Automatic Detection of Anatomical Landmarks in Three-Dimensional MRI

Hannes Järrendahl
Master of Science Thesis in Electrical Engineering

Automatic Detection of Anatomical Landmarks in Three-Dimensional MRI

Hannes Järrendahl

LiTH-ISY-EX--16/4990--SE

Supervisor: Andreas Robinson

ISY, Linköpings universitet

Magnus Borga, Thobias Romu

AMRA

 Examiner: Maria Magnusson

ISY, Linköpings universitet

Computer Vision Laboratory
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Copyright © 2016 Hannes Järrendahl
Abstract

Detection and positioning of anatomical landmarks, also called points of interest (POI), is often a concept of interest in medical image processing. Different measures or automatic image analyzes are often directly based upon positions of such points, e.g. in organ segmentation or tissue quantification. Manual positioning of these landmarks is a time consuming and resource demanding process.

In this thesis, a general method for positioning of anatomical landmarks is outlined, implemented and evaluated. The evaluation of the method is limited to three different POI; left femur head, right femur head and vertebra T9. These POI are used to define the range of the abdomen in order to measure the amount of abdominal fat in 3D data acquired with quantitative magnetic resonance imaging (MRI). By getting more detailed information about the abdominal body fat composition, medical diagnoses can be issued with higher confidence. Examples of applications could be identifying patients with high risk of developing metabolic or catabolic disease and characterizing the effects of different interventions, i.e. training, bariatric surgery and medications.

The proposed method is shown to be highly robust and accurate for positioning of left and right femur head. Due to insufficient performance regarding T9 detection, a modified method is proposed for T9 positioning. The modified method shows promises of accurate and repeatable results but has to be evaluated more extensively in order to draw further conclusions.
Acknowledgments

I would like to thank my supervisor Andreas Robinson and my examiner Maria Magnusson at ISY for continuous feedback and support. I would also like to thank my supervisors Magnus Borga and Thobias Romu for all the help and guidance, as well as giving me the opportunity to do my thesis work at AMRA.

Linköping, August 2016
Hannes Järrendahl
Contents

Notation ix

1 Introduction 1
  1.1 Motivation ................................................. 1
  1.2 Aim ......................................................... 4
  1.3 Limitations ............................................... 4

2 Theory 5
  2.1 Related work ............................................... 5
  2.2 Magnetic resonance imaging .............................. 6
    2.2.1 Spin physics ............................................ 6
    2.2.2 Radio frequency pulse ................................ 7
    2.2.3 Gradient encoding ..................................... 8
    2.2.4 Pulse sequences ....................................... 9
    2.2.5 Dixon imaging ......................................... 9
    2.2.6 Intensity correction ................................... 10
  2.3 Image registration ......................................... 12
    2.3.1 Linear registration .................................... 12
    2.3.2 Morphon registration .................................. 13
    2.3.3 Local phase ............................................. 14
    2.3.4 Scale space ............................................. 16
    2.3.5 Deformation ............................................. 16
    2.3.6 Regularization ......................................... 17
  2.4 Feature extraction ......................................... 17
    2.4.1 Haar features .......................................... 17
    2.4.2 Sobel operators ....................................... 18
  2.5 Random regression forests ............................... 21
    2.5.1 Regression trees ....................................... 21
    2.5.2 The ensemble model .................................... 23
    2.5.3 Randomness in random regression forests ............ 24
    2.5.4 Feature selection ..................................... 25
    2.5.5 Forest parameters .................................... 25
3 Method
   3.1 Proposed method ................................................. 27
      3.1.1 Search space initialization ............................... 27
      3.1.2 Determining search space size ............................. 29
      3.1.3 Filter bank generation .................................... 29
      3.1.4 Feature extraction ......................................... 30
      3.1.5 Forest training ............................................. 30
      3.1.6 Regression voting ......................................... 30
   3.2 Modified method for T9 detection .............................. 31

4 Results
   4.1 Experimental setup ............................................. 33
   4.2 Results right and left femur .................................. 34
   4.3 Results T9 ...................................................... 44
   4.4 Results modified method for T9 ............................... 50

5 Discussion
   5.1 Method evaluation ............................................. 51
   5.2 Further work .................................................. 54

6 Conclusion .......................................................... 55

Bibliography ............................................................ 57
Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAT</td>
<td>Visceral adipose tissue</td>
</tr>
<tr>
<td>ASAT</td>
<td>Abdominal subcutaneous adipose tissue</td>
</tr>
<tr>
<td>MRI</td>
<td>Magnetic resonance imaging</td>
</tr>
<tr>
<td>FID</td>
<td>Free induction decay</td>
</tr>
<tr>
<td>LF</td>
<td>Left femur</td>
</tr>
<tr>
<td>RF</td>
<td>Right femur</td>
</tr>
<tr>
<td>CNN</td>
<td>Convolutional neural network</td>
</tr>
<tr>
<td>POI</td>
<td>Point(s) of interest</td>
</tr>
<tr>
<td>NCC</td>
<td>Normalized cross-correlation</td>
</tr>
</tbody>
</table>
1.1 Motivation

In medical image processing, detection and positioning of specific anatomical landmarks, or points of interest (POI), is often an important issue. Different measures or automatic image analyzes might be directly based upon positions of such points, e.g. in organ segmentation. Manual positioning of these landmarks is a time consuming and resource demanding process.

AMRA is a medical technology company, specializing in image processing, that provides a service for classification and quantification of fat and muscle groups. The body fat can be divided into visceral (VAT), subcutaneous (ASAT) and ectopic adipose tissue. VAT refers to intra-abdominal fat which surrounds abdominal organs. Large amounts of VAT has been shown to be the most significant risk factor for cardiovascular events that is related to body composition. It is also connected to diabetes, liver disease and cancer [1]. ASAT on the other hand has been observed to provide different health benefits. These include decreased risk of abnormal glucose metabolism, decreased cardiometabolic risk and improved insulin sensitivity [2]. Ectopic fat accumulates in regions where it is not supposed to be, such as in the liver and in muscle tissue. Being able to get more accurate knowledge about the body fat composition opens up many possibilities. Some examples include identifying patients that have high risk of developing metabolic or catabolic disease and characterizing the effects of different interventions i.e. training, bariatric surgery and medications.

The fat and muscle segmentation is based on non-rigid registration between atlas segmented prototypes and the target volume in which quantification of fat and/or muscles is wanted. The segmentation is an automatic process to a large
extent [3]. However, AMRA’s anatomical definition of the abdominal range for determination of VAT and ASAT is based upon three landmarks, the upper edges of the left and right femur heads and the upper edge of vertebra T9. These POI are placed manually today. The POI configurations in the different anatomical planes can be seen in figures 1.1 and 1.2. An example of fat and muscle segmentations can be seen in figure 1.3.

**Figure 1.1:** Positions of the top of femur heads according to AMRA’s anatomical definition. To the left the femur positions are shown in the coronal plane. To the right the femur positions are shown in the transverse plane. Image source: AMRA.

**Figure 1.2:** Positions of vertebrae Sacrum-T9 according to AMRA’s anatomical definition. To the left a sagittal cross section of the spine with the position of every vertebra is shown. To the upper right the position of vertebra Sacrum is shown in the transverse plane. To the lower right the position of vertebra T9 is shown in the transverse plane. Image source: AMRA.
1.1 Motivation

The conditions for this master thesis are unique because of the large amount of data that AMRA provides. AMRA has access to thousands of quantitative fat and water signals acquired with 3D magnetic resonance imaging (MRI) by using the two point Dixon technique [4], which is described in section 2.2.5. Since the data is quantitative, the voxel intensities in the fat signal directly represent the amount of fat in a certain voxel. A method that can identify anatomical landmarks by incorporating quantitative MRI, machine learning and image registration is of interest since it is not an extensively tested approach to solve the landmark detection problem.

Figure 1.3: Coronal cross sections of a patient with fat and muscle segmentations. To the left, ASAT is highlighted in light blue and VAT in light red. To the right, different muscle groups are highlighted in different colors. Image source: AMRA.
1.2 Aim

The main objective of this master thesis was to develop a method for automatic POI placement in 3D MRI using machine learning and image processing. The method should take into account that anatomical a priori information is available in the form of prototypes provided by AMRA [5]. Comparison and evaluation of the results when using different training data, as well as implementing selected algorithms in Python was part of the project goal. The master thesis answers the following questions:

- How can a general method for detection of anatomical landmarks be designed and implemented?
- How good performance can be achieved with a general method for detection of the previously specified anatomical landmarks?

1.3 Limitations

The method should be as general and adaptive as possible for positioning of different anatomical landmarks. Validation will however be restricted to evaluation of the landmarks mentioned previously (right and left femur and vertebra T9). These positions are of special interest for AMRA and data labeled with these landmarks is available for training and validation. Training and validation have mainly been conducted on data of the same resolution. The data used to evaluate the method was limited to be quantitative fat and water signals acquired with 3D MRI.
This chapter provides the theory behind concepts that are vital for the thesis and the proposed method outlined in section 3. The chapter aims to give an overview of related work (section 2.1) and theory behind the following topics; magnetic resonance imaging (section 2.2), image registration (section 2.3), feature extraction (section 2.4) and random regression forests (section 2.5).

2.1 Related work

Anatomical landmark detection in 3D images is a growing field of work and many methods have been developed to solve this problem. Existing learning-based methods are typically either classification-based [6], [7] or regression-based [8], [9], [10]. In classification-based landmark detection, voxels near the POI to be detected are labeled as positives and the remaining voxels are labeled as negatives. Typically, a single strong classifier is then trained to distinguish voxels that are close to the landmark. For example, Yang et al. [7] proposed a classification-based approach using a convolutional neural network (CNN) for landmark localization on the distal femur surface by treating the problem as a binary classification. This was done by converting the 3D volume into three sets of 2D images (one for each dimension x, y, z) and labeling image slices that are close to the wanted landmark as positives and the other slices as negatives.

Regression on the other hand tries to predict a continuous value instead of assigning samples with discrete classes. For example, Gau et al. [9] proposed a regression-based approach using a random regression forest to detect prostate landmarks. The use of random forests in computer vision and machine learning
Theory

Applications has increased dramatically in recent years. This is especially true for medical imaging applications due to fast run times even for large amounts of data [11]. Classification and regression trees (CART) have been around for many years and were first introduced by Breiman et al. in [12]. In the early 2000s, Breiman introduced the ensemble method of random forests in [13] as an extension to his previous work on decision trees. Random forests have proven to compare favourably over other popular learning methods, often with preferable generalization properties [11].

In this thesis, a random regression forest approach to the landmark detection problem was chosen. The random forest model is more extensively described in section 2.5. CNN may achieve state of the art accuracy in many applications, but random forests have the potential to be more efficient while still performing results in the same order of success [14]. The random regression forest approach was mainly chosen in favor of CNN due to the method’s previously shown success in medical applications.

2.2 Magnetic resonance imaging

Magnetic resonance imaging (MRI) is a non-invasive medical imaging technique that is used for soft tissue imaging. The most fundamental principle of MRI is the physical phenomenon of magnetic resonance (MR), which enables generation and acquisition of radio frequency (RF) signals. Since the theory behind MRI is extensive, it will only be covered briefly. The following sections (2.2.1 - 2.3) are based on the descriptions in [15] and [16].

2.2.1 Spin physics

The human body consists of approximately 60% water and consequently large amounts of hydrogen nuclei (i.e. protons). In quantum mechanics, nuclear particles can be described in terms of different quantum properties. A vital property for MRI is the spin. The spin property can be visualized as the nucleus spinning around its own axis, inducing a small magnetic field. The resulting magnetic field from a single proton is almost undetectable but the total net magnetization \( M_0 \) from sufficiently many protons is large enough to generate a detectable signal. In the original energy state, the protons net magnetization is zero since the spin axes are oriented in arbitrary directions. However, by adding a strong external magnetic field \( B_0 \), the proton spins will align parallel or anti-parallel to \( B_0 \). Protons that are aligned anti-parallel to \( B_0 \) are on a higher energy level than those that are parallel to \( B_0 \). Due to the protons constantly flipping between energy states, but with a slightly higher probability of ending in the lower energy state, there will be a small abundance of spins aligning parallel to \( B_0 \). This subset of excess protons in the lower energy state prevents the net magnetization from being...
ing cancelled out and adds up to a net magnetic vector pointing in the direction of the $B_0$ field.

### 2.2.2 Radio frequency pulse

Apart from aligning the proton spins, the external magnetic field will also cause the protons to precess around the $B_0$-axis with a certain frequency $\omega$. The precession frequency is called the Larmor frequency and is given by

$$\omega = \gamma B_0, \quad (2.1)$$

where $\gamma$ is the nucleus specific gyro-magnetic ratio and $B_0$ is the magnetic field strength given in tesla ($T$). Common field strengths used in medical imaging are 1.5 T and 3 T. Stronger field strength will result in stronger MR signals. From now on, the direction of the $B_0$ field will be denoted as the z-direction. An illustration of the proton precession can be seen in figure 2.1.

![Figure 2.1: An illustration of the proton spin precession around the z-axis with the Larmor frequency $\omega$. Image source: [16].](image)

In order to detect the MR signal, a receiver coil is used to convert the magnetic changes into electrical signals along a certain axis. Initially there is no magnetization orthogonal to the $B_0$ field since the projections of the magnetic vectors precessing in the z-direction cancel out each other in the $xy$-plane. By applying
a radio frequency (RF) pulse with the same frequency as the Larmor frequency, some of the protons will go into the higher energy level, aligning anti-parallel to the external magnetic field $B_0$. The magnetization in the z-direction therefore decreases to zero. The RF pulse will also cause the protons to synchronize and precess coherently. This leads to the magnetization in the z-direction $M_z$ to flip down into the $xy$-plane, establishing a transverse magnetization vector $M_{xy}$. The moving transverse magnetic field will induce a current in the receiver coil. This current is the MR signal and is called free induction decay (FID). Directly after the RF pulse has flipped the net magnetization vector into the $xy$-plane ($M_{xy}$) the signal will have highest amplitude. The signal magnitude will decrease as time progresses while the frequency stays the same. An illustration of how the RF pulse affects the net magnetization can be seen in figure 2.2.

Since the $B_0$ field is static, the net magnetization in the z-direction ($M_z$) will eventually recover to full strength while the magnetization in the $xy$-plane decays until it disappears. These processes are called T1 and T2 relaxation and will occur with different rates depending on the surrounding tissue. The recovery of $M_z$ is given by

$$M_z = M_0(1 - e^{-t/T_1}),$$

(2.2)

where $t$ is time, $M_0$ is the total magnetization and $T_1$ is a tissue dependent time constant. The decay of $M_{xy}$ due to the spins getting out of phase is given by

$$M_{xy} = M_0e^{-t/T_2},$$

(2.3)

where $t$ is time, $M_0$ is the total magnetization and $T_2$ is a tissue dependent time constant. $T_1$ is defined as the time when 63% of the magnetization in the z-direction has recovered. $T_2$ is defined as the time it takes for the transverse magnetization to have decreased by 37%. $T_1$ is also dependent on the strength of the magnetic field, a strong field will result in longer $T_1$ times.

### 2.2.3 Gradient encoding

In order to generate the MR image, spatial encoding is needed to determine the signal origin. By applying gradient fields along the different axes using gradient coils, the total magnetic field is altered spatially. A slice-selection gradient is used to change the magnetic field linearly along the z-axis. The Larmor frequency will now vary along the z-axis, implicating that by using RF pulses with different frequencies, spatial encoding in the z-direction can be achieved. In order to determine the signal origin within the z-slice, a phase encoding gradient and a frequency encoding gradient are used. The phase encoding gradient is turned on directly after the RF pulse for a very short period of time along the y-axis. This
Figure 2.2: An illustration of how the RF pulse affects the net magnetization. In the left image, the magnetization in the z-direction is flipped down into the xy-plane. In the right image, $M_z$ has started to recover its original magnitude. Image source: [16].

will change the phases of the precessing protons in the $y$-direction. The position dependent phase-shift enables spatial encoding along the $y$-axis. The frequency encoding gradient changes the proton frequency along the $x$-axis such that the signal contains frequencies below and above the Larmor frequency. The signal is now spatially encoded along all three directions.

2.2.4 Pulse sequences

A pulse sequence is a repetition of RF pulses. Two important concepts are the repetition time $T_R$ and echo time $T_E$. $T_R$ defines the time period between the RF pulses and $T_E$ is the time between the FID generation and the MR signal acquisition. A commonly used pulse sequence is the spin echo. This sequence is initiated with a $\pi/2$-pulse that flips the $M_z$ magnetization into the $xy$-plane. The $\pi/2$-pulse is repeated with time $T_R$. The proton spins will begin to dephase in the transverse plane due to $T_2$ relaxation. At the time $T_E/2$, a $\pi$-pulse is used to invert the magnetization from $M_x$ to $-M_x$. Thus the spins of the protons will start to rephase. By applying the $\pi$-pulse, the FID signal intensity will not decay as fast as it would have else wise. Without the use of the $\pi$-pulse, the $T_2$ relaxation is referred to as $T_2^*$ decay. After time $T_E$, the spins are re-phased which removes the $T_2^*$ effect. An illustration of the spin echo sequence can be seen in figure 2.3.

2.2.5 Dixon imaging

The Dixon imaging technique can be used to produce two separate images from the MR signal; one of the voxel water content and one of the voxel fat content. This method was first proposed by Dixon in [4]. Reasons for separating the MR signal into these two components is that quantitative MR images can be obtained
Figure 2.3: The spin echo sequence. First the $\pi/2$-pulse flips the magnetization from the z-axis (A) to the xy-plane (B). The spins then begin to dephase due to T2 relaxation (C). After time $T_E/2$, the spins are out of phase (D) and a $\pi$-pulse is applied (E), inverting the spins. The spins are now starting to rephase (F) and are in phase again at time $T_E/2$. Image source [16].

by doing intensity homogeneity correction. This is further explained in section 2.2.6.

The basic procedure of Dixon imaging is to acquire two MR images, one where the fat and water signal are in phase ($I_1$) and one where they are out of phase ($I_2$) by $\pi$ radians. This is given by

\begin{align*}
I_1 &= f_w - f_f, \\
I_2 &= f_w + f_f.
\end{align*}

By taking the difference and the sum of $I_1$ and $I_2$ respectively, the water signal $f_w$ and the fat signal $f_f$ can be obtained [5]. This is however only the ideal case, in reality the phase often varies over the images due to magnetic field inhomogeneities. To compensate for this, robust phase correction has to be performed before calculating $f_w$ and $f_f$.

2.2.6 Intensity correction

Common problems with MR images are intensity inhomogeneities due to different factors, such as magnetic field strength alterations and variations in coil sensitivity. By making the valid assumption that pure fat voxels should have equal intensity, independent of voxel position, these errors can be corrected [17]. Based on the observations, an intensity bias field $\tilde{s}$ can be estimated and used for correction of the original image according to
2.2 Magnetic resonance imaging

Determination of pure fat voxels is based on the relation between the fat signal intensity in a certain voxel and the sum of the water and fat voxels. Voxels where this normalized fat signal intensity is above some threshold are classified as pure fat voxels. From this classification, a binary fat certainty mask can be obtained. The mask is set to 1 where for all pure fat voxels and 0 otherwise. The certainty mask is then used to perform normalized averaging (2.7) in order to retrieve the bias field $\tilde{s}$. After correction with the bias field has been performed, the data is normalized such that voxels classified as pure fat voxels get unit intensity [17].

The estimation of the bias field can be done using a normalized averaging approach which works by computing local weighted averages of the fat intensities. This can be seen as a special case of normalized convolution. This is given by

$$I_{\text{corr}} = \frac{I_{\text{original}}}{\tilde{s}}.$$  (2.6)
\[ s = \frac{(s \cdot c) \ast a}{c \ast a}. \]  

Here \( a \) denotes an applicability function which corresponds to a low pass filter kernel when used in normalized averaging and \( c \) is the data certainty mask. Pointwise multiplication is denoted by \( \cdot \) and \( \ast \) denotes convolution. An extension to the normalized averaging method presented above is to construct a scale space of \( s \cdot c \) and \( c \) as described in [17]. This method is called MANA and allows for more robust generation of intensity-corrected, quantitative MR data. For every scale, the result is used to update the bias field \( \bar{s} \). If no scale space is used, \( s \) is simply equal to \( I_{original} \).

## 2.3 Image registration

The main objective of image registration for 3-dimensional data is to geometrically align two volumes by finding a point to point transformation field between a prototype volume \( I_p \) and a target volume \( I_t \). The registration can be performed using either a linear (e.g. affine or rigid) or a non-linear approach. Using a linear transformation model is less computationally demanding but non-linear approaches tend to yield more accurate registrations.

### 2.3.1 Linear registration

Many recent registration methods are based upon the linear optical flow equation, first described in [18],

\[ \nabla I^T v = \Delta I, \]  

(2.8)

where \( \nabla I \) denotes the volume gradient, \( v \) is the motion vector field describing how the volumes differ and \( \Delta I \) is the difference in intensity between \( I_p \) and \( I_t \). Due to the aperture problem, \( v \) can not be found directly by solving (2.8). By introducing a 12-dimensional parameter vector \( p = [p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}, p_{12}]^T \) and a base matrix \( B(x) \), the motion field can be modelled as

\[ v(x) = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} + \begin{bmatrix} p_4 & p_5 & p_6 \\ p_7 & p_8 & p_9 \\ p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = Bp, \]  

(2.9)

where \( B \) is the 3x12 basis matrix and \( x = [x, y, z]^T \). The first three parameters in the equation represent a translation and the last nine make up a transformation
matrix [19]. By using the motion field model, the error measure over the entire volume can be written as

\[ \epsilon^2 = \sum_i (\nabla I(x_i)^T B(x_i)p - \Delta I(x_i))^2. \]  

(2.10)

By differentiating the error measure with respect to the parameter vector and setting the derivative to zero, a linear equation system consisting of 12 equations is obtained. This equation system can now easily be solved and the best parameters can be found since \( L_2 \) norm minimization is used.

### 2.3.2 Morphon registration

The optical flow equation provides the fundamental basics for many registration methods. In this thesis, the main focus will be on the Morphon, first proposed in [20]. The Morphon is a non-linear registration method and is thus more advanced and computationally demanding than the method described in the previous section. The estimation of the motion field is based on local phase from quadrature filters, rather than from gradients. An advantage of using local phase is that the method becomes insensitive to variations in voxel intensity between \( I_p \) and \( I_t \). In other words, the brightness consistency constraint does not have to be fulfilled as in the case of gradient based registration. The concept of local phase is further explained in section 2.3.3.

Morphon registration is an iterative method and the pipeline can be divided into three main steps; estimation of the displacement field, accumulation of the displacement field and deformation [21]. In the displacement estimation step, the goal is to find a local description of how to deform the prototype to match the target image better. A commonly used similarity measure is the normalized cross-correlation (NCC). The NCC of two signals \( f \) and \( g \) with means \( \bar{f} \) and \( \bar{g} \) is defined as

\[ \langle \frac{f - \bar{f}}{||f||}, \frac{g - \bar{g}}{||g||} \rangle, \]  

(2.11)

where \( \langle \cdot, \cdot \rangle \) denotes the inner product and \( ||\cdot|| \) is the \( L_2 \) norm. As previously mentioned, the estimates are based upon differences in local phase. The displacement estimates are then used to update the deformation field. This step also involves a regularization which is described in section 2.3.6. The last step is to transform the original prototype with respect to the accumulated deformation field. This is commonly done using conventional methods such as linear interpolation. A Morphon registration example is shown in figure 2.5. The following sections (2.3.3-2.3.6) are based on the work in [19], [20], [21] and [22].
2.3.3 Local phase

Estimating optical flow between images is a common issue in image processing. As mentioned previously, advantages of doing registration based on local phase from quadrature filters are that it works for weak gradients and variations in image intensity. This means that the registration is dependent on structural similarities rather than on similarities in the intensity values.

The quadrature filter is complex in the spatial domain and gives different structural interpretations depending on the relation between the real and imaginary components in the filter response. The real part of the filter works as a line detector and the imaginary part works as an edge detector. This is illustrated in figure 2.6. Matching lines to lines and edges to edges results in a robust registration method.

Since phase by definition is one-dimensional, the phase has to be calculated in different directions to have a meaningful interpretation for multi-dimensional sig-

Figure 2.5: Coronal cross sections of target, prototype and deformed prototype obtained after Morphon registration. The cut feet in 2.5c is probably due to problems with edge effects in the Morphon registration.
2.3 Image registration

Figure 2.6: An illustration of local phase $\phi$ from quadrature filter response. A completely real filter response indicates that a line has been detected. A completely imaginary filter response indicates that an edge has been detected. Image source: [19].

In the linear registration algorithm, three quadrature filters oriented along the x, y and z directions are used and one equation system is solved for the whole volume. The Morphon instead uses six quadrature filters and solves one equation system for each voxel. The filters are evenly distributed on the half sphere of an icosahedron. The resulting displacement field estimate is proportional to the local difference in phase in every direction $k$. The displacement field $d$ can be found by minimizing

$$
e^2 = \sum_k^N (c_k T (\Delta \phi_k \hat{n}_k - d))^2 ,$$

(2.12)

where $\hat{n}_k$ is the direction of filter $k$, $N$ is the number of quadrature filters, $\Delta \phi_k$ is the difference in phase between the volumes, $T$ is the local structure tensor, $d$ is the sought after displacement field to be optimized and $c_k$ is a certainty measure for filter $k$. By using the magnitude of the filter responses as certainty measure, an indication of how much the estimated displacement can be trusted is given. A large filter response indicates that the phase estimate is more accurate than if it is small. For a small filter response, the phase is more or less just noise.

The structure tensor $T$ represents signal magnitude and orientation in every neighborhood. From the filter responses of the six quadrature filters, the tensor can be calculated as
\[ T = \sum_{k} |q_k| \left( \frac{5}{4} \hat{n}_k \hat{n}_k^T - \frac{1}{4} I \right) \]  

(2.13)

where \( q_k \) is the quadrature filter response and \( I \) is the identity tensor. By including the tensor in the error measure, displacement estimates perpendicular to lines and edges are reinforced.

### 2.3.4 Scale space

As long as the displacement estimates indicate that the result can be improved by additional morphing, the algorithm is further iterated. The method also incorporates a scale space scheme to allow fast and successful morphing even for quite dissimilar images. The morphing is first done at a very coarse scale and iteratively performed on finer levels to refine the displacement estimations from each scale. This enables performing good estimations of both large displacements, as well as local displacements.

### 2.3.5 Deformation

By minimizing (2.12), a displacement field \( d \) can be found for the current scale or iteration. Since every interpolation step has a low pass filtering effect on the data, the displacement fields should be accumulated into a total field. By morphing the original prototype with the accumulated field in every iteration and comparing the result to the target data, a displacement field for the current iteration can be estimated. The accumulated field \( d_a' \) should be updated according to

\[ d_a' = \frac{d_a c_a + (d_a + d_i) c_i}{c_a + c_i}, \]  

(2.14)

where \( d_a \) is the latest accumulated field, \( d_i \) is the estimated displacement field for the current iteration \( i \) and \( c_a, c_i \) are the corresponding certainty measures (i.e. the magnitude of the quadrature filter responses) for both displacement fields. Updating the certainty of the accumulated displacement field is somewhat troublesome but one approach is to update the certainty according to

\[ c_a = \frac{c_a^2 + c_i^2}{c_a + c_i} \]  

(2.15)

However, it might be reasonable to increase the weight of \( c_i \) for the finer resolution scales.
2.3.6 Regularization

One way to interpret the displacement estimates is to think of them as local forces that are trying to pull the prototype in different directions. Preferably, the simplest deformation field possible is wanted. Letting the local estimates transform the prototype entirely according to the displacement field would most probably lead to a disrupted result. By introducing a regularization factor to the deformation, this problem can be avoided. A common approach is to use normalized convolution with respect to the certainty measure $c_i$ in combination with low pass filtering using a Gaussian kernel. The regularization is given as

$$d_i = \frac{(d_i \cdot c_i) \ast g}{c_i \ast g},$$

(2.16)

where $g$ is a Gaussian kernel that defines the elasticity of the displacement field. The Gaussian filter works as a low pass filter and thus suppression of large variations in $d_i$ will depend on the filter size. By increasing the Gaussian filter size, the deformation field is restricted to become stiffer and a small kernel allows for more elastic deformations.

2.4 Feature extraction

In order to find a non-linear mapping between local volume appearance and voxel POI displacement, feature extraction design is vital. Choices of features that are commonly used in machine learning applications when regarding 2D images are extensive. Apart from low-level features such as color or intensity, many different filters are often used to extract features. In this thesis, Haar feature extraction from the raw water or fat signal and from gradient filtered data will be used as input features to the system. The theory behind Haar feature extraction and the Sobel operator is given in the following section 2.4.1 and 2.4.2.

2.4.1 Haar features

An attractive approach when working with random forests is to use a large quantity of randomly generated context features. This is due to the random forest's property of sorting out what features that are important or not for the total success of the regression. This is more thoroughly described in section 2.5.4. Haar filters can be used to extract features that remind of Haar basis functions as described in [23]. Haar features are commonly used in computer vision applications for detection of different objects such as faces or cars. The Haar filter in 2D most commonly consists of two rectangular regions. When applied to an image, the filter computes the difference between the mean of the pixel intensities within the
two regions. The Haar filter definition can be further extended for use of more rectangular regions. Examples of 2D Haar filters can be seen in figure 2.7.

![Figure 2.7: Three examples of 2D Haar filters. The Haar feature is extracted by applying the filter to an image and taking the intensity difference between the dark and the bright regions.](image)

The 3D generalization of the Haar filter is often defined as the local mean intensity in a cuboid region or as the intensity difference over two randomly positioned cuboid regions within a certain patch $R$, centered around a voxel $x$. This can be expressed as

$$f(u, v) = \frac{1}{|R_1|} \sum_{u \in R_1} I(u) - p \frac{1}{|R_2|} \sum_{v \in R_2} I(v), R_1 \in R, R_2 \in R, p \in \{0, 1\}$$  \hspace{1cm} (2.17)

where $R_1$ and $R_2$ are cuboid regions centered around voxel positions $u$ and $v$ and $I$ is the intensity within a certain region. $|\cdot|$ denotes cardinality, i.e. the number of voxels $R_1$ and $R_2$ respectively. The stochastic variable $p$ determines whether the difference between two randomly positioned cuboid regions should be computed or if only the local mean of a single cuboid region should be computed. $p$ assumes the value 0 or 1 with different probabilities. The use of 3D Haar features, defined as in (2.17), has proven to be successful in different classification and regression applications, including the work described in [6], [9] and [24]. In [6], 3D Haar features are used in combination with machine learning in order to capture the non-linear relationship between local voxel appearance and different types of brain tissue. An example of a 3D Haar filter can be seen in figure 2.8.

### 2.4.2 Sobel operators

First order derivatives filters are commonly used in image processing to enhance edge structures. For a 3-dimensional signal $f$, the gradient is defined as

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{bmatrix}$$  \hspace{1cm} (2.18)
2.4 Feature extraction

Figure 2.8: A conceptual example of a 3D Haar feature for $p = 1$, illustrated in the coronal plane. The green rectangle represents the search space and the white cross indicates the position of a voxel $x$. $x$ is centered in the middle of the Haar patch $R$, which is represented by the turquoise square. The Haar feature is computed as the mean intensity difference over two randomly positioned, cuboid regions, $R_1$ and $R_2$ (indicated by the orange and red rectangles), that lie within $R$.

The gradient vector is oriented in the direction where the rate of change is greatest. The magnitude of $\nabla f$ is given as

$$M(x, y, z) = \sqrt{f_x^2 + f_y^2 + f_z^2}. \quad (2.19)$$

Discrete approximations of the first order derivatives can for example be implemented with the Sobel operator [25]. The resulting images from Sobel filtering in the $x$ respectively $y$ direction and the gradient magnitude are shown in figure 2.9.

The Sobel operator in 2D consists of two separable operations; smoothing with a triangle filter orthogonal to the derivative direction and taking the central difference in the derivative direction. In 2D, the Sobel operator only uses the intensity values in a $3 \times 3$ region around each pixel in order to approximate the gradient. The Sobel operators in the $x$ respectively $y$ direction are given as

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (2.20)$$
Figure 2.9: Coronal cross sections of a MR water signal with the Sobel operator applied in the x and y directions. In 2.9c the magnitude of the combined derivatives (x, y, z) is shown. In 2.9a and 2.9b gray represents 0, whereas in 2.9c black represents 0.

The concept can be extended to multiple dimensions. For 3-dimensional data, the Sobel operator is a $3 \times 3 \times 3$ filter kernel. For example, the 3D Sobel operator in the z-direction is given as

$$
G_z(-1) = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}, \quad G_z(0) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad G_z(1) = \begin{bmatrix} -1 & -2 & -1 \\ -2 & -4 & -2 \\ -1 & -2 & -1 \end{bmatrix}
$$

The kernel in (2.21) approximates the derivative in the z-direction and applies smoothing in the x and y directions.
2.5 Random regression forests

The basic idea of random forests is to combine multiple weak learner trees into one strong predictor. In this section, focus will be on regression since classification is not relevant for the thesis. The main principle of regression and classification forests is however the same. Regression aims at predicting a continuous value while classification divides the data into discrete class bins. The following section is based on [11] and [13].

2.5.1 Regression trees

To understand how regression forests work, it is necessary to understand the functionality of the individual trees. The trees make up for a hierarchical structure where each leaf contains a prediction. The final prediction in a certain leaf is obtained by aggregating over all training data samples that reach that leaf in the training phase. During testing, every input data sample is passed through the tree, splitting into the right or left branch at every node according to the optimized split parameters $\theta_j$. When the data sample has reached a leaf node, a predicted value is obtained. At each node $j$ in the tree, the best data split parameters $\theta_j$ are determined during the training stage by maximizing an objective function $I$. Only subsets of the training data reach the different tree branches, so for instance, $S_1$ denotes the subset of the training data $S$ that reaches the first node. The following properties applies to the tree subsets: $S_j = S_j^L \cup S_j^R$, $S_j^L \cap S_j^R = 0$, where $S_j$ denotes the set of the training data $S$ that reaches node $j$ and $S_j^L, S_j^R$ denotes the sets of $S_j$ that split up into the left respectively right sub branches. A general tree structure with associated training data can be seen in figure 2.10. The optimal split parameter function can be written as

$$\theta_j = \arg \max I(S_j, \theta), \quad (2.22)$$

where $\theta$ is a discrete set of possible parameter settings. The split parameters of the weak learner model (i.e. the parameters of the binary node split) can be denoted as $\theta = (\phi, \psi, \tau)$, where $\phi = \phi(v)$ selects some features from the feature vector $v$, $\psi$ determines the geometric properties of how the data is separated (e.g. by using an axis aligned hyperplane as shown in figure 2.10) and $\tau$ is the actual threshold determining where to split the data.

The concepts of entropy and information gain provide the foundation for decision tree learning. Entropy is an uncertainty measure associated with the stochastic variable that is to be predicted and information gain is a measure of how well a certain split separates the data $S_j$. Information gain is often defined as
Figure 2.10: The dark circles in (a) represent training data. In this example, the input feature space $y = v(x)$ is one-dimensional. The position along the $y$ axis is the associated continuous ground truth label. The light gray circle and the dashed gray line represents a previously unseen test input. The $y$ value is not known for this sample. In (b), a binary regression tree is shown. The amount of training data passing through the tree is proportional to the thickness of every branch. High entropy is denoted with red and low entropy is denoted with green. A set of labeled training samples $S_0$ is used to optimize the splitting parameters of the tree. As the tree grows from the root towards the leaves, the confidence of the prediction increases. Image source: [11].

\[
I = H(S) - \sum_{i \in \{L,R\}} \frac{|S_i|}{|S|} H(S_i),
\]

(2.23)

where $H$ denotes the entropy. $|\cdot|$ denotes cardinality, i.e., the number of elements in a set. In other words, the information gain can be formulated as the difference between the entropy of the parent and the weighted sum of the entropy of the children. The entropy for continuous distributions is defined as

\[
H(S) = - \int y p(y) \log(p(y)) dy,
\]

(2.24)

where $y$ is the continuous variable that is to be predicted and $p$ is the probability density function which can be estimated from the set of training data $S$. Another simpler choice is to approximate the density $p(y)$ with a Gaussian-based model. When only a point estimate of $y$ is wanted rather than a fully probabilistic output, it is common in regression tree applications to use variance reduction as information gain instead of the definition given in (2.23). The variance reduction can be expressed as a least-squares error reduction function, which is maximized in order to find the optimal split parameters. This can be written as
\[ I = \sum_{v \in S_j} (y_k - \bar{y}_j)^2 - \sum_{i \in \{L, R\}} \left( \sum_{v \in S_{j,i}} (y_k - \bar{y}_j)^2 \right), \quad (2.25) \]

where \( S_j \) denotes the set of the training data that reaches node \( j \), \( \bar{y}_j \) is the mean of all training data points \( y_k \) arriving at the \( j^{th} \) node. \( v = (x_1, ..., x_n) \) is a multi-dimensional feature vector. The expression in (2.25) can easily be extended for multivariate output.

### 2.5.2 The ensemble model

The combination or ensemble of several regression trees is typically referred to as a regression forest. Random regression forests have been shown to achieve a higher degree of generalization than single, highly optimized regression trees. The output from the regression forest is simply the average over all tree outputs, which is illustrated in figure 2.11. The averaging is given by

\[ p(y|v) = \frac{1}{T} \sum_{t=1}^{T} p_t(y|v), \quad (2.26) \]

where \( T \) is the total number of trees in the forest and \( p(y|v) \) the estimated probability function.

**Figure 2.11:** Illustration of the ensemble model. In this toy example, a forest consisting of three trees is present. The output is simply the average over all tree predictions. Image source: [11].
2.5.3 Randomness in random regression forests

The reason for incorporating randomness in the random regression forest model is essentially to increase the generalization of the learner. Bagging and randomized node optimization are two different approaches to reduced overfitting and achieve increased generalization. However, they are not mutually exclusive. According to Breiman [13], bagging in combination with random feature selection generally increases the accuracy of the final prediction.

Bagging is an abbreviation of the term bootstrap aggregating. A randomly sampled subset $S_t$ (also called a bootstrap subset) of the training data set $S$ is drawn for each tree in the forest. About one third of the samples in the total training data set are usually left out in each bootstrap subset. By only training each tree with its corresponding subset $S_t$, specializing the split parameters $\theta$ to a single training set is avoided. The training efficiency is also greatly increased when not the whole set $S$ is used. Another reason for using bagging is that a generalization error of the combined tree ensemble can be estimated during training. For each data sample $x$ in the training set $S$, the output is aggregated only over the trees for which $S_t$ does not contain $x$. The result is called an out-of-bag estimate of the generalization error.

The idea behind random node optimization is to only consider a random subset of the features when determining the optimal splitting parameters $\theta$ at each node. By doing so, overfitting can be reduced and the training efficiency increased since the set of features often is very large. The amount of randomness is determined by the parameter $\rho = |T_j|$, where $T_j$ is a subset of all features $T$. With the extreme case of $\rho = |T|$, no randomness is introduced to the model which means that all trees will look the same if bagging is not used. On the other hand, if $\rho = 1$, the tree generation is fully random. This is illustrated in figure 2.12.

![Figure 2.12:](image-source)

---

**Figure 2.12:** In (a) approximately the whole feature set $T$ is used to determine the optimal split parameters. This results in highly correlated trees which is not desirable. In (b) the splits are fully random which results in low tree correlation. Image source: [11].
2.5.4 Feature selection

During tree training, how much each feature increases the weighted tree information gain can be computed according to (2.25). After averaging over the information increase for each feature in the forest, it is reasonable to rank the importance of the features according to this measure. This information can be used to remove redundant features and thus decreasing computational complexity. The importance of the $j^{th}$ feature can also be measured after training by randomly permuting the values of the $j^{th}$ feature among the training data. An out-of-bag error is then computed for the perturbed data set and the original data. By averaging the difference in out-of-bag error over all trees, before and after the feature permutation, an importance measure for the $j^{th}$ feature can be obtained. The standard deviation of the differences is used to normalize the importance score. A large value for feature $j$ indicates that it is an important feature that carries much information.

2.5.5 Forest parameters

The most important parameters that influence the behaviour of a random regression forest are the forest size $T$, the maximum tree depth $D$ and the amount of randomness $\rho$. These parameters are to be handled with care and depend to a large extent on the properties of the training data. Testing accuracy theoretically increases with forest size $T$. The increase will however flatten after a certain forest size. Very deep trees are prone to overfitting but very large amounts of training data usually suppresses this effect. The amount of features $\rho$ to consider when determining a split is commonly set to about half of $|T_j|$. This is just a guideline since the choice for $\rho$ is not very crucial.
In this section, the experimental setup and final method are presented. A general outline of the system pipeline implemented in Python is also given as pseudo code.

### 3.1 Proposed method

The proposed method was mainly evaluated for detection of femur heads and vertebra T9 according to the thesis objectives specified in section 1.2. T9 detection was found to be problematic and therefore some minor modifications to the proposed method had to be done. A detailed description of the general method is described in sections 3.1.1 - 3.1.6 and the modifications for T9 detection are specified in section 3.2. The Python implementation consists of three separate modules: filter pre-selection, training and validation (testing). In the three modules, different non-overlapping target data subsets are used. A pseudo code overview of the implementation is given in algorithm 1.

#### 3.1.1 Search space initialization

In order to efficiently determine the position of a POI in 3D data, reduction of the search space was essential. An initial guess of the target POI position was used to center the search space in the target data. In the initialization step during validation (step 25 in algorithm 1), *a priori* information in the form of atlas segmented prototypes were used in combination with Morphon registration (see section 2.3)
Algorithm 1 General algorithm

1: procedure POI detection (pre-selection, training and testing)  
2:  
3: pre-selection:  
4:   filter_bank ← generateHaarFilters( filter_parameters )  
5: for target in pre-selection_subset do  
6:   load target  
7:   init_pos ← ground_truth + positional_noise  
8:   data ← searchSpaceExtraction( target, init_pos )  
9:   features ← data * filter_bank  
10:  randomForest() ← trainRandomForest( features, ground_truth )  
11:  filter_importances ← randomForest()  
12:  filter_bank ← filterReduction( filter_bank, filter_importances )  
13:  
14: training:  
15: for target in training_subset do  
16:   load target  
17:   init_pos ← ground_truth + positional_noise  
18:   data ← searchSpaceExtraction( target, init_pos )  
19:   features ← data * filter_bank  
20:  randomForest() ← trainRandomForest( features, ground_truth )  
21:  
22: testing:  
23: for target in testing_subset do  
24:   load target  
25:   init_pos ← morphonInitialization( target, prototypes )  
26:   data ← searchSpaceExtraction( target, init_pos )  
27:   features ← data * filter_bank  
28:   prediction ← randomForest( features )  
29:   pos ← regressionVoting( prediction )  
30:  return pos

in order to get a rough initial guess of the wanted POI position. By performing registration between the target and multiple prototypes, a displacement field was obtained for each prototype. The similarity in a local region around the transformed POI between the prototype and target should reasonably give an indication of how successful the final deformation was. To evaluate the degree of similarity, the NCC measure (2.11) was used. Local similarity should reasonably be more relevant than global similarity for the purpose of determining the initial POI position since the Morphon registration aims at minimizing a global error function. This implies that even if the global correlation between target and deformed prototype was high, the registration might have been unsuccessful locally. The target and prototype data was low pass filtered with a small Gaussian kernel before calculating the NCC in order to prevent noise and small structures.
from affecting the result negatively. The deformed ground truth POI in the prototype with the highest NCC value was chosen as the initial POI position in the target data. In the registration step, 29 unique prototypes with well defined POI positions for right femur (RF), left femur (LF) and vertebra T9 were used. The prototypes were provided by AMRA and consisted of 3D MRI data, containing neck to knee information.

Since volume registration is a time consuming process, search space initialization in the pre-selection and training steps was performed by adding a positional Gaussian noise to the ground truth POI position in the target data. The ground truth POI were deployed in the data sets by experienced analysts according to the anatomical definitions given in section 1.1. Registration was thus only used in the testing (validation) step. The noise was designed to have a mean of the same magnitude as the mean deviation from the ground truth POI position when using the Morphon. The noise variance was set to match the variance of the mean deviation when using the Morphon.

### 3.1.2 Determining search space size

Suitable search space sizes for each POI (LF, RF, T9) were determined by calculating the NCC between a number of targets and their corresponding deformed prototypes for a range of different sizes. The NCC measures were then ordered with respect to the deviation from the true POI position for every target. By ensuring that a high NCC measure corresponded to a small deviation from the POI ground truth, the search space was assumed to be of suitable size for that POI. Search spaces of the determined sizes were then used in the separate modules.

### 3.1.3 Filter bank generation

In order to maximize the probability of successfully capturing the non-linear relationship between local voxel appearance and the distance to a certain POI, many Haar filters were randomly generated. The filter generation was implemented in Python according to (2.17). After the filter generation, pre-selection was performed according to the method described in section 2.5.4 in order to reduce computational complexity. The `RandomTreesRegressor` class from the Python library *scikit-learn* [26] was used to perform filter pre-selection. Initially, 10000 Haar filters were generated and reduced to about 200 according to the feature importance measure described in section 2.5.4. A subset of 1000 feature filters were considered in the random node optimization to reduce computational complexity. The resulting 200 filters were then used to extract features from the training data.
3.1.4 Feature extraction

After search space extraction, convolution of the extracted data with the generated Haar filter bank was performed. This was then performed for every input target. The final method was evaluated on both the water signal and the fat signal as well as Sobel filtered data. The filtered data was stored in an array with final dimension \(n \times m\) where \(n\) is equal to the number of voxels in the reduced search space times the number of targets and \(m\) is the number of extracted features, which is equal to the number of selected Haar filters.

3.1.5 Forest training

The forest training was implemented using the Python machine learning library scikit-learn [26]. The ExtraTreesRegressor class was used in both training and validation. The difference from the RandomForestRegressor class is that additional randomness is introduced to the model. Instead of always choosing the best split according to the information gain measure, thresholds are drawn at random for each feature and the best of these randomly generated thresholds is chosen as the splitting point [27]. As input during forest training, the extracted feature data and corresponding ground truth was used. As ground truth, the displacement vector between every voxel in the limited search space and wanted POI position was used. This vector regression (multivariate) approach was found to improve on the results when compared to only predicting the Euclidean distance to the target POI. This can be seen in the results in section 4.

3.1.6 Regression voting

The resulting estimated landmark position for a target was obtained by performing regression voting (step 29 in algorithm 1). Every voxel in the search space casts one vote to the discrete position closest to \(x + d\) where \(x\) is a voxel position and \(d\) is the corresponding predicted displacement vector. The discrete voxel position that receives the most votes determines the landmark location. If multiple positions get approximately the same number of votes, the mean of these positions is chosen as the predicted POI. The results were validated by testing the trained regression forest on features extracted with the generated filter bank on previously unseen target data sets. No cross validation was needed for detection of femur heads and T9 since sufficient amounts of data was available. The deviation in Euclidean distance from the true POI position was calculated for all test targets and the mean deviation calculated as

\[
\bar{x} = \frac{1}{N} \sum_{k=1}^{N} \|r_k - t_k\|,
\]  

(3.1)
3.2 Modified method for T9 detection

where \( r_k \) is the POI position given by the regression voting and \( t_k \) is the true POI position for target \( k \).

The standard deviation \( \sigma \) was calculated as

\[
\sigma = \sqrt{\frac{\sum_{k=1}^{N} (x_k - \bar{x})^2}{N - 1}},
\]

(3.2)

where \( x_k = \|r_k - t_k\| \).

3.2 Modified method for T9 detection

Due to unsatisfying results regarding the T9 detection, as can be seen in table 4.3, an alternative approach to the proposed method described in section 3.1 was outlined. Additional vertebra positions for Sacrum - T10 were manually deployed in 84 data sets by experienced analysts with anatomical knowledge. The configuration of these landmarks in the spine can be seen in figure 1.2. The mean vector between every vertebra POI was calculated in 30 of the data sets. These vectors were then used to incrementally initialize the mid point position of the search space. By doing so, initialization using the Morphon could be preformed only for LF and/or RF instead. This was an advantage since the POI initialization using the Morphon was found to be more robust for LF and RF than for T9, which can be seen by comparing the results in tables 4.3 and 4.2, 4.1. The standard deviation was low for the mean vectors which indicated that it was reasonable to base the POI initialization upon the addition of the previous POI and corresponding mean vector. For each vertebra, a separate filter bank and regression forest was trained to refine the POI position given from the vector addition. To summarize, LF/RF was first initialized using Morphon registration. This position was then refined with a trained random forest and corresponding filter bank. The mean vector between LF/RF and Sacrum was added to the LF/RF POI to initialize the center of the search space for vertebra Sacrum. This procedure was then repeated until T9 was reached. Pseudo code of the algorithm used in the modified method can be seen in algorithm 2.

Other attempts were also made to initialize the center of the search space more accurately in the \( z \)-direction for T9 detection. The correlation between a patients height and the distance between the top of femur head and T9 was found to be fairly strong. Based on this observation, attempts to determine if the POI initialization was within reasonable range of T9 in the \( z \)-direction were made. Unfortunately the correlation was not sufficiently strong to be able to draw any conclusions regarding the Morphon initialization.
Algorithm 2 Modified algorithm for T9 detection

1: procedure MODIFIED T9 (TESTING)
2:     load target
3:     poi = LF
4:     init_pos ← morphonInitialization( target, prototypes, poi )
5:     j = 0;
6:     for poi in poi_list do
7:         load filter_bank && randomForest()
8:         data ← searchSpaceExtraction( target, init_pos )
9:         features ← data * filter_bank
10:        prediction ← randomForest( features, ground_truth )
11:        pos ← regressionVoting( prediction )
12:        if poi != T9 then
13:            pos ← pos + mean_vec[j]
14:            j++;  
15:    else
16:        return pos
This chapter presents the results obtained when using the method outlined in chapter 3. Used parameter settings are presented in section 4.1. Both quantitative and qualitative results are given.

### 4.1 Experimental setup

The voxel resolution of the data sets that were used for pre-selection, training and validation was approximately $3 \times 2 \times 2$ mm. The patch size of the Haar filters was set to $7 \times 7 \times 7$ voxels, which corresponds to $21 \times 14 \times 14$ mm in the given resolution. This filter size was observed to give a suitable trade-off between time consumption and performance. It was sufficiently small to get low computational complexity when performing multiple convolutions in the feature extraction step and still large enough to capture contextual information in the search space. This patch size was also suggested in other similar applications [6], [28]. Testing with patch sizes of size $9 \times 9 \times 9$ and $11 \times 11 \times 11$ voxels were performed and did not improve the results but increased the computation time. The search space was set to $45 \times 45 \times 45$ mm for LF and RF since it is a sufficiently large region to encapsulate the top of the femur head and fulfilled the conditions mentioned in section 3.1.2. For T9, the search space was designed to have dimensions $30 \times 45 \times 45$ to limit the extension in the z-direction. The reason for this was mainly due to problems with the repeating structure of the spine in the z-direction. Also, the patch size was increased to $9 \times 9 \times 9$ to fully be able to capture important context information in the xy-plane of the T9 vertebra. For pre-selection of features to keep, 20 data sets were used.
Regarding the regression forest parameters, no maximum depth or minimum node split conditions were used. No tree pruning will result in longer training times and the model will be more prone to capturing noise in train data. Restraining these parameters was however observed to have a negative effect on the performance. 50 trees were trained in both the feature selection forest and the final random forest. More trees should theoretically give more robust results but will also increase training times. However, no distinct improvement was seen when increasing the number of trees over 50. Both bagging and randomized node optimization were used in the forest training phase. A subset of half the features were used in each split for every tree, according to the theory in section 2.5.5.

### 4.2 Results right and left femur

The following results were obtained when testing the trained regression forest with corresponding filter bank on 119 unique, previously unseen target data sets. During filter pre-selection, 20 unique data sets with ground truth positions for LF and RF were used. During training, 127 unique data sets with ground truth positions for LF and RF were used. The pre-selection, training and testing sub-sets contained both male and female targets of varying ages. The results are presented in terms of the deviation in Euclidean distance from the ground truth POI position, maximum error and median error.

<table>
<thead>
<tr>
<th>Error</th>
<th>Morphon init</th>
<th>Water scalar</th>
<th>Water multi</th>
<th>Fat multi</th>
<th>Gradient multi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean ± σ</td>
<td>7.17 ± 3.73</td>
<td>4.53 ± 3.31</td>
<td>2.97 ± 2.00</td>
<td>3.44 ± 1.85</td>
<td>3.15 ± 2.15</td>
</tr>
<tr>
<td>Median</td>
<td>6.77</td>
<td>4.35</td>
<td>2.23</td>
<td>3.16</td>
<td>3.00</td>
</tr>
<tr>
<td>Max</td>
<td>19.70</td>
<td>21.08</td>
<td>9.26</td>
<td>9.26</td>
<td>11.43</td>
</tr>
</tbody>
</table>

In table 4.1 it is clear that the proposed method improves on the Morphon initialization. However, the max error of the scalar regression is larger than max error of the Morphon. The vector regression outperforms scalar regression both in terms of mean, median and max error. Training the system on Haar features extracted from the water, fat or gradient data does not seem to have a significant impact on the final results.
Table 4.2: Results for left femur (LF) given in mm. Multi refers to vector (multivariate) regression. $\sigma$ denotes standard deviation.

<table>
<thead>
<tr>
<th>Error</th>
<th>Morphon init</th>
<th>Water scalar</th>
<th>Water multi</th>
<th>Fat multi</th>
<th>Gradient multi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean $\pm \sigma$</td>
<td>7.57 $\pm$ 3.85</td>
<td>4.18 $\pm$ 2.47</td>
<td>3.67 $\pm$ 2.30</td>
<td>3.22 $\pm$ 2.22</td>
<td>3.44 $\pm$ 2.18</td>
</tr>
<tr>
<td>Median</td>
<td>6.99</td>
<td>3.73</td>
<td>3.16</td>
<td>3.16</td>
<td>3.16</td>
</tr>
<tr>
<td>Max</td>
<td>22.13</td>
<td>14.43</td>
<td>12.67</td>
<td>10.03</td>
<td>11.64</td>
</tr>
</tbody>
</table>

The results for LF in table 4.2 are very similar to those in table 4.1. The same pattern can be seen. Differences are that the performance of the scalar regression is closer to that of the vector regression. Also, the overall error is larger in the multivariate case. An overview of the results for LF and RF are given in figure 4.1. Figures 4.2 - 4.9 show examples of qualitative results. Figures 4.2 - 4.5 show a clear improvement on the Morphon initialization. Figures 4.6 - 4.9 show an example of when the regression based prediction seems to be more accurate than the manually positioned ground truth.

![Figure 4.1](image-url)
Figure 4.2: Qualitative test results for RF using vector regression and the water signal. Circles indicate landmark positions and the blue bounding box represents the search space. Red: Morphon estimation. Blue: Regression based prediction. Green: Ground truth landmark position. Close-ups in the coronal and sagittal planes are shown in figure 4.3.
4.2 Results right and left femur

Figure 4.3: Close ups of the qualitative results presented in figures 4.2a and 4.2b. Circles indicate landmark positions and the blue bounding box represents the search space. Red: Morphon estimation. Blue: Regression based prediction. Green: Ground truth landmark position.
Figure 4.4: Resulting scalar regression maps for RF in the different planes corresponding to the results presented in figure 4.2. Small values (blue) indicate that the position is predicted to be close to the target POI. Note that the ellipsoid appearance of the regression maps in the sagittal and coronal planes is a result of the lower resolution in the z-direction.
Figure 4.5: Example of resulting vector regression voting maps for RF in the different planes corresponding to the results presented in figure 4.2. The regression maps are scaled with the max number of votes given to a certain position. Dark red indicates that many votes were cast to this position and vice versa for dark blue. As can be seen, the voting distribution is concentrated around a few pixels.
Figure 4.6: Qualitative test results for RF using vector regression and the water signal. Circles indicate landmark positions and the blue bounding box represents the search space. Red: Morphon estimation. Blue: Regression based prediction. Green: Ground truth landmark position. Close-ups in the coronal and sagittal planes are shown in figure 4.7.
Figure 4.7: Close ups of the qualitative results presented in figures 4.6a and 4.6b. Circles indicate landmark positions and the blue bounding box represents the search space. **Red**: Morphon estimation. **Blue**: Regression based prediction. **Green**: Ground truth landmark position.
(a) Regression map in the coronal plane

(b) Regression map in the sagittal plane

(c) Regression map in the transverse plane

Figure 4.8: Resulting scalar regression maps for RF in the different planes corresponding to the results presented in figure 4.6. Small values (blue) indicate that the position is predicted to be close to the target POI. Note that the ellipsoid appearance of the regression maps in the sagittal and coronal planes is a result of the lower resolution in the z-direction.
4.2 Results right and left femur

Figure 4.9: Example of resulting vector regression voting maps for RF in the different planes corresponding to the results presented in figure 4.6. The regression maps are scaled with the max number of votes given to a certain position. Dark red indicates that many votes were cast to this position and vice versa for dark blue. As can be seen, the voting distribution is concentrated around a few pixels.
4.3 Results T9

The following results were obtained when testing the trained regression forest and corresponding filter bank on 119 unique, previously unseen target data sets. During filter pre-selection, 20 unique data sets with ground truth positions for T9 were used. During training, 127 unique data sets with ground truth positions for T9 were used. The pre-selection, training and test subsets contained both male and female targets of varying ages. The results are presented in terms of the deviation in Euclidean distance from the ground truth POI position, maximum error and median error.

Table 4.3: Results for T9 given in mm. Multi refers to vector (multivariate) regression. \( \sigma \) denotes standard deviation.

<table>
<thead>
<tr>
<th>Error</th>
<th>Morphon init</th>
<th>Water scalar</th>
<th>Water multi</th>
<th>Fat multi</th>
<th>Gradient multi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean ( \pm \sigma )</td>
<td>20.52 ( \pm ) 17.42</td>
<td>19.62 ( \pm ) 17.19</td>
<td>17.94 ( \pm ) 18.04</td>
<td>17.68 ( \pm ) 18.24</td>
<td>18.21 ( \pm ) 18.20</td>
</tr>
<tr>
<td>Median</td>
<td>17.43</td>
<td>14.87</td>
<td>12.06</td>
<td>15.71</td>
<td>15.68</td>
</tr>
<tr>
<td>Max</td>
<td>166.30</td>
<td>147.40</td>
<td>168.45</td>
<td>177.97</td>
<td>168.45</td>
</tr>
</tbody>
</table>

As can be seen in table 4.3, the results are very poor for T9 detection. The proposed method barely improves on the Morphon initialization, which is highly inaccurate in the first place. The max error of the Morphon is as much as 166 mm, and likewise, the max error of the proposed method is in the same range for all cases of data.

Table 4.4: Results for T9 with outlier removal given in mm. Multi refers to vector (multivariate) regression. \( \sigma \) denotes standard deviation.

<table>
<thead>
<tr>
<th>Error</th>
<th>Morphon init</th>
<th>Water scalar</th>
<th>Water multi</th>
<th>Fat multi</th>
<th>Gradient multi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean ( \pm \sigma )</td>
<td>11.80 ( \pm ) 5.01</td>
<td>10.78 ( \pm ) 6.65</td>
<td>8.34 ( \pm ) 3.81</td>
<td>\textbf{7.82} ( \pm ) \textbf{3.85}</td>
<td>8.78 ( \pm ) 5.61</td>
</tr>
<tr>
<td>Median</td>
<td>11.38</td>
<td>8.05</td>
<td>7.80</td>
<td>7.34</td>
<td>7.47</td>
</tr>
<tr>
<td>Max</td>
<td>25.14</td>
<td>32.04</td>
<td>19.23</td>
<td>19.96</td>
<td>26.20</td>
</tr>
</tbody>
</table>

With outliers from the Morphon initialization removed, the results for T9 are
given in table 4.4. If the deviation from the target POI in the z-direction was off by more than 15 mm after Morphon initialization, the result was sorted out. According to this definition, 68 out of the 119 validations were kept as inlier results. As can be seen, the proposed method improves on the initialization. As in the case with the femur heads, the vector regression is more robust than the scalar regression. The fat signal marginally gives the best performance. An overview of the results for T9 after outlier removal are given in figure 4.10. Figures 4.11 - 4.14 show examples of qualitative results. A clear improvement on the Morphon initialization can be seen.

Figure 4.10: Results for T9 after outlier removal. The bars show the mean deviation from the target POI in mm ± one standard deviation. The bars labeled Water, Fat and Sobel are results obtained using vector regression.
Figure 4.11: Qualitative test results for T9 using vector regression and the water signal. Circles indicate landmark positions and the blue bounding box represents the search space. Red: Morphon estimation. Blue: Regression based prediction. Green: Ground truth landmark position. Close-ups in the coronal and sagittal planes are shown in figure 4.12.
Figure 4.12: Close ups of the qualitative results presented in figures 4.11a and 4.11b. Circles indicate landmark positions and the blue bounding box represents the search space. Red: Morphon estimation. Blue: Regression based prediction. Green: Ground truth landmark position.
Figure 4.13: Resulting scalar regression maps for T9 in the different planes corresponding to the results presented in figure 4.11. The scalar regression maps for T9 does not have as well defined appearances as the regression maps for the femur heads in figures 4.4 and 4.8. This is especially true in the coronal and sagittal planes.
Figure 4.14: Resulting vector regression voting maps for T9 in the different planes corresponding to the results presented in figure 4.11. The regression maps are scaled with the max number of votes given to a certain position. Dark red indicates that many votes were cast to this position and vice versa for dark blue. The votes are more scattered and not as concentrated as in the case of the femur heads.
4.4 Results modified method for T9

The following results were obtained when the modified version of the proposed method described in section 3.2 was evaluated on 9 unique target data sets. Training was performed on 35 data sets and filter pre-selection on 10 data sets. A separate forest and filter bank were trained for each POI (LF, Sacrum, L5, L4, L3, L2, L1, T12, T11, T10, T9). The same forest and filter parameters as for T9, mentioned in section 4.1, were used for the vertebrae (Sacrum - T9). Only Haar extraction from the water signal in combination with vector regression was used.

<table>
<thead>
<tr>
<th>Error</th>
<th>Sacrum</th>
<th>L5</th>
<th>L4</th>
<th>L3</th>
<th>L2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean ± σ</td>
<td>8.08 ± 5.56</td>
<td>6.26 ± 3.12</td>
<td>4.41 ± 2.45</td>
<td>3.83 ± 0.91</td>
<td>6.36 ± 4.02</td>
</tr>
<tr>
<td>Median</td>
<td>6.70</td>
<td>4.35</td>
<td>4.46</td>
<td>4.35</td>
<td>6.70</td>
</tr>
<tr>
<td>Max</td>
<td>20.00</td>
<td>10.98</td>
<td>8.05</td>
<td>4.99</td>
<td>13.39</td>
</tr>
</tbody>
</table>

Table 4.5: Results for vertebrae Sacrum-L2 given in mm. σ denotes standard deviation.

<table>
<thead>
<tr>
<th>Error</th>
<th>L1</th>
<th>T12</th>
<th>T11</th>
<th>T10</th>
<th>T9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean ± σ</td>
<td>4.04 ± 1.65</td>
<td>5.09 ± 2.15</td>
<td>4.63 ± 0.81</td>
<td>4.50 ± 2.10</td>
<td>5.14 ± 1.72</td>
</tr>
<tr>
<td>Median</td>
<td>3.16</td>
<td>4.35</td>
<td>4.46</td>
<td>3.74</td>
<td>4.46</td>
</tr>
<tr>
<td>Max</td>
<td>6.70</td>
<td>9.26</td>
<td>5.82</td>
<td>8.05</td>
<td>7.67</td>
</tr>
</tbody>
</table>

Table 4.6: Results for vertebrae L1-T9 given in mm. σ denotes standard deviation.

The modified method for T9 detection was not thoroughly tested due to time and data limitations. However, according to the few qualitative tests performed it is showing potential to improve the detection accuracy.
In this chapter, the proposed method and the results presented in chapter 4 are discussed. Further developments are also discussed.

5.1 Method evaluation

As can be seen in the results, the proposed method is highly robust for LF and RF detection. The proposed method lowers the error of the Morphon initialization by more than half and also reduces the standard deviation for both RF and LF. However, it does not perform very well for vertebra T9. The T9 detection problem was expected to be harder to solve due to the more complex structural surroundings. There are many factors which make the case of T9 more complicated than the femur heads. The main problem with T9 is probably the repeating structural appearance in the spine. The T9 vertebra is almost identical with adjacent vertebrae T8 and T10. The design of the search space and the initialization from the Morphon registration is thus much more crucial than in the case of the femur heads. As can be seen in table 4.3, unfortunately the Euclidean error from the Morphon initialization is significantly larger than for the femur heads. Also, the standard deviation is larger. The reason for this is probably that the Morphon deforms prototypes that are very different from the target such that the T9 position gets incorrectly positioned on an adjacent vertebra. Since a local similarity measure is used to determine the most successful registration, an incorrectly deformed prototype might still get a high NCC score. This results in a large deviation from the correct position in the z-direction. Supposedly, a large initialization error is mostly problematic in the z-direction since the structure is repeating in that orientation. This problem is hard to overcome since a local
similarity measure still is wanted.

The combination of using registration and random regression forest for landmark detection is interesting since they are two very different methods. Registration is a heavily time consuming process and thus, the choice of combining the two might seem a bit ad hoc. The main reason behind this decision is that the Morphon registration is the most vital component in AMRA's product pipeline. For every target, displacement fields are always acquired in the Morphon registration step. It would be wasteful not to use these intermediate results to limit the search space and reduce computational complexity. The choice of using 29 prototypes for the Morphon registration might be a bit excessive for LF and RF detection. The motivation for the large number was mainly that those 29 prototypes are used by AMRA in their standard registration protocol.

In tables 4.1, 4.2 and 4.3, it is clear that vector regression is performing better than the scalar regression for all three POI. This is not very surprising since every voxel in the limited search space contributes to the final POI prediction in the multivariate case. When the local appearance is non-linearly mapped to a scalar, indicating the Euclidean distance to the target POI, less information from the output regression is used to predict the POI position. Only the position with lowest estimated distance is regarded and chosen. This introduces instability to the system since an incorrect prediction of a single voxel greatly can affect the final result. Another disadvantage with the scalar regression is that the POI has to be within the search space in order to be correctly positioned. Inaccurate Morphon initializations may therefore affect the result negatively. A larger search space could solve this problem to some extent but would also increase computational complexity and run times. By instead predicting a distance vector for every voxel in the search space, a few wrongly predicted vectors will not affect the final result. Vector regression is also capable of predicting POI positions outside of the search space and is thus not depending on the Morphon initialization as much. An improvement to the scalar regression could reasonably be achieved by applying a regularization. Since the distance should increase radially from the predicted POI, a simple regularization could be used to enforce the distance map to have approximately a spherical appearance. This was however not tested since the results from using the vector regression were sufficiently good.

No significant distinction between the multivariate results can be seen for any of the evaluated POI. It does not seem to matter whether Haar features are extracted from water, fat or gradient data. The water signal was thought to be more robust since the distribution of fat tissue in the body is more varying from individual to individual than the placement of muscles. However, since the search space is limited to small regions containing mostly bone for the evaluated POI, the fact that no significant difference between using the fat or water signal can be seen in the results is not surprising. Both the water and fat signal seem to provide sufficient contrast between bone and surrounding tissue in order to capture the non-linear relationship between local voxel appearance and distance to the target POI. The reason behind applying Haar feature extraction from gradient data was
to remove possibly unnecessary information to get more distinct features. This does not seem to have affected the results significantly in comparison to performing Haar extraction directly from the water and fat data. However, the standard deviation is higher for evaluation on T9. This could be due to the more complex structures in the spine generating confusing and noisy gradients.

The results are clearly better for RF and LF than for T9, even when T9 results from initialization outliers are removed. This result can not only be explained by the repeating structure in the spine. The most probable explanation is that the appearance of the top of T9 is not as well defined as the top of the femur heads. The surrounding area is much less characteristic and not as distinct. As a consequence of this, the non-linear mapping is harder to generalize and the ground truth POI for T9 may have larger errors than for the left and right femur. For example, in figures 4.6 - 4.9 the regression based prediction seems to be more accurate than the manually positioned ground truth. The ground truth is clearly off in the z-direction according to the definition in section 1.1. If faulty POI ground truths are present for the femurs, faulty T9 POI ground truths are guaranteed to exist in the data as well. This is not a problem if only a few data sets have inaccurate POI ground truths. Many faulty data sets will however introduce a great amount of noise to the system. This will obviously affect the result negatively. The ground truth POI positions in the data sets used in training and validation were not controlled to be correct. Even though the ground truths were placed by experienced personnel, human errors are prone to occur. This is especially true for T9 in combination with low resolution data since incorrectly placing the ground truth on an adjacent vertebra is easily done even for a trained eye. The data used for training and evaluation has quite low resolution, $3 \times 2 \times 2$ mm per voxel. Data with higher resolution would reasonably give better performance but also increase computational complexity since more voxels are contained in the same search space. If time resources were less limited, an extensive review of the ground truths in the training and test data would have been favorable to rule out human errors.

There are many parameter settings to consider when designing the system. This include forest parameters (number of trees, minimum splitting constraint, maximum depth, feature subset size), filter parameters (Haar patch size, number of filters, filters to keep after pre-selection) and data parameters (search space size, number of data sets used for training and pre-selection). No extensive evaluation was performed in order to find the optimal values for these parameters due to time limitations. Optimizing these parameters is a lot of work and therefore parameter choices were mostly based on smaller test results and findings in previous work on similar topics. This is not fully optimal since the results probably would have been slightly better with more extensive parameter tuning.

The modified method for T9 detection was implemented at the end of the project due to the observed problems with the Morphon initialization. Time was limited and therefore no extensive evaluation or optimization was done. Since a random forest with a separate filter bank has to be trained for each vertebra, the process
of optimization becomes very circumstantial. This is obviously not optimal but was still tested since it is a straightforward approach that could be implemented in a short period of time. The choice of adding the mean vector between every vertebra to initialize the center of the search space might seem a bit sloppy. This solution was motivated by the fact that the standard deviation of the vector distances were low. However, only 30 data sets were used to calculate the mean vectors. Reasonably, a very tall person has a longer spine and bigger gaps in between the vertebrae than a shorter person. If this would become a problem, scaling the means with patient height could give more accurate initializations.

5.2 Further work

In order to get the T9 detection more robust, further development is needed. The modified method has shown to have potential for detection improvement but has to be evaluated more extensively in order to draw further conclusions about its performance. As has been established previously, the main problem with T9 seems to be the initialization of the limited search space using the Morphon. The reason for performing the search space limitation is obviously that features from many training subjects are needed in order for the system to be able to generalize and detect POI in previously unseen data. Since a data set consists of millions of voxels, feature extraction and training on the whole full body is not practically viable. An alternative idea could be to incorporate a scale space scheme in order to achieve the same result. By starting at a coarsely downsampled version of the original data and iteratively training a forest for each scale refinement towards the original, the position of the target landmark could be incrementally refined. Further development of either the modified method for T9 detection or implementing the scale space learner in combination with more extensive parameter tuning optimization have potential to improve on T9 detection.
The proposed method shows great robustness and high performance for detection of left and right femur, a clear improvement of the Morphon prediction is achieved. Robust detection of T9 was observed to be much more problematic due to many different factors. The main reason for the unsuccessful results regarding T9 is due to insufficient precision of the search space initialization in the z-direction. In order to avoid this problem, an alternative to using the Morphon image registration for search space initialization is necessary. The detection of T9 using the proposed method was observed to improve on the Morphon prediction when results from initialization outliers were removed. After outlier removal, the results for T9 were still not as robust as for detection of right and left femur. The conclusion of this is that to achieve better performance regarding T9 detection, more extensive parameter optimization seems to be needed. Multivariate vector regression was observed to outperform scalar regression for all three POI. No significant difference in the results for LF and RF can be distinguished regarding whether Haar feature extraction is performed on water, fat or gradient data. For T9, the performance when using the gradient data was observed to be slightly less robust in comparison to performing feature extraction on the raw fat or water signal. The modified method shows promises of accurate and repeatable results but has to be evaluated more extensively in order to draw further conclusions.


