulte, U. Schumacer, and L. Ross, *Head and Neuroanatomy.* THIEME Atlas of Anatomy Series, Thieme Medical Publishers Inc, 3 2007.

[12] T. Fletcher and A. Weber, *Veterinary Developmental Anatomy, Veterinary Embryology Class notes.* No. CVM 6100, College of Veterinary Medicine, University of Minnesota, 2009.

[13] C. Clemente, *Gray's Anatomy of the Human Body (30th Edition).* Lea and Febiger, 1985.

[14] D. Atwood, "Postextraction Changes in the Adult Mandible as Illustrated by Microradiographs of Midsagittal Sections and Serial Cephalometric Roentgenograms," *The Journal of Prosthetic Dentistry*, vol. 13, no. 5, pp. 810 – 824, 1963.

[15] M. Naitoh, K. Nakahara, Y. Suenaga, K. Gotoh, S. Kondo, and E. Ariji, "Comparison Between Cone-Beam and Multislice Computed Tomography Depicting Mandibular Neurovascular Canal Structures," *Orthodontics and Craniofacial Research*, vol. 109, pp. 25–31, Jan 2010.

[16] G. Juodzbalys, H. Wang, and G. Sabalys, "Anatomy of Mandibular Vital Structures. Part I: Mandibular Canal and Inferior Alveolar Neurovascular Bundle in relation with Dental Implantology.," *Journal of Oral and Maxillofacical Research*, vol. 1, no. 1, 2010.

[17] R. Langlais, R. Broadus, and B. Glass, "Bifid Mandibular Canals in Panoramic Radiographs," *The Journal of the American Dental Association*, vol. 110, pp. 923–926, June 1985.

[18] M. Naitoh, Y. Hiraiwa, H. Aimiya, and E. Ariji, "Observation of Bifid Mandibular Canal Using Cone-Beam Computerized Tomographys," *Journal of Oral and Maxillofacial Implants*, vol. 24, pp. 115–9, 2009.

[19] E. Siebert, G. Bohner, M. Dewey, F. Masuhr, K. Hoffmann, J. Mews, F. Engelken, H. Bauknecht, S. Diekmann, and R. Klingebiel, "320-slice CT Neuroimaging: Initial Clinical Experience and Image Quality Evaluation ," *The British journal of radiology*, vol. 82, pp. 561–70, July 2009.

[20] M. Kachelrieb and W. Kalender, "Advanced Single-slice Rebinning in Cone-beam Spiral CT ," *Medical Physics*, vol. 27, pp. 754–772, 2000.

[21] M. Loubele, R. Bogaerts, E. Van Dijck, R. Pauwels, S. Vanheusden, P. Suetens, G. Marchal, G. Sanderink, and R. Jacobs, "Comparison Between Effective Radiation Dose of CBCT and MSCT Scanners for Dentomaxillofacial Applications," *European Journal of Radiology*, vol. 71, no. 3, pp. 461 – 468, 2009. Osteoporosis.

[22] L. Feldkamp, L. Davis, and J. Kress, "Practical Cone-Beam Algorithm," *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, vol. 1, pp. 612–619, 1984.

[23] W. Chlewicki, C. Badea, and P. N., "Cone Based 3D Reconstruction: A FDK-SART Comparison for Limited Number of Projections," in *MEDICON*, 2001.

[24] S. Baker, R. Gross, and I. Matthews, “Lucas-Kanade 20 Years On: A Unifying Framework,” *International Journal of Computer Vision*, vol. 56, pp. 221–255, 2004.

[25] T. Cootes, C. Taylor, D. Cooper, and J. Graham, “Active Shape Models - Their Training and Application,” *Computer Vision, Graphics and Image Processing*, vol. 61(1), pp. 38–59, January 1995.

[26] T. Cootes, G. Edwards, and C. Taylor, “Active Appearance Models,” in *European Conference on Computer Vision*, vol. 2, pp. 484–498, H. Burkhardt & B. Neumann, 1998.

[27] W. Stein, S. Hassfeld, and J. Muhling, “Tracing of Thin Tubular Structures in Computer Tomographic Data,” *Computer Aided Surgery*, vol. 3, pp. 83–88, 1998.

[28] N. Hanssen, Z. Burgielski, T. Jansen, M. Lievin, L. Ritter, B. von Rymon-Lipinski, and E. Keeve, “Nerves - Level Sets for Interactive 3D Segmentation of Nerve Channels,” in *Biomedical Imaging: Nano to Macro, 2004. IEEE International Symposium on*, vol. 1, pp. 201–204, April 2004.

[29] T. Kondo, S. Ong, and K. Foong, “Computer-based Extraction of the Inferior Alveolar Nerve Canal in 3D Space.,” *Computer Methods and Programs in Biomedicine*, vol. 76, no. 3, pp. 181–191, 2004.

[30] S. Rueda, J. Gil, R. Pichery, and A. Raya, “Automatic Segmentation of Jaw Tissues in CT Using Active Appearance Models and Semi-automatic Landmarking,” in *Medical Image Computing and Computer-Assisted Intervention*, pp. 167–174, 2006.

[31] S. Sotthivirat and W. Narkbuakaew, “Automatic Detection of Inferior Alveolar Nerve Canals on CT Images,” in *Biomedical Circuits and Systems Conference, 2006. BioCAS 2006. IEEE*, pp. 142 –145, 29 2006-dec. 1 2006.

[32] H. Yau, Y. Lin, L. Tsou, and C. Lee, “An Adaptive Region Growing Method to Segment Inferior Alveolar Nerve Canal from 3D Medical Images,” *Computer-Aided Design and Applications*, vol. 5, no. 5, pp. 743–752, 2008.

[33] D. Kainmueller, H. Lamecker, H. Seim, M. Zinser, and S. Zachow, “Automatic Extraction of Mandibular Nerve and Bone from Cone-Beam CT Data,” in *MICCAI 2009*, vol. 5762 of *Lect. Notes Comput. Sci.*, pp. 76–83, Springer Berlin, Heidelberg, 2009.

[34] H. Lamecker, S. Zachow, A. Wittmers, B. Weber, H. Hege, B. Elsholtz, and M. Stiller, “Automatic Segmentation of Mandibles in Low-Dose CT-Data,” *International Journal Computer Assisted Radiology and Surgery* , vol. 1, p. 393395, 2006.

[35] G. Kim, J. Lee, H. Lee, J. Seo, Y. Koo, Y. Shin, and B. Kim, "Automatic Extraction of Inferior Alveolar Nerve Canal Using Feature-Enhancing Panoramic Volume Rendering," *Biomedical Engineering, IEEE Transactions on*, vol. 58, pp. 253 –264, feb. 2011.

[36] J. Roberts, N. Drage, J. Davies, and D. Thomas, "Effective Dose from Cone-Beam CT Examinations in Dentistry," *The British Journal of Radiology*, vol. 82, pp. 35–40, 2009.

[37] J. Sunnegardh, *Iterative Filtered Backprojection Methods for Helical Cone-Beam CT* . PhD thesis, Computer Vision , The Institute of Technology, 2009.

[38] S. Awate and R. Whitaker, "Unsupervised, Information-Theoretic, Adaptive Image Filtering for Image Restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 364–376, 2006.

[39] P. Perona and J. Malik, "Scale-space and Edge Detection using Anisotropic Diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 629–639, 1990.

[40] J. Weickert, *Anisotropic Diffusion in Image Processing.* PhD thesis, University of Copenhagen, Department of Computer Science, 1998.

[41] A. Mendrik, E. Vonken, A. Rutten, M. Viergever, and van Ginneken. B., "Noise Reduction in Computed Tomography Scans using 3D Anisotropic Hybrid Diffusion with Continuous Switch," *IEEE Transactions on Medical Imaging*, vol. 28, pp. 1585–1594, October 2009.

[42] J. Weickert and H. Scharr, "A Scheme for Coherence-Enhancing Diffusion Filtering with Optimized Rotation Invariance," *Journal of Visual Communication and Image Representation*, vol. 13, no. 1, pp. 103–18, 2002.

[43] A. Frangakis and R. Hegerl, "Noise Reduction in Electron Tomographic Reconstructions using Nonlinear Anisotropic Diffusion," *Journal of Structural Biology*, vol. 135, no. 135, pp. 239–250, 2001.

[44] M. Felsberg, "On the Relation between Anisotropic Diffusion and Iterated Adaptive Filtering," in *Proceedings of the 30th DAGM symposium on Pattern Recognition*, (Berlin, Heidelberg), pp. 436–445, Springer-Verlag, 2008.

[45] J. Lagarias, J. Reeds, M. Wright, and P. Wright, "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions," *SIAM Journal on Optimization*, vol. 9, no. 1, pp. 112–147, 1998.

[46] D. Shanno, "Conditioning of Quasi-Newton Methods for Function Minimization," *Mathematics of Computation*, vol. 24, no. 111, pp. 647–656, 1970.

[47] M. Ahmed, S. Yamany, A. Farag, and T. Moriarty, “Bias Field Estimation and Adaptive Segmentation of MRI Data Using a Modified Fuzzy C-Means Algorithm,” in *In Proceedings IEEE International Conference Computer Vision and Pattern Recognition*, pp. 250–255, 1999.

[48] L. Dario, “Implant Placement above a Bifurcated Mandibular Canal: a Case Report.,” *Implant Dentistry*, vol. 11, no. 3, pp. 258–256, 2002.

[49] G. Hager and P. Belhumeur, “Efficient Region Tracking With Parametric Models of Geometry and Illumination,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 10, pp. 1025–1039, 1998.

[50] D. Schreiber, “Robust Template Tracking with Drift Correction,” *Pattern Recognition Letters*, vol. 28, no. 12, pp. 1483–1491, 2007.

[51] I. Matthews, T. Ishikawa, and S. Baker, “The Template Update Problem,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 810–815, June 2004.

[52] H. Akhoondali, R. Zoroofi, and G. Shirani, “Fully Automatic Extraction of Panoramic Dental Images from CT-Scan Volumetric Data of the Head,” *Journal of Applied Sciences*, vol. 9, no. 11, pp. 2106–2114, 2009.

[53] C. Iacobellis, A. Bulzacchi, and A. Rioda, “CT Evaluation of Regenerated Osseous Segments Following Bone Transport,” *Journal of Orthopaedics and Traumatology*, vol. 5, pp. 178–184, 2004. 10.1007/s10195-004-0067-0.

[54] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active Contour Models,” *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[55] C. Lürig, L. Kobbelt, and T. Ertl, “Hierachical Solutions for the Deformable Surface Problem in Visualization,” *Graphical Models*, vol. 62, no. 1, pp. 2–18, 2000.

[56] D. L. Pham, C. Xu, and J. L. Prince, “A Survey of Current Methods in Medical Image Segmentation,” in *Annual Review of Biomedical Engineering*, vol. 2, pp. 315–338, Annual Review of Biomedical Engineering, 1998.

[57] V. Wyawahare, P. M. Patil, and H. K. Abhyankar, “Image Registration Techniques: An Overview,” *International Journal of Signal Processing,Image Processing*, vol. 2, pp. 11–28, Sept. 2009.

[58] K. Pearson, “On lines and Planes of Closest Fit to Systems of Points in Space,” *Philosophical Magazine*, vol. 2, no. 6, pp. 559–572, 1901.

[59] B. Horn, “Closed-form Solution of Absolute Orientation Using Unit Quaternions,” *Journal of the Optical Society of America A*, vol. 4, no. 4, pp. 629–642, 1987.

[60] D. Rueckert, L. Sonoda, C. Hayes, D. Hill, M. Leach, and D. Hawkes, “Non-Rigid Registration using Free-Form Seformations: Application to Breast MR Images,” *IEEE Transactions Medical Imaging*, vol. 18, no. 8, pp. 712–721, 1999.

[61] S. Lee, G. Wolberg, and S. Y. Shin, “Scattered Data Interpolation with Multilevel B-Splines,” *IEEE Transactions Visualization and Computer Graphics*, vol. 3, pp. 228–244, 1997.

[62] M. Stegmann, R. Fisker, and B. Ersbøll, “Extending and Applying Active Appearance Models for Automated, High Precision Segmentation in Different Image Modalities,” in *Proceedings 12th Scandinavian Conference on Image Analysis - SCIA 2001, Bergen, Norway* (I. Austvoll, ed.), (Stavanger, Norway), pp. 90–97, NOBIM, jun 2001.

[63] T. Coleman and Y. Li, “An Interior Trust Region Approach for Nonlinear Minimization Subject to Bounds,” tech. rep., Cornell University, Ithaca, NY, USA, 1993.

[64] T. Coleman and Y. Li, “On the convergence of reflective newton methods for large-scale nonlinear minimization subject to bounds,” 1992.

[65] D. J. Kroon, E. S. B. van Oort, and C. H. Slump, “Multiple Sclerosis Detection in Multispectral Magnetic Resonance Images with Principal Components Analysis,” in *3D Segmentation in the Clinic: A Grand Challenge II: MS lesion segmentation, New-York, USA*, (Website), pp. 604–617, Kitware, September 2008.

[66] J. Thirion, “Image Matching as a Diffusion Process: an Analogy with Maxwell’s Demons,” *Medical Image Analysis*, vol. 2, pp. 243–260, September 1998.

[67] F. Castro, C. Pollo, O. Cuisenaire, J. Villemure, and J. Thiran, “Validation of Experts Versus Atlas-Based and Automatic Registration Methods for Subthalamic Nucleus Targeting on MRI,” *International Journal of Computer Assisted Radiology and Surgery*, vol. 1, no. 1, pp. 5–12, 2006.

[68] M. Bro-Nielsen and C. Gramkow, “Fast Fluid Registration of Medical Images,” in *Proceedings Visualization in Biomedical Computing*, pp. 267–276, Springer-Verlag, 1996.

[69] H. Wang, L. Dong, J. O’Daniel, R. Mohan, A. Garden, K. Ang, D. Kuban, J. Bonnen, M. Chang, and R. Cheung, “Validation of an Accelerated ’Demons’ Algorithm for Deformable Image Registration in Radiation Therapy,” *Physics in Medicine and Biology*, vol. 50, no. 12, p. 28872905, 2005.

[70] P. Cachier, X. Pennec, and N. Ayache, “Fast Non-Rigid Matching by Gradient Descent: Study and Improvement of the Demons Algorithm,” 1999.

[71] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache, "Non-parametric Diffeomorphic Image Registration with the Demons Algorithm," *Medical Image Computing and Computer-Assisted Intervention MICCAI 2007*, vol. 2, pp. 319–326, 2007.

[72] C. Cocosco, V. Kollokian, R.-S. Kwan, G. Pike, and A. Evans, "Brainweb: Online Interface to a 3D MRI Simulated Brain Database," *NeuroImage*, vol. 5, p. 425, 1997.

[73] R. Kwan, A. Evans, and G. Pike, "MRI Simulation-Based Evaluation of Image-Processing and Classification Methods," *IEEE Transactions on Medical Imaging*, vol. 18, no. 11, pp. 1085–1097, 1999.

[74] F. Chen and K. Arunachalam, "Geometric Transformation Pinched Hallway and its Restoration," 2001.

[75] E. Haber and J. Modersitzki, *Bildverarbeitung für die Medizin 2005*, vol. 5, pp. 350–354. Springer-Verlag, 2005.

[76] B. Zitova and J. Flusser, "Image Registration Methods: a Survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977 – 1000, 2003.

[77] W. Crum, T. Hartkens, and D. Hill, "Non-Rigid Image Registration: Theory and Practice," *Br J Radiol*, vol. 77, pp. S140–153, Dec. 2004.

[78] J. Schnabel, C. Tanner, A. Castellano-Smith, A. Degenhard, M. Leach, D. Hose, D. Hill, and D. Hawkes, "Validation of Nonrigid Image Registration Using Finite-Element Methods: Application to Breast MR Images," *Medical Imaging, IEEE Transactions on*, vol. 22, pp. 238 –247, feb. 2003.

[79] J. Pluim, J. Maintz, and M. Viergever, "Mutual-Information-Based Registration of Medical Images: a Survey," *Medical Imaging, IEEE Transactions on*, vol. 22, pp. 986 –1004, aug. 2003.

[80] G. Penney, J. Weese, J. Little, P. Desmedt, D. Hill, and D. hawkes, "A Comparison of Similarity Measures for use in 2-D-3-D Medical Image Registration," *Medical Imaging, IEEE Transactions on*, vol. 17, pp. 586 –595, aug. 1998.

[81] J. Nocedal, "Updating Quasi-Newton Matrices with Limited Storage," *Mathematics of Computation*, vol. 35, no. 151, pp. 773–782, 1980.

[82] D. Rueckert, P. Aljabar, R. A. Heckemann, J. V. Hajnal, and A. Hammers, "Diffeomorphic Registration Using B-Splines," in *Proc Intern Conf on Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (R. Larsen, M. Nielsen, and J. Sporring, eds.), vol. 4191 of *LNCS*, pp. 702–709, Springer, 2006.

[83] Y. Choi and S. Lee, "Injectivity Conditions of 2D and 3D Uniform Cubic B-spline Functions," *Graphical Models*, vol. 62, pp. 411–427, December 2000.

[84] I. Barandiaran, I. Macia, E. Berckmann, D. Wald, M. Dupillier, C. Paloc, and M. Grana, "An automatic segmentation and reconstruction of mandibular structures from CT-data," in *Proceedings of the 10th IDEAL*, IDEAL'09, (Berlin, Heidelberg), pp. 649–655, Springer-Verlag, 2009.

[85] M. Styner, K. Rajamani, L. Nolte, G. Zsemlye, G. Szekely, C. Taylor, and R. Davies, "Evaluation of 3D Correspondence Methods for Model Building," in *IPMI*, pp. 63–75, 2003.

[86] R. Davies, C. Twining, T. Cootes, J. Waterton, and C. Taylor, "3D Statistical Shape Models Using Direct Optimisation of Description Length," in *ECCV 2002* (A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, eds.), vol. 2352 of *Lect. Notes Comput. Sci.*, pp. 1–17, Springer Berlin, Heidelberg, 2002.

[87] H. Thodberg, "Minimum Description Length Shape and Appearance Models," in *IPMI*, pp. 51–62, Springer, 2003.

[88] S. Belongie, J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Transactions Pattern Analysis Machine Intelligence*, vol. 24, pp. 509–522, April 2002.

[89] J. Munkres, "Algorithms for the Assignment and Transportation Problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, pp. 32–38, March 1957.

[90] T. Rohlfing, C. Maurer, D. Bluemke, and M. Jacobs, "Volume-Preserving Nonrigid Registration of MR Breast Images Using Free-Form Deformation with an Incompressibility Constraint," *IEEE Transactions Medical Imaging*, vol. 22, pp. 730–741, 2003.

[91] M. Kortgen, M. Novotni, and R. Klein, "3D Shape Matching with 3D Shape Contexts," in *7th Central European Seminar on Comp. Graph.*, 2003.

[92] J. Friedman, J. Bentley, and R. Finkel, "An Algorithm for Finding Best Matches in Logarithmic Expected Time," *ACM Transactions Mathematical Software*, vol. 3, pp. 209–226, Sept. 1977.

[93] C. Maurer, R. Qi, and V. Raghavan, "A Linear Time Algorithm for Computing Exact Euclidean Distance Transforms of Binary Images in Arbitrary Dimensions," *IEEE Transactions Pattern Analyis Machine Intelligence*, vol. 25, pp. 265–270, February 2003.

[94] A. Brett and C. Taylor, "A Method of Automated Landmark Generation for Automated 3D PDM Construction," *Image and Vision Computing*, vol. 18, pp. 739–748, 1998.

[95] Y. Wang, B. Peterson, and L. Staib, "Shape-Based 3D Surface Correspondence Using Geodesics and Local Geometry," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 644–651, 2000.

[96] T. Heimann, I. Wolf, T. Williams, and H. Meinzer, “3D Active Shape Models Using Gradient Descent Optimization of Description Length,” in *in Proceedings IPMI*, pp. 566–577, Springer, 2005.

[97] C. Brechbuhler, G. Gerig, and O. Kubler, “Abstract Parametrization of Closed Surfaces for 3-D Shape Description,” 1996.

[98] E. Dijkstra, “A note on Two Problems in Connexion With Graphs,” *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[99] C. Xu and J. Prince, “Gradient Vector Flow: A New External Force for Snakes,” in *IEEE Proceedings Conf. On*, pp. 66–71, 1997.

[100] M. Hassouna and A. Farag, “Multistencils Fast Marching Methods: A Highly Accurate Solution to the Eikonal Equation on Cartesian Domains,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 1–12, 2007.

[101] X. Liang, R. Jacobs, B. Hassan, L. Li, R. Pauwels, L. Corpas, P. Souza, W. Martens, M. Shahbazian, A. Alonso, and I. Lambrichts, “A Comparative Evaluation of Cone Beam Computed Tomography (CBCT) and Multi-Slice CT (MSCT): Part I. On Subjective Image Quality,” *European Journal of Radiology*, vol. 75, no. 2, pp. 265 – 269, 2010.

[102] N. Gerlach, G. Meijer, T. Maal, J. Mulder, F. Rangel, W. Borstlap, and S. Berge, “Reproducibility of 3 Different Tracing Methods Based on Cone Beam Computed Tomography in Determining the Anatomical Position of the Mandibular Canal.,” *Journal of Oral and Maxillofacial Surgery*, vol. 68, no. 4, pp. 811–7, 2010.

[103] Bicon Dental Implants. `http://www.bicon.com/product_info/pi_implants.html`, 501 Arborway, MA 02130, Boston, 2011.

Disseminations

1. A. Klein, D. J. Kroon, Y. Hoogeveen, L. J. Schultze Kool, W. K. J. Renema, and C. H. Slump, "Multimodal Image Registration by Edge Attraction and Regularization Using a B-spline Grid," in *Medical Imaging 2011: Image Processing, Lake Buena Vista, Florida, USA* (D. M. Dawant and D. R. Haynor, eds.), vol. 7962 of *Proceedings of SPIE*, (Bellingham, USA), p. 796220, SPIE, February 2011.

2. Y. Mazaheri, L. Bokacheva, D. J. Kroon, O. Akin, H. Hricak, D. Chamudot, S. Fine, and J. A. Koutcher, "Semi-automatic Deformable Registration of Prostate MR Images to Pathological Slices," *Journal of Magnetic Resonance Imaging*, vol. 32, pp. 1149–1157, September 2010.

3. D. J. Kroon, C. H. Slump, and T. J. J. Maal, "Optimized Anisotropic Rotational Invariant Diffusion Scheme on Cone-Beam CT ," in *13th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI 2010), Beijing, China*, vol. 6363 of *Lecture Notes in Computer Science*, (Berlin), pp. 221–228, Springer Verlag, September 2010.

4. D. J. Kroon and C. H. Slump, "Coherence Filtering to Enhance the Mandibular Canal in Cone-Beam CT Data," in *Proceedings of the 4th Annual Symposium of the IEEE-EMBS Benelux Chapter, Enschede*, (Enschede), pp. 41–44, IEEE EMBS Benelux Chapter, 2009.

5. D. J. Kroon and C. H. Slump, "MRI Modality Transformation in Demon Registration," in *IEEE International Symposium on Biomedical Imaging: From Nano to Macro, ISBI '09, Boston, MA*, (USA), pp. 963–966, IEEE Signal Processing Society, 2009.

6. D. J. Kroon, E. S. B. van Oort, and C. H. Slump, "Multiple Sclerosis Detection in Multispectral Magnetic Resonance Images with Principal Components Analysis," in *3D Segmentation in the Clinic: A Grand Challenge II: MS lesion segmentation, New-York, USA*, (Website), pp. 604–617, Kitware, September 2008.

7. D. J. Kroon, C. H. Slump, M. Sluzewski, and W. J. J. van Rooij, "Image Based Hemodynamic Modelling of Cerebral Aneurysms and the Determination of the Risk of Rupture," in *Medical Imaging 2006: Physiology, Function, and Structure from Medical Images, San Diego, USA*, vol. 6143 of *Proceedings of SPIE*, pp. 718–727, The International Society for Optical Engineering (SPIE), February 2006.

Software Publications

All software is written in Mathworks Matlab and C++.
URL: www.mathworks.com/matlabcentral/fileexchange/authors/29180

Rendering

Viewer3d*, medical volume data rendering. Shaded, maximum intensity and slice rendering. Including annotation and segmentation tools.*

Showvol Isosurface Render*, interactive ISO-surface rendering.*

OpenGl .Net Examples*, Microsoft .Net based openGL rendering in Matlab.*

Patch Software Render*, software renderer, supporting textured polygons, and shadow volume calculation.*

File Format

Dicom Toolbox, Tags Read and Write*, read and write Dicom volumes, and read write private tags.*

Read Medical Data 3D*, read medical file formats, for example Philips V3D, HDR analyze and VMP brainvoyager.*

Wave front Obj Toolbox*, read and write wavefront object files, with geometric data, such as polygons.*

Image Filtering

Fast Non-Local Means 1D, 2D color and 3D*, edge preserving Non-Local means filtering.*

Image Edge Enhancing Coherence Filter Toolbox*, edge enhancing, non-anisotropic diffusion filtering, optimized for rotational invariance.*

Hessian Based Frangi Vesselness filter*, vessel enhancing filter based on local eigenvalue analysis.*

Bias Field Corrected Fuzzy C-Means*, bias field correction for MRI and clustering of image intensities.*

Registration and Matching

Shape Contex Based Corresponding Point Models*, Construct point corresponding model through shape-context point matching constrained with a B-spline registration grid.*

Bspline Grid, Image and Point based Registration*, Rueckert free form deformation (FFD) image registration, and Lee et al. point registration. Including Jacobian constraints, affine registration, and mutual information.*

Multimodality non-rigid demon algorithm image registration*, basic and optimizer based demon image registration. Including code to register multi-modal images.*

Fast/Robust Template Matching*, Fast SSD and normalized cross correlation, for 2D and 3D template matching .*

Lucas Kanade affine template tracking*, Lucas Kanade template tracking.*

Object Detection and segmentation

Active Shape Model (ASM) and Active Appearance Model (AAM)*, 2D and 3D active shape and active appearance model, training and segmentation.*

Viola Jones Object Detection*, Viola jones, object and face detection.*

OpenSurf*, Scale invariant feature point detection and matching.*

Snake Active Contour*, 2D and 3D object contour segmentation with active contour.*

Mesh Tools

Mesh2Tetra*, Fit tetrahedrons inside a triangulated mesh*

Smooth Triangulated Mesh*, curvature based smoothing of a triangulated mesh.*

Triangle Area and Angles*, accurate angles and area calculation for triangulated data.*

Triangular Mesh Refinement*, smooth 4-split refinement of triangulated mesh.*

Finite Iterative Closes Point*, register point clouds affine.*

Polygon2Voxel*, Converts a triangulated mesh into a binary volume.*

Optimization and classification

Classic Adaboost classifier*, Two class Adaboost classifier.*

Basic PCA based log-Likelihood classifier*, Maximum log-likelihood classifier.*

Fminlbfgs : Fast Limited Memory Optimizer*, quasi Newton optimizer.*

Uncategorized

Accurate Fast Marching*, Shortest path detection and skeletonize of objects.*

DTI and Fiber Tracking*, Diffusion tensor imaging of nerve-fibers in the brain.*

Isocontour*, Marching squares, 2D implementation of fast marching algorithm.*

Hist Connect*, Improved histograms for sparse spatial correlated 1D and 2D data.*

Separate Kernel in 1D kernels*, Separate ND kernels for faster image filtering.*

Region growing*, Region growing 2D image segmentation.*

Web Server, *Matlab html webserver, supporting uploads and m-file execution.*

Kinect Matlab, *Kinect wrapper code, depth camera and supporting skeleton tracking*

Summary

Accurate information about the location of the mandibular canal is essential in case of dental implant surgery [48], because entering the canal space during implant placement can damage the inferior alveolar nerve or blood vessels. Cone beam CT (CBCT) is becoming an increasingly utilized imaging modality in dental examinations, with a factor of ten lower dose than multi-slice CT [36]. The goal of our research is to find an automatic method which can segment the mandibular canal in CBCT. This allows an accurate safety margin around the canal in surgery planning.

In the last thirteen years, eight methods for automatic mandibular canal segmentation were published. The first methods focus on segmentation of the canal from CT [27] [28] [29] [30] [31] [32] and the later methods are developed for lower dose CBCT data [33] [34] [35]. The first approaches do not include shape constraints from training, and need user based initialization. In 2006, Rueda et al. [30] have produced a model with a priori shape information based on 2D active appearance models (AAM), followed by Kainmueller et al. [33] in 2009 with an 3D active shape model (ASM) model. The mean distance to manual annotation of the mandibular canal of the method of Kainmueller is around $1.1mm$. Almost all methods from literature, assume a black canal surrounded by higher intensities, and use fast marching [100] or Dijkstra's algorithm [98] to find a more accurate position of the mandibular canal. In 2011 Kim et al. [35] have published a paper which does not use shape-constraints, but out performs existing methods with a mean distance to ground truth nerve location of around $0.7mm$

In this thesis we develop and evaluate five methods for mandibular canal localization. The methods, Lukas Kanade tracking (LK) [49], B-spline registration [60], demon registration [66], 3D active shape model (ASM) [25] and active appearance model (AAM) [26].

The first method does not use training information and is based on LK template tracking of the canal. The methods starts tracking from a user selected point inside the mandibular canal. The two registration methods are demon-registration and B-spline registration which are used for mandibular bone and canal segmentation. In a pre-process we automatically select from a list of manually segmented images the one which closest matches the test image. Then we register the images to obtain the mandibular canal position.

The fourth method, is a 3D ASM model which extends on work of Kainmueller et al. [33]. An ASM models learns variations between corresponding points in training data sets, which are used as shape constraints during segmentation. We can extend the model by learning the combination of variations in shape and appearance data,

which is our last method, the AAM.

We test several methods to find the corresponding points. The first is the minimum description length (MDL) [86] [96] method which finds point correspondences by optimizing the point positions to minimize the model variance and obtain the most compact ASM. MDL is mainly suitable for sphere like objects therefore we introduce a new method. Our method finds correspondences based on surface registration in which we use shape context (SC) [88] as match criterion.

We have obtained 13 cone-beam CT (CBCT) scans from the department of oral and maxillofacial surgery, Radboud university Nijmegen medical centre. The mandibular canals are annotated in these scans by a medical expert, and the mandibular bone is also segmented.

In the results we measure the bone segmentation performance of our methods. This is done by calculating the overlap between automatically segmented mandibular bone and manually segmentations. The B-spline registration, demon registration ASM and AAM method have approximately the same score for bone segmentation with a Dice coefficient of around 0.75.

We also measure the performance of mandibular canal localization. This is done by calculating the distance between manual annotation of the mandibular canal and automatic localization. The registration methods have the worst performance, with a mean distance of more than $4mm$. The LK tracking method has a mean distance error of $3.0mm$, which is mainly due to template drift at the ends of the mandibular canal. The AAM and ASM are robust and have a mean distance to manually annotation of respectively $2.0mm$ and $2.3mm$.

The mean distance error of the ASM is high compared to the latest literature. Therefore we investigate a few post processing methods to increase the accuracy.

The first is filtering the CBCT data with an edge preserving optimized anisotropic diffusion filter, to remove noise and enhance the bony edges of the mandibular canal. The second, is 3D snake [55] [56] optimization of the ASM surface describing the bone. In our case we have a low number of training images, thus our ASM does not incorporate all the possible variations between the mandibles. A snake is a deformable model, constrained by internal energy such as bending energy. With the snake we optimize the surface description from ASM, and obtain a mean Dice coefficient of 0.8. As final optimization we use fast marching to find the dark tunnel through the bone. This tunnel often corresponds to the mandibular canal. Because of the low contrast between canal and surroundings, we first filter the data with a circular filter to lower the intensities in the canal. We limit the fast-marching to the inside of the bone, by setting all pixels outside the bone to a high intensity using the bone-shape obtain before. The average mean distance between expert and ASM segmentation of mandibular canal does not decrease and remains $2.3mm$.

The AAM is extended using the mandible AAM as initialization of an AAM of only the mandibular canal. By flipping the left and right canals we obtain twice as much data sets. If we look at results some good localization results improve, but the other canal localization results decrease. Finally we use a method in which we increase the influence of pixels close to the mandibular canal, on the AAM of the mandible. This increases our canal localization accuracy to an average mean distance to expert segmentation of 1.88mm and mean STD of 1.05mm.

In conclusion, our ASM and AAM method outperform the mandibular canal localization of LK tracking, B-spline and demon registration methods. The accuracy of the ASM method on bone segmentation can be increased by adding a snake post-processing step. Fast marching only increases the accuracy of already good ASM mandibular localizations, and decreases the accuracy from bad ASM segmentations. Our AAM with weighted intensities has a mean distance error of 1.88mm which is higher than the 1.3mm inter observer distance [102] of human segmentation. This is also higher than the best results in literature, 0.7mm. The main reasons are probably the low number of training data sets and low contrast between canal and surroundings due to a low dose. The accuracy of the introduced ASM and AAM method can probably be increased to 1.1mm [33], if we use more training data sets.

Samenvatting

Nauwkeurige informatie over de locatie van het mandibular kanaal is essentieel in mond en kaak chirurgie [48]. Want als het kanaal word beschadigd tijdens het plaatsen van een implantaat, kan de inferior alveolar zenuw of bloedvaten beschadigd raken. Cone beam CT (CBCT) word steeds vaker gebruikt om een beeld te krijgen van de kaak voor een dergelijke tandartsingreep. CBCT heeft een factor 10 lagere dosis dan normale CT [36]. Het doel van ons onderzoek, is het ontwerpen van een automatische methode, voor het segmenteren van het mandibular kanaal in CBCT. Dit zal een nauwkeurige veiligheidsmarge geven, die gebruikt kan worden bij het plannen van de ingreep.

In de afgelopen dertien jaren zijn acht methoden gepubliceerd voor het automatisch segmenteren van het mandibular kanaal. De eerste methoden focussen op het segmenteren van het kanaal uit CT scans [27] [28] [29] [30] [31] [32], maar latere methoden zijn ontworpen voor lage dosis CBCT data [33] [34] [35]. De eerste methoden gebruiken geen aangeleerde mandibula vormen om de segmentatie te ondersteunen. Vaak moet ook het begin van het kanaal door de gebruiker worden aangegeven. In 2006, introduceerde Rueda et al. [30] een model die van te voren aangeleerde geometrische informatie gebruikt tijdens de segmentatie. Deze informatie werd aangeleerd met behulp van een 2D active appearance model (AAM). Deze publicatie werd gevolgd door een publicatie van Kainmueller et al. [33] die in 2009 een 3D active shape model (ASM) voor segmentatie van de mandibula en kanalen gebruikte. De gemiddelde afstand van automatische tot handmatig annotatie van het the mandibular kanaal is in de methode van Kainmueller ongeveer $1.1mm$. Bijna alle literatuur gaat er van uit dat het kanaal een lage intensiteit heeft en omgeven is door pixels met een hogere intensiteit. Daardoor is het mogelijk fast marching [100] of Dijkstra's algoritme [98] te gebruiken om een nauwkeuriger positie van mandibular kanaal te vinden. In 2011 publiceerde Kim et al. [35] een paper met een methode waarin geen aangeleerde kaakvormen worden gebruikt. Deze methode gaf betere resultaten dan alle andere methoden, met een gemiddelde afstand tot expert annotatie van ongeveer $0.7mm$

In dit proefschrift beschrijven wij het ontwerp en de evaluatie van vijf methoden voor het lokaliseren van het mandibular kanaal. Namelijk Lukas Kanade tracking (LK) [49], B-spline registratie [60], demon registratie [66], 3D active shape model(ASM) [25] en een active appearance model (AAM) [26].

De eerste methode gebruikt geen van te voren aan geleerde informatie en is gebaseerd op LK template tracking van het kanaal. De methode begint het kanaal te volgen vanaf een door de gebruiker geselecteerd punt in het mandibular kanaal. De twee registratie methoden zijn demon-registratie en B-spline registratie, deze worden gebruikt om mandibula en het mandibular kanaal te segmenteren. Van te voren selecteren we uit een lijst van handmatig gesegmenteerde kaken, diegene die het best past bij de test scan. Daarna registeren we deze gesegmenteerde scan op de test scan om zo de locatie van het mandibular kanaal te verkrijgen.

Onze vierde methode is de 3D ASM methode die door Kainmueller et al. [33] is geïntroduceerd en door ons uitgebreid. Een ASM model leert de geometrische variaties tussen correspondeerde punten in training data. De aangeleerde variatie en correlatie wordt tijden de segmentatie gebruikt om de segmentatievorm te begrenzen. Deze limitatie houd in dat alleen object vormen mogelijk zijn die overeen komen met de variatie in de trainingdata. We kunnen dit model uitbreiden, door het model ook textuur en grijswaarde informatie aan te leren. Dit doen we met onze laatste methode, de AAM.

Voor het leren van de vorm variaties van de mandibular zijn corresponderende punten tussen de oppervlaktes van onderkaaken nodig. We hebben verschillende methoden getest om deze punten te vinden. De eerste methode is minimum description length (MDL) [86] [96]. Deze methode vindt corresponderende punten door de punten zo te verplaatsen dat de variatie tussen de punten zo klein mogelijk wordt, en de beschrijving door de ASM zo compact mogelijk. MDL is voornamelijk bruikbaar voor bolvormige objecten, daarom introduceren we een nieuwe methode. Onze methode vindt corresponderende punten door middel van oppervlakte registratie, waarbij we gebruik maken van shape context (SC) [88] om de overeenkomende plekken op de oppervlakten te vinden.

We hebben 13 cone-beam CT (CBCT) datasets ontvangen van de mond, kaak- en aangezichtschirurgie afdeling van het Radboud university nijmegen medical centre. De mandibular kanalen zijn geannoteerd door een medische specialist. Ook de mandibula is daarna gesegmenteerd.

In de resultaten in hoofdstuk 10, meten we de prestatie van de methoden op het gebied van segmentatie van de mandibula. Dit wordt gedaan door de overlap tussen automatisch en handmatig gesegmenteerde mandibula botten te vergelijken. De B-spline registratie, demon registratie, ASM en AAM methode hebben ongeveer dezelfde bot segmentatie Dice coëfficiënt van ongeveer 0.75.

We meten ook de prestatie van de algoritmes als het gaat om het lokaliseren van het mandibular kanaal. Dit wordt gedaan door de afstand te meten tussen de handmatige annotatie en automatische annotatie van het kanaal. De registratie methoden presenteren het slechtst, met een gemiddelde afstand van meer dan $4mm$ tussen de verschillende annotaties. De LK tracking methode heeft een gemiddelde afstandsfout van $3.0mm$. Deze fout komt voornamelijk door dat de method van het kanaal wegdrijft tijdens het volgen, waardoor de nauwkeurigheid aan het eind van het mandibular kanaal laag is. De AAM en ASM methode zijn robuust maar hebben een gemiddelde afstand tot handmatige annotatie van respectievelijk 2.0 en $2.3mm$.

De gemiddelde afstandsfout van het ASM model is hoog in vergelijking met recente literatuur. Daarom onderzoeken we een aantal nabewerkingsmethoden om de nauwkeurigheid te verhogen.

De eerst stap, is het filteren van CBCT data met een geoptimaliseerd rand beschermend anisotroop diffusie filter. Dit filter verwijderd ruis in de scan terwijl het de uit bot bestaande randen van het mandibular kanaal verduidelijkt. De tweede optimalisatie methode is een 3D snake [55] [56]. Deze optimaliseert het door het ASM verkregen oppervlakte van het bot. In ons geval hebben we erg weinig training scans, daarom is het onmogelijk voor onze ASM methode om all mogelijke variaties tussen mandibula botten te leren kennen. Een snake is een vervormbaar model, wat vervormd onder druk van interne energieën zoals buig energie, en externe energie zoals randen in de scan. Met de snake verbeteren we de oppervlakte beschrijving die we met de ASM hebben verkregen. Resulterend in een gemiddelde Dice coëfficiënt van 0.8. Als laatste optimalisatie gebruiken we fast marching. Deze methode vindt de donkerste tunnel door de mandibula, wat vaak gelijk is aan de locatie van het mandibular kanaal. Maar omdat in onze data het contrast tussen kanaal en omgeving erg laag is, gebruiken we eerst een circulair convolutie filter om de grijswaarden in het kanaal te verlagen. We limiteren fast-marching tot de binnenkant van het bot, door alle pixels buiten het bot een hoge grijswaarde te geven. Hiervoor gebruiken we de botvorm die we eerder met behulp van de ASM hebben verkregen. De afstand tussen expert segmentatie en automatisch segmentatie van het mandibular kanaal, neemt niet af en blijft gemiddeld $2.3mm$.

We breiden ook de AAM methode uit, door de mandibula AAM te gebruiken als initialisatie van een AAM van alleen het mandibular kanaal. Door het spiegelen van de kanalen krijgen we twee keer zoveel data sets. Deze kanaal AAM blijkt sommige al goede resultaten te verbeteren maar verslechterd de andere kanaal lokalisatie resultaten. Als laatste testen we ook een AAM methode waarin we pixels dicht bij het kanaal een grotere invloed geven dan de andere mandibula model pixels. Dit blijkt goed te werken en we verkrijgen een gemiddelde afstandsfout van $1.88mm$ en gemiddelde STD van $1.05mm$.

Conclusie, onze ASM en AAM methode presenteren beter op het gebied van mandibular kanaal lokalisatie dan LK tracking, B-spline en demon registratie methoden. De nauwkeurigheid, van de ASM methode op het gebied van bot segmentatie kan worden verhoogd door een snake te gebruiken na het ASM model. Fast marching verhoogt alleen de nauwkeurigheid van scans waarin het kanaal al redelijk nauwkeurig was gelokaliseerd, en verlaagd de nauwkeurigheid van matig gelokaliseerde kanalen. Onze uitgebreide AAM methode heeft een gemiddelde afstandsfout van $1.88mm$ wat hoger is dan het verschil tussen experts $1.3mm$ [102] en slechter dan het beste resultaat in literatuur met een nauwkeurigheid van $0.7mm$.

De belangrijkste reden voor de lage nauwkeurigheid, is waarschijnlijk het kleine aantal training scans, en het lage contrast tussen kanaal en omgeving als gevolg van een lage CT dosis. De nauwkeurigheid van de geïntroduceerde ASM en AAM modellen kunnen waarschijnlijk worden verhoogd naar $1.1mm$ [33], als er meer datasets beschikbaar zijn.

Dankwoord

Dit promotieonderzoek had niet tot stand kunnen komen zonder financiële steun van de provincie Overijssel, voor het innovatieve project van technische geneeskunde de Virtuele Patiënt. Dit project werd mede mogelijk gemaakt door de faculteit Technische Natuurwetenschappen (TNW) en de faculteit Elektrotechniek, Wiskunde en Informatica (EWI).

Maar minstens zo belangrijke was de steun, hulp en interesse die ik ontving van de mensen om mij heen. In de hoop dat ik niemand vergeet, wil ik enkelen hiervan hieronder persoonlijk bedanken.

Allereerst wil ik mijn begeleider en promotor, Kees Slump bedanken, voor al de leuke projecten waar ik in alle vrijheid aan mocht werken, maar ook voor alle tijd en geduld die hij heeft gestoken in het corrigeren van mijn papers en proefschrift.

Van het ziekenhuis Nijmegen wil ik Thomas Maal en Stefaan Bergé bedanken voor hun inzet en het mee kunnen kijken in de keuken van de kaak, mond en aangezichtschirurgie. Het onderwerp van mijn thesis is ontstaan tijdens een bezoek van Niek Gerlach aan onze vakgroep, die ook onderzoek deed naar het mandibular canal. Hem wil ik bedanken voor alle medische data, mandibular canal annotaties en informatie.

De prettige werksfeer gedurende al die jaren, heb ik te danken aan mijn collega's van de leerstoel Signalen en Systemen. In het bijzonder wil ik mijn oud kamergenoten bedanken, Rene voor het goede wiskundige advies en leuke vakantie trips, en Joost als kamer oudste voor alle tips en gezelligheid. Daarnaast wil ik Bas en Anne bedanken voor alle gezelligheid, samen volleyballen en snowboarden. Gerbert en Almar voor de leuke conferenties, bijbehorende vakanties, en het samenwerken en de gezelligheid binnen onze medische subgroep. In het bijzonder wil ik ook Anneke, Sandra, Geert-Jan en Henny bedanken voor alle zaken, die het werk merkbaar gemakkelijker maakten en alle keren gezellig koffie drinken.

Verder wil ik mijn familie en vrienden, voor hun interesse in mijn werk, maar ook voor hun bijdrage aan alle gave momenten en gezellige momenten buiten werktijd. Vooral mijn ouders, die er altijd voor mij waren en die mij er opwezen dat ik toch echt dat proefschrift een keer moest afmaken. Ook wil ik mijn huisgenoten bedanken, en speciaal Richard voor al die gave Hup-reizen.

Ten slotte, mijn vriendin Patrica, bedankt voor al jouw liefde, samen de toekomst tegemoet.