MASTER THESIS

PROSTATE TRACKING USING 3D ACTIVE APPEARANCE MODELS WITH 2D FITTING:

FEASIBILITY TESTING WITH DEFORMABLE PROSTATE PHANTOM

Peng JIN

ELECTRICAL ENGINEERING, MATHEMATICS AND COMPUTER SCIENCE
SIGNALS AND SYSTEMS

EXAMINATION COMMITTEE
Dr. ir. F. van der Heijden
Prof. dr. ir. C. H. Slump
Dr. J. J. Fütterer
Dr. S. Misra.

DOCUMENT NUMBER
EWI/SAS – 2013-057

UNIVERSITY OF TWENTE.

13/09/2013

Abstract

Motivation: The prostate biopsy and needle intervention surgery are the important procedure of prostate cancer diagnosis and therapy. Different from the conventional manually needle intervention in prostate surgery, MIRIAM project aims at minimally invasive robotics in MR environment to give an efficient and accurate clinical solution. Accordingly, a fully automatic prostate segmentation and 3D reconstruction in real-time intervention procedure is required for positioning and visualizing the prostate to get a correct needle path plan and be able to track this.

Problems: The lack of knowledge of prostate position and deformation in 3D space during needle intervention hinders the active appearance models (AAM) from being trained. Besides, the conventional 3D model fitting is not applicable in a real time needle intervention since the 3D image acquisition is time-consuming. As a result, the feasibility of using a phantom to build the deformed prostate data set and a 2D slices based model search will be examined.

Approach: A deformable prostate phantom was made for MRI guided needle intervention simulation and data acquisition. The phantom was designed anthropomorphically and made of soft PVC. Full 3D MR images of the phantom were collected for training AAM. Before building AAM, a group-wise parameterization based shape correspondence was establishment. A 2D slices based AAM search approach was proposed to get a 3D shape segmentation and texture reconstruction. After the AAM was established and trained, the leave-all-in and leave-one-out schemes were applied for validation. Some metrics were also calculated for results evaluation.

Results: The phantom showed a good bio-mechanical property and MR compatibility, where the deformations were clearly observed. For the segmentation results, leave-all-in scheme validate the feasibility of 2D slices based AAM search, but the leave-one-out scheme showed worse results which lost the deformations for both 3D and 2D fitting. This could result from the lack of training data. The further research and verifications would be taken with larger data set and more considerations should be taken on generalization from the phantom to the real case.

Acknowledgement

I was introduced to this Master project in the spring of 2013, and it appears to be a challenging task. The result would never have been the same without the support, encouragement and assistance from the following people.

Ferdi van der Heijden, my first supervisor in my assignment, is always there for giving a hand. It is he who brings me to the image processing field and always inspires me with new ideas . I am very grateful with his supervision and assistance, which made me concentrate on the assignment and finish it. I also appreciate Jurgen Fütterer and Martin van Amerongen, who gave me so many helps in medical knowledges and MR scan experiments as well as advices in making the phantom. Without these helps, it's impossible to carry out the experiments.

I would like to give my special gratitude to Henny, the technician in SAS group. It is he who helped me with the phantom fabrication, material purchase, molds manufacture. I also thank Geert Jan for the technical support. Their assistances have contributed a lot to my progress.

Thank my friends Eirk, Gert, Meiru and Rita for their help and supports, and thank Bianca for explaining her previous work. Thank Sandra and Kees for her help with my study issues.

Besides, I would like to thank Lu, Xiaochen, Xichen, Yin, Yen and other friends for the support in daily life. Finally I want to thank my parents, who support my study and care me so much from thousands miles away.

This master assignment is derived from the MIRIAM project. It was carried out at the chair of Signals and Systems, University of Twente (Enschede, the Netherlands), and cooperated with Radiology department of University Medical Center Nijmegen St. Radboud (Nijmegen, the Netherlands).

Contents

Abbreviations & Acronyms

AAM Active Appearance Models

ACM Active Contour Models

AD Absolute Distance

ASM Active Shape Model

BEAT_IRTTT Interactive, Real-time, Multi-slice TrueFISP Squences

BPH Benign Prostatic Hyperplasia

CCA Canonical Correlation Analysis

CIRS Computerized Imaging Reference Systems, Inc.

CPS Clamped-Plate Spline

CT Computerized Tomography

DSC Dice Similarity Coefficent

HFS Head First Supine

ICA Independent Components Analysis

ICP Iterative Closest Points

IFE Interactive Front End

LSE Least Square Estimate

MAD Mean Absolute Distance

MDL Minimum Description Length

MICCAI Medical Image Computering and Computer Assisted Intervention

MIRA Institute for Biomedical Technology and Technical Medicine

MIRIAM Minimally Invasive Robotics In An MR environment

MLE Maximum Likelihood Estimate

MLR Multivariate Linear Regression

MR Magnetic Resonance

MRI Magnetic Resonance Imaging

PASM Probabilistic Active Shape Models

PCA Principal Components Analysis

PCa Prostate Cancer

PDM Point Distribution Models

PSA Prostate-Specific Antigen

PVA Polyvinyl Acetate

PVC Polyvinyl Chloride

RMS Root Mean Square

SAS Chair of Signals and Systems in University of Twente

SD Standard Deviation

SPHARM Spherical Harmonics

SSM Statistical Shape Models

SVD Singular Value Decomposition

TPS Thin-Plate Spline

TRUS Trans-rectal Ultrasound

UMC University Medical Center/Universiteit Medisch Centrum

UT University of Twente/Universiteit Twente

Chapter 1

Introduction

Prostatic diseases such as prostate cancer (PCa) and benign prostatic hyperplasia (BPH) are common among men, especially in Europe and America. An early stage diagnosis can help to make an appropriate therapy plan. Hereby, it is common to inspect whether the prostate is cancerous or benignly enlarged by biopsies. In the needle intervention, magnetic resonance (MR) imaging and ultrasound are necessary for locating the prostate and needle. Nevertheless, some studies have shown a high false negative rate in trans-rectal ultrasound (TRUS) guided biopsies[1], which missed up to 31% of prostate cancer because some zones of the prostate cannot be detected by using ultrasound. Alternatively, the MRI gives more attractions in prostate biopsy and local therapy since its hight soft-tissue contrast and multi-parametric capabilities.

Conventionally, the biopsies are taken manually and all the imaging methods (e.g. MR and ultrasound) are helping the doctor to locate the needle and the prostate. Presently, the robot-assisted biopsy device is favorable to increase the target accuracy. Minimally Invasive Robotics In An MR environment, MIRIAM project is concentrating on this, which, consequently, requires an automatic prostate segmentation and image reconstruction during the needle intervention. In this chapter, we will briefly introduce the project and some relative knowledge, and review the common used approaches for the prostate segmentation. Based on these, we will put forward the research questions and objectives. In the end, we will give the framework of the whole article.

1.1 Project Overview

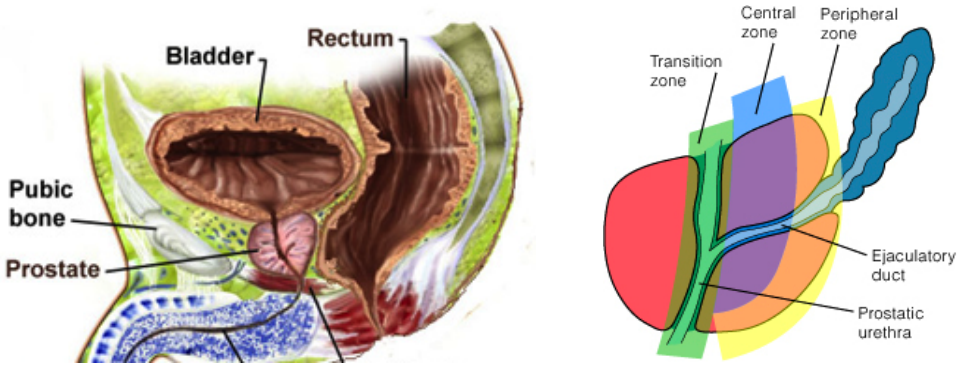
MIRIAM project is aiming at developing new solutions for minimally invasive surgical procedures in an MR environment. This project is collaborated by the University of Twente (UT) MIRA, the University Medical Center (UMC) St. Radboud Nijmegen Radiology department, Demcon, Siemens and XiVent Medical. Finally a MRI compatible device can lead to better diagnosis of prostate cancer, which will decrease the overtreatment of patients due to the limitations associated with prostate-specific antigen (PSA) test while reducing the incidence of false-negative

biopsies [2]. Meanwhile, it will lead to a better treatment decision and focal therapies with fewer side effects.

The major task of the UT-SAS group is to develop a detection system of the needle in the MR images for adjustment of the scanning planes, a prediction of prostate deformation and position, and 3D prostate reconstruction. Because the needle path is planned in a 3D space, it requires a 3D prostate representation and a 3D tracking during the insertion, since the prostate might deform and move during intervention. However, taking a 3D volume image is time consuming. It is not applicable to stop the biopsy for image acquisition. However three orthogonal slices can be taken in real time for the radiologist to see where the needle and prostate are. This inspires us to segment prostate only relying on these 2D MR image slices. Before we can figure out how to accomplish it, we would like to gain some anatomy and MRI knowledge and review the commonly used approaches for segmentation of the prostate, especially the MR modality.

1.2 Prostate Anatomy

The prostate is a relatively small organ in the pelvis and is part of the male reproductive system [3]. It is located under the bladder and against the rectum. Pubis is in front of it and the urethra goes through. The anatomic annotation of the male pelvis part is shown in Figure 1.1(a)[1] .



(a) Male pelvis anatomical illustration (b) Prostate schematic drawing

Figure 1.1: Prostate anatomy illustration

From this geometry, the prostate apex refers to the caudal part, and the prostate base refers to the head part. The prostate can also be divided into three zones in terms of functions: Transition zone, central zone and peripheral zone, as Figure 1.1(b)[2] shows. The central zone surrounds the ejaculatory ducts, and the transition zone is located around the urethra, where the BPH happens most probably; the peripheral zone is wrapping at the posterolateral part and comprises the majority of

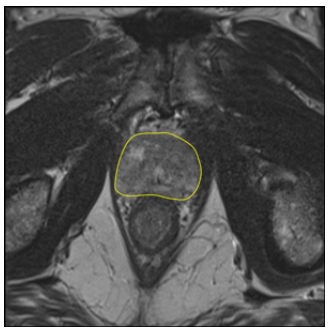
[1]Source: http://www.yoursurgery.com/ProcedureDetails.cfm?BR=7&Proc=55
[2]Source: http://www.pathologyoutlines.com/topic/prostateanatomy.html

prostatic glandular tissue [3]. Some study reveals that 65% to 74% of the prostate tumors are located in the peripheral zone [4], which would be firm, hard or stony compared with the health tissue.

1.3 MR Imaging Principle

An MRI scanner has large homogenous magnets, which generate the magnetic filed that tends to line up all the magnetic moments of the nuclei, the spin of the nuclei and the hydrogen in the human body where water is concentrated. When a radio frequency pulse is put on, the magnetization of the hydrogen nuclei can be turned 90 degrees away from the direction of the magnetic field. Then a radio frequency signal, as the protons return to original equilibrium will be induced in the coils around. For determining where the signal comes from, another magnetic field gradient is put on the top of the homogeneous one, with different strength and different directions, causing nuclei at different locations to precess at different speeds. All the data are recorded and the measured signals are Fourier transformed to recover the spatial information.

There are some basic MRI scan types, such as T1-weighted, T2-weighted, T2* etc. For diagnosis of PCa, T2-weighted image has been used for many years [3]. A T2-weighted MRI scan mainly shows the tissue difference in terms of transverse relaxation time. As the PCa tissue may appear a similarity with BPH in the MR scan, it is necessary to take biopsies to increase the diagnosis accuracy. Figure 1.2(a)[1] shows an instance of T2-weighted image slice of male pelvis, and the prostate is enclosed with yellow line. Figure 1.2(b)[2] shows a sagittal real time slice with manually annotations in a MRI-guided cryotherapy.



(a) An example of MR image slice of male pelvis.

(b) An annotation of sagittal MR image in the cryotherapy. (1. Prostate; 2. Bladder; 3. Pubis; 4. Rectum; 5. Needle)

Figure 1.2: MR image example of prostate

[1]Source: http://promise12.grand-challenge.org/
[2]Source: UMC St. Radboud Nijmegen

Nowadays MRI can scan a full 3D image, making the 3D prostate detection easier. By viewing the axial, sagittal, coronal or with any other orientation of the image, it can show the prostate completely so that the biopsy or treatment can be planned precisely.

1.4 Prostate Segmentation Methods

There have been a lot of prostate segmentation researches in TRUS and MR images. The methods can be classified into different strategies: contour and shape based, region based, classification based approach, and hybrid method [5]. Here we give a short review of these approaches.

1.4.1 Contour and Shape Based Approach

The contour and shape based segmentation is the most commonly used method. It exploits contour features and shape information to segment the prostate. As for the edge based approach, Liu *et al.* [6] developed a new method for automatic prostate boundary detection in ultrasound images. Kwoh *et al.* [7] implemented a harmonics method using Fouriere information for noise removal and boundary encoding. Aarnink *et al.* [8] used the local standard deviation to identify homogenous and heterogeneous regions in a multi-resolution image framework, and Pathak *et al.* [9] reduced the speckle by applying a stick filter based on the non-zero correlation value of the speckle over large distances. There are of course many other articles addressed in this approach. An advantage of the edge detection based method is that it can neglect the prior knowledge of shapes. It fits our task because we do not have the knowledge of prostate deformation in the biopsy. Even though, the edge detection based approach is still not applicable, since we can only get the edges in the slices extract for segmentation but it is not possible to use several slices to reconstruct a 3D segmentation without any priori knowledge.

A similar problem also occurs in the method based on probabilistic filtering, though it has been applied in image segmentation successfully. This method features a moving object tracking constrained by a dynamic model, such as the work of Abolmaesumi *et al.* [10], which aims at projecting equi-spaced radii from an arbitrary seed point inside the prostate cavity towards its boundary. However, these methods suffer from a sensitivity to initialization and dependence of image completeness which is not the case in our task.

Most researches are concentrating on the deformable model based segmentation. In the early stage, Kass *et al.* proposed active contour models (ACM), which also called snakes algorithm. It controls continuous closed curves composed by discrete points, and evaluates an energy function for matching. The ACM method has been applied in many fields. As to segment the prostate, Knoll *et al.* [11] proposed the technique for elastic deformation restriction to particular object shapes of any closed planar curve using localized multi-scale contour parameterization based on

the 1D dyadic wavelet transform. Ladak *et al.* [12] proposed a semi-automatic segmentation of the prostate with the initialization by selecting four points from which the outline of the protate can be estimated using cubic interpolation and shape information. The estimated contour is then deformed automatically to better fit the image. In contrast, Ding *et al.* [13] developed a 3D segmentation method. They used a cardinal spline to construct the initial contour and then the internal and external forces were calculated. They propagated the final contour slices by slices until all boundaries are segmented then a 3D mesh can be formed for calculating the volume of prostate. Jendoubi *et al.* [14] experimented with an improved deformable 2D snakes modeling technique with median filtering for speckle noises removal and gradient vector flow computed from the gradient map obtained using Sobel and Laplacian of Gaussian as external force to drive the active contour towards the boundary of the prostate. Most of these applications were in TRUS images segmentation. As an evolution of ACM, statistical shape models (SSM) were soon proposed. The research then was concentrated on two typical SSM methods, active shape models (ASM) and active appearance models (AAM).

ASM, proposed by Cootes *et al.* [16], worked in the deformable framework maintaining the principal modes of shape variation of the anatomical structures. By assuming the input data have Gaussian distributions, ASM applies principal components analysis (PCA) of the aligned shapes to find the principal shape variances and point distribution models (PDM) to construct the linear relationship between the weights and shape vectors. For the model searching, gray level information, normal to the contour point, is acquired offline. Then by minimizing the difference between the detected gray levels and the modeled ones, the object shape can be obtained by updating the pose and weight parameters iteratively [17]. Cootes *et al.* also applied it to many applications [15], including segmenting MR slices [18]. Zhu *et al.* [19] proposed a hybrid ASM approach for prostate segmentation. This approach was investigated when the volumetric data, namely the prostate slices are sparse, so that a 3D ASM tends to be less robust, where a 2D plus 3D methodology can be employed. This application resembles ours but differs. Although we both aim to use 2D information for a 3D construction, their approach is suitable for the training set and testing set are equivalently sparse. And this method is actually still a 2D ASM. However, we can get a full 3D data for training but less data for testing, compared with Zhu's application. Kirschner *et al.* [21] has proposed a automatic prostate segmentation approach in MR images with a probabilistic ASM (PASM). They employed a boosted prostate detector to locate the prostate in the image, followed by a PASM for the delineation of its contour. The full 3D ASM is applicable to our condition, but the texture information relies on the position of landmarks, which might not be included in the extracted slices.

AAM, as an advanced version of ASM, also proposed by Cootes *et al.* [20], incorporates the texture information as well as shape information to construct the linear relation between a combined weights and appearance (shape and texture). Because of more accurate performance, AAM replaced ASM gradually and it was

applied widely in medical image segmentation. In the 2012 international conference on Medical Image Computering and Computer Assisted Intervention (MICCAI) challenge, some participants have used AAM in their system, such as the work of Vincent *et al.* [23] and Maan *et al.* [22]. A 3D AAM can help us build the model. In the model searching, not like ASM searching, there are always texture information available in the extracted slices. This feature gave us the chance to consider our problem, and we will discuss them in the latter chapters.

Another two techniques from the deformable model based method are using edge based level set and curve fitting. Although they have shown a good performance in TRUS, they were rarely implemented with MR images.

1.4.2 Region Based Approach

In addition to the contour and shape based segmentation, the region based one is mainly exploiting the atlas or a graph partition or a region based level sets. An atlas approach converts the segmentation problem to a image registration problem since the registration can help to find a transformation to map a pre-segmented image to a new image. Klein *et al.* [24] presented a three dimensional multi-atlas approach to segment the prostate, measuring a localized version of mutual information, and considering a majority voting rule and the "simultaneous truth and performance level estimation" algorithm for atlas fusion then producing the final segmentation decision. Dowling *et al.* [25] improved the Klein's result [24], and introduced the diffeomorphic demons method for non-rigid registration and a comparison of alternate metrics for atlas selection. Many other relevant researches on graphic partition and region based level sets haven been investigated, and most of them were considered in a TRUS environment. Nevertheless, this method is not suitable since the image for segmentation only contains several 2D slides, while this approach requires the full 3D information. However, it provides a good choice for building a prediction model when it can be incorporated with probabilistic methods.

1.4.3 Classification Based Approach

The classification based approach is aiming at obtaining a partition of the feature space into a set of labels for different regions[5], where a classifier or clustering based method can make the system learn the features from labeled the training set, then label the testing data set. Maan *et al.* [26] has shown a new prostate segmentation approach using multispectral MRI and a statistical pattern classifier. By examining the Bayesian-quadratic classifier and assuming a normal density likelihood, and a k-nearest-neighbor classifier, they segmented the prostate precisely with the knowledge from multispectral MR images. Other similar methods have been proposed for TRUS images. However, in the feature extraction part, the problems from edge detection part also exists in this approach. As for our task, it cannot be well deployed in 3D condition.

1.4.4 Hybrid Approach

The hybrid approach may be any combination of the methods listed in the preceding parts, which aims at improving the accuracy. Martin *et al.* [27] applied a hybrid registration framework which coupled an intensity-based registration with a robust point-matching algorithm for both atlas building and atlas registration. Tsai *et al.* [28] proposed a shape and region based level set framework to segment the 3D prostate MR images. They made the PCA coefficients of aligned shapes incorporated in the level set function, and segment the prostate by minimizing this function. Firjani *et al.* [30] proposed a new segmentation framework based on maximum a posteriori estimation of appearance descriptors, spatial rotation variant and shape descriptor. Toth *et al.* [29] used mean, standard deviation, range, skewness and intensity values and its kurtosis in a local neighborhood to propagate ASM automatically initialized from MR spectroscopy information. MR spectroscopy information was clustered using replicated k-means clustering to identify the prostate in a mid slice to initialize multi feature ASM [5]. There are more studies concerned with different hybrid approaches, which can be referred to the study of Ghose *et al.* [5] for a detailed summary.

1.5 Research Questions and Objectives

As we discussed in the previous sections, we wish to track the prostate during the needle insertion in order to update the needle path to target the correct position of prostate. Briefly stated, the task is to segment the prostate and reconstruct the 3D representation in the real time intervention. However, two major problems arise. First, due to a lack of 3D image data of prostate in needle intervention, we do not know how the prostate will deform or move during the intervention; second, only three orthogonal slices could be available in the intervention since the full 3D image acquisition is time consuming. Accordingly, we can put forward our research questions:

- Is it feasible to learn the prostate deformation and motion during needle intervention in 3D space?
- Is it feasible to segment the prostate based on three orthogonal slices in a 3D MR environment?

Currently, we cannot get the knowledge by taking 3D images of patients in the intervention, as the needle intervention is fast and it will not stay in the body for long. The acquisition of full 3D MR images will take 2 to 4 minutes depending on the resolution. Moreover, it is impossible to repeat the procedure in the patient to collect a statistical data set. As a result, we resort to a anthropomorphic phantom. The phantom should be representative, namely have resemblance with a real prostate in terms of mechanical properties, and also MR compatible.

As to the second question, according to the review of the prostate segmentation methods, we turn to a full 3D AAM relying on the deformable prostate

phantom, and then test the feasibility of model search based on three orthogonal 2D slices. Therefore, our objectives are:

- Make an anthropomorphic deformable prostate phantom to simulate the prostate motion in needle intervention and collect MR image data with respect to prostate deformations.
- Build active appearance models using the collected data and segment the prostate by means of three orthogonal 2D slices based model search.

1.6 Thesis Outline

According to the research questions and objectives, this thesis is organized as follows:

- Chapter 2 introduces our innovative design of prostate phantom based on the prior works. It includes material selection, structural design and other relevant issues.
- Chapter 3 reviews the standard principles of basic AAM including the shape, texture and appearance modeling specifications and model search procedure.
- Chapter 4 discusses the correspondence problem and addresses our choice of correspondence establishment method, namely parameterization based group-wise minimum description length approach.
- Chapter 5 proposes our innovative 2D slices based AAM search, including the mathematical principles and algorithm procedures.
- Chapter 6 specifies the experimental setup and implementation issues including the phantom fabrication details, data collection experiments, AAM modeling procedures, and shape segmentation and texture reconstruction results.
- Chapter 7 evaluates the phantom properties and segmentation results in quantitive metrics and 3D visual representations.
- Chapter 8 summarizes the whole work and gives the general remarks and recommendations for future work.

Chapter 2

Prostate Phantom

According to the previous chapter, an effective anthropomorphic phantom is required to mimic the needle intervention for visualization of 3D deformations. There are plenty of phantoms built in different research orientations. It is necessary to learn from the retrospective literatures that can provide valuable suggestions and essential advices. In the following section, a brief literature review for the prior arts is firstly presented. Afterwards, the detailed annotation and description of our phantom will be proposed.

2.1 Prior Arts

The previous phantoms are varying from simple single component phantom to complex multiple components that imitate the prostate and surrounding tissues. For the sake of more accurate simulation of needle insertion, a complex phantom with multiple components will be studied more.

Ho *et al.* [31] investigated the prostate intervention device. They evaluated the software's safety, and workflow protocol by modeling a phantom that was made of egg. Specifically, they suspended a shelled hard boiled chicken egg into transparent gelatin, which were placed in a plastic box. This prostate phantom was tested in the biopsy gun insertion guided by ultrasound. For an economic sake, Wilkin *et al.* [32] used cooked beets and corned beef or gelatin mixture to make a cheap prostate phantom for the application in measuring prostate volume and taking prostate biopsies guided by ultrasound. Heikkilä *et al.* [33] has proposed a technique for needle insertion in prostate seed implantation, where a corresponding phantom has been made of bolus material building the perineal surface, a plastic tube for rectum, and gelatin for the prostate which is fixed to the case. However, the mechanical properties and visibility in MRI has not been tested for these ultrasound oriented phantoms. What is more, the motion of the prostate in the insertion of biopsy needle/gun was not taken into consideration as well, which are necessary for building the phantom in our experiment.

Similar problems have also happened in some commercial prostate phantom prod-

ucts fabricated by CIRS (Computerized Imaging Reference Systems, Inc.). The most popular prostate phantoms are the 053, 058 and 066 series[1]. These phantoms have modeled the prostate, the rectum, the perineum wall, the urethra and the seminal vesicles, most of which are made of Zerdine®. This is a patented solid elastic material that has good elastic property, is not sensitive to temperature variation and has good ultrasonic, MRI and CT performance. Some phantoms have been employed in medical and imaging researches such as MRI and Ultrasound fusion [34][35] and are valid in performance. Although these commercialized phantoms are aimed at modeling the application of biopsy, sonoelastography, cryosurgery, brachytherapy and other general purposes, they are neither designed for MRI modality (except 053-MM [2]), nor for simulating the realistic prostate motions. In addition, these phantoms are expensive (e.g. $480 for CIRS 053-MM model, $602.67 for CIRS 058 model. etc[3]), where price is also an important factor. Another product from CIRS is the pelvic phantom, CIRS 048 [4], which is a more complete version of prostate phantom. It adds pubic bone/pelvis that is made of tissue equivalent epoxy resin, and bladder made of Zerdine®, compared with their prostate phantom, but loses the perineum part. This phantom shows good performance in MRI, CT and ultrasound, while it has difficulty in trans-perineal biopsy intervention because of the absence of perineum. In addition, the price is over $3300 [5], which makes it unsuitable for destructive experiments.

Zerdine® would be a good choice of material for the prostate phantom, but it cannot be purchased separately and very expensive. Besides, it does not have evident deformation during the needle insertion. Dehghan *et al.*[36] have selected polyvinyl chloride (PVC) plasticizer to make the prostate model, and also the surrounding tissues with a mixture with softener. This phantom also contains a rectum into which a stiff cylinder made of hard plastic was inserted to simulate the rectal probe. Besides, the prostate was connected with the base. These designs are made for estimating the rotation and motion of prostate around the pubic bone and the effects from the rectal probe. The investigation of motion and deformation of prostate is prior to other studies. A well fitted model and method have been established to estimate and measure the interaction between the needle and tissue. It also proposed an appropriate adjustment for improve the simulator. It is reported that the displacement of the prostate in axial direction is less than 4 mm during the needle insertion. However, it was not indicated how much this phantom fits to the realistic condition and was not tested in MR environment.

Sherman *et al.* [37] has made a phantom that created the true prostate environment for experimenting an unconventional prostate stabilizing technique to decrease the motion of prostate during the brachytherapy procedure. The phantom

[1] http://www.cirsinc.com/products/all

[2] http://www.cirsinc.com/products/modality/64/multi-modality-prostate-training-phantom

[3] http://www.pnwx.com/Accessories/Phantoms/Radiology/Ultrasound/CIRS/Prostate/

[4] http://www.cirsinc.com/products/modality/62/multi-modality-pelvic-phantom/?details=specs

[5] http://www.supertech-to-go.com/mupeph.html

consisted of three distinct regions, the perineum, the prostate and the surrounding fascia tissue, made of different concentrations of PVC. It is important to indicate that the proportion of hardener and plasticizer of these layers were based on prior experimentation with PVC, which attempted to obtain the force and displacement seen in the operation room table, namely *in vivo* measurement, during brachytherapy procedures [38]. Moreover, in order to determine the extent of the prostate displacement accurately using ultrasound scanner, vitamin E capsules were placed along the outer region of the prostate. The phantom was used in a vertical orientation with two orthogonal ultrasound probes for visualization. The experiment indicated that the motion is up to 11 mm during the needle insertion. Unfortunately, this phantom has not been tested in MRI either.

On the basis of the preceding two mobile phantoms, Hungr *et al.* [39] has built a more realistic and deformable prostate phantom. This phantom consists of a harder prostate embedded in a softer surrounding, both of which are made of PVC with different mixture proportions. Besides, a rectum structure was constructed as an air tube for the placement of ultrasound probe and the perineum is part of the frame which has the same material as the prostate. With the phantom, the speed of sound, stress-strain relationship, the imaging characteristics in CT, MRI, ultrasound, needle insertion force and prostate motion have been examined respectively. It is worth mentioning that most of the experimental data were comparable with the results tested with the tissue *in vivo* or *in vitro*. For instance, the stress-strain relationships for PVC with different proportions were published with the corresponding *in vitro* prostate tissue, and the needle axial force measured with the phantom was comparable to the *in vitro* liver puncture curve by Maurin *et al.* [40], the dog *in vivo* prostate test by Kataoka *et al.* [41], as well as the *in vivo* force measurements by Podder *et al.* [38]. Apart from that, the experiment revealed that the movement of the phantom prostate along the needle insertion was between 2 and 4 mm, with rotations between 0° and 2°, which were comparable with the range of $3 - 10$ mm of translation and 0°-10° of rotation published *in vivo* [42]. But the rotation of the prostate was tended to be downward (toward the rectum) rather than upward toward (the pubic arch). This phantom provides sufficient technical instructions and supports since it satisfies most requirements concern with our research compared with other phantoms.

2.2 Phantom Specifications

The phantom in our research concerns with the motion and deformation of prostate in the needle insertion. As stated above, the phantom made by Hungr *et al.* [39] was selected as our phantom prototype. Our phantom also includes a frame, a rectum, a prostate, surroundings. But we added a pubic bone and replaced the homogeneous structure of the prostate with a two-layer structure (capsule plus inner core). In this section, the phantom details in terms of material, structure and components, as well as some relevant issues will be described.

2.2.1 Phantom Material

The choice of material is very important, for it determines most mechanical properties of the phantom. Three major issues are concerned, namely the kind of material, the appropriate concentration for these materials, and the material costs.

2.2.1.1 Material Type

Based on the preceding section, most commonly used materials for the phantom include agar, gelatin, PVA, PVC, silicone, Zerdine®, etc. In the design of Hungr *et al.* [39], a detailed comparison of these materials has been listed in terms of density, Young's modulus etc., which has shown that agar and gelatin are too fragile for motion particularly in condition of soft surrounding materials; Zerdine®, as discussed, is neither economic nor suitable for evident deformation; while PVA is too complicated to prepare so that it increases the difficulty in fabrication, as well as it has inappropriate acoustic property if the phantom would be used in ultrasound environment. Accordingly, considering the trade-off between mechanical and imaging characteristics, as suggested by Hungr *et al.* [39], soft PVC is the best choice at present.

The soft PVC used in our phantom is a mixture of PVC polymer solution and the softener diethyl hexyl adipate. Both the polymer solution and softener were ordered from the company specialized in fishing bait construction, Bricoleurre (Mont Saint Aignan, France), same as the phantom of Hungr *et al.* [39] did. Table 2.1 shows the ingredient proportion of 4 premixed polymer-softener mixture products called Plastileurre®. Each product can be either ordered from Bricoleurre company directly, or generated by mixing hardener (i.e. Super Rigide) and softener (i.e. d'assouplissant) with the specific ratio. This proprietary recipe can also be found on their official website [1].

Table 2.1: Ingredients proportion of soft PVC mixture products

Plastileurre® Products	**Hardener(%)**	**Softener(%)**
Super rigide (Hardener)	100	0
Rigide (Hard)	75	25
Standard	50	50
Soft	25	75

2.2.1.2 Material Concentrations

As for the concentration of the mixture, its elasticity property should be comparable to the human tissue. Generally, the Young's modulus is an important factor for evaluating the elasticity of any tissue. Normally it can be obtained by measuring the stress-strain relationship, and the higher the value, the stiffer the material.

[1] http://www.bricoleurre.com/index.php?main_page=index&cPath=1

Krouskop *et al.* [43] tested the elastic moduli in normal and abnormal (PCa and BPH prostate tissues *in vitro*) tissues. For the normal tissue the moduli were at 63 ± 18 kPa (anterior) and 70 ± 14 kPa (posterior); for BPH tissue, they were 36 ± 11 kPa; but for cancerous tissue they were 221 ± 32 kPa which showed the cancerous tissue had a measurable elevated modulus compared with that of the healthy tissue. As for the stiffness of same specimen, Kemper *et al.* [46] has investigated the stiffness of healthy human prostates with an *in vivo* magnetic resonance elastography (MRE), where the result showed the peripheral portion of the prostate was stiffer than the central portion. In the research of Phipps *et al.* [44], the Young's moduli of cancerous prostate glands from 9 specimens (4 Treated and 5 untreated) ranged from $68 - 168$ kPa. While Zhang *et al.* [45] did the unconstrained stress-relaxiation tests on 8 normal prostate specimens and 9 cancerous prostate specimens *in vitro*, and the measured Young's moduli were 15.9 ± 5.9 kPa and 40.4 ± 15.7 kPa respectively. Yang *et al.* [47] has used an electro-mechanical shaker to do a macro and micro-mechanical testing *in vitro*, and the Young's modulus was up to 200 kPa.

To make an appropriate phantom, the prostate capsule and the surrounding material should be distinctive. Apart from the 4 Plastileurre® products, Hungr *et al.* [39] defined the super soft mixture for making the surrounding tissue that on most condition is fat. As proposed, the super soft mixture is the blend of 10% hardener and 90% softener. Then, Hungr *et al.* [39] have measured the stress-strain relationships for these five mixtures with different polymer-softener concentrations, and the Young's moduli for super rigide, rigide, standard, soft and super soft were approximate 200, 150, 100, 50, 3 kPa respectively. Combining the measured Young's Moduli of prostate *in vitro* [48, 49, 43] and that of soft PVC mixtures, Hungr *et al.* suggested the frame and prostate in the phantom can be made of the standard mixture corresponding to a combination of 50% hardener and 50% softener, and the surrounding part can be filled with super soft mixture. It was fair enough for selecting standard mixture also to make the frame, since it can be stiff enough not only for supporting the whole phantom, but also for simulating the perineal skin layer and the anus area. Krouskop *et al.* [43] also tested the normal fat in the breast tissue, and the Young's moduli range from 11 to 30 kPa, whereas the modulus for super soft PVC mixture is only around 3 kPa. It seems to be too soft, but soft enough for the prostatic mobility during needle insertion and ultrasound probe placement.

Although Hungr *et al.* indicated Plastileurre® standard can be generated by mixing the hardener and softener with equal proportion, the outcome of the self mixture appears not to be the same as the premixed standard mixture as indicated by our experiments. In preliminary research (B. Maan, personal communication, September, 2012) , the testing results showed that the Young's modulus of premixed standard PVC mixture is around 86.5 kPa instead of 100 kPa that is the reference from the website, and the modulus of super rigide (hardener) is 208.8 kPa that is close to the reference value 200 kPa. While the Young's modulus of self mixed standard PVC based on the given proportion is approximate 23.5 kPa, which is

too small (too soft) for making the frame.

In summary, it can be concluded that, firstly our prostate capsule and phantom frame can be made from the same material, namely the standard mixture according to the elastic moduli of prostatic peripheral tissue and soft PVC mixture; secondly, using the premixed standard mixture product is recommended rather than using mixing hardener and softener manually. As for the prostate inner part, because the capsule is harder than the inner part, we choose a 62.5% premixed soft mixture with 37.5% softener to fill the inner part of prostate.

We would like to form a rectum part in the phantom because it can be feasible in trans-rectal biopsy, and also help to locate the prostate. Due to the stiffness of the frame, we build the rectum in the frame as an air tube instead of filling with any materials.

As for the surrounding material, which simulates the fat tissue surrounding the prostate, the super soft mixture proposed by Hungr. *et al.* appealed to be too soft compared with the real fat in terms of hand feelings and the Young's modulus. Therefore it is better to use a harder mixture. Originally, we advised to use 40% premixed soft mixture with 60% softener to make the super soft mixture proposed by Hungr *et al.*. Nevertheless, we find this mixture is too soft compared with fat. With several experiments we slightly change the recipe to 60% soft mixture with 40% softener.

The pubic bone is taken into consideration for the possibility of its influence on the prostate mobility during the needle insertion, because it is close to the prostate and also a fixed part. The material for making the pubis should be hard firstly, and also contains no metal components that can cause artifact in the MRI scanner. Initially the pubis was constructed as a simplified shaped block with dried clay bought from toy shop (Intertoys™). However, it caused a lot of air bubbles around the its surface and showed artifact in the MR images due to the material itself. Instead, we selected Fullcare® VeroWhite Plus™ or VeroClear™ material specific for Objet Eden series 3D printer. This material can not only stay stable when we pouring the surrounding material in a high temperature, but also show few artifact in the MRI scanner. Actually other materials such as perspex, delrin, PVC black, PVC gray, and teflon were also tested. They performed well in the MR images, but they have high melting points that makes shaping difficult. For obtaining an anthropomorphic pubis, it is preferred to print it directly as the printing material is suitable. To sum up, the materials used in our phantom and proportion of ingredients are listed in Table 2.2.

2.2.1.3 Material Price

The cost of one phantom does also matter. Because some components, such as all the apparatus, molds and pubis model are retainable, they are expensive though, we put them into one time cost. Only the soft PVC mixtures that cannot be reusable

Table 2.2: Phantom ingredients and proportion.

Components	**Ingredients**
Frame	Plastileurre® Standard mixture
Prostate capsule	Plastileurre® Standard mixture
Inner Prostate	62.5% Plastileurre® Soft mixture+37.5%Softener
Surroundings	60% Plastileurre® Soft mixture+40%Softener
Pubic bone	Fullcure® VeroWhite Plus™/VeroClear™
Rectum	Air

would be considered here as the variable cost. Unlike Zerdine®, the premixed soft PVC mixtures from Bricoleurre company is much cheaper. From the website, the price per liter can be easily found. Table 2.3 is a brief summary.

Table 2.3: Premixed soft PVC mixtures and softener price (Currency: Euro).

PVC Mixture	**Volume**			
Plastileurre® Product	**0.5L**	**1L**	**3L**	**5L**
Standard	–	19	53	84
Soft	–	19	53	84
Softener(d'assouplissant)	12	20	54	–

We have estimated the frame and prostate will take around 1 liter standard mixture and surrounding part will take approximate 1 liter soft mixture plus softener, therefore each phantom will roughly cost €40 for PVC mixtures, which makes the phantom more practical and economic.

2.2.2 Phantom Structure and Components

The phantom was designed based on the patients specimens to make it more anthropomorphic, including the shape of pubis and prostate, and the relative position of them. As said in the preceding section, the phantom has the following components: the prostate (capsule and inner tissue), the pubis, the rectum tube and the frame.

2.2.2.1 Prostate

The prostate model comprises a hard capsule and a soft inner part. The capsule is made of standard mixture and the inner part is made of a relative soft mixture. Hence, for casting the model, an outer matrix and an inner mold should be made beforehand, We first made two prostate molds of the same shape but with different sizes. The bigger one was for making the matrix to cast the outer surface of prostate capsule, and the smaller one is for saving the space for the inner prostate and casting the inner surface of capsule.

The prostate shape was a mean of 10 specimens shapes segmented from MR images [1]. The relevant data sets information is listed in Appendix Table A.1. After the construction of each prostate specimen, their surfaces were normalized to voxel spacing of $1 \times 1 \times 1\ mm^3$ and translated to the origin. Then each of them was smoothed by interpolation between slices. The interpolation was carried out by resampling the contours using Fourier descriptors, and normalizing the starting point. After resampling, the interpolated contours were generated using PCA by varying the first weight factor of eigenvector linearly. When all the prostate specimens were interpolated to get the same number of vertices, the positions of corresponding vertices were averaged and then each slice was masked to construct the mean shape of the prostate. The small mold is 30% smaller than the big mold in each dimension. Figure 2.1 shows the 3D draft of prostate in Solidworks.

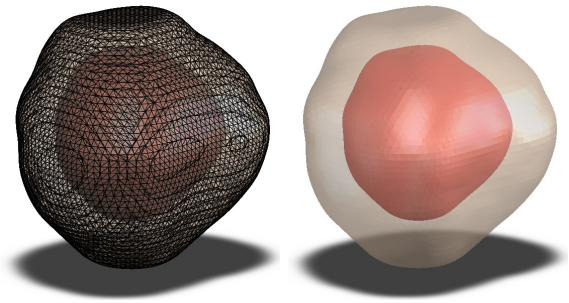


Figure 2.1: Capsule with inner part

2.2.2.2 Pubis and Rectum

The shape of the pubis in the phantom was derived from nine male specimens which were segmented manually in the CT abdominal tomography images. [2]. The specimen image information are summarized in Appendix Table A.2. The surfaces of these sample images were normalized by rotation and translation, and then smoothed by interpolation the same as the prostate specimens. Finally the voxel spacing was normalized to $1 \times 1 \times 1\ mm^3$. The pubis model applied in the phantom was the mean shape of these specimens. The constructed pubis model was then presented in Solidworks as Figure 2.2 shows. Initially most parts of pubis were segmented, including superior rami, pubis symphysis, and inferior rami. Because of phantom size limit, only pubic tubercles, pubis symphysis and parts of inferior rami remained. The rectum in the phantom is an air tube. The mold for casting the tube is finalized the as Figure 2.3 shows, a curved pipe and a hemisphere at the end.

[1] The images were provided by radiology department of UMC St. Radboud in Nijmegen
[2] The datasets is provided by Radiology department in UMC St. Radboud Nijmegen

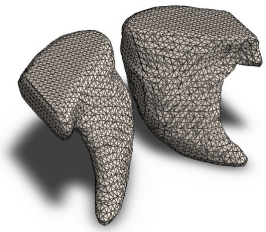


Figure 2.2: 3D pubis model

Figure 2.3: 3D rectum mold

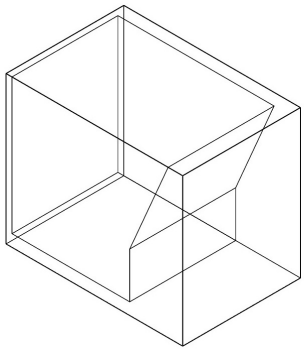


Figure 2.4: 3D Frame sketch

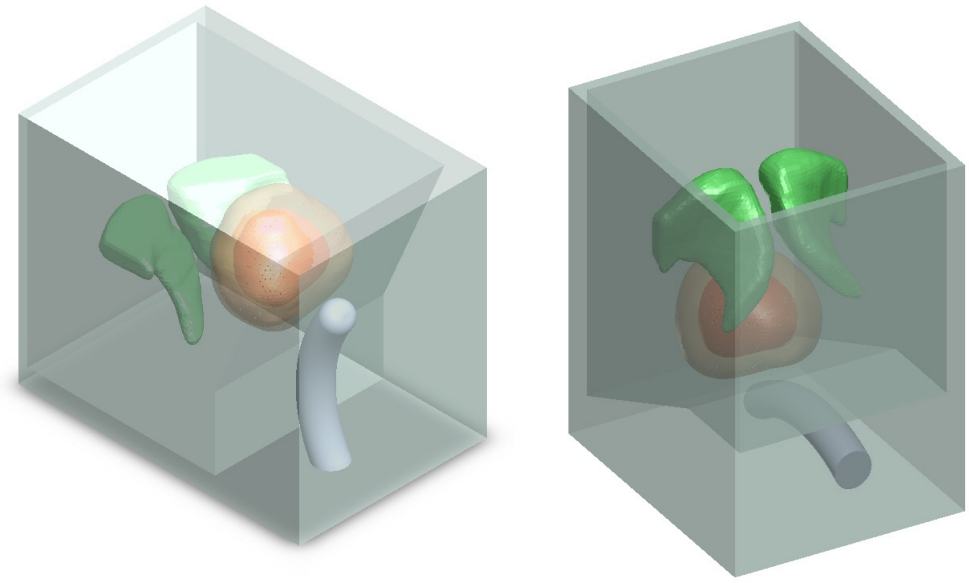
2.2.2.3 Phantom Frame

The frame is the skeleton of the phantom, which has three main functions. First, it works as a container to support the phantom and maintain all the components in the correct positions. Second, one side of the wall mimics the perineum which has to be stiffer as the perineal skin and hypoderm when the needle is inserted. Third, the base of the phantom should form an air tube as the rectum which should be strong enough in case the air tube collapses. Therefore, the frame of the phantom was made similar to the frame by Hungr *et al.* [39], but with a slope face at the bottom considering the relative positions of phantom components. The relative positions statistics among these components were collected by UMC St. Radboud from 100 patients. A detailed annotation and statistical dimensions are shown in Appendix Table A.3 and Figure A.1. The frame sketch is presented in Figure 2.4

2.3 Summary

From all the reviewed phantoms, the art from Hungr *et al.* [39] was selected as the prototype of our phantom. We redesigned the prostate with a two layer structure to give more deformation and added a pubis in the phantom. The phantom would be composed fully as shown in Figure 2.5(a). During the construction and testing, the phantom would be placed as Figure 2.5(b), which corresponds to the patient position and orientation, namely Head First Supine (HFS). The full dimension

specifications of the phantom are attached in Appendix Figure A.2.



(a) Composed phantom sketch

(b) Composed phantom sketch with HFS postion

Figure 2.5: Complete Phantom

The phantom can be made based on this information. The detailed mold and phantom fabrication procedures will be presented in chapter 6. The bio-mechanical and MRI properties will be evaluated in chapter 7.

Chapter 3

3D Active Appearance Models

In this chapter, a 3D AAM approach will be presented, which is based on [20] and more details will be explained. Firstly, we review AAM in medical imaging application briefly. Secondly, we show how to build the 3D appearance[1] model based on a 3D shape model and texture model. Thirdly, the standard 3D AAM Search is introduced.

3.1 AAM in Medical Imaging Applications

As discussed in the first chapter, 3D AAM is selected as the 3D prostate segmentation approach. The classic AAM was firstly proposed by Cootes *et al.* [20]. It was originally meant for an application on 2D face segmentation with some extension to medical image applications such as knee joint MR image slices. It was further developed in [50, 51, 52]. Later on, its application in multiple modalities medical images begins to thrive and also from 2D to 3D. Mitchell *et al.* [53] applied the 3D AAM in segmentation of cardiac MR and ultrasound images. Vincent *et al.* have performed a fully automatic segmentation of the knee Joint [54] and 3D MR prostate images [23]. Kroon *et al.* investigated the segmentation in knee cartilage [55] and mandibular canal [56] of 3D CT data. Maan *et al.*, as mentioned in the previous chapter, proposed a segmentation of 3D prostate MR data as well [22].

Besides the full 3D AAM, a multi-view AAM approach was proposed to represent the 3D pose and shape variance in a particular view-point and the correlations between models of different view-points[57, 58]. Lelieveldt *et al.* built the left ventricle shape and appearance models from multiple standardized views. Üzümcü *et al.* [60] presented an uncoupled multi-view AAM for simultaneous segmentation of cardiac MR images. Oost *et al.* [61] proposed an automated contour detection of the left ventricle with coupled multi-view AAM and dynamic programming for relaxing the constrains on shape change and 3D motions, and so did Leung *et al.* [63]. Hansegård *et al.* [62] have shown an multi-view and multi-frame AAM applied in triplane echocardiograms.

[1] The appearance, in our context, means the combined shape and texture.

As for our work, since the scanned MR 2D slices are different from case to case, namely it is not like echocardiograms where the triplanes and multi-view points are relatively fixed, a multi-view AAM could not be a good option. Therefore, we will build the full 3D AAM.

3.2 3D AAM Modeling

The AAM modeling includes the shape modeling and texture modeling using variance statistics. A combined model can be synthesized afterwards, namely the appearance models. The following context will introduce them in turn.

3.2.1 Shape Modeling

In the three dimensional space, the shape of an object can be represented as 3D vertices and a closed surface mesh[1] connecting these vertices. Shapes are independent on rotation, scaling and translation of the object. Ideally, the vertices of each shape are representative as landmark points. Generally, landmarks are featured as anatomical structural meaningful positions, junctions connecting distinguished boundaries, geometrically defined point collections, or evenly spaced points on an objects surface between existing landmarks. Moreover, landmark points in the shape set should have a good correspondence. However, for a 3D shape it is impossible to label the landmarks manually. As a result, it's essential to use an automatic approach to get landmark points of each shape with a good correspondence between different shapes. We will discuss how to establish the correspondence in the next chapter. Now it is assumed that all the shapes in the training set have a good correspondence, which means all the shapes are represented with well labelled landmarks of the same number and connectivity topology (i.e. mesh topology). The landmark points in each shape are represented by vertices with three dimensional Cartesian coordinates.

3.2.1.1 Shape Alignment

Initially, the shapes can differ with respect to size, orientation and position in the 3D space. This does not contribute anything to the shape variation. In order to get the shape variance statistics, we have to scale, rotate and translate all the shapes into a common coordinate. This is what shape alignment will do. The shape alignment can be done with a Procrustes analysis.

We define a single shape S as a collection of n_p points[2], which can be denoted as a $n_p \times 3$ matrix $\mathbf{X}$, where each row vector $\mathbf{x}_i = (x_i^{(\mathbf{x})}, y_i^{(\mathbf{x})}, z_i^{(\mathbf{x})})$ represents the Cartesian coordinates of the i^{th} points in this shape and $i \in \{1, 2, \cdots, n_p\}$. Let us

[1] Here the mesh in our work is a triangular mesh

[2] In the following context of this chapter, shape points, vertices, and landmark points are the same.

consider two shapes represented as matrix $\mathbf{X}$ and $\mathbf{Y}$, where shape $\mathbf{Y}$ is fixed but shape $\mathbf{X}$ is moving. We wish to find a transformation $\mathcal{T}$ to align $\mathbf{X}$ with respect to $\mathbf{Y}$, which can be written as

$$\mathcal{T}(\mathbf{X}) = s \cdot \mathbf{X}\mathbf{R} + \mathbf{j}\mathbf{t}^T \qquad w.r.t. \quad \mathbf{Y} \tag{3.1}$$

where s is a scaling factor, $\mathbf{R}$ is a 3×3 rotation matrix in 3D space, and $\mathbf{t}$ is a column translation vector $(t_x, t_y, t_z)^T$, $\mathbf{j}$ is a n_p length unit vector $(1, 1, \cdots, 1)^T$. We consider two common methods for finding an appropriate pose parameter set $\{s, \mathbf{R}, \mathbf{t}\}$, namely orthogonal Procrustes analysis and unit quaternions based procrustes analysis.

1. Orthogonal Procrustes Analysis

The basic orthogonal Procrustes analysis problem was discussed and solved in [64]. Generally, this can be identified as an extended orthogonal Procrustes problem, which holds the residual matrix $\mathbf{L} = s\mathbf{X}\mathbf{R} + \mathbf{j}\mathbf{t}^T - \mathbf{Y}$. The objective is to find the least square solution $\{s, \mathbf{R}, \mathbf{t}\}$ to minimize the sum of squares of the residual matrix, which can be written as

$$\{s, \mathbf{R}, \mathbf{t}\} = \arg\min trace\{\mathbf{L}^T\mathbf{L}\} \tag{3.2}$$

$$= \arg\min trace\{(s\mathbf{X}\mathbf{R} + \mathbf{j}\mathbf{t}^T - \mathbf{Y})^T(s\mathbf{X}\mathbf{R} + \mathbf{j}\mathbf{t}^T - \mathbf{Y})\} \tag{3.3}$$

where *trace* means the sum of the diagonal elements in a square matrix. The problem also has anther implicit condition, which is an orthogonal rotation matrix, namely $\mathbf{R}^T\mathbf{R} = \mathbf{I}$, where $\mathbf{I}$ is an identity matrix. Then we can write the Lagrangean function and set its the derivatives with respect to $\{s, \mathbf{R}, \mathbf{t}\}$ to zero in order to obtain a least squares estimate (LSE). Afterwards, we can get the solution as follows:

$$\mathbf{R} = \mathbf{U}\mathbf{V}^T \tag{3.4}$$

$$s = \frac{trace\left\{\mathbf{R}^T\mathbf{X}^T\left(\mathbf{I} - \frac{\mathbf{j}\mathbf{j}^T}{n_p}\right)\mathbf{Y}\right\}}{trace\left\{\mathbf{X}^T\left(\mathbf{I} - \frac{\mathbf{j}\mathbf{j}^T}{n_p}\right)\mathbf{X}\right\}} \tag{3.5}$$

$$\mathbf{t} = (\mathbf{Y} - s\mathbf{X}\mathbf{R})^T\frac{\mathbf{j}}{n_p} \tag{3.6}$$

where $\mathbf{U}$ and $\mathbf{V}$ are two orthogonal matrices derived from the Singular Value Decomposition (SVD) of matrix $\mathbf{S}$ that is,

$$\mathbf{S} = \mathbf{X}^T\left(\mathbf{I} - \frac{\mathbf{j}\mathbf{j}^T}{n_p}\right)\mathbf{Y} \tag{3.7}$$

and its SVD holds the format $\mathbf{S} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where Σ is a diagonal matrix. Actually, $(\mathbf{I} - \mathbf{j}\mathbf{j}^T/n_p)\mathbf{Y}$ performs the translation of $\mathbf{Y}$ from its centroid to the coordinate origin. In practice, we prefer to normalize $\mathbf{X}$ and $\mathbf{Y}$ first then calculate the rotation

matrix. For more details, it can be referred to Appendix B.

When ASM was proposed by Cootes *et al.* [16], a 2D weighted orthogonal Procrustes analysis was performed using a linear equations set to find the solution. The weights for each point can give more significance to the points which are more stable across the set. However, it is hard to find a solution with respect to 3 dimensional space. In [65], the solution of a weighted orthogonal Procrustes analysis was also given, which can be derived from the unweighted one with minor changes. The details can also be referred to Appendix B.

2. Unit Quaternions based Procrustes Analysis

Alternatively, the unit quaternions based approach can give an absolute orientation [66]. To be specific, the objective is also to minimize the sum of squares of residual errors, as equation 4.3 shows. We also define $\mathbf{X}$ and $\mathbf{Y}$ as $\mathbf{X} = \{\mathbf{x}_i = (x_i^{(\mathbf{x})}, y_i^{(\mathbf{x})}, z_i^{(\mathbf{x})}) | i = 1, 2, \cdots, n_p\}$ and $\mathbf{Y} = \{\mathbf{y}_i = (x_i^{(\mathbf{y})}, y_i^{(\mathbf{y})}, z_i^{(\mathbf{y})}) | i = 1, 2, \cdots, n_p\}$ where each row is one point coordinate in each shape. The absolute translations are the centroid coordinates as

$$\overline{\mathbf{x}} = \frac{1}{n_p} \sum_{i=1}^{n_p} \mathbf{x}_i, \qquad \overline{\mathbf{y}} = \frac{1}{n_p} \sum_{i=1}^{n_p} \mathbf{y}_i \tag{3.8}$$

Now the centered shapes are (This is the same as using $(\mathbf{I} - \mathbf{j}\mathbf{j}^T/n_p)$ to left multiply $\mathbf{X}$ and $\mathbf{Y}$, as last section states.)

$$\mathbf{x'}_i = \mathbf{x}_i - \overline{\mathbf{x}}, \quad \mathbf{y'}_i = \mathbf{y}_i - \overline{\mathbf{y}}, \qquad i = 1, 2, \cdots, n_p. \tag{3.9}$$

The relative translation $\mathbf{t}$ is actually the difference of the centroid of $\mathbf{Y}$ and the scaled and rotated centroid of $\mathbf{X}$, namely,

$$\mathbf{t} = \overline{\mathbf{y}} - s\overline{\mathbf{x}}\mathbf{R} \tag{3.10}$$

Once we get the scale s and the rotation matrix $\mathbf{R}^T$ we can calculate the translation $\mathbf{t}$. The objective function now can be written as

$$\sum_{i=1}^{n_p} \|\mathbf{y'}_i - s\mathbf{x'}_i\mathbf{R}\|^2 \tag{3.11}$$

According to the derivation in [66], the scale can be obtained by

$$s = \frac{\sum_{i=1}^{n_p} \left[\mathbf{y'}_i \cdot (\mathbf{x'}_i\mathbf{R})\right]}{\sum_{i=1}^{n_p} \|\mathbf{x'}_i\|^2} \tag{3.12}$$

This expression is similar with equation 3.5. As for the rotation, we aims at finding

$\mathbf{R}$ that maximizes $\sum_{i=1}^{n_p} [\mathbf{y'}_i \cdot (\mathbf{x'}_i \mathbf{R})]$. Now we use the quaternions approach to solve this problem, which is to maximize $\dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}}$ by finding the optimum quaternion $\dot{\mathbf{q}}$. For more explanations, [66] can be referred. Technically, we first introduce a 3×3 matrix $\mathbf{S}$ which contains all the information required to solve the least squares problem for rotation. $\mathbf{S}$ holds the form

$$\mathbf{S} = \sum_{i=1}^{n_p} \mathbf{x'}_i^T \mathbf{y'}_i \quad or \quad \mathbf{S} = \mathbf{X'}^T \mathbf{Y}, \quad \mathbf{S} = \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{bmatrix} \tag{3.13}$$

where $\mathbf{X'}$ is the collection of $\mathbf{x'}_i$ and each row is the row vector $\mathbf{x'}_i$. It is the same with $\mathbf{Y'}$ and $\mathbf{y'}_i$. Then we form the matrix $\mathbf{M}$ represented as $\mathbf{M} =$

$$\begin{bmatrix} S_{xx}+S_{yy}+S_{zz} & S_{yz}-S_{zy} & S_{zx}-S_{xz} & S_{xy}-S_{yx} \\ S_{yz}-S_{zy} & S_{xx}-S_{yy}-S_{zz} & S_{xy}+S_{yx} & S_{zx}+S_{xz} \\ S_{zx}-S_{xz} & S_{xy}+S_{yx} & -S_{xx}+S_{yy}-S_{zz} & S_{yz}+S_{zy} \\ S_{xy}-S_{yx} & S_{zx}+S_{xz} & S_{yz}+S_{zy} & -S_{xx}-S_{yy}+S_{zz} \end{bmatrix} \tag{3.14}$$

Afterwards, we can calculate the normalized eigenvectors and eigenvalues $\{\mathbf{n}_j, \lambda_j | j = 1,2,3,4\}$ of $\mathbf{M}$, where $\mathbf{M}\mathbf{n}_j = \lambda_j \mathbf{n}_j$ and $\|\mathbf{n}_j\|^2 = 1$ are satisfied. Then we assume the maximal (most positive) eigenvalue is λ_{max} which holds $\lambda_{max} = max\{\lambda_j | j = 1,2,3,4\}$, and the corresponding eigenvector is $\mathbf{n}_{max}$, which satisfies $\mathbf{M}\mathbf{n}_{max} = \lambda_{max}\mathbf{n}_{max}$. It can be proved that the quadratic form $\dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}}$ can be maximized when $\dot{\mathbf{q}} = \mathbf{n}_{max}$, where $\dot{\mathbf{q}}$ is the unit quaternion representing the rotation information.

Now we assign $\{q_0, q_x, q_y, q_z\}$ constructing the unit quaternion $\dot{\mathbf{q}}$ with the four elements in $\mathbf{n}_{max}$ respectively, then the transpose of the rotation matrix can be written as

$$\mathbf{R}^T = \begin{bmatrix} q_0^2+q_x^2-q_y^2-q_z^2 & 2(q_x q_y - q_0 q_z) & 2(q_x q_z + q_0 q_y) \\ 2(q_y q_x + q_0 q_z) & q_0^2-q_x^2+q_y^2-q_z^2 & 2(q_y q_z - q_0 q_x) \\ 2(q_z q_x - q_0 q_y) & 2(q_z q_y + q_0 q_x) & q_0^2-q_x^2-q_y^2+q_z^2 \end{bmatrix} \tag{3.15}$$

or in matrix notation as

$$\mathbf{R}^T = \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix}^T + \begin{bmatrix} q_0 & -q_z & q_y \\ q_z & q_0 & -q_x \\ -q_y & q_x & q_0 \end{bmatrix}^2 \tag{3.16}$$

With the rotation matrix $\mathbf{R}$ we can calculate the scale s and translation vector $\mathbf{t}$, then we can obtain the aligned shape as represented by equation 3.1.

Later on Horn *et al.* [67] proposed another closed form solution for absolute orientation using orthonormal matrices, which is an advance of the quaternions based approach.

We define n_s shapes $\{S_i | i = 1, 2, \cdots, n_s\}$ in the training set, and the shape points in shape S_i can be denoted as $\mathbf{X}_i$. Then we use an iterative way to align all the shapes as follows:

(1) Set the first shape $\mathbf{X}_1$ as the mean shape $\overline{\mathbf{X}}$.

(2) Align all the shapes $\mathbf{X}_i$ with respect to $\overline{\mathbf{X}}$ to the aligned shape points $\mathbf{X}_i^{(a)} = \mathcal{T}_i(\mathbf{X}_i)$ as well as the pose parameter sets $\{s_i, \mathbf{R}_i, \mathbf{t}_i\}$,

(3) Calculate the mean shape $\overline{\mathbf{X}}$ from the aligned shapes $\mathbf{X}_i^{(a)}$, and mean pose parameter set $\{\overline{s}, \overline{\mathbf{R}}, \overline{\mathbf{t}}\}$ over the training set and save them.

(4) Transform the mean shape $\overline{\mathbf{X}}$ inversely with inverse pose parameter set $\{1/\overline{s}, \overline{\mathbf{R}}^{-1}, -\overline{\mathbf{t}}\}$ to update the mean shape $\overline{\mathbf{X}}$, namely $\mathcal{T}^{-1}(\overline{\mathbf{X}})$.

(5) Return to step (2) with the updated mean shape $\overline{\mathbf{X}}$, until stable.

(6) Align all the shapes $\mathbf{X}_i$ with respect to the stable mean shape $\overline{\mathbf{X}}$, save the aligned shape $\mathbf{X}_i^{(a)}$ and the pose parameter sets $\{s_i, \mathbf{R}_i, \mathbf{t}_i\}$,

Actually both of the approaches discussed above are available, but the quaternion based approach has the advantage of using less parameters (only 4 elements can represent the rotation) to control the transformation, which is more preferable in AAM.

3.2.1.2 Shape Variance Modeling

Now we suppose all the shapes have been aligned. As introduced in [20], we define a $3n_p$ dimensional shape vector $\mathbf{x}$ in 3D space to represent the aligned shape points $\mathbf{X}^{(a)}$ in a single shape S as

$$\mathbf{x} = (x_1, x_2, \cdots, x_{n_p}, y_1, y_2, \cdots, y_{n_p}, z_1, z_2, \cdots, z_{n_p})^T \tag{3.17}$$

where (x_j, y_j, z_j) is the Cartesian coordinates of the j^{th} points and $j \in \{1, 2, \cdots, n_p\}$. Therefore, the shape points of n_s shapes can be represented as a column vector set $\{\mathbf{x}_i | i = 1, 2, \cdots, n_s\}$. Then we can apply PCA to construct the point distribution model for modeling the shape statistics. It is worth noting that if the scattering of shape points is Gaussian in the space, the PCA can find the optimal principal axes. If the data do not have a Gaussian distribution, then the variance cannot be used as criterion of evaluating the component significance, and PCA decomposition is not the optimal approach, though it may work well. In this case, the independent components analysis (ICA) could be applied.

We consider a data set with Gaussian distribution. Usually we first calculate the centroid of all shape vectors. The maximum likelihood estimate (MLE) of the

covariance can be given as

$$\overline{\mathbf{x}} = \frac{1}{n_s}\sum_{i=1}^{n_s}\mathbf{x}_i, \qquad \mathbf{S} = \frac{1}{n_s}\sum_{i=1}^{n_s}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T = \frac{1}{n_s}\mathbf{D}\mathbf{D}^T \tag{3.18}$$

where $\mathbf{D}$ is $3n_p \times n_s$ matrix as $\mathbf{D} = (\mathbf{x}_1 - \overline{\mathbf{x}}, \mathbf{x}_2 - \overline{\mathbf{x}}, \cdots, \mathbf{x}_{n_s} - \overline{\mathbf{x}})$.

By PCA, we can get the $3n_p \times 3n_p$ diagonal matrix $\mathbf{\Lambda}_s$ in which the diagonal elements are eigenvalues, and the $3n_p \times 3n_p$ orthogonal matrix $\mathbf{P}_s$ which is composed by $3n_p$ column-wise corresponding eigenvectors. This can be denoted as

$$\mathbf{S}\mathbf{P}_s = \mathbf{P}_s\mathbf{\Lambda}_s, \quad \mathbf{P_s} = (\mathbf{p}_1^{(s)}, \mathbf{p}_2^{(s)}, \cdots, \mathbf{p}_{3n_p}^{(s)}), \qquad \mathbf{\Lambda}_s = \begin{bmatrix} \lambda_1^{(s)} & & & \\ & \lambda_2^{(s)} & & \\ & & \ddots & \\ & & & \lambda_{3n_p}^{(s)} \end{bmatrix} \tag{3.19}$$

We arrange the eigenvalues in a descending order, and corresponding eigenvectors, and we assume now $\lambda_1^{(s)} \geq \lambda_2^{(s)} \geq \cdots \lambda_{3n_p}^{(s)}$. Based on these parameters, we can have a linear representation for a shape instance, as

$$\mathbf{x} = \overline{\mathbf{x}} + \mathbf{P}_s\mathbf{b}_s \tag{3.20}$$

where $\mathbf{b}_s = \{b_1^{(s)}, b_2^{(s)}, \cdots, b_{3n_p}^{(s)}\}$ is shape model parameter, or mode weight vector. There are $3n_p$ elements in $\mathbf{b}_s$ and each of them gives a weight to one mode. Because the shape data are satisfying the Gaussian distribution, the weight should be constrained by $-3\sqrt{\lambda_m^{(s)}} \leq b_m^{(s)} \leq +3\sqrt{\lambda_m^{(s)}}$ where $\sqrt{\lambda_m^{(s)}}$ is the standard deviation of the parameters and $m \in \{1, 2, \cdots, 3n_p\}$ is the mode index. For any shape points from the training set, the mode weight vector can be obtained by

$$\mathbf{b}_i^{(s)} = \mathbf{P}_s^T(\mathbf{x}_i - \overline{\mathbf{x}}) \qquad i = 1, 2, \cdots, n_s \tag{3.21}$$

When there are many points in the shape, we would like to take the modes with more significances to represent the shape and the rest ones are seen as noise. Whereas the explained variances of each eigenvector is equal to the corresponding eigenvalues and we have arranged the eigenvalues in a descending order, so it is practical to take the first t_s modes. We introduce p to represent the percentage of retained variation, and t_s can be chosen as

$$\sum_{m=1}^{t_s}\lambda_m^{(s)} \geq \frac{p}{100\%}\sum_{m=1}^{3n_p}\lambda_m^{(s)} \tag{3.22}$$

Actually, when the number of shapes in the training set n_s is much smaller than the shape vector dimension $3n_p$, it is computationally expensive to get the eigenvalues

and eigenvectors since the covariance matrix is huge. Because if there are n_s shapes, there are at most $n_s - 1$ non zero eigenvalues (non computational zero value). As a result in the case of $n_s - 1 < 3n_p$, we can calculate the eigenvalues and eigenvectors faster in the following way:

We define a $n_s \times n_s$ matrix $\mathbf{S}'$ as

$$\mathbf{S}' = \frac{1}{n_s}\mathbf{D}^T\mathbf{D} \tag{3.23}$$

Then we find the eigenvectors and corresponding eigenvalues of $\mathbf{S}'$ as $\mathbf{P}'_s = (\mathbf{p}'^{(s)}_1, \mathbf{p}'^{(s)}_2, \cdots, \mathbf{p}'^{(s)}_{n_s})$ and $\mathbf{\Lambda}'_s = diag\{\lambda'^{(s)}_1, \lambda'^{(s)}_2, \cdots, \lambda'^{(s)}_{n_s}\}$ respectively. According to [16], the first n_s eigenvalues of $\mathbf{S}$ (which are already in a descending order) are same as the eigenvalues of $\mathbf{S}'$, and the corresponding eigenvectors of $\mathbf{S}$ are linear compositions of eigenvectors of $\mathbf{S}'$. This can be represented as

$$\lambda^{(s)}_i = \lambda'^{(s)}_i \qquad i = 1, 2, \cdots, n_s \tag{3.24}$$

$$\mathbf{p}^{(s)}_i = \frac{1}{\sqrt{\lambda'^{(s)}_i n_s}}\mathbf{D}\mathbf{p}'^{(s)}_i \tag{3.25}$$

Finally with the first t_s modes, the shape instance is

$$\mathbf{x} \approx \overline{\mathbf{x}} + \mathbf{P}_{s,t_s}\mathbf{b}_{s,t_s} = \overline{\mathbf{x}} + \sum_{m=1}^{t_s} \mathbf{p}^{(s)}_m b^{(s)}_m \tag{3.26}$$

Briefly $\mathbf{P}_s$ is used to briefly denote $\mathbf{P}_{s,t_s}$ which contains t_s column-wise eigenvectors.

3.2.2 Texture Modeling

Similarly to shapes, a linear relationship can be formed to obtain the texture instance. The texture information in our implementation is the image gray value, i.e. intensity, in each voxel. In the basic AAM, we do not extract all the texture information in each image, and only the voxels enclosed by the shape points will be used. To be specific, we should first warp all the shapes in the training set to match the mean shape $\overline{\mathbf{x}}$ in order to build 'correspondence' between the candidate voxel positions in different shapes. Then, like the alignment in shapes, the gray values should also be normalized. Finally, the texture statistics will be constructed using PCA.

3.2.2.1 Obtaining Texture

In the basic 3D AAM, we first take the mean shape $\overline{\mathbf{x}}$ points as base points and normalize them to a unit cubic space. If the base points can be denoted as $\overline{\mathbf{x}} = \{(\overline{x}_i, \overline{y}_i, \overline{z}_i) | i = 1, 2, \cdots, n_p\}$ where n_p is the number of landmarks, then the

normalization can be made by

$$(\overline{x}_i, \overline{y}_i, \overline{z}_i) \leftarrow \frac{(\overline{x}_i, \overline{y}_i, \overline{z}_i) - (xyz_{min}, xyz_{min}, xyz_{min})}{xyz_{max} - xyz_{min}} \tag{3.27}$$

where xyz_{min} is the minimal value of all the three coordinates in all points of the base points, and xyz_{max} is the maximal value of all point three coordinates. Then, we map this normalized base points to a cubic grid space of which the size is predefined as texture size. We call the mapped points inside the grid of texture size the *mapped base points*. Afterwards, we use this *mapped base points* to calculate (1) a voxel template, and (2) tetrahedrons in the shape.

The voxel template specifies the voxel positions where the texture information would be taken out. By examining the triangular faces of the *mapped base points*, we divided each triangular face into two triangular sub-faces from the longest edge in each face if the longest edge is larger than half of a voxel size. After splitting the faces, we assigned all the face vertices to the grid to get the voxel positions of the shape surfaces in the voxel template. After emphasizing the voxels inside the surface, we can finally obtain the voxel template, where all the voxels on and inside the *mapped base points* are labelled 1 and others are 0, which is like a segmented image. Figure 3.1 uses a 2D circle to illustrate a pixel template.

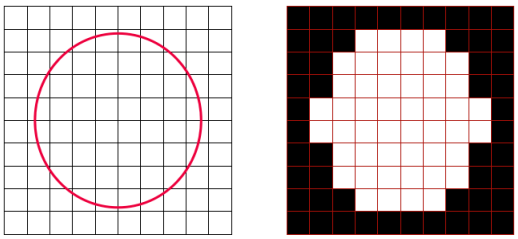


Figure 3.1: A pixel template of a circle in 2D

However different shapes have different texture sizes. In order to get the equivalent texture information, namely voxel intensities following the order as in the voxel template, we should transform the voxel in each shape into the voxel in *mapped base points*, which can be done by image warping techniques. It is worth noting that the shape points in shape S from the training set should not be aligned, namely they are original shape points different from the shape represented by shape vector $\mathbf{x}_i$.

1. Image Warping

In general, Image warping is to build the spatial relationship between two different images. To be specific, in the 3D shapes, we would like to map the spatial position of voxels inside the *mapped base points* to relative spatial positions in a given shape, namely in all the shapes in the training set. Mathematically, if we define the voxel points inside the *mapped base points* as a set $\{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{n_g}\}$, where $\mathbf{v}_i = \{(x_i, y_i, z_i) | i = 1, 2, \cdots, n_g\}$ is the spatial coordinate of each point and n_g is the total number of voxels, then we can define a mapping function $\boldsymbol{f}$ which can

map $\mathbf{v}_i$ from the *mapped base points* to the relative position $\mathbf{v'}_i$ inside the a shape S, namely,

$$\boldsymbol{f}(\mathbf{v}_i) = \mathbf{v'}_i \qquad i = 1, 2, \cdots, n_g \tag{3.28}$$

For simplicity, we perform the image warping by a piecewise affine transformation and assume the mapping $\boldsymbol{f}$ is linear. In [68] the 2D warping specifications were introduced. For a 3D case, we apply the affine transformation between two tetrahedrons instead of delaunay triangles. Figure 3.2 shows an illustration of 3D image warping.

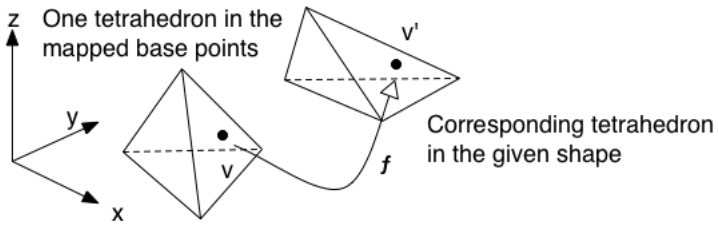


Figure 3.2: Tetrahedron transformations

We assume a single tetrahedron from the *mapped base points* with four vertices defined in the Cartesian coordinates as $\mathbf{v}_i = \{(x_i, y_i, z_i)^T | i = 1, 2, 3, 4\}$, then we can define the tetrahedral coordinate system with three edge vectors in the tetrahedron, For instance, we define the three vectors as $\mathbf{e}_i = \{(\mathbf{v}_i - \mathbf{v}_4) | i = 1, 2, 3\}$. So for any point $\mathbf{v} = (x, y, z)^T$, it can be represented as

$$\mathbf{v} = \mathbf{v}_4 + \alpha_1\mathbf{e}_1 + \alpha_2\mathbf{e}_2 + \alpha_3\mathbf{e}_3 \tag{3.29}$$

$$= \mathbf{v}_4 + \alpha_1(\mathbf{v}_1 - \mathbf{v}_4) + \alpha_2(\mathbf{v}_2 - \mathbf{v}_4) + \alpha_3(\mathbf{v}_3 - \mathbf{v}_4) \tag{3.30}$$

$$= \alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \alpha_3\mathbf{v}_3 + \alpha_4\mathbf{v}_4 \tag{3.31}$$

where $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1$. Therefore, we should find the representation of α_i using the Cartesian coordinates of $\mathbf{v}_i$ and $\mathbf{v}$. We first rewrite the equation 3.29 as

$$\mathbf{v} = \begin{bmatrix}\mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3\end{bmatrix} \begin{bmatrix}\alpha_1 & \alpha_2 & \alpha_3\end{bmatrix}^T + \mathbf{v}_4 \tag{3.32}$$

Hence, we can derive that

$$\begin{bmatrix}\alpha_1 & \alpha_2 & \alpha_3\end{bmatrix}^T = \begin{bmatrix}\mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3\end{bmatrix}^{-1} (\mathbf{v} - \mathbf{v}_4) \qquad \alpha_4 = 1 - \alpha_1 - \alpha_2 - \alpha_3 \tag{3.33}$$

For a singe tetrahedron represented as $\{\mathbf{v'}_i | i = 1, 2, 3, 4\}$ in another shape S, we can find the Cartesian coordinate corresponding point of $\mathbf{v}$, the $\mathbf{v'} = (x', y', z')^T$ as

$$\mathbf{v'} = \alpha_1\mathbf{v'}_1 + \alpha_2\mathbf{v'}_2 + \alpha_3\mathbf{v'}_3 + \alpha_4\mathbf{v'}_4 \tag{3.34}$$

If the point $\mathbf{v}$ is inside the tetrahedron, the tetrahedral coordinates should suffice $0 \leq \alpha_i \leq 1$, for $i = 1, 2, 3, 4$. So we can use this criterion to find all the tetrahedral

coordinates of the voxels inside all the tetrahedrons constructed from the *mapped base points*. With the same tetrahedral coordinates $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ the corresponding positions in the each shape S in the training set can be obtained.

However, these found corresponding points positions $\mathbf{v}'$ in image of shape S could not right lie in the voxel grid. For obtaining the voxel intensity in that position, we will apply the trilinear voxel interpolation to these corresponding points positions.

2. Trilinear Voxel Interpolation[1]

The trilinear interpolation is identical to three successive linear interpolations. Figure 3.3 interprets how to obtain the interpolated gray value at an arbitrary position. We suppose the point V_{xyz} is the point position in shape S corresponding

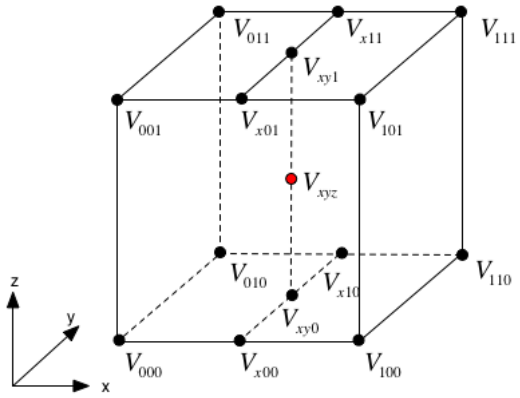


Figure 3.3: Trilinear Voxel Interpolation

to the voxel inside the reference, the *mapped base points*. The cubic lattice represents the eight adjacent voxels points V_{ijk} around point V_{xyz}, which hold the coordinates as $\{(x_i, y_j, z_k)|i, j, k \in \{0, 1\}\}$. The gray values in the eight voxels are g_{ijk} (e.g. the intensity in voxel V_{000} is g_{000} and so forth). Firstly, we calculate the relative position ratio of V_{xyz} as

$$\gamma_x = \frac{x - x_0}{x_1 - x0} \quad \gamma_y = \frac{y - y_0}{y_1 - y_0} \quad \gamma_z = \frac{z - z_0}{z_1 - z_0} \tag{3.35}$$

Then by linear interpolation, the gray values in V_{x00}, V_{x10}, V_{x01}, V_{x11} can be calculated respectively as

$$g_{x00} = g_{000}(1 - \gamma_x) + g_{100}\gamma_x \tag{3.36}$$
$$g_{x10} = g_{010}(1 - \gamma_x) + g_{110}\gamma_x \tag{3.37}$$
$$g_{x01} = g_{001}(1 - \gamma_x) + g_{101}\gamma_x \tag{3.38}$$
$$g_{x11} = g_{011}(1 - \gamma_x) + g_{111}\gamma_x \tag{3.39}$$

[1]This can be referred to http://en.wikipedia.org/wiki/Trilinear_interpolation

Similarly, the intensities in V_{xy0} and V_{xy1} are

$$g_{xy0} = g_{x00}(1-\gamma_y) + g_{x10}\gamma_y \tag{3.40}$$
$$g_{xy1} = g_{x01}(1-\gamma_y) + g_{x11}\gamma_y \tag{3.41}$$

Finally the intensity in V_{xyz} is

$$g_{xyz} = g_{xy0}(1-\gamma_z) + g_{xy1}\gamma_z \tag{3.42}$$

After obtaining all the interpolated intensities inside all the tetrahedrons of each shape from the training set, it is easy get a texture map $\mathbf{J}_i$ inside a set of fixed shape points for each shape S_i. One texture map $\mathbf{J}_i$ is a collection of corresponding intensities[1] from a shape in training set, and the fixed shape points positions, i.e. *mapped base points*, here $i = 1, 2, \cdots, n_s$. Accordingly, we use the voxel template to resample all the texture maps to ensure all textures have the same voxel number and order. After vectorization of these textures, for a shape S_i, we obtain a $n_g \times 1$ shape texture vector $\mathbf{g}_i$, where n_g is the number of voxel/intensity values in the shape and $i = 1, 2, \cdots, n_s$.

3.2.2.2 Texture Normalization

Like the alignment in shape points, texture, namely gray values of the shape should also be normalized to dismiss the variance in the global illumination. Conventionally, this procedure would be done in an iterative way [20, 68]. Briefly, the normalized intensity vector of shape S_i can be processed as following.

(1) Calculate the mean of all texture vectors, namely, $\overline{\mathbf{g}} = \frac{1}{n_s}\sum_{i=1}^{n_s}\mathbf{g}_i$.

(2) Standardize the mean texture by $\overline{\mathbf{g}} \mapsto \frac{1}{\sigma^2}(\overline{\mathbf{g}} - \overline{g}\mathbf{1})$, where $\overline{g}$ is the mean estimate of the mean texture vector as $\overline{g} = \frac{1}{n_g}\overline{\mathbf{g}} \cdot \mathbf{1}$, and σ^2 is the variance estimate represented as $\sigma^2 = \frac{1}{n_g}(\overline{\mathbf{g}} - \overline{g}\mathbf{1}) \cdot (\overline{\mathbf{g}} - \overline{g}\mathbf{1})$

(3) For all the texture vectors of all shapes, $\mathbf{g}_i$, $i = 1, 2, \cdots, n_s$, update the normalized gray value vector as

$$\mathbf{g}_i \mapsto \frac{\mathbf{g}_i - \beta\mathbf{1}}{\alpha} \tag{3.43}$$

where $\alpha = \mathbf{g}_i \cdot \overline{\mathbf{g}}$, and $\beta = (\mathbf{g}_i \cdot \mathbf{1})/n_g$.

(4) Return to step (1), until $\overline{\mathbf{g}}$ is stable.

[1] They are the intensities at the corresponding points in the shape, which correspond to the voxels inside the mapped base points

Alternatively, if the illumination variance is minor, we can use a more simple way to normalize the texture vectors. We still apply the equation 3.43 but assign α with the standard deviation of $\mathbf{g}_i$ and β with the mean gray value of the n_g intensities in $\mathbf{g}_i$ [56].

3.2.2.3 Texture Variance Modeling

With the same procedure as processing the shape points data, we apply the PCA to the texture vectors set as well. Briefly, we define the $n_g \times n_s$ matrix $\mathbf{G} = (\mathbf{g}_1 - \overline{\mathbf{g}}, \mathbf{g}_2 - \overline{\mathbf{g}}, \cdots, \mathbf{g}_{n_s} - \overline{\mathbf{g}})$, then calculate the covariance or use the SVD to get the eigenvectors $\mathbf{P}_g = \{\mathbf{p}_i^{(g)} | i = 1, 2, \cdots, t_g\}$ and the eigenvalues $\mathbf{\Lambda}_g = \{\lambda_i^{(g)} | i = 1, 2, \cdots, t_g\}$ where t_g represents the number of modes used. Using the PDM we can obtain the texture instance $\mathbf{g}$ with a linear model written as

$$\mathbf{g} \approx \overline{\mathbf{g}} + \mathbf{P}_{g,t_g}\mathbf{b}_{g,t_g} = \overline{\mathbf{g}} + \sum_{m=1}^{t_g} \mathbf{p}_m^{(g)} b_m^{(g)} \tag{3.44}$$

where $\mathbf{b}_{g,tg}$ is the mode weight vector containing t_g weight scalar $b_m^{(g)}$ for each mode. Due to the statistic feature, each weight should be constrained by $-3\sqrt{\lambda_m^{(g)}} \leq b_m^{(g)} \leq +3\sqrt{\lambda_m^{(g)}}$ for the mode $m \in \{1, 2, \cdots, t_g\}$. The weight vector $\mathbf{b}_i^{(g)}$ of any texture instance from the training set can be obtained as

$$\mathbf{b}_i^{(g)} \approx \mathbf{P}_{g,t_g}^T(\mathbf{g}_i - \overline{\mathbf{g}}) \tag{3.45}$$

If there is no special explanation $\mathbf{P}_g$ will stand for $\mathbf{P}_{g,t_g}$.

3.2.3 Appearance Modeling

From the last two sections, it is easy to build the shape models and textures models given the mean and eigenvectors and the mode weights (also called model parameters). If the variance of shape landmark points and the variance of shape texture are correlated, it is possible to build an appearance model, synthesizing the shape and texture model.

3.2.3.1 Shape and Texture Combination

For a certain appearance instance, we wish to use a model parameter $\mathbf{b}$ to generate the appearance model. Simply $\mathbf{b}$ is concatenated from weighted shape parameters $\mathbf{b}_s$ and the texture parameters $\mathbf{b}_g$, as

$$\mathbf{b} = \begin{bmatrix} \mathbf{W}_s\mathbf{b}_s \\ \mathbf{b}_g \end{bmatrix} = \begin{bmatrix} \mathbf{W}_s\mathbf{P}_s^T(\mathbf{x} - \overline{\mathbf{x}}) \\ \mathbf{P}_g^T(\mathbf{g} - \overline{\mathbf{g}}) \end{bmatrix} \tag{3.46}$$

where $\mathbf{W}_s$ is a diagonal matrix of weights for each shape parameter to unify the shape and gray values. If we apply PCA to the combined parameters, a synthesized

parameter $\mathbf{c}$ can be obtained by

$$\mathbf{b} = \overline{\mathbf{b}} + \mathbf{Q}_{t_b}\mathbf{c} \tag{3.47}$$

where $\mathbf{Q_{t_b}}$ is t_b eigenvectors of appearance model corresponding to the first t_b largest eigenvalues, and $\mathbf{c}$ is the model parameter. Then for a given shape appearance from the training set, the model parameter $\mathbf{c}$ can be obtained by $\mathbf{c} = \mathbf{Q}_{t_b}^T(\mathbf{b} - \overline{\mathbf{b}})$. Approximately $\overline{\mathbf{b}}$ is zero since both $\mathbf{b}_s$ and $\mathbf{b}_g$ have zero means theoretically. Hence, in some literature, the appearance model could be represented as

$$\mathbf{b} = \mathbf{Q}_{t_b}\mathbf{c} \tag{3.48}$$

If there is no extra interpretation, $\mathbf{Q}$ will stand for $\mathbf{Q}_{t_b}$.

3.2.3.2 Parameter Weights Selection

For determining the weight matrix $\mathbf{W}_s$, a common way is simply using unified weights placed in the diagonal [68], namely,

$$\mathbf{W}_s = \begin{bmatrix} \frac{\lambda_g}{\lambda_s} & & \\ & \ddots & \\ & & \frac{\lambda_g}{\lambda_s} \end{bmatrix} \quad \lambda_g = \sum_{m=1}^{t_g} \lambda_m^{(g)} \quad \lambda_s = \sum_{m=1}^{t_s} \lambda_m^{(s)} \tag{3.49}$$

A more general way is to give a small change in shape parameter $\mathbf{b}_s$ and to find how much the texture parameter $\mathbf{b}_g$ will change. Since $\mathbf{P}_g$ is an orthogonal matrix, an unit change in texture vector $\mathbf{g}$ is identical to unit change in texture parameter $\mathbf{b}_g$. Therefore, the root mean square (RMS) change of $\mathbf{g}$ when there is unit change of shape parameter $\mathbf{b}_s$ forms the weights $\mathbf{W}_s$ [20, 68].

Specifically, we first calculate the original gray vectors $\mathbf{g}_i$ in each shape with the optimum shape parameters. As extracting the gray values, the shape points should be aligned inversely (adding the scale, rotation and translation) for finding the right shape in the original image. We assume t_s modes can be used. Then, for each shape mode weight, we change around the optimum value ($b_m^{(s)} \pm 0.5$) and calculate the displaced shape points respectively. We use these inversely aligned displaced shape points to sample the image to obtain a displaced gray value vector $\mathbf{g'}_i$. Afterwards we calculate the RMS_i between $\mathbf{g'}_i$ and $\mathbf{g}_i$, and get a mean RMS for $i = 1, 2, \cdots, n_s$, as a result the mean RMS of gray value displacements from all n_s instances is actually the weights element $w_s^{(m)}$ with respect to mode m. Finally we can get the weights matrix accordingly.

3.3 Standard AAM Search

After establishing the shape, texture, and appearance model based on the training set, it is important to consider how to utilize this information to perform a model search given a new image. The model search[1] should provide a way to find the appropriate pose parameters and appearance parameters to segment the object as accurate as possible. The standard AAM search algorithm was also proposed in [20] and implemented in many applications. The basic approach is to minimize the objective function, which is defined as

$$\Delta = |\delta \mathbf{g}|^2 = |\mathbf{g}_s - \mathbf{g}_m|^2 \tag{3.50}$$

where $\mathbf{g}_s$ is the sampled intensity values given the current found shape points, and $\mathbf{g}_m$ is the gray values derived from the current model parameters. So we wish to update the appearance parameters $\mathbf{c}$ and the pose parameter set $\{s, \mathbf{R}, \mathbf{t}\}$ iteratively to minimize the magnitude of the difference vector $\delta\mathbf{g}$. Hence it is of importance to learn the relationship between the offset of appearance model parameters plus pose parameters i.e. $\delta\mathbf{c}$ and $\{\delta s, \delta\mathbf{R}, \delta\mathbf{t}\}$, and the intensity difference i.e. $\delta\mathbf{g}$.

3.3.1 Parameter Update Modeling

We define a parameter vector $\mathbf{p}$ is the combination of appearance parameter $\mathbf{c}$, the translation $\mathbf{t}$ and the scaled rotation parameters $\mathbf{q}$ (the quaternions with scale[2]). It can be written as $\mathbf{p} = (\mathbf{c}^T, \mathbf{t}^T, \mathbf{q}^T)^T$. As for the relationship between the parameter $\mathbf{p}$ and intensity difference $\delta\mathbf{g}$, we define a mapping $\mathbf{r}$ for the intensity residual, which suffices

$$\mathbf{r}(\mathbf{p}) = \mathbf{g}_s(\mathbf{p}) - \mathbf{g}_m(\mathbf{p}) = \delta\mathbf{g} \tag{3.51}$$

If we change the parameter $\mathbf{p}$ with a difference $\delta\mathbf{p}$, we wish to minimize the sum of squares of intensity residual $\mathbf{r}^T\mathbf{r}$. Following the Gauss-Newton optimization, we use the first order Taylor expansion,

$$\mathbf{r}(\mathbf{p} + \delta\mathbf{p}) \approx \mathbf{r}(\mathbf{p}) + \frac{\partial \mathbf{r}}{\partial \mathbf{p}}\delta\mathbf{p} \tag{3.52}$$

After an update of $\mathbf{p}$, we ideally wish $\mathbf{r}$ to be zero, then we get

$$\mathbf{r}(\mathbf{p} + \delta\mathbf{p}) = 0 \Rightarrow \mathbf{r}(\mathbf{p}) = -\frac{\partial \mathbf{r}}{\partial \mathbf{p}}\delta\mathbf{p} \tag{3.53}$$

where $\partial\mathbf{r}/\partial\mathbf{p}$ is the Jacobian at $\mathbf{p}$. From equation 3.53, given $\mathbf{r}(\mathbf{p})$ i.e. the intensity difference $\delta\mathbf{g}$, we can calculate the optimal parameter update $\delta\mathbf{p}$ with parameter

[1] Model search will be also denoted as model fitting, or simply fitting.

[2] Here we only use quaternions multiplied by scale to denote the rotation and scaling, the full rotation matrix can be recovered using the formulas discussed in shape alignment.

offset to intensity displacement matrix, which can be denoted as $\mathbf{A}$, namely,

$$\delta\mathbf{p} = -\mathbf{A}\mathbf{r}(\mathbf{p}) \tag{3.54}$$

where $\mathbf{A}$ can calculated using the Mooore-Penrose pseudo-inverse, as

$$\mathbf{A} = \left(\frac{\partial\mathbf{r}}{\partial\mathbf{p}}\right)^{\dagger} = \left[\left(\frac{\partial\mathbf{r}}{\partial\mathbf{p}}\right)^{T}\left(\frac{\partial\mathbf{r}}{\partial\mathbf{p}}\right)\right]^{-1}\left(\frac{\partial\mathbf{r}}{\partial\mathbf{p}}\right)^{T} \tag{3.55}$$

where columns of $\partial\mathbf{r}/\partial\mathbf{p}$ are linearly independent. Instead of recalculating $\partial\mathbf{r}/\partial\mathbf{p}$ at every searching step, it can be computed offline using numeric differentiation from the training set. Actually when the AAM was first introduced [69] , a multivariate linear regression (MLR) approach was applied to calculate the linear transform matrix $\mathbf{A}$, which was interpreted further in [68], where it built the parameter and intensity displacement matrices and used a principal component regression approach. This MLR approach is slower than the numeric differentiation method, but both of them can deduce a similar result.

Specifically, for an instance S from the training set, given the parameter $\mathbf{p}$, we can extract the appearance model parameter $\mathbf{c}$, translation $\mathbf{t}$ and rotation quaternions with scale $\mathbf{q}$. Then, we can calculate the shape model parameters $\mathbf{b}_s$ and texture parameters $\mathbf{b}_g$ by

$$\begin{bmatrix}\mathbf{W}_s\mathbf{b}_s \\ \mathbf{b}_g\end{bmatrix} = \overline{\mathbf{b}} + \mathbf{Q}\mathbf{c} \tag{3.56}$$

With the shape model parameters $\mathbf{b}_s$, we can calculate the shape point instance $\mathbf{x}$ by $\mathbf{x} = \overline{\mathbf{x}} + \mathbf{P}_s\mathbf{b}_s$. Then we align the shape inversely to the real coordinate $\mathcal{T}^{-1}(\mathbf{x})$, warp the image within the inversely aligned shape points to *the mapped base points* and normalize it to get the texture instance $\mathbf{g}_s$. Then, we use the sampled intensity vector $\mathbf{g}_s$ to find the model parameter $\mathbf{c}'$ by

$$\mathbf{c}' = \mathbf{Q}^T\left(\begin{bmatrix}\mathbf{W}_s\mathbf{P}_s^T(\mathbf{x} - \overline{\mathbf{x}}) \\ \mathbf{P}_g^T(\mathbf{g}_s - \overline{\mathbf{g}})\end{bmatrix} - \overline{\mathbf{b}}\right) \tag{3.57}$$

Then by applying

$$\begin{bmatrix}\mathbf{W}_s\mathbf{b'}_s \\ \mathbf{b'}_g\end{bmatrix} = \overline{\mathbf{b}} + \mathbf{Q}\mathbf{c}' \tag{3.58}$$

we can get the modeled intensity $\mathbf{g}_m = \overline{\mathbf{g}} + \mathbf{P}_g\mathbf{b'}_g$

If we give a small offset $\delta\sigma_i$ to the i^{th} parameter p_i in parameters $\mathbf{p}$ from the training set S, where δ is a small value between -0.5 and 0.5, and σ_i is the standard deviation of one parameter p_i (for appearance model parameters $\mathbf{c}$, $\sigma_i = \sqrt{\lambda_i}$, and for pose parameters it is set to 1), we can get the permuted parameters $\dot{\mathbf{p}}$ where $\dot{p}_i = p_i + \delta\sigma_i$. Then we warp the image within the inversely aligned shape

points of $\dot{\mathbf{x}}$ (i.e. $\mathcal{T}^{-1}(\dot{\mathbf{x}})$) derived from the permuted parameters $\dot{\mathbf{p}}$ to the *mapped base points* obtaining the sampled texture $\mathbf{g'}_s$. Similar with equation 3.57 and 3.58, we convert the sampled texture back to modeled texture $\mathbf{g'}_m$.

Therefore, we can permute each component in parameter $\mathbf{p}$ around the known optimal value, and get the instensity displacement to parameter offset ratio (with respect to $p_i \in \mathbf{p}$) by

$$\frac{\partial \mathbf{r}}{\partial \mathbf{p}_i} = \frac{\mathbf{r}(\mathbf{p} + \delta\sigma_i) - \mathbf{r}(\mathbf{p})}{\delta\sigma_i} w_i' \approx (\mathbf{g'}_s - \mathbf{g'}_m) w_i \tag{3.59}$$

where w_i is weight factor $w_i = exp\left(-\delta^2/(2\sigma_i^2)\right)/\delta$ and p_i is the i^{th} parameter. In practice, δ is taken symmetrically around 0 with ± 0.5 in turn to get different $\partial\mathbf{r}/\partial\mathbf{p}_i$ and we assign it a mean value. For all n_s images, we calculate the j^{th} ratio $\partial\mathbf{r}_j/\partial\mathbf{p}$ where $j = 1, 2, \cdots, n_s$ and get the mean by

$$\frac{\partial \mathbf{r}}{\partial \mathbf{p}} = \frac{1}{n_s} \sum_{j=1}^{n_s} \frac{\partial \mathbf{r}_j}{\partial \mathbf{p}} \tag{3.60}$$

Afterwards the linear transform matrix $\mathbf{A}$ can be obtained by equation 3.55.

3.3.2 Parameter Optimization

Given a new image, we will apply an iterative approach to optimize the displacement of parameters $\delta\mathbf{p}$ for minimizing difference between the sampled and the modeled textures. The following steps show the procedure.

(1) Begin with the mean appearance parameter $\mathbf{c} = \overline{\mathbf{c}}$, . Given the parameter number (mode number for shape and texture) t_s and t_g from the training set, extract the shape parameter $\mathbf{b}_s$ and $\mathbf{b}_g$ with the help of $\mathbf{W}_s$. Then we can give an initial shape points by $\mathbf{x}_s = \mathcal{T}^{-1}(\overline{\mathbf{x}} + \mathbf{P}_s\mathbf{b}_s)$. The inverse transformation $\mathcal{T}^{-1}$ uses the mean pose parameters obtained in the mean shape alignment.

(2) Find the initial parameter $\mathbf{p}$. Given the initial shape $\mathbf{x}_s$, align $\mathbf{x}_s$ to $\mathbf{x} = \mathcal{T}(\mathbf{x}_s)$ and get the pose parameters $(\mathbf{t}^T, \mathbf{q}^T)^T$. Meanwhile, warp the image with respect to *mapped base points* to exact the normalized texture $\mathbf{g}_s$ within $\mathbf{x}_s$. Use the equation 3.57 to get the appearance model parameter $\mathbf{c}$. Constrain $\mathbf{c}$ within $\pm 3\sqrt{\lambda_i^{(c)}}$ and form the parameter $\mathbf{p} = (\mathbf{c}^T, \mathbf{t}^T, \mathbf{q}^T)^T$.

(3) Start the parameter update iteratively. The pseudocode is

   (I) Initialize the damping factor $k = 1$, the maximum iteration number N_{iter}, parameter $\mathbf{p}_o$. Set the initial intensity difference magnitude $E_o = \infty$.

   (II) **for** $iter = 1 : N_s$

      (i) Split the parameter $\mathbf{p}$ into appearance model parameter $\mathbf{c}$, the pose parameters $(\mathbf{t}, \mathbf{q})$.

(ii) Apply equation 3.56 to calculate the shape parameter $\mathbf{b}_s$ and then get the aligned shape points $\mathbf{x} = \overline{\mathbf{x}} + \mathbf{P}_s\mathbf{b}_s$, then calculate the real position in image by $\mathbf{x}_s = \mathcal{T}^{-1}(\mathbf{x})$, where the transformation $\mathcal{T}^{-1}$ is constructed by the pose parameters $(\mathbf{t}, \mathbf{q})$.

(iii) Warp the image to *mapped based points* to obtain the sampled normalized texture $\mathbf{g}_s$ within $\mathbf{x}_s$.

(iv) Apply equation 3.57 to get a new appearance model parameter $\mathbf{c}'$ and apply equation 3.58 to get the texture parameter $\mathbf{b}'_g$ then calculate the modeled texture by $\mathbf{g}_m = \overline{\mathbf{g}} + \mathbf{P}_g\mathbf{b}'_g$.

(v) Evaluate the intensity difference $\delta\mathbf{g} = \mathbf{g}_s - \mathbf{g}_m$ and its magnitude $E = \delta\mathbf{g}^T\delta\mathbf{g}$.

(vi) **if** $E > E_o$

Update damping factor $k \leftarrow 0.9k$, the parameter $\mathbf{p} \leftarrow \mathbf{p}_o$, and found shape points $\mathbf{x}_s \leftarrow \mathbf{x}_o$;

else

$k \leftarrow 1.1k$, $E_o \leftarrow E$.

end if

(vii) Update the parameter $\mathbf{p}_o \leftarrow \mathbf{p}$ and found shape points $\mathbf{x}_o \leftarrow \mathbf{x}_s$.

(viii) Permute the parameter $\mathbf{p}$ with the predicted offset $\delta\mathbf{p} = -\mathbf{A}\delta\mathbf{g}$, by $\mathbf{p} \leftarrow \mathbf{p} + k \cdot \delta\mathbf{p}$ and constrain the updated $\mathbf{p}$ within $\pm 3\sigma_i$.

end for

(III) **if** $E > E_o$,

update the parameter $\mathbf{p} \leftarrow \mathbf{p}_o$, the found shape points $\mathbf{x}_s \leftarrow \mathbf{x}_o$ and the intensity difference magnitude $E \leftarrow E_o$.

end if

(IV) Optimize the parameter $\mathbf{p}$ using the local minimization algorithm regarding the intensity difference E.

(4) Once the optimal $\mathbf{p}$ is found, execute step (i) to (iv) to get the sampled texture $\mathbf{g}_s$, the modeled texture $\mathbf{g}_m$, and the found shape points $\mathbf{x}_s$.

(5) Realign the texture vector $\mathbf{g}_m$ with respect to the voxel template, and inversely normalize the intensity, warp the image with shape points of *mapped base points* and texture of realigned $\mathbf{g}_m$, with respect to the found shape points $\mathbf{x}_s$. Then we can get the segmented image.

(6) Use the technique of obtaining the voxel template, we can get the discrete voxel based segmentation.

This standard model fitting is based on the linear modeling, which makes it only valid for the object locating in the neighborhood. Besides, this approach requires a full image in which all the voxels inside the shape can be reached to calculate the texture information. Actually there are more approaches for AAM search approaches, and we will discuss them in the latter chapter as well as the method on the condition when only three 2D slices are available.

Chapter 4

Shape Correspondence

Since we choose to use a typical statistical shape model, AAM, to learn the appearance including shape and texture variation, two major important issues should be considered: how to find the appropriate landmark points in a 3D shape. And how to build the correspondence between the landmarks in different shapes from the training set. In this chapter, we aim to solve these problems by a combined approach. Firstly, before introducing the approach, we will reveal the correspondence problem. Then, we will briefly review the approaches of correspondence establishment. Afterwards, the parameterization based group-wise MDL approach will be interpreted in details.

4.1 Correspondence Problem

As shown in the previous chapter, good shape variation modeling must rely on a good landmark selection and a good correspondence establishment. It means, first, the landmarks should be chosen carefully considering the geometric features and mathematical meanings. Second, the landmarks from different shape vectors at the same position should represent the same feature in the objects, namely have the correct correspondence. If there is an inappropriate correspondence, for instance, it will lead some 'illegal' representation, which means the produced shape variation modes cannot interpret the actual object variation. Figure 4.1 shows an example of landmark correspondence. The colored dots are landmarks manually labeled on the contours of bolts. The first and second shape have a consistent correspondence, but the third one has not, where the points with same color indicates the correspondence.

With the bad correspondence, the shape alignment will go wrong. Even though the shapes are aligned when small deviation happens, the landmarks will be scattered widely, which makes the allowable shape domain larger. All of these will affect the PCA and bad principal axes will be generated, resulting in bad models.

In 2D images, manually labeling landmarks is feasible only for small amount of points and images and also for obvious visible features. In 3D images, it is

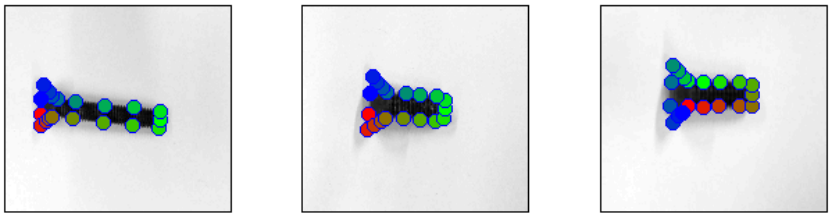


Figure 4.1: Left and Middle: The shape landmarks of bolts have consistent correspondence. Right: The landmarks lose consistent correspondence with the left one.

impossible to manually place the landmarks, because it is difficult to define the reproducible points in smooth surface and locate the points with visualization. In addition, manual landmarking is tedious and time-consuming. ASM and AAM would help to accomplish it. However these method are based on statistic prior knowledge. Therefore an interpretation of images for training is needed and these training images should be segmented manually beforehand. For medical images, a specialist should help to segment the shapes. In obtaining the training set, it is essential to get the manual segmentation of objects, which can be represented as a binary matrix where 1 stands for the pixels of object and 0 is the background. In a 3D case, each slice image in 2D will be labeled manually to get a 2D segmentation. All the binary slices will be in a stack to form a 3D volume. To construct the surface, iso-surface extraction using the marching cube algorithm [70] could be applied. Accordingly, the vertices and mesh faces will be constructed and the landmarks can be a subset of these vertices. Next we should consider (1) how to select the landmarks among candidate points, then establish the correspondence, and (2) how to manipulate the landmarks to optimize the correspondence.

4.2 Common Approahes

The correspondence establishment can be divided into many categories in terms of different standards, such as manual landmarking, feature-based approach, parameterization based approach, physical properties based method, and image based correspondence [79], or using another categorizing criterion such as the mesh to mesh approach, mesh to volume, volume to volume, parameterization to parameterization approaches etc.[88]. However, all the approaches are not individually applied, they are always combined for a better result. Besides, the correspondence establishment is related largely with image registration methods, both rigid and non-rigid image registration. As a result we will not make an explicit distinction between these methods. Here we just pick some typical methods for a brief review.

Early methods focused more on **rigid registration**. Some were based on equally spacing labeling on a mesh surface to get the landmarks and build the correspondence, such as the method of Bastuscheck *et al.* [72]. They used fast Fourier transform to determine the least square difference between the sequences of points

sampled in equal spacing on the 3D curves. Then they used the returned rotation and translation to register the curves. Besl *et al.*[71] create the iterative closest points (ICP) algorithm to align the unlabelled points together by means of minimizing the distances between pair of points iteratively with similarity transformation. This algorithm also works for surfaces with potential different number of vertices before the initialization of building the correspondence. Many derived simpler algorithms and modified algorithms have been implemented. However, because of the restrictions to these algorithms using similarity transformation and based on distance between the equivalent points, these approaches are not suitable for a non-rigid registration.

For **non-rigid registration**, Bookstein [76] proposed thin-plate spline (TPS). Soon Bookstein [77] used it to adjust the landmarks along the tangential directions of the shape contour to minimize the correspondence error defined by thin-plate bending energy to control the correspondence. Richardson *et al.* [78] improved this method with sliding, inserting and deleting landmarks to minimize the landmark correspondence error and representation error, and to improve the representation compactness.

Declerck *et al.* [74] and Subsol *et al.* [73] adapted ICP algorithm together with a B-spline transformation to minimize the influence of an erroneous matched points. Similarly, Rueckert *et al.* [75] have proposed the B-splines based free-form deformation, applied on the breast MR images. They applied the global transformation with rigid image registration algorithm such as ICP, but local deformation using B-spline. In contrast to TPS, B-splines are locally controlled, resulting in an efficient computation for large amount of control points. The free-form deformation registration is robust to noise and popular in medical imaging, but not accurately invertible. So for generating a diffeomorphic deformation, a novel interpolating spline, the clamped-plate spline (CPS) has been introduced by Twining *et al.* [85]. This approach can guarantee diffeomorphic warps between pairs of images with a dense pixel to pixel correspondence, which overcomes the drawbacks of B-splines and TPS. Marsland *et al.* [87] has implemented a polyharmonic CPS to perform the non-rigid registration group-wise.

An alternative is based on shape context registration proposed by Belongie *et al.* [81]. This method also allows the input points with different number of vertices. It firstly constructs the feature vector in each data set, then matches these vectors by minimizing the defined cost function. Kroon *et al.* [55] has applied this approach to the correspondence establishment of MR cartilage images and further to the mandibular image registration [56] with the histogram enhancement and B-splines approximation, and so did Maan *et al.* [22].

The approaches above manipulate the correspondence directly on the object surface, no matter if they are based on distance or features. Another group of distinctive methods are based on the **parameterization** of shapes, which manipulate the

correspondence in the parameterization instead of the shape itself. Kelemen *et al.* [82] proposed a spherical harmonics (SPHARM) representation of shape surface providing the parametric descriptions of object shapes, where the coefficients of a parameterized basis function are fit to surface. Then the correspondence of object shapes are consistent with the parameterization. The parameterizations are developed mainly based on the work of Brechbühler *et al.* [83] or the developed version with a hierarchical optimization by Quicken *et al.* [84]. These methods were later used for the initialization of correspondence. Davies *et al.* [86] used the spherical parameterization to initialize the correspondence, and optimized them using a cauchy kernel representation with respect to minimize the description length. Heimann *et al.* [88] used a conformal mapping based on the spherical parameterization and minimum description length (MDL) as the objective function[1] with a gradient descent optimization to evaluate the correspondence. Davies *et al.* [89] have also proposed an automatic model building by direct optimization of description length where the CPS was applied for manipulating the parameterization[2]. Kirschner *et al.* [90] suggested to use the ICP alignment before parameterization, and to remove the area distortion [91] to correct and refine the parameterization for correspondence. They also investigated the work from Davies *et al.* and made a comparison. Vincent *et al.* have adopted the parameterization based group-wise MDL registration to build the correspondence of landmarks to utilize the active appearance models for cartilage [54] and prostate [23] segmentation respectively.

Some approaches above are performed pairwise, such as the ICP, B-splines based free-form deformation [75], shape context approach [81] etc. But some are group-wise performed, such as the polyharmonic CPS non-rigid registration [87] and MDL based approaches [86]. Actually, the pairwise algorithm is problematic for some cases, even though most of them work fine. From a theoretical viewpoint, the match between pair of shapes needs to be determined within the wider context of the group [79], since some ambiguity can be removed by the information introduced from the group instead of the pair. The group-wise registration mainly differs in the objective function where the global information would be considered. The group-wise objective functions are featured to be based on the models, such as the determinant of the model covariance, or based on the information theoretical measurement such as MDL. Because we apply the AAM which requires a consistent correspondence across the data set, a group-wise registration algorithm for correspondence establishment and optimization would be preferable.

The correspondence optimization can be regarded as an optimization problem since it can be seen that the aim of the approaches above are to minimize some objective functions. Therefore, we can generalize the framework of correspondence optimization into three phases [79]: manipulating the correspondence, calculating

[1] MDL is similar with the cost function, energy function, erroneous function or distance function in the former methods. This is also used to evaluate the shape correspondence

[2] The parameterization as a meshed sphere can also be manipulated using CPS, so all the methods for manipulating the shape mesh can also be applied to control the parameterization mesh

the objective functions, and optimizing the correspondence. Figure 4.2 illustrates this framework. This flow can be conducted iteratively until we get a satisfying result. There are multiple choices of approaches in each phase. For instance, CPS, symmetric theta transformation can both achieve the correspondence manipulation; either MDL or the model covariance determinant can be the objective function.

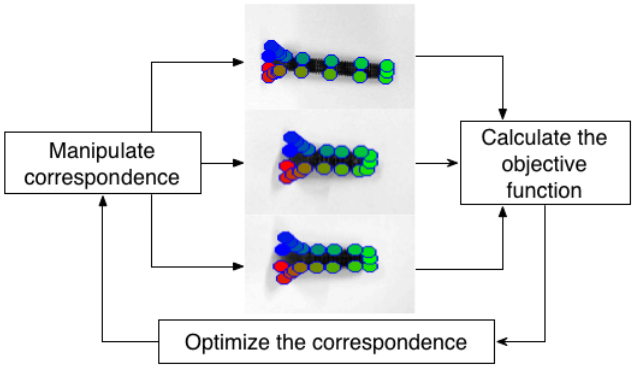


Figure 4.2: Framework for correspondence optimization

We choose the group-wise correspondence optimization based on parameterization of shapes with the MDL objective function. The following sections will interpret the whole procedure in detail.

4.3 Correspondence Establishment and Optimization

Unlike the shape context and B-splines manipulating the shape vertices directly, the basic idea is mapping all the shapes to a spherical space, where each point on the sphere corresponds to a vertex in the object shape. This approach performs better for objects with spherical topology, hereby it is suitable for the prostate. An initial correspondence can be established by means of propagating the consistent landmarks over all the shapes and their parameterizations (i.e. the points on sphere). Furthermore, these parameterized landmark points can be manipulated to minimize the MDL objective function, to select the optimum model which has the optimum correspondence. The previous works from Davies *et al.* [79, 86, 89] and Kirschner *et al.* [90, 91] are fundamental to this section.

4.3.1 Spherical Parameterization and Re-parameterization

Because we use the spherical parameterization based representation of shape points [83], some concepts and definitions should be clarified. We define a single shape $S \in \mathbb{R}^3$ which contains n_v vertices with Cartesian coordinates

$$\mathcal{V} = \{\mathbf{v}_i = (x_i, y_i, z_i) | i = 1, 2, \cdots, n_v\} \tag{4.1}$$

and the topology of mesh faces

$$\mathcal{F} = \{\mathbf{t}^\alpha = (\mathbf{v}_A^\alpha, \mathbf{v}_B^\alpha, \mathbf{v}_C^\alpha) | \mathbf{v}_A^\alpha, \mathbf{v}_B^\alpha, \mathbf{v}_C^\alpha \in \mathcal{V}\} \tag{4.2}$$

where $\mathbf{t}^\alpha$ stands for the α triangular face in the surface mesh, and A, B, C are the three vertices indices in $\mathcal{V}$ in the triangle.

Then we define a parameterization space P, where all the points are scattered on the unit spherical surface. These points are vertices of the sphere, which can be denoted using Cartesian coordinates and polar coordinates respectively as

$$\mathcal{X} = \{\mathbf{x}_i = (x_i, y_i, z_i) | x_i^2 + y_i^2 + z_i^2 = 1, i = 1, 2, \cdots, n_v\} \tag{4.3}$$

$$= \{\mathbf{x}_i = (1, \theta_i, \psi_i) | i = 1, 2, \cdots, n_p\} \tag{4.4}$$

If a common topology $\mathcal{F}$ is maintained in both shape space S and parameterization space P, then a mapping $\mathfrak{X}$ from the parameterization space to the shape space can be represented as

$$\mathfrak{X}: \quad P \longmapsto S \quad , \quad \mathcal{V} = \mathfrak{X}(\mathcal{X}) \tag{4.5}$$

The mapping from the shape space to the parameterization space is called the *parameterization*[1], which is a inverse mapping of $\mathfrak{X}$, denoted as $\mathfrak{X}^{-1}$, namely $\mathcal{X} = \mathfrak{X}^{-1}(\mathcal{V})$. An example of relationship between $\mathcal{V}$ and $\mathcal{X}$ is illustrated in Figure 4.3. The tetrahedron represents a object shape and the red nodes on

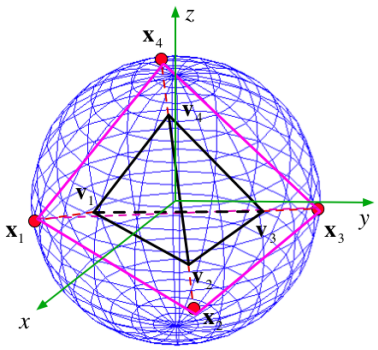


Figure 4.3: An example of the mapping between the shape vertices and parameterization on the spherical surface

the sphere are its parameterization. Now we have a training set with n_s shapes $\{S_i | i = 1, 2, \cdots, n_s\}$, we can parameterize all the shapes to their parameterization, namely

$$\mathcal{X}_i = \mathfrak{X}^{-1}(\mathcal{V}_i) = \{\mathbf{x}_j^{(i)} = \mathfrak{X}^{-1}\left(\mathbf{v}_j^{(i)}\right) | i = 1, 2, \cdots, n_s, j = 1, 2, \cdots, n_v\} \tag{4.6}$$

[1]Parameterization has two meanings here. One is the mapped points on the unit sphere and the other is the mapping from shape space to these unit spherical space.

The *re-parameterization* means to manipulate the parameterized sphere points $\mathcal{X}$ to new positions on the same sphere $\mathcal{X}'$, which can be represented as

$$\mathcal{X}' = \phi(\mathcal{X}) \tag{4.7}$$

If we have a same parameterization $\mathcal{X}$ for all the object shapes $\{\mathcal{V}_i, i = 1, 2, \cdots, n_s\}$, the object shapes will have a consistent correspondence, namely

$$\mathfrak{X}_i(\mathcal{X}) \sim \mathfrak{X}_j(\mathcal{X}) \longmapsto \mathcal{V}_i \sim \mathcal{V}_j \quad i, j \in \{1, 2, \cdots, n_s\} \tag{4.8}$$

4.3.1.1 Initial Parameterization

We follow the methods from Brechbühler *et al.* [83] to parameterize the shapes. In most conditions, shape points $\mathcal{V}$ are represented in the Cartesian coordinates, and unit spherical parameters $\mathcal{X}$ can be represented using Polar coordinates, so we can form the mapping with the help of the geographic coordinates system of the earth, and θ is defined as latitude, and ψ is defined as the longitude, where they suffice $\theta \in [0, \pi]$, $\psi \in [0, 2\pi)$. We take a shape with n_p vertices and topology $\mathcal{F}$ as shown in equation 4.1 and 4.2 as an example, and the steps are interpreted as follows.

(1) Define 'south pole' and 'north pole' in the shape vertices. In a simple way, the vertex $\mathbf{v}_S$ with the smallest z coordinate value is defined as the 'south pole' which is assigned with latitude $\theta_S = 0$, and the vertex $\mathbf{v}_N$ with highest z value and assigned with $\theta_N = \pi$. This way is simple but not sufficient for an object with non-spherical topology, for which a Dijkstra's algorithm can be used for searching the 'north pole'.

(2) Assign all the shape points $\mathbf{v}_i$ with latitude values θ_i. The corresponding continuous problem can be formulated as Laplace's equation $\nabla^2\theta = 0$ with Dirichlet conditions $\theta_S = 0$ and $\theta_N = \pi$. This means the latitude θ_i assigned to the vertices $\mathbf{v}_i$ ($i \in \{1, \cdots, n_v\}$ and it is neither 'north pole' vertex nor 'south pole' vertex) is the average of the latitudes of its neighbor points. Specifically, it can be solved using the following pseudo-codes.

The latitudes can be calculated by solving the equation $\mathbf{A}\boldsymbol{\theta} = \mathbf{b}$, where $\mathbf{A}$ is a $n_v \times n_v$ matrix and both $\mathbf{b}$ and $\boldsymbol{\theta}$ are $n_v \times 1$ column vectors. The steps for constructing $\mathbf{A} = \{a_{i,j} | i, j = 1, 2, \cdots, n_v\}$ as

(I) Initialize $\mathbf{A}$ as a zero matrix, namely $a_{i,j} = 0$ for $i, j \in \{1, 2, \cdots, n_v\}$.

(II) **for** i from 1 to n_v

  (i) **If** the i^{th} vertex $\mathbf{v}_i$ in $\mathcal{V}$ is $\mathbf{v}_S$ or $\mathbf{v}_N$, set $a_{i,i}$ to 1;
  otherwise, set $a_{i,i}$ with the number of direct neighbor vertices, which can be obtained from $\mathcal{F}$.

  (ii) **for** j from 1 to n_v where $j \neq i$ & $\mathbf{v}_i \neq \mathbf{v}_S$ and $\mathbf{v}_i \neq \mathbf{v}_N$
  If the vertex $\mathbf{v}_j$ is the direct neighbor vertex of $\mathbf{v}_i$, set $a_{i,j} = -1$;
  otherwise set $a_{i,j} = 0$.

For constructing $\mathbf{b} = \{b_i | i = 1, 2, \cdots, n_v\}$

(I) Initialize $\mathbf{b}$ as a zero vector, namely $b_i = 0$ for $i = 1, 2, \cdots, n_v$.

(II) **for** i where $\mathbf{v}_i = \mathbf{v}_N$, set $b_i = \pi$.

The latitude values $\boldsymbol{\theta} = \{\theta_i | i = 1, 2, \cdots, n_v\}$ can be obtained by $\boldsymbol{\theta} = \mathbf{A}^{-1}\mathbf{b}$.

---

(3) Determine the meridian line[1]. We define the current candidate meridian vertex is $\mathbf{v}_c$. Among the latitude value $\{\theta_i\}$ where $\{\mathbf{v}_i\}$ are the direct neighbor vertices of $\mathbf{v}_c$, find the minimal latitude $\theta_{min} \in \{\theta_i\}$, then the vertex $\mathbf{v}_{min}$ is the next meridian point candidate, then perform this procedure again. We start from the 'north pole' $\mathbf{v}_N$ and until it reaches the 'south pole' $\mathbf{v}_s$. Then we will have a meridian vertex sequence $\{\mathbf{v}_{n_k} | k = 1, 2, \cdots, m, \mathbf{v}_{n_1} = \mathbf{v}_N, \mathbf{v}_{n_m} = \mathbf{v}_S\}$, where m is the meridian vertices number.

(4) Assign all the shape points $\mathbf{v}_i$ with longitude values ψ_i. First we dismiss the connections to the poles. Similar with the latitude, the longitude value for a vertex $\mathbf{v}_i$ is the average of the longitudes of its neighbor vertices. But there is a leap of 2π from the east to the west. So we have to define whether the vertices connected to the meridian points are western vertices or eastern ones. Then for the meridian points, every western neighbor connection should be added 2π and for the western points, each meridian neighbor connection should be subtracted 2π. We also have to define the zero value to a meridian point. The detailed pseudo-codes is as follows:

Similarly with calculating the latitude, longitude values will be also derived from a modified equation $\mathbf{A}'\boldsymbol{\psi} = \mathbf{b}'$. For getting $\mathbf{A}'$, it can be done by modifying $\mathbf{A}$ in step (2).

(I) Initialize $\mathbf{A}'$ with $\mathbf{A}$, namely $\mathbf{A}' = \mathbf{A}$, $a'_{i,j} = a_{i,j}$, for $i, j \in \{1, 2, \cdots, n_v\}$.

(II) **for** i from 1 to n_v where $\mathbf{v}_i \neq \mathbf{v}_N$ & $\mathbf{v}_i \neq \mathbf{v}_S$,

  (i) **if** $\mathbf{v}_S \in \{\mathbf{v}^{(i)}_{neighbor}\}$ or $\mathbf{v}_N \in \{\mathbf{v}^{(i)}_{neighbor}\}$, where $\{\mathbf{v}^{(i)}_{neighbor}\}$ are the direct neighbor vertices of $\mathbf{v}_i$,

    set $a'_{i,i} = a_{i,i} - 1$.

(III) **for** $i = n_2$, where $\mathbf{v}_{n_2}$ is the meridian point next to $\mathbf{v}_N$,

  (i) set $a'_{i,i} = a_{i,i} + 2$

(IV) **for** i, where $\mathbf{v}_i = \mathbf{v}_N$ or $\mathbf{v}_i = \mathbf{v}_S$,

  (i) **for** j from 1 to n_p

    set $a'_{i,j} = 0$ and $a'_{j.i} = 0$.

We should define if the vertex connected to the meridian points is in the west or east before we compute $\mathbf{b}'$. For one meridian vertex $\mathbf{v}_{n_k}$, the two other meridian vertices are $\mathbf{v}_{n_{k-1}}$ and $\mathbf{v}_{n_{k+1}}$. For one direct neighbor vertex

[1]This meridian line is different from the geographic meridian. In the geographic definition, longitude increases to the east until π, and decreases to the west until $-\pi$; here we define the longitude ψ increases from 0 to 2π along the east direction.

$\mathbf{v}$ to the $\mathbf{v}_{n_k}$, we define $\mathbf{v}$ as a western point sufficing the one of following conditions.

- Both the three vertices of triangle $\{\mathbf{v},\ \mathbf{v}_{n_{k-1}},\ \mathbf{v}_{n_{k+1}}\}$ and the three vertices of triangle $\{\mathbf{v}, \mathbf{v}_{n_{k-1}}, \mathbf{v}_{n_k}\}$ are clockwise[1] placed.
- Both the three vertices of triangle $\{\mathbf{v},\ \mathbf{v}_{n_{k-1}},\ \mathbf{v}_{n_{k+1}}\}$ and the three vertices of triangle $\{\mathbf{v}, \mathbf{v}_{n_k}, \mathbf{v}_{n_{k+1}}\}$ are clockwise placed.
- Both the three vertices of triangle $\{\mathbf{v},\ \mathbf{v}_{n_k},\ \mathbf{v}_{n_{k+1}}\}$ and the three vertices of triangle $\{\mathbf{v}, \mathbf{v}_{n_{k-1}}, \mathbf{v}_{n_k}\}$ are clockwise placed.

Then for constructing $\mathbf{b}'$, the steps are

(I) Initialize $\mathbf{b}'$ as a zero vector, namely $b'_i = 0$ for $i = 1, 2, \cdots, n_v$.

(II) **for** k from 2 to $m-1$, namely n_k from n_2 to n_{m-1}, assuming there are m meridian points as in step (3), and $\mathbf{v}_{n_1} = \mathbf{v}_N$,$\mathbf{v}_{n_m} = \mathbf{v}_S$.

(i) **for** every j where $\mathbf{v}_j \in \{\mathbf{v}^{(n_k)}_{neighbor}\}$ & $\mathbf{v}_j \neq \mathbf{v}_{n_{k-1}}$ & $\mathbf{v}_j \neq \mathbf{v}_{n_{k+1}}$
($\{\mathbf{v}^{(n_k)}_{neighbor}\}$ are the set of direct neighbor points of $\mathbf{v}_{n_k}$)
if vertex $\mathbf{v}_j$ is a western point
$b'_j \leftarrow b'_j + 2\pi$
$b'_{n_k} \leftarrow b'_{n_k} - 2\pi$

(III) Applying $\mathbf{A}'$ and $\mathbf{b}'$, the longitude values $\boldsymbol{\psi} = \{\psi_i | i = 1, 2, \cdots, n_v\}$ can be obtained by $\boldsymbol{\psi} = \mathbf{A'}^{-1}\mathbf{b}'$. The longitude values in both poles are implicitly assigned to 0.

---

(5) From the preceding steps the parameterization can be written in polar coordinates and Cartesian coordinates as

$$\mathcal{X} = \{\mathbf{x}_i = (1, \theta_i, \psi_i) | i = 1, 2, \cdots, n_v\} \tag{4.9}$$
$$= \{\mathbf{x}_i = (\sin\theta_i \cos\psi_i, \sin\theta_i \sin\psi_i, -\cos\theta_i) | i = 1, 2, \cdots, n_v\} \tag{4.10}$$

This procedure will yield a unique solution for the initial parameterization. An example of initial parameterization of the prostate is shown in Figure 4.4.

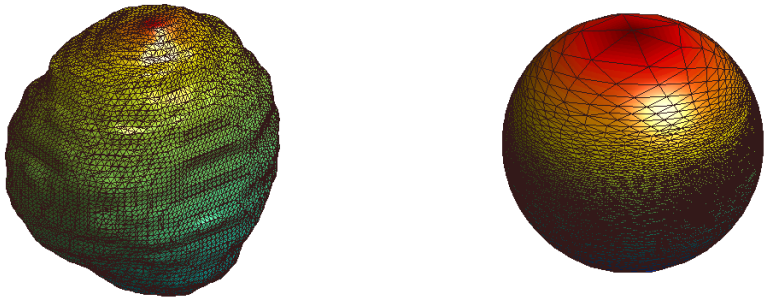
4.3.1.2 Re-parameterization Approaches

Because we want to manipulate the parameterization to change the correspondence, a re-parameterization approach should be considered. Actually there were many re-parameterizations introduced in [79], but we mainly use two approaches.

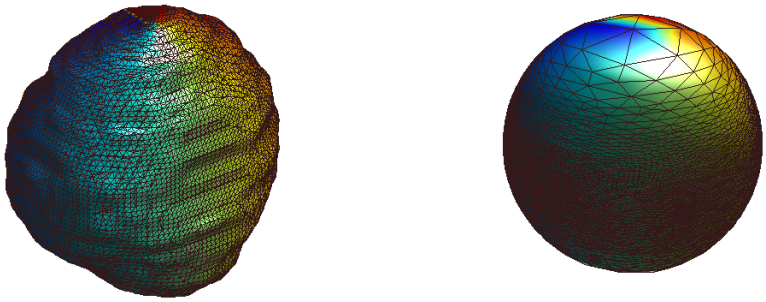
1. Sigmoid Transformation

This approach has been proposed in [91] for a fast initial area preservation. This transformation only affects on the latitude dimension, so the clusters of sphere nodes will be dissolved in latitude directions. If we use the polar coordinates

[1]The viewing point should be outside the object surface.



(a) The example of a prostate shape and parameterization, where color going from blue to red is along the increment of latitude.



(b) The example of a prostate shape and parameterization, where color going from blue to red is along the increment of longitude.

Figure 4.4: The initial parameterization example

in parameterization as $\mathcal{X} = \{\mathbf{x}_i = (1, \theta_i, \psi_i) | i = 1, 2, \cdots, n_v\}$, the transformed parameterization can be represented as

$$\mathcal{X}' = \phi(\mathcal{X}) = \{\mathbf{x}'_i = (1, f(\theta_i), \psi_i) | i = 1, 2, \cdots, n_v\} \tag{4.11}$$

where

$$f(\theta_i) = \frac{\pi}{1 + e^{-\alpha(\theta_i - \theta_{ip})}} \tag{4.12}$$

In equation 4.12, α is a weight scalar, and θ_{ip} is the sigmoid inflection point. When $\theta_i > \theta_{ip}$, the parameterization point will be mapped to the northern hemisphere, and when $\theta_i < \theta_{ip}$, it will be to the southern hemisphere. Normally, we take the mean value, namely,

$$\theta_{ip} = \frac{1}{n_p} \sum_{i=1}^{n_v} \theta_i \tag{4.13}$$

This re-parameterization transformation is particularly for a fast spread of the sphere nodes to the to poles. From the examples in Figure 4.4, it can be seen that the nodes in initial parameterizations are clustered near the 'equator'. Therefore this transformation can make the parameterization more uniformly distribute on

the sphere.

2. Symmetric Theta Transformation

The symmetric theta transformation is one basic re-parameterization method based on the Cauchy distribution. The idea is to smoothly adjust the points along the series of great circles (like the longitude circles) intersecting at the kernel and its antipodal points on the sphere. The geometric interpolation is based on the Cauchy distribution and a sum of probability density functions will be used for a general case. First we consider when the kernel center is at the 'south pole' where $\theta = 0$, then the re-parameterization will take place on θ only. The re-parameteriation function [79] is defined as

$$f(\theta) = \frac{1}{1+A}\left(\theta + A \arccos\left(\frac{(1+\Omega^2)\cos\theta - 2\Omega}{1+\Omega^2 - 2\Omega\cos\theta}\right)\right) \tag{4.14}$$

where $\Omega \in (0,1)$ is the width parameter and $A > 0$ is the amplitude of the wrapped Cauchy. At the kernel center and its antipodal point, the values do not change, namely when kernel center is at $\theta = 0$, then $f(\theta) \equiv 0$ and $f(\pi) \equiv \pi$.

For an arbitrary kernel center, the equation 4.14 should be changed. Supposing we have the kernel center defined in the Cartesian coordinates as $\mathbf{a} \in \mathbb{R}^3$ where $|\mathbf{a}|^2 = 1$, we will have the representation of transformation for an arbitrary point $\mathbf{x}$ defined also in Cartesian coordinates on the sphere as

$$\cos\theta' = \mathbf{a} \cdot \mathbf{x} \tag{4.15}$$

$$\mathbf{x}' = \mathbf{a}\cos f(\theta') + \frac{\sin f(\theta')}{\sin\theta'}(\mathbf{x} - \mathbf{a}\cos\theta') \tag{4.16}$$

where $\mathbf{x}'$ is the new position where $\mathbf{x}$ will move after the transformation. Given a fixed width, we can get different re-parameterizations by adjusting the amplitude A. In practice, multiple kernels would be selected to perform the re-parameterization for a better result.

There are also similar transformations such as asymmetric theta transformations and shear transformations [79], which provide more complicated patterns to control the parameterizations. In practice, the symmetric theta transformation will be sufficient. Figure 4.5 shows examples of sigmoid transformation and the symmetric theta transformation, where we set α in the sigmoid transformation to 5, the amplitude, width in the symmetric theta transformation are 4 and 0.25 respectively, the kernel center is at the 'north pole' (i.e. $[0, 0, 1]$).

4.3.1.3 Area Distortion Removal

It has been indicated that the parameters are gathered around the 'equator', but rarely cover around 'poles', which is a major drawback using such a mapping. If we use a uniform distributed vertices to sample the spherical surface, there

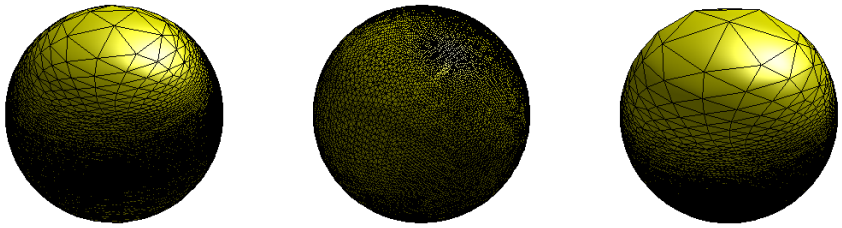


Figure 4.5: The original parameterization(Left), the re-parameterizations after sigmoid transformation (middle) and after symmetric theta transformation.

are considerable areal distortions. One solution is to use re-parameterization approaches to remove the area distortions for area preservation. By means of re-parameterizing the nodes on the sphere, we try to make each triangular face have a similar area proportion among all the faces, with the area proportion of the corresponding triangular face in the object shape mesh. To be specific, we define the α^{th} triangular face in the object shape mesh is $\mathbf{t}_{\mathbf{v}}^{\alpha} = (\mathbf{v}_A^{\alpha}, \mathbf{v}_B^{\alpha}, \mathbf{v}_C^{\alpha})$ and that in the spherical parameter mesh is $\mathbf{t}_{\mathbf{x}}^{\alpha} = (\mathbf{x}_A^{\alpha}, \mathbf{x}_B^{\alpha}, \mathbf{x}_C^{\alpha})$, and there are β triangular faces. So we can write the objective function as

$$\mathcal{L} = \sum_{\alpha=1}^{\beta} \left(\frac{Area(\mathbf{t}_{\mathbf{v}}^{\alpha})}{\sum_{\alpha=1}^{\beta} Area(\mathbf{t}_{\mathbf{v}}^{\alpha})} - \frac{Area(\mathbf{t}_{\mathbf{x}}^{\alpha})}{\sum_{\alpha=1}^{\beta} Area(\mathbf{t}_{\mathbf{x}}^{\alpha})} \right)^2 \tag{4.17}$$

where $Area(\mathbf{v}^{\alpha})$ means area of triangular face $\mathbf{t}^{\alpha}$. Given the objective function, we can use the representation functions to adjust the parameters on the sphere for minimizing $\mathcal{L}$. Similar with [91], we use two stages to reduce the area distortion. We do not use the refinement in [91] since the first two stages are sufficient.

In the first stage, we apply the sigmoid transformation to re-arrange the latitude first considering the 'equator clustering' feature of the allocation of the points in the initial parameterization. By using a line searching algorithm, we can find an optimal α in equation 4.12 minimizing the objective function $\mathcal{L}$. In the second stage, we perform the symmetric theta transformation for removing areal distortions. A fixed width Ω is selected based on the case, and several uniformly distributed spherical points are generated as the kernel centers. The uniformly distributed sphere points can be obtained by subdividing the faces of an octahedron or icosahedron into smaller triangles then normalizing the points vectors to the unit sphere[1]. We also use a searching algorithm to find the optimal amplitude A in equation 4.14 that can minimize $\mathcal{L}$. Instead of selecting the nodes merely in the cluster as kernel centers [91], we apply the transformation on all candidate kernel centers iteratively. Finally we will get an area preserved parameterization. The parameterization sphere of the example in Figure 4.4 will become as shown in Figure 4.6 after the area preservation operation.

[1]Illustrations will be shown in the following sections

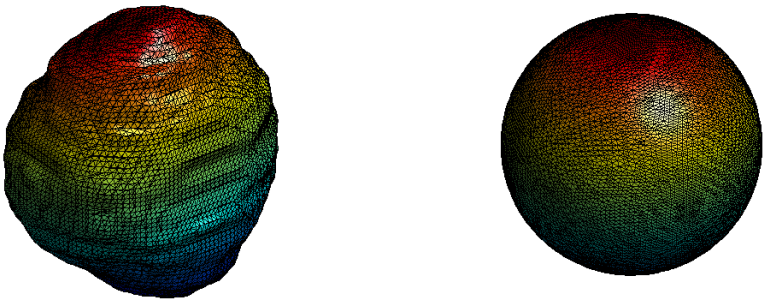


Figure 4.6: The shape mesh and its parameterization mesh after area preservation

4.3.2 Correspondence Initialization

All the processes discussed above are done for a single shape. In order to establish the correspondence, we should operate the training set pairwise or group-wise. Normally, we select one shape as a reference, and then we can propagate the correspondence to the rest shapes in the training set. However, all the training shapes may not have the same number of vertices, and also there are too many to process. As a result, we will decimate the number of vertices, namely select the same number of landmarks during the propagation to get a coarse initial correspondence.

4.3.2.1 Landmark Sampling

After a reference shape is selected, we can sample the landmarks on the reference to give an initial correspondence with respect to the reference for other shapes in the training set. We suppose the reference shape S_{ref} has vertices $\mathcal{V}^o_{ref}$, and the area preserved parameterization $\mathcal{X}^o_{ref}$. We choose to use uniform spherical mesh nodes for sampling the landmarks. It was found that 20% of the original shape nodes was sufficient [79]. In practice, we will decimate the vertices in the reference shape firstly, namely sample $\mathcal{X}^o_{ref}$ and then find the corresponding landmarks on the reference shape. In the following part, we will show a more general case that sampling an arbitrary parameterization $\mathcal{X}^o$ and sampling its original shape $\mathcal{V}^o$.

The uniform nodes on an unit sphere for sampling, denoted as $\mathcal{X} = \{\mathbf{x}_i | i = 1, 2, \cdots, n_p{}^1\}$, can be derived similarly as the kernel centers in the re-parameterization part. The octahedron or icosahedron derived uniform spheres are both available. For instance, we obtain the uniformly distributed sphere nodes by subdividing an icosahedron four times. As a result, there are 2562 uniform sphere nodes as the sampling points/positions. Figure 4.7 shows this process.

We can then sample an arbitrary parameterization $\mathcal{X}^o$ with this uniform nodes

[1] n_p is defined as the number of sampled landmarks, i.e. the node number of the uniformly distributed sphere for sampling. The original vertices number is n_v.

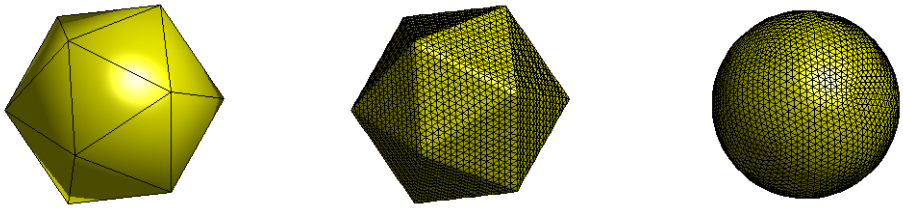


Figure 4.7: The derivation of uniform spherical points from an icosahedron

$\mathcal{X}$. [1] We use Figure 4.8 to illustrate the sampling process. The yellow points in

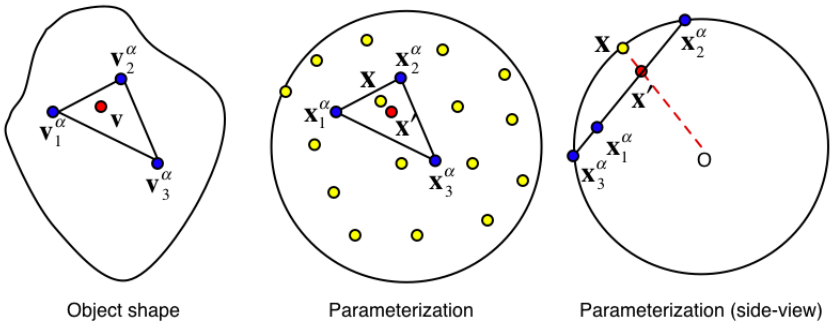


Figure 4.8: The example of landmark sampling

the parameterization refer to the sampling positions $\mathcal{X}$, which are generated from the subdivided octahedron/icosahedron. The red point $\mathbf{x}'$ is the intersection of the line connecting one sampling point $\mathbf{x} \in \mathcal{X}$ and the origin, and the plane defined by the α^{th} triangular face $\mathbf{t}_{\mathbf{x}}^{\alpha} = (\mathbf{x}_A^{\alpha}, \mathbf{x}_B^{\alpha}, \mathbf{x}_C^{\alpha})$ where the nodes $\mathbf{x}_A^{\alpha}, \mathbf{x}_B^{\alpha}, \mathbf{x}_C^{\alpha} \in \mathcal{X}^o$ are blue in the figure. Then we will have three sub-triangular faces $\mathbf{t}_{\mathbf{x}_A}^{\alpha} = (\mathbf{x}', \mathbf{x}_B^{\alpha}, \mathbf{x}_C^{\alpha})$, $\mathbf{t}_{\mathbf{x}_B}^{\alpha} = (\mathbf{x}_A^{\alpha}, \mathbf{x}', \mathbf{x}_C^{\alpha})$, $\mathbf{t}_{\mathbf{x}_C}^{\alpha} = (\mathbf{x}_A^{\alpha}, \mathbf{x}_B^{\alpha}, \mathbf{x}')$. Therefore, the barycentric coordinates $(a^{\alpha}, b^{\alpha}, c^{\alpha})$ of $\mathbf{x}'$ with respect to the triangle $\mathbf{t}_{\mathbf{x}}^{\alpha}$ can be defined as

$$a^{\alpha} = \frac{Area(\mathbf{t}_{\mathbf{x}_A}^{\alpha})}{Area(\mathbf{t}_{\mathbf{x}}^{\alpha})} \quad b^{\alpha} = \frac{Area(\mathbf{t}_{\mathbf{x}_B}^{\alpha})}{Area(\mathbf{t}_{\mathbf{x}}^{\alpha})} \quad c^{\alpha} = \frac{Area(\mathbf{t}_{\mathbf{x}_C}^{\alpha})}{Area(\mathbf{t}_{\mathbf{x}}^{\alpha})} \tag{4.18}$$

where $Area()\dot{}$ calculates the area. If $\mathbf{x}'$ lies in the triangular face $\mathbf{t}_{\mathbf{x}}^{\alpha}$, as Figure 4.8 shows, these barycentric coordinates will suffice

$$\mathbf{x}' = a^{\alpha}\mathbf{x}_A^{\alpha} + b^{\alpha}\mathbf{x}_B^{\alpha} + c^{\alpha}\mathbf{x}_C^{\alpha} \tag{4.19}$$

$$a^{\alpha} + b^{\alpha} + c^{\alpha} = 1 \;\; \& \;\; a^{\alpha}, b^{\alpha}, c^{\alpha} \in [0, 1] \tag{4.20}$$

So the corresponding point of $\mathbf{x}'$ in the object shape is the red node $\mathbf{v}$ surrounded by three shape points in triangle $\mathbf{t}_{\mathbf{v}}^{\alpha} = (\mathbf{v}_A^{\alpha}, \mathbf{v}_B^{\alpha}, \mathbf{x}_C^{\alpha})$. Its position can be determined

[1] Here we are discussing a special case namely a uniform sampling. Actually $\mathcal{X}$ can also be non-uniformly distributed sphere node set.

using the same barycentric coordinates as

$$\mathbf{v} = a^{\alpha}\mathbf{v}_A^{\alpha} + b^{\alpha}\mathbf{v}_B^{\alpha} + c^{\alpha}\mathbf{v}_C^{\alpha} \tag{4.21}$$

Based on this knowledge, we can sample any shape (with points $\mathcal{V}^o$) by sampling its parameterization sphere (with nodes $\mathcal{X}^o$). Firstly we have to find all the barycentric coordinates of $\mathcal{X}$ in $\mathcal{X}^o$. This can be achieved by

(1) finding the triangles in the parameterization $\mathcal{X}^o$ close to $\mathbf{x}_i \in \mathcal{X}$;

(2) calculating the intersection points of ray starting from the origin to $\mathbf{x}_i$, and planes defined by these triangles;

(3) calculating the barycentric coordinates of these intersection points;

(4) selecting the barycentric coordinates and triangles that suffice condition 4.20.

Secondly, we apply the found barycentric coordinates and triangle labels to obtain the landmark vertices in the shape. These sampled landmarks correspond to $\mathcal{X}$, denoted as $\mathcal{V} = \{\mathbf{v}_i | i = 1, 2, \cdots, n_p\}$.

When building the correspondence, we apply the sampling on the reference shape firstly. We use the uniform sampling nodes $\mathcal{X}$ to sample the preserved parameterization $\mathcal{X}^o_{ref}$, then we will get the sampled reference shape with shape landmarks $\mathcal{V}_{ref}$ and parameterization $\mathcal{X}_{ref}(= \mathcal{X})$. The example of sampled reference shape is shown in Figure 4.9.

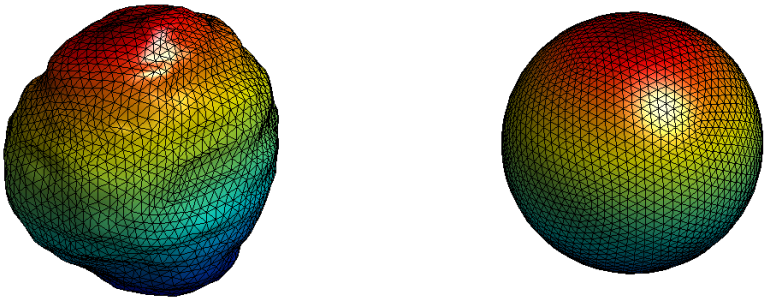


Figure 4.9: The reference shape after uniform landmark sampling and its sampled parameterization

4.3.2.2 Consistent Parameterization Propagation

After the landmark sampling on the reference shape, we would like to have consistent landmarks labeled in the rest shapes so that an initial correspondence can be built. It is assumed that the shapes are aligned[1], so the initial correspondence can be built by finding the shape vertices in $S_i (i \neq ref \ \ \& \ \ i \in \{1, 2, \cdots, n_s\})$ closest to

[1]Before the propagation, the original shapes should be aligned as chapter 3 states. If the shapes have different vertices number, we can apply ICP first instead of the Procrustes alignment

the landmarks in the reference shape. We use Figure 4.10 to illustrate this process. For a single training shape $S_i \neq S_{ref}$, let $\mathcal{V}_i^o = \{\mathbf{v}_i^{o,B} | B = 1, 2, \cdots, n_v\}$ (the green

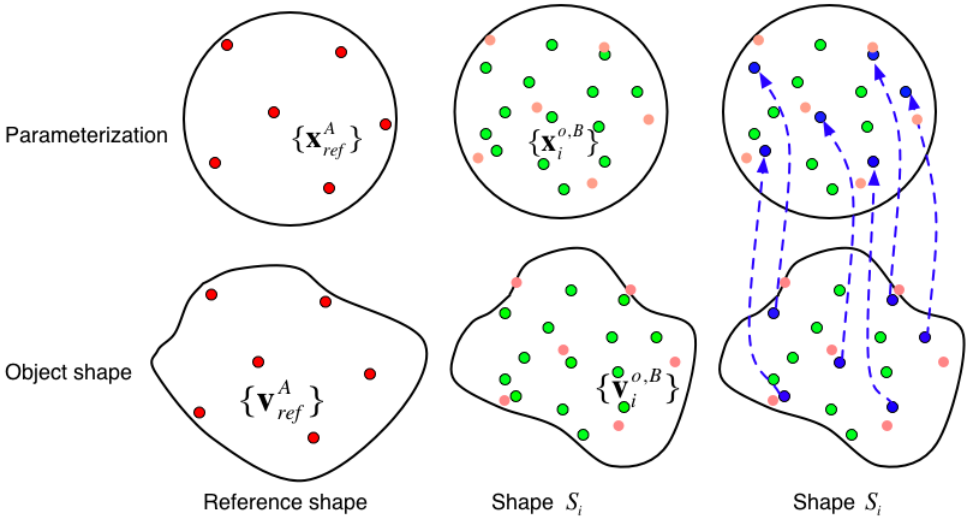


Figure 4.10: Illustration of initial correspondence establishment.

points in the figure) denotes the original shape vertices before sampling, and $\mathcal{X}_i^o = \{\mathbf{x}_i^{o,B} | B = 1, 2, \cdots, n_v\}$ represents the parameterization. $\mathcal{V}_{ref} = \{\mathbf{v}_{ref}^A | A = 1, 2, \cdots, n_p\}$ and $\mathcal{X}_{ref} = \{\mathbf{x}_{ref}^A | A = 1, 2, \cdots, n_p\}$ are the sampled shape vertices (landmarks) and parameterization of the reference shape respectively (the red and the light red points in the figure). The initial correspondence can be defined with respect to the vertex index, written as

$$I_i(A) = B, \text{ where } B = \arg\min \|\mathbf{v}_{ref}^A - \mathbf{v}_i^{o,B}\| \tag{4.22}$$

We then use the re-parameterization to minimize the sum of squared Euclidian distance between the sampled parameterization $\mathcal{X}_{ref}$ of the reference shape, and the corresponding parameterization $\mathcal{X}_i^{o,\{B\}}$ of the shape S_i, where $B = I_i(A)$, to make the parameterizations consistent. In Figure 4.10, it can be annotated as minimizing the distance between the blue points and the corresponding light red points in the parameterization. The objective function can be written as

$$\mathcal{L} = \sum_A \|\mathbf{x}_{ref}^A - \mathbf{x}_i^{o,B}\|^2 \tag{4.23}$$

In short, we will re-parameterize all the parameterizations $\{\mathcal{X}_i^o | i = 1, 2, \cdots, n_s\}$ (expect the reference) to have a consistent correspondence with the sampled parameterization of reference $\mathcal{X}_{ref}$. The re-parameterization uses the symmetric theta transformation with a uniformly distributed kernels and a fixed width. By a line searching algorithm, we manipulate the amplitude to minimize the objective function. This pairwise correspondence initialization will be processed iteratively

until the value of objective function is under some threshold. To propagate the correspondence, we use the same sampling $\mathcal{X}$ to all the consistent parameterizations $\mathcal{X}_i^o$ and then we will get the sampled shape landmarks $\mathcal{V}_i = \{\mathbf{v}_i^A | A = 1, 2, \cdots, n_p\}$ and sampled parameterization $\mathcal{X}_i = \{\mathbf{x}_i^A | A = 1, 2, \cdots, n_p\}$, where n_p is the landmark number.

Davies *et al.*[79] have suggested to apply the area preservation on the reference shape alone and to manipulate the parameterization in other shapes in the propagation afterwards. So we can perform a sigmoid transformation and a rotation in longitude followed by the iterative symmetric theta transformation to propagate the area preservation. After that, we can sample all the shapes to have a consistent initial correspondence. In practice, this propagation will be faster and more reliable.

4.3.3 Correspondence Optimization

After the initial correspondence is established, a group-wise registration algorithm will help to optimize the correspondence. Because we are aiming at building good statistical shape models, the correspondence can be evaluated using the properties of good models. These properties can be summarized as [92]

- **generalization ability** it can represent any instance of the class of object.
- **specificity** it can only represent valid instances of the modeled class of object.
- **compactness** the model can be represented with few parameters.

Here we choose to use the Minimum Description Length (MDL) principle as the objective function, as suggested by Davies *et al.* [92].

4.3.3.1 Description Length Principles

The basic of MDL is to transmit the coded training set with a minimum description length. According to the information theory, if we want to transmit a number within some range of width R and a precision δ, it requires a codeword of length:

$$l(x; R, \delta) = -\log \frac{\delta}{R}, \quad y < x < y + R, \tag{4.24}$$

To be more general, the shannon codeword length for transmission of an event α with the occurrence probability P_α can be written as:

$$l_\alpha = -\log P_\alpha. \tag{4.25}$$

Based on this knowledge, we can calculate the description length, namely the codeword length, of transmitting a set of shapes represented by statistical shape models, which can be modeled using a multivariate Gaussian model [79].

For the statistical shape models, we have explained the detail in chapter 3. We

can use equation 3.20 to build any shape instance, and use the method shown in equation 3.23 to get the simplified way of getting compact models, since there are always at most $n_s - 1$ non-zero eigenvalues when there are limited number of shapes (i.e. $n_s \ll n_p$). So we rewrite it as

$$\mathbf{v}_i = \overline{\mathbf{v}} + \mathbf{P}\mathbf{b}_i = \overline{\mathbf{v}} + \sum_{m=1}^{n_s-1} \mathbf{p}^m b_i^m \tag{4.26}$$

where the shape landmarks in S_i are all vectorized and aligned as column vectors $\{\mathbf{v}_i | i = 1, 2, \cdots, n_s\}$, $\overline{\mathbf{v}}$ is the mean shape vector, $\mathbf{p}^m$ is the m^{th} eigenvector, b_i^m is the mode weight (model parameter) for a shape instance S_i. If the shape vector elements are bounded by $-r/2 \leq v_{i,j} \leq r/2, (v_{i,j} \in \mathbf{v}_i, i = 1, 2, \cdots, n_s, j = 1, 2, \cdots, 3n_p)$, the mode weights b_i^m also have a strict bound as $R = r\sqrt{3n_p}$, which suffices $|b_i^m| \leq R$. For the transmission, given a fixed precision Δ and R, the number of modes $n_s - 1$, and the dimensionality of the shape vector $3n_p$, the description length can be formed as:

$$\mathcal{L}_{total} = f(R, \Delta, n_s) + \sum_{m=1}^{n_s-1} (\mathcal{L}_{params}^{(m)} + \mathcal{L}_{data}^{(m)}) = f(R, \Delta, n_s) + \sum_{m=1}^{n_s-1} \mathcal{L}^{(m)}. \tag{4.27}$$

where $f(R, \Delta, n_s)$ is the fixed description length of the mean shape $\overline{\mathbf{v}}$, the PCA directions $\{\mathbf{p}^m\}$, and other fixed parameters etc. For each mode m, $\mathcal{L}_{params}^{(m)}$ is the description length for the parameters of the m^{th} centered Gaussian in the direction $\mathbf{p}^m$, and $\mathcal{L}_{data}^{(m)}$ is the description length for the data in this direction, namely the mapping of the centered shape $\{\mathbf{v}_i - \overline{\mathbf{v}}\}$ onto the principal axes $\{\mathbf{p}^m\}$, $\{\hat{\mathbf{b}}_i = \mathbf{P}^T(\mathbf{v}_i - \overline{\mathbf{v}})\}$.

For $\mathcal{L}_{params}^{(m)}$, each centered Gaussian has one parameter, the width (standard deviation) σ^m, which can be calculated as,

$$(\sigma^m)^2 = \frac{1}{n_s} \sum_{i=1}^{n_s} (\hat{b}_i^m)^2, \tag{4.28}$$

We quantize it to some precision δ, and set a upper bound σ_{max} to R and a lower bound σ_{min} to 2Δ (It was numerically proven in [79, 86, 92] that this approximation is good enough). For each direction, the description length[1] can be written as

- If $\sigma^m \geq \sigma_{min}$, $\mathcal{L}^{(m)} = \mathcal{L}_1^{(m)}(\sigma^m, n_s, R, \Delta)$. where,

$$\begin{aligned} \mathcal{L}_1^{(m)} =& 1 + \ln\left(\frac{\sigma_{max} - \sigma_{min}}{\delta}\right) + |\ln \delta| - n_s \ln \Delta \\ &+ \frac{n_s}{2} \ln(2\pi(\sigma^m)^2) + \frac{n_s}{2} + \frac{n_s \delta^2}{12(\sigma^m)^2} \end{aligned} \tag{4.29}$$

[1] It is not the full description length since some constant terms are ignored.

where the optimum parameter accuracy δ is $\delta^*(\sigma, n_s) = \min\left(1, \sigma\sqrt{\frac{12}{n_s}}\right)$.

- If $\sigma^m < \sigma_{min}$ but the range of $b^m = \{b_i^m | i = 1, 2, \cdots, n_s\} > \Delta$, $\mathcal{L}^{(m)} = \mathcal{L}_2^{(m)}(\sigma^m, n_s, R, \Delta)$, where,

$$\begin{aligned}\mathcal{L}_2^{(m)} =& 1 + \ln\left(\frac{\sigma_{max} - \sigma_{min}}{\delta}\right) + |\ln\delta| - n_s \ln\Delta \\ &+ \frac{n_s}{2}\ln(2\pi\sigma_{min}^2) - \frac{n_s\delta^2}{24\sigma_{min}^2} + \frac{n_s(\sigma^m)^2}{2\sigma_{min}^2}\left(1 + \frac{\delta^2}{4\sigma_{min}^2}\right)\end{aligned} \tag{4.30}$$

  where $\delta = \delta^*(\sigma_{min}, n_s)$.

- Else $\mathcal{L}^{(m)} = 0$.

In the description length $\mathcal{L}^{(m)}$, the first three terms are the code length of the quantized parameter $\sigma^{\hat{m}}$ and the quantization precision, which comprise $\mathcal{L}_{params}^{(m)}$. The rest terms are the code length of data $\mathcal{L}_{data}^{(m)}$. For a detailed derivation, [79, 86, 92] can be referred.

4.3.3.2 MDL Objective Function

Based on the last section, we can calculate the total description length for all modes. For a given training set, if the correspondence is well established, the variance of the points will be small, leading to the small description length; otherwise, the description length will be large. Therefore, we are aiming at minimizing the total description length. Then the objective function constructed from the full expression of description length can be written as

$$\mathcal{L} = \sum_{p=1}^{n_g} \mathcal{L}_1^{(p)}(\sigma^p, n_s, R, \Delta) + \sum_{q=n_g+1}^{n_g+n_{min}} \mathcal{L}_2^{(q)}(\sigma^q, n_s, R, \Delta) \tag{4.31}$$

where n_g is the number of directions for $\sigma^p \geq \sigma_{min}$, and n_{min} is the number of directions for $\sigma^q < \sigma_{min}$ but the range of $b^q > \Delta$.

Apart from the full expression, there are some approximations of MDL objective function, such as the case assuming $\Delta \to 0$ and $n_s \to \infty$ by Davies *et al.*[79], and the case limiting the criteria by Thodberg [93]. Moreover, the gradient of simplified MDL objective functions have also been investigated such as [79] and [88].

4.3.3.3 Optimization Procedure

After establishing the initial correspondence, we have the sampled shapes with equal number of landmarks, and the consistent sampled parameterizations, which are most probably the same as the sampling sphere. We select a shape (except the reference shape) in the training set in turn to perform the optimization, which aims at

minimizing the total description length of training set by varying/re-parameterizing the parameterization of each shape. The re-parameterization approach is mainly based on symmetric theta transformation, where the kernel centers and width are fixed in the optimization. By varying the amplitude, the parameterization will be changed. The changed parameterization is used to sample the original parameterization (before the sampling in the correspondence propagation) again to get new sampled shape vertices. This new sampled shape together with other sampled shapes are used to evaluate the total description length. We use a linear searching optimization algorithm for finding the optimum value of amplitude which can give a minimum total description length. The specific steps are listed as follows:

For $i = 1, 2, \cdots, n_s$, the sampled shape vertices are $\mathcal{V}_{ref}$ and $\mathcal{V}_i$, the sampled parameterization nodes are $\mathcal{X}_{ref}$ and $\mathcal{X}_i$; The original shape vertices and parameterizations are $\mathcal{V}^o_{ref}$, $\mathcal{V}^o_i$, and $\mathcal{X}^o_{ref}$, $\mathcal{X}^o_i$ respectively. We choose a shape $S_i, i \neq ref$ to start the optimization.

(1) Fix a Cauchy kernel center position and a width ω, re-parameterize $\mathcal{X}_i$ using the symmetric theta transformation with a varying amplitude A. to get $\mathcal{X}'_i$.

(2) Use this re-parameterized $\mathcal{X}'_i$ to sample the un-sampled parameterization $\mathcal{X}^o_i$ and the original shape surface $\mathcal{V}^o_i$, then get the new sampled shape vertices $\mathcal{V}'_i$ and parameterization $\mathcal{X}'_i$.

(3) Form the shape vectors from $\mathcal{V}_{ref}$, $\mathcal{V}'_i$ and $\{\mathcal{V}_j | j = 1, 2, \cdots, n_s; \ \& \ j \neq i\}$ as the one in equation 4.26, which are $\mathbf{v}_{ref}$, $\mathbf{v}'_i$ and $\{\mathbf{v}_j | j = 1, 2, \cdots, n_s; \ \& \ j \neq i\}$.

(4) Concatenate these shape vectors together, apply the PCA to calculate the eigenvectors $\{\mathbf{p}^m | m = 1, 2, \cdots, n_s - 1\}$, mean shape vector $\overline{\mathbf{v}}$ and model parameters $\{b^m_i | i = 1, 2, \cdots, n_s, m = 1, 2, \cdots, n_s - 1\}$.

(5) Fix a quantization parameter Δ and use the model obtained in step (4) to calculate the description length and evaluate the MDL objective function.

(6) Use a line searching algorithm to iteratively conduct step (1) to step (5) to find the optimum A that can make the objective function value to a minimum.

(7) Use the obtained optimum A to re-parameterize $\mathcal{X}_i$ using symmetric theta transformation with the same kernel and width as step (1).

(8) Repeat step (2) to get the sampled shape vertices $\mathcal{V}'_i$.

(9) Assign $\mathcal{V}'_i \rightarrow \mathcal{V}_i$ and $\mathcal{X}'_i \rightarrow \mathcal{X}_i$.

(10) Move to the next kernel center among a set of uniform sphere nodes, repeat the step (1) to (9) until all kernels are processed.

(11) Move to the next shape S_i to repeat step (1) to (10).

(12) Iteratively conduct step (1) to (11).

Chapter 5

2D Slices based AAM Fitting

In this chapter, we will discuss the AAM search further. Firstly, a brief survey on the AAM search approaches will be given, and then we will introduce the novel algorithm used in our project, which is a 2D-slice based AAM search approach, where only three 2D MR image slices are used for model fitting.

5.1 AAM Search Overview

In chapter 3, we have reviewed the standard AAM search algorithm. This works only when the object is located in the neighborhood. When there is a large difference (offset, scaling, rotation) between the initial shape (usually the mean shape) and the actual object shape, the standard algorithm will lead to a mismatch with the object. In that case, a multi-scale model can be made to apply a multi-resolution search. In the low resolution image, a coarse contour would be found, then the refinement will be processed using the higher resolution image. This multi-resolution search was first introduced for ASM search [94] and implanted in AAM later. Cootes *et al.* [95] have also proposed a way of reducing the texture information in finding the transfer matrix $\mathbf{A}$ as updating the parameter $\mathbf{p}$. They believed there are redundant texture information not representative to the texture change. As a result, they picked out the 'important' texture for parameter update modeling. Meanwhile, they also suggested an approach to establish the relationship between the texture change and the shape parameters directly, but it works better only in some limited conditions. Similarly, Stegmann [68] investigated a border AAMs, where only the texture around the shape border is extracted, instead of the texture inside the shape. This approach solved the problem of shrinking in the AAM search, and is also faster. Moreover, Stegmann [68] has also proposed a constrained AAM search that forms some bounds for the parameter changes. Actually these were improved later in the standard algorithm using a numeric differentiation method. Matthews *et al.* [96] suggested an inverse compositional alignment algorithm based approach to improve the AAM fitting efficiency. They also presented an analytical derivation for gradient descent searching, an efficient ad-hoc fitting algorithm that overcomes the inefficiency in standard gradient descent algorithm. This approach was extended to a 3D case [97]. Beichel *et al.*

[98] proposed a novel robust AAM matching algorithm, which contains two stages. The first one is for analyzing the initial residuals with a mean-shape-based mode detection. The second one is for utilizing an objective function for the selection of a mode combination not representing the gross outliers. Donner *et al.* [99] employed a fast canonical correlation analysis (CCA) based search to calculate the derivative matrix with the reduced rank estimates obtained by CCA, instead of the standard linear least square regression estimates. This approach has shown faster training with fewer synthetic difference images and faster convergence with equal accuracy. Tresadern *et al.* [100] has investigated additive predictors as a substitute for the linear predictor in order to improve the AAM search accuracy and efficiency. They have demonstrated that the proposed method is faster and more accurate as well, and the linear models performs the same with non-linear ones.

Most of these works are for 2D AAM search. Actually in 3D AAM, the majority of model fitting approaches are still based on the standard one, such as [53, 22, 23]. However, in terms of our project, the full 3D data is not available since the acquisition of full 3D data is time consuming which is not acceptable in the real time biopsy. For this purpose, we would like to figure out a method that can utilize three orthogonal slices in the AAM search, with which we can reconstruct the full texture map.

5.2 PCA based Prediction

In AAM modeling, both shape variance and texture variance was analyzed using PCA to form a PDM. For instance, in texture modeling, we have the representation as

$$\mathbf{g} \approx \overline{\mathbf{g}} + \mathbf{P}_g \mathbf{b}_g = \overline{\mathbf{g}} + \sum_{m=1}^{t_g} \mathbf{p}_m^{(g)} b_m^{(g)} \tag{5.1}$$

where $\mathbf{g}$ is a $n_g \times 1$ texture vector, $\overline{\mathbf{g}}$ is the mean texture vector from

$$\overline{\mathbf{g}} = \frac{1}{n_s} \sum_{i=1}^{n_s} \mathbf{g}_i \tag{5.2}$$

where $\mathbf{g}_i$ is the normalized texture vector taken from image S_i in the training set. $\mathbf{P}_g$ is the eigenvectors representing the principal axes in the training set, $\mathbf{b}_g$ includes the weight vector for each mode, called texture parameters. Given an instance $\mathbf{g}_i (i \in \{1, 2, \cdots, n_s\})$, we can get the texture parameter $\mathbf{b}_g$ by approximately,

$$\mathbf{b}_{g,i} = \mathbf{P}_g^T (\mathbf{g}_i - \overline{\mathbf{g}}) \tag{5.3}$$

Inversely, if $\mathbf{b}_{g,i}$ is given, we can also recover the original normalized texture vector $\mathbf{g}_i$ using equation 5.1.

Now we look how equation 5.3 is derived. As we review chapter 3, it is obvi-

ous that originally $\mathbf{P}_g$ is a $n_g \times n_g$ square matrix $(\mathbf{p}_1, \cdots, \mathbf{p}_m, \cdots, \mathbf{p}_{n_g})$, where each column is an unit (normalized) eigenvector and each eigenvector is always orthogonal to any other (i.e. $\mathbf{p}_m^T \mathbf{p}_n = \delta_{m,n}$). Therefore we can hold the following equation

$$\mathbf{g}_i - \overline{\mathbf{g}} = \mathbf{g}_{c,i} = \mathbf{P}_g \mathbf{b}_{g,i} = \sum_{m=1}^{n_g} \mathbf{p}_m^{(g)} b_{i,m}^{(g)} \tag{5.4}$$

where $\mathbf{g}_{c,i}$ is the centered vector. In this case, we can calculate $\mathbf{b}_{g,i}$, which is the vector of $\mathbf{g}_i - \overline{\mathbf{g}}$ mapping to the principal axes, as

$$\mathbf{b}_{g,i} = \mathbf{P}_g^{-1}(\mathbf{g}_i - \overline{\mathbf{g}}) = \mathbf{P}_g^T(\mathbf{g}_i - \overline{\mathbf{g}}) = \mathbf{P}^T \mathbf{g}_{c,i} \tag{5.5}$$

where $\mathbf{P}_g^{-1} = \mathbf{P}_g^T$ since $\mathbf{P}_g$ is an orthogonal matrix. However, when we use the simplified method for calculating the principal axes (as the latter part of section 3.2.1.2 states), we only get the first n_s eigenvectors and there are actually $n_s - 1$ non-zero eigenvalues so that only $n_s - 1$ eigenvectors will be used. This gives $\mathbf{P'}_g = (\mathbf{p}_1, \cdots, \mathbf{p}_m, \mathbf{p}_{n_s-1})$ as a $n_g \times (n_s - 1)$ matrix where all column vectors are still unit orthogonal to each other. And now equation 5.4 becomes

$$\mathbf{g}_i - \overline{\mathbf{g}} = \mathbf{g}_{c,i} = \mathbf{P'}_g \mathbf{b}_{g,i} = \sum_{m=1}^{n_s-1} \mathbf{p}_m^{(g)} b_{i,m}^{(g)} \tag{5.6}$$

If we want to get $\mathbf{b}_{g,i}$, we can use the Moore-Penrose pseudo-inverse of $\mathbf{P}_g$ since $\mathbf{P}_g$ is not a square matrix any more. We can use the LSE[1] to get the Moore-Penrose pseudo-inverse, namely minimizing the objective function

$$\mathcal{L} = \|\mathbf{g}_{c,i} - \mathbf{P'}_g \mathbf{b}_{g,i}\|^2 = (\mathbf{g}_{c,i} - \mathbf{P'}_g \mathbf{b}_{g,i})^T (\mathbf{g}_{c,i} - \mathbf{P'}_g \mathbf{b}_{g,i}) \tag{5.7}$$

$$= \mathbf{g}_{c,i}^T \mathbf{g}_{c,i} - 2\mathbf{b}_{g,i}^T \mathbf{P'}_g^T \mathbf{g}_{c,i} + \mathbf{b}_{g,i}^T \mathbf{P'}_g^T \mathbf{P'}_g \mathbf{b}_{g,i} \tag{5.8}$$

by differentiating it with respect to $\mathbf{b}_{g,i}$, we get

$$\frac{\partial \mathcal{L}}{\partial \mathbf{b}_{g,i}} = -2\mathbf{P'}_g^T \mathbf{g}_{c,i} + 2\mathbf{P'}_g^T \mathbf{P'}_g \mathbf{b}_{g,i} \tag{5.9}$$

For getting the minimum value, equation 5.9 should be zero, namely

$$\hat{\mathbf{b}}_{g,i} = (\mathbf{P'}_g^T \mathbf{P'}_g)^{-1} \mathbf{P'}_g^T \mathbf{g}_{c,i} \tag{5.10}$$

While we can also see $\mathbf{p}_m^T \mathbf{p}_n = \delta_{m,n}$, so $\mathbf{P'}_g^T \mathbf{P'}_g = \mathbf{I}_{n_s-1}$ (i.e. identity matrix). In conclusion we get

$$\hat{\mathbf{b}}_{g,i} = \mathbf{P'}_g^T \mathbf{g}_{c,i} = \mathbf{P'}_g^T (\mathbf{g}_i - \overline{\mathbf{g}}) \tag{5.11}$$

[1] LSE has also been implemented in orthogonal Procrustes analysis and AAM parameter update modeling, where the derivations are not given.

It also works when $\mathbf{P}_g$ has less eigenvectors such as $t_g \leq n_s - 1$, as long as the column vectors are orthogonal. Then usually we get equation 5.3 depending on this.

Next we consider the following form

$$\mathbf{g}_{c,i} = \mathbf{g}_i - \overline{\mathbf{g}} = \begin{bmatrix} \mathbf{g}_{1,i} \\ \mathbf{g}_{2,i} \\ \mathbf{g}_{3,i} \end{bmatrix} - \begin{bmatrix} \overline{\mathbf{g}}_1 \\ \overline{\mathbf{g}}_2 \\ \overline{\mathbf{g}}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{g}_{c,1,i} \\ \mathbf{g}_{c,2,i} \\ \mathbf{g}_{c,3,i} \end{bmatrix} \tag{5.12}$$

where we divide the vector $\mathbf{g}_{c,i}$ into three column vectors of length n_1, n_2, n_3 respectively and $n_1 + n_2 + n_3 = n_g$. Similarly we divide $\mathbf{P}_g$[1] as well, which is

$$\mathbf{P}_g = \begin{bmatrix} \mathbf{P}_{g,1} \\ \mathbf{P}_{g,2} \\ \mathbf{P}_{g,3} \end{bmatrix} \tag{5.13}$$

where $\mathbf{P}_{g,1}$ is a $n_1 \times t_g$ matrix, $\mathbf{P}_{g,2}$ is of dimension $n_2 \times t_g$, $\mathbf{P}_{g,3}$ is of dimension $n_3 \times t_g$. And we have known that $\mathbf{b}_{g,i}$ is a column vector of size $t_g \times 1$. Therefore, from equation 5.1, we can derive the following three equations

$$\mathbf{g}_{c,1,i} = \mathbf{P}_{g,1}\mathbf{b}_{g,i}, \quad \mathbf{g}_{c,2,i} = \mathbf{P}_{g,2}\mathbf{b}_{g,i}, \quad \mathbf{g}_{c,3,i} = \mathbf{P}_{g,3}\mathbf{b}_{g,i} \tag{5.14}$$

Using the least square estimate solution shown in equation 5.10, we can obtain

$$\hat{\mathbf{b}}_{g,i} = (\mathbf{P}_{g,1}^T\mathbf{P}_{g,1})^{-1}\mathbf{P}_{g,1}^T\mathbf{g}_{c,1,i} \tag{5.15}$$

$$= (\mathbf{P}_{g,2}^T\mathbf{P}_{g,2})^{-1}\mathbf{P}_{g,2}^T\mathbf{g}_{c,2,i} \tag{5.16}$$

$$= (\mathbf{P}_{g,3}^T\mathbf{P}_{g,3})^{-1}\mathbf{P}_{g,3}^T\mathbf{g}_{c,3,i} \tag{5.17}$$

It should be noted that $\mathbf{P}_{g,1}^T\mathbf{P}_{g,1} \neq \mathbf{I}$ as well as $\mathbf{P}_{g,2}$ and $\mathbf{P}_{g,3}$. These results show that, when part of the texture vector (e.g. $\mathbf{g}_{1,i}$, $\mathbf{g}_{2,i}$ or $\mathbf{g}_{3,i}$) is available, given the mean texture vector, and the corresponding part of principal axes, we can always predict the texture parameter and then give a full realization of $\mathbf{g}_i$ by

$$\hat{\mathbf{g}}_i = \overline{\mathbf{g}} + \mathbf{P}_g\hat{\mathbf{b}}_{g,i} \tag{5.18}$$

As a result, this knowledge proved the possibility of only using part of the texture information, such as three orthogonal slices, to accomplish the AAM matching.

5.3 2D Slices based Model Fitting Procedure

Because of the unpredictable position where the needle would be inserted, we do not know which slices would be scanned for model fitting. Therefore, we cannot build AAM with respect to any triple slices texture information. This is also why the border AAM [68] or the texture reduced AAM [95] are not applicable in terms

[1] We assume here $\mathbf{P}_g$ has t_g column vectors and t_g could be either smaller than $n_s - 1$ or between n_s and n_p.

of our task. Hereby we should build the full 3D AAM for training to give the complete information.

Suppose that the 3 slices for model fitting are placed in correct position and orientation in 3D space. When updating the parameters to minimize the intensity difference $\delta\mathbf{g}$, only the intensities in the voxels from these slides will be examined. The full texture parameters can be predicted using these known intensities and the principal axes as well as the mean texture. Then the whole texture map can be recovered using these predicted parameters and the models. The following steps implement the whole procedure.

(1) Begin with the mean appearance parameter $\mathbf{c} \leftarrow \overline{\mathbf{c}}$, . Given the parameter number t_s and t_g from the training set, extract the shape parameter $\mathbf{b}_s$ and $\mathbf{b}_g$. Then we can give initial shape points by $\mathbf{x}_s = \mathcal{T}^{-1}(\overline{\mathbf{x}} + \mathbf{P}_s\mathbf{b}_s)$.

(2) Given the initial shape $\mathbf{x}_s$, get the aligned shape vector $\mathbf{x} = \mathcal{T}(\mathbf{x}_s)$ and get the pose parameters $(\mathbf{t}^T, \mathbf{q}^T)^T$.

(3) Prepare two testing 3D images. One is that three slices are placed in the correct position with the background filled with zero. The other is that three slices are located in the correct position, and the background is composed of the mean of all images in the training set. warp the both the test images with respect to *mapped base points* to exact the normalized texture $\mathbf{g}_{s,1}$ and $\mathbf{g}_{s,2}$ within $\mathbf{x}_s$, respectively. Use the $\mathbf{g}_{s,1}$ which declares three slices to only get the voxel positions, and use $\mathbf{g}_{s.2}$ which has the background of image mean to extract the intensity in these slices, and get $\mathbf{g}_s$.

(4) Get the parameters $\mathbf{p}$. Apply the same position of $\mathbf{g}_s$ (same as $\mathbf{g}_{s,1}$) of $\mathbf{P}_g$, denoted as $\mathbf{P}_{g,s}$. Then we calculate

$$\mathbf{c} = \mathbf{Q}^T \left(\begin{bmatrix} \mathbf{W}_s\mathbf{P}_s^T(\mathbf{x} - \overline{\mathbf{x}}) \\ (\mathbf{P}_{g,s}^T\mathbf{P}_{g,s})^{-1}\mathbf{P}_{g,s}^T(\mathbf{g}_s - \overline{\mathbf{g}}_s) \end{bmatrix} - \overline{\mathbf{b}} \right) \tag{5.19}$$

where $\overline{\mathbf{g}}_s$ is the subset of the mean texture vector from the training set, of the positions from $\mathbf{g}_s$. Constrain $\mathbf{c}$ within $\pm 3\sqrt{\lambda_i^{(c)}}$ and form the parameter $\mathbf{p} = (\mathbf{c}^T, \mathbf{t}^T, \mathbf{q}^T)^T$.

(5) Start the parameter update iteratively.

(I) Initialize the damping factor $k = 1$, the maximum iteration number N_{iter}, parameter $\mathbf{p}_o$. Set the initial intensity difference magnitude $E_o = \infty$.

(II) **for** $iter = 1 : N_s$

(i) Split the parameter $\mathbf{p}$ into the appearance model parameter $\mathbf{c}$ and the pose parameters $(\mathbf{t}^T, \mathbf{q}^T)$.

(ii) Calculate the shape parameter $\mathbf{b}_s$ from $\overline{\mathbf{b}}, \mathbf{W}_s, \mathbf{c}, \mathbf{P}_s$ and then get the aligned shape points $\mathbf{x} = \overline{\mathbf{x}} + \mathbf{P}_s\mathbf{b}_s$, and the real position in image by $\mathbf{x}_s = \mathcal{T}^{-1}(\mathbf{x})$.

(iii) Warp the both test images to *mapped based points* to obtain the sampled texture $\mathbf{g}_{s,1}$ and $\mathbf{g}_{s,2}$ within $\mathbf{x}_s$, and get the real normalized slice intensity vector $\mathbf{g}_s$ as in (3).

(iv) Apply equation 5.19 to get a new appearance model parameter $\mathbf{c}'$ and apply equation 3.58 to get the texture parameter $\mathbf{b}'_g$ then calculate the modeled texture by $\mathbf{g}_m = \overline{\mathbf{g}} + \mathbf{P}_g\mathbf{b}'_g$, but only the intensities in slices position are taken as $\mathbf{g}_{m,s}$.

(v) Evaluate the intensity difference $\delta\mathbf{g} = \mathbf{g}_s - \mathbf{g}_{m,s}$ and its magnitude $E = |\delta\mathbf{g}|^2 = \delta\mathbf{g}^T\delta\mathbf{g}$.

(vi) **if** $E > E_o$

Update damping factor $k \leftarrow 0.9k$, the parameter $\mathbf{p} \leftarrow \mathbf{p}_o$, and found shape points $\mathbf{x}_s \leftarrow \mathbf{x}_o$;

else

$k \leftarrow 1.1k$, $E_o \leftarrow E$.

end if

(vii) Update the parameter $\mathbf{p}_o \leftarrow \mathbf{p}$ and found shape points $\mathbf{x}_o \leftarrow \mathbf{x}_s$.

(viii) Permute the parameter $\mathbf{p}$ with the predicted offset $\delta\mathbf{p_s} = -\mathbf{A}_s\delta\mathbf{g}_s$, by $\mathbf{p} \leftarrow \mathbf{p} + k \cdot \delta\mathbf{p}$ and constrain the updated $\mathbf{p}$ within $\pm 3\sigma_i$.

end for

(III) **if** $E > E_o$,

update the parameter $\mathbf{p} \leftarrow \mathbf{p}_o$, the found shape points $\mathbf{x}_s \leftarrow \mathbf{x}_o$ and the intensity difference magnitude $E \leftarrow E_o$.

end if

(IV) Optimize the parameter $\mathbf{p}$ using the local minimization algorithm regarding the intensity difference E.

(6) Once the optimal $\mathbf{p}$ is found, execute step (i) to (iv) to get the sampled slices texture $\mathbf{g}_s$, the modeled full image texture $\mathbf{g}_m$, and the found shape points $\mathbf{x}_s$.

(7) Realign the texture vector $\mathbf{g}_m$ with respect to the voxel template, and inversely normalize the intensity, warp the image with shape points of *mapped base points* and texture of realigned $\mathbf{g}_m$, with respect to the found shape points $\mathbf{x}_s$. Then we can get the segmented image.

(8) Using the technique of obtaining the voxel template, we can get the discrete voxel based segmentation.

In these procedure, we can see that it still follows the standard search approach, but with only the voxel positions in the given slices are taken into consideration. we added a fake background besides the given slices for decreasing the influence from the surrounding voxels on the intensities in the positions of these slices.

Although it works, this 2D slices based method suffers from a slow convergence since the parameter updating is insufficient in every iteration, resulting from the intensity displacements are partially applied.

Chapter 6

Experimental Setup and Implementations

Based on the knowledge from previous chapters, we synthesized all the works and implemented the phantom and AAM to conduct experiments. In this chapter, we will mainly show the procedures and results of phantom fabrications, MRI settings and 3D image acquisition, shape correspondence establishment and AAM modeling as well as the shape and texture reconstruction based on the 2D fitting.

6.1 Phantom Fabrication

The phantom fabrication consists of mold establishment for making the model and model fabrication for completing the phantom. The structure and materials are chosen as chapter 2 states. The dimension for each part is shown in Appendix A.

6.1.1 Building Phantom Molds

Due to the special shapes of our phantom components, especially for the prostate and pubis, corresponding molds were built. For the prostate, according to the suggested 3D sketch, we changed the design where a cylinder with the diameter of 8 mm was added on the top (prostatic base) of the big prostate mold for reserving a hole on the matrix for casting. A similar tail with the diameter of 4.3 mm was added on the small prostate mold for suspending the small mold in the matrix. Then we utilized the 3D printer[1] to get the precise shape for the larger prostate mold and the small prostate mold. The printed bigger prostate mold and smaller mold are shown in Figure 6.1.

The matrix for casting prostate was made of a cold vulcanizing silicone-rubber by mixing the silicone-rubber and the corresponding hardener. This matrix can resist high temperature of 160-180 ℃ when the soft PVC mixture is poured in. We

[1]The Objet EDEN 250 3D printer is in UT Carré 3rd floor RAM lab.

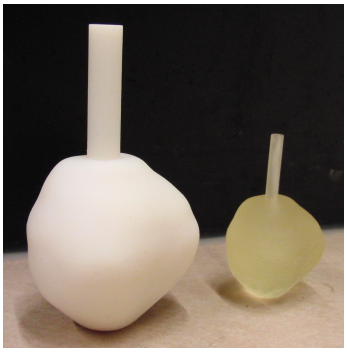
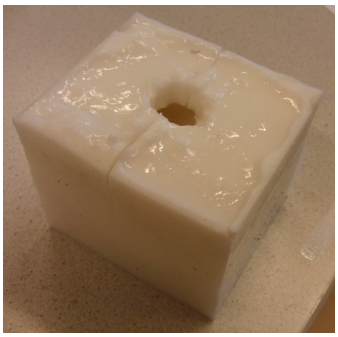
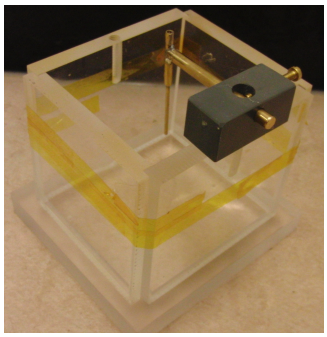


Figure 6.1: 3D printed bigger and smaller prostate molds with cylinders attached on the top

chose the Siliconenrubber 620/TL95 [1] and its hardener T58 [2] produced by Harjon Polyster B.V. in Netherlands. The exterior container for the matrix was made of perspex and the printed prostate mold was suspended in the middle. We mixed the silicone-rubber and the hardener with a ratio of 20:1, and vacuumed this mixture, then poured the mixture into the container and wait for over 1 day. The matrix can be obtained after taking out the prostate mold by cutting the silicone-rubber in the middle. Figure 6.2 shows the completed prostate matrix for casting the outer capsule, and the plexiglass box with a needle on the top for containing the matrix and suspending the small prostate mold for saving space for the inner prostate.



(a) Prostate matrix made of silicone-rubber.



(b) Plexiglass container with needle for containing matrix and suspending smaller prostate.

Figure 6.2: Prostate matrix and container

For the pubic bone, we added an additional connecting part and a cylinder to connect the left and right ramus. We also printed the model which was directly

[1] http://www.harjonpolyester.nl/index.php?pgname=SILICONENRUBBER%20620/TL95&pid=535&cat_id=220

[2] http://www.harjonpolyester.nl/index.php?pgname=HARDER%20T58%20VOOR%20SILICONENRUBBER%20620/TL95&pid=536&cat_id=220

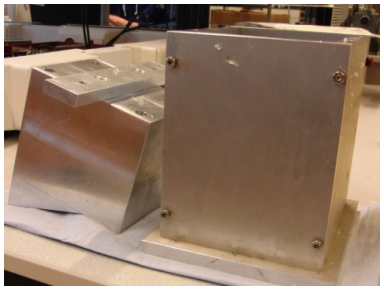
put in the phantom. The printed pubis model is shown in Figure 6.3(a).

Due to the rectum in the phantom is an air tube, considering the art processing, a strong, heat-resisting and smooth rectum mold made of copper pipe as Figure 6.3(b) was fixed for saving the space. When the phantom was done, the rectum mold should be removed to form the air tube in the frame.

To make the frame, a corresponding aluminium frame mold with a matched inner casting block was constructed. The inner dimension of aluminium bin corresponds to the outer size of frame, and the dimension of inner block corresponds to the inner frame size. The copper rectum mold was be screwed on the 100 mm $\times$150 mm wall at the higher base, and above the bottom at 30 mm height approximately. The final aluminium mold and inner block are shown in Figure 6.3(c).

(a) 3D printed pubis model

(b) Copper rectum mold



(c) Aluminium mold for frame making

Figure 6.3: Pubis model, rectum mold and aluminium mold

6.1.2 Building Phantom

We built the phantom in the laboratory[1] in UT. The apparatus includes: a heating plate, a BINDER oven/vacuum chamber, a ventilating duct, a aluminium pot over 1liter, a stiring rod made of teflon, screws and a screw driver, a electronic scale, a thermometer, a pair of oven mitten. Apparently the molds and models are necessary, including the aluminium bin with the inner block, the silicone-rubber matrix for prostate, the printed pubis model. The materials, as stated, include Plastileurre® standard, premixed soft and softener for 3 liter respectively. The whole procedure has two major steps. First, make the frame and prostate; second, fill up the phantom surroundings.

6.1.2.1 Frame and Prostate Capsule Making

Firstly, we put the pot on the scale and weighed around 800 gram standard mixture solution while pouring the mixture in the pot. The oven was set to 200 ℃ (It should be over 160 ℃ but below 250 ℃ considering the polymerization temperature

[1]IMS (Inorganic Materials Science) Lab in Carré 3rd floor

is around 160 ℃ and the heating tolerance of silicone matrix is 250 ℃). Then the small prostate mold was suspended by inserting the needle to its tail, and was put in the matrix contained by the matrix plexiglass box. The small prostate mold should not touch the wall and bottom of the inner surface of the matrix. The matrix container was bounded by heat resisting tapes in case of collapse. The aluminium bin and the inner metal block as well as the matrix box (with matrix and small prostate mold) were put into the oven for preheating in case of fast curdling of polymer.

Secondly, the pot was put on the heating plate that was set to 260 ℃ (In spite of the polymerization happens at 160 ℃, if the heating temperature is under 200 ℃ the process of polymerization would take too long; while if the temperate is too high, such as over 300 ℃, the polymer would easily be burned). The polymerization took approximately 30 minutes. During the heating, the solution was stirred slowly with a teflon rod to make the material well heated. We put the ventilating duct right above the pot to exhaust the unpleasant smell from heating the solution. The standard solution at the beginning was white like milk, and it turned clear as the heating went on, finally into some clear viscous creamy liquid, like glue.

Thirdly, the preheated aluminium bin, the inner metal block, and matrix box were taken out onto a cleaning paper. Meanwhile, the pot with polymerized soft PVC mixture was transferred to the oven with the same temperature. Then the vacuum was turned on to exhaust the air bubbles in the mixture. At the beginning of the vacuuming, the bubbles gathered on the surface, and they popped after a while. The vacuuming took around 1 hour, until there were no obvious bubbles on the surface. On the other side, the aluminium bin and inner block were put on the heating plates to keep them warm.

Fourthly, after the vacuuming was done, the aluminium bin was kept on the plate, the pot was taken out from the oven quickly. Then the solution was poured into the aluminium bin until the solution was just above the rectum copper model. The aluminium bin was slanted to let the solution drop along the wall to avoid introducing air bubbles. After the base part was filled, the inner block was hung at the correct place and was screwed tight on the wall to keep it from moving during the pouring and vacuum. Then the frame wall was filled with the PVC mixture.

Finally, the matrix was filled up through the hole on top of matrix with the standard PVC mixture. Then the filled bin and matrix were put carefully to the oven for vacuuming around 40 minutes, because some bubbles were generated during the pouring. At the same time, the pot stayed on the heating plate to avoid PVC mixture curdling. This vacuuming took less time than before. After taking out the aluminium bin and matrix, they were filled up again using the rest materials. The cooling to a room temperature took more than 3 hours.

6.1.2.2 Inner Prostate Filling

Firstly, after the prostate matrix cooled down, we took out the prostate capsule with the small mold from the matrix. Because the capsule was wrapping on the small prostate mold, we used a knife to cut up the top part (close to the tail) with a small crevice and took the small mold out. Then we put this capsule back to the matrix and bounded the matrix with taps. The matrix with the capsule was put into the oven for a short time preheating with around 160 ℃.

Secondly, around 100 grams premixed soft solution together with 60 grams softener were mixed in the pot. The pot was heated at 260 ℃ on the heating plate until the mixture became clear. It took about 25 minutes. After the polymerization completed, we took out the matrix from the oven then vacuumed the pot for around 10 minutes for exhausting the air bubbles. After the vacuuming was finished, we took out the pot and poured the mixture solution slowly to the cavity of the prostate capsule to fill it up. Afterwards, we put the capsule back to the oven for vacuuming another 10 minutes. Then we took it out and waited until it cooled down.

6.1.2.3 Surroundings Filling

Following a similar procedure, after the aluminium bin and prostate matrix cooled down, we took out the prostate model and cut off the tail, and also took out the inner block from the aluminium bin. We detached the walls of the bin for taking out the inner block more easily. Afterwards, the walls were screwed on to support the phantom.

Firstly, 600 grams premixed soft solution together with 400 grams softener were mixed in the pot. The bin and frame were put into the oven for preheating at 180 ℃. The pot was put on the heating plate at 260 ℃, and the mixed solution was stirred. The polymerization took around 25 minutes. After the mixture became transparent and the temperature in the oven reached over 160 ℃, the bin and frame were taken out, then we put the pot into oven for vacuuming around 10 minutes.

Secondly, we tilted the bin and poured the solution in the bin slowly along the frame wall to form the first thin layer. After it cooled down, the bin was tilted in the opposite side and we poured mixture to form the valley at the bottom. The cooling procedure took about 15 minutes. And the rest material was put on the heating plate to keep warm.

Thirdly, the prostate model was put at the correct position in the valley. The cylinder part of pubis was tapped for fixing on the metal strip. The pubis model was hung at the correct place and the metal strip was fixed on the wall with tapes. Then we filled the phantom up with the remaining soft solution. The whole phantom was kept in the oven and vacuumed for 1 hour. Afterwards, the phantom was taken out carefully to cool down over night.

6.1.2.4 Phantom Completion

After the phantom cooled down to room temperature, the tape on the pubis and the metal strip that hang over the walls were removed. The walls and bottom of the aluminium bin were screwed off and the gel phantom was taken out. We made a transparent perspex box to contain the phantom. We added a cover on the top with a hole that made the cylinder of pubis get through for fixing. The plexiglass in perineal part was spared for needle insertion. All parts were composed together with plastic screws to complete the phantom. The constructed phantom is shown in Figure 6.4. There are some additional remarks on phantom making in Appendix.

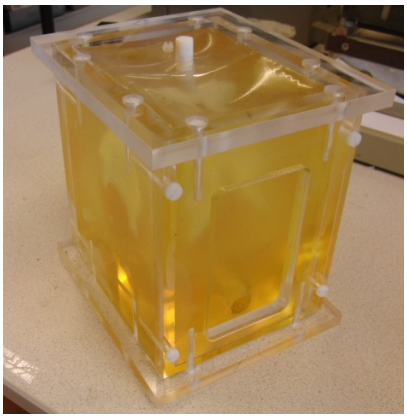


Figure 6.4: Constructed phantom

6.2 MRI Setup and Image Acquisition

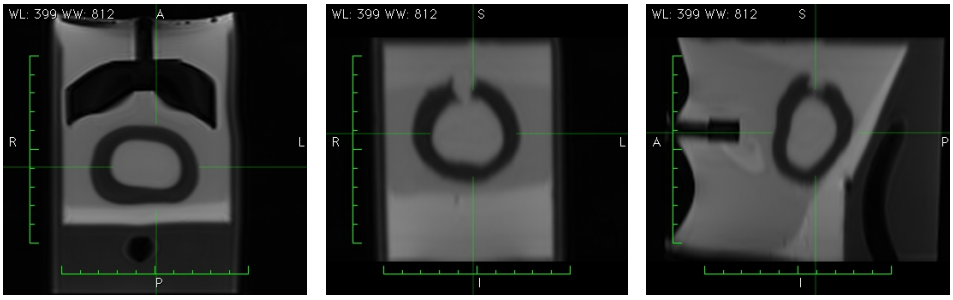
We brought the constructed phantom to UMC St. Radboud, Radiology department, and put it in the MRI scanner for acquiring the full 3D images. The 3 Tesla Skyra Siemens wide-bore MRI scanner was used for the image visualization and acquisition. We used 3D T2-weighted scans the same as in clinic applications. Table 6.1 shows the relevant settings in the MRI scanner.

The voxel spacing was chosen under the consideration of the trade off between the image resolution and the acquisition time. If we want to obtain a more precise image (e.g. $0.5mm \times 0.5mm \times 0.5mm$), it would take too long to keep the deformation. But if the resolution is low, the image would be less accurate and perhaps the deformation would disappear.

The scan was full 3D, and Figure 6.5 shows three example slices in axial, coronal and sagittal plane respectively. Figure 6.6 shows the corresponding 3D placement of these slides, where the horizontal plane is the axial plane. The corresponding coordinates are also shown.

Table 6.1: MRI Scanner Settings for 3D Image Acquisition

Magnetic Field Strength	3 Tesla
Total Acquisition Time	4 min
Echo Time (TE) / Repetition TIme (TR)	112 ms / 1600 ms
Image Size	192×192
Voxel Dimensions	$1mm \times 1mm \times 2mm$
Slices Thickness	$2mm$
Image Format	Dicom
Patient Position	HFS
Number of Slices	60



(a) Axial (b) Coronal (c) Sagittal

Figure 6.5: The example MR Image slices in 3 planes

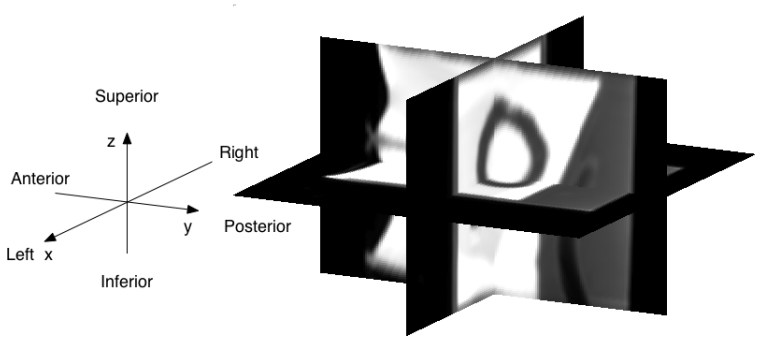


Figure 6.6: The example of 3D MR Image

As for the image acquisition, we scheduled 9 directions for inserting the biopsy gauge needle after discussing with the radiologist and as we realized that the cancerous tissue could happen to any zone. The 9 insertion directions are: (1) from caudal (down, inferior) central to head (up, superior) central, (2) from caudal central to ventral (anterior, frontal), (3) from caudal central to dorsal (posterior, or back), (4) from caudal central to right (of patient), (5) from caudal central to left (of patient), (6) from caudal ventral to head central (7) from caudal dorsal to head central (8) from caudal right to head central, (9) from caudal left to head central. The schematic figure 6.7 will illustrate these directions.

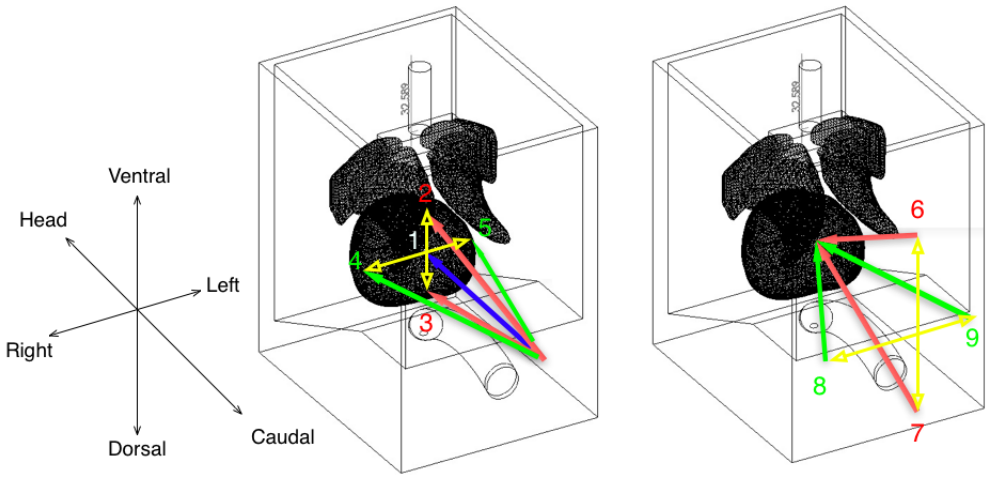


Figure 6.7: Illustration of needle insertion direction. The numbers are correspondent to the text above. Blue direction is perpendicular to axial plane, red directions are in sagittal plane, green directions are in coronal plane, yellow arrows are coordinates in axial plane.

In addition, we also took two scans without the needle. So there were totally eleven full 3D images. Actually, these directions are way too less than the clinical conditions. The selection of insertion positions obeys a uniform distribution, we intend to include as many cases as possible. Due to the time limitation, only the eleven images were scanned.

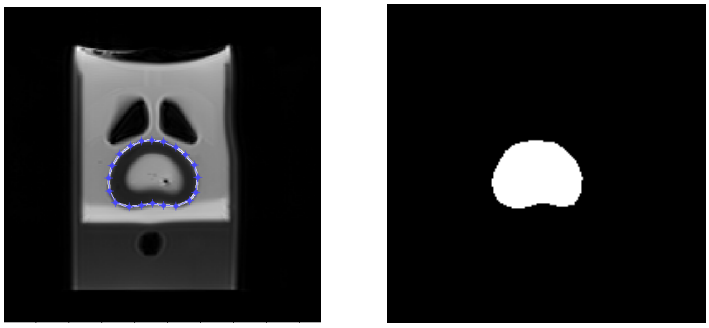
It is also noted that the needle should be held by people during the whole acquisition time, since the deformations would be gone if the tension is lost. Besides, it was difficult to acquire a series of images to show the whole fashion of deformation in time sequence. The reason is once the needle is inserted, the needle will move slowly further if the tension is held. However the deformation can only happen when the tip of needle goes into the capsule until most part of the needle is in the prostate. This whole period will take all the time of acquisition, when only one 3D image can be obtained.

6.3 AAM Modeling

After the images were scanned, we extracted the images from the Dicom files using Matlab. The whole process includes: manual segment shape surface, build the correspondence and sample the shape surface to get landmarks, build shape and texture models then synthesize the appearance models, and then use the full 3D model search and 2D slices based search for testing. The following parts annotate each course specifically.

6.3.1 Surface Acquisition

First we have to manually segment each image with respect to the shape of prostate. The manual segmentation was taken slice by slice, where a serial points was labelled around the contour of prostate and the inner part was masked 1's and the outside part was 0's. The labeling and binary outline are shown in Figure 6.8.



(a) Labeling the contour of prostate

(b) Binary segmentation of prostate outline.

Figure 6.8: Slices segmentation for shape acquisition

It is noticed that the voxel size is not equal, so we secondly resized the voxel size from $1mm \times 1mm \times 2mm$ to $1mm \times 1mm \times 1mm$ by interpolation in the z direction. Then the image size, for both binary and the original scanned image, were transformed to $192 \times 192 \times 120$. By stacking the binary slices together, we can reconstruct the 3D surfaces using the marching cube algorithm [70]. Then each shape was assigned a vertices matrix and triangular faces matrix connecting the vertices. But the correspondence was not established, so the next phase is building the correspondence.

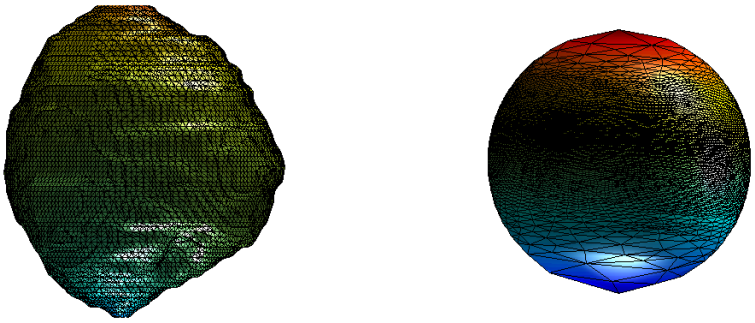
6.3.2 Establishing Correspondence

As stated in chapter 4, the correspondence establishment has three phases: first, parameterizing the reference shape and removing the area distortion and also parameterizing the rest shapes; second, sampling the landmarks from the reference parameterization and reference shape, then initializing the correspondence and propagating the parameterization by sampling the rest images the same as the

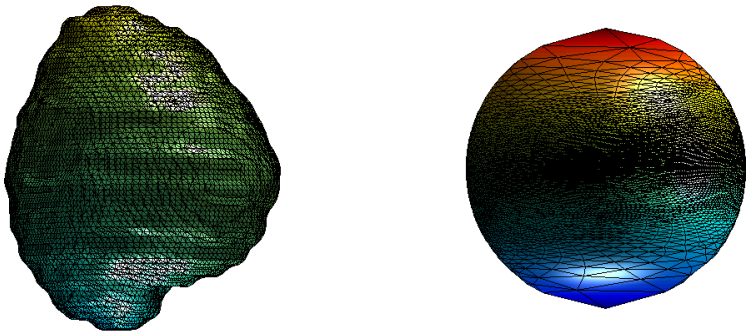
reference; third, optimizing the correspondence with MDL objective functions. We illustrate each phases separately.

6.3.2.1 Initializing Parameterizations

We arranged the training set in the order as: first two images are scans without needle, and the third to the eleventh images are scans with nine needle insertion directions, as section 6.2.2 states with the same label order. Then we chose the first scan, namely the image without the intervention, as the reference image. The reference has 13296 vertices and 26588 triangular faces on the surface initially. We performed the parameterization using the approach in section 4.3.1.1. The parameterized reference shape and its parameterization are shown in Figure 4.4. With the same procedure, we parameterized the remaining training images (second to eleventh). We take 2 examples to show the results, which are caudal central to left (Figure 6.9(a)) and caudal left to head central (Figure 6.9(b)) respectively. The same color in the shape surface and parameterization stands for the correspondence.



(a) Image 7 (caudal central to left).



(b) Image 11 (caudal left to central).

Figure 6.9: Original shape surfaces and Initial parameterizations

From Figure 6.9 it can be seen that the initial parameterizations are not area preserved, and the nodes are clustered near the 'equator' in the parameterization. Besides, the deformations caused by the needle insertion are shown clearly. Then we

only removed the area distortion in the reference shape (image 1). The result is in Figure 4.6. We did not remove the areal distortions in the initial parameterization of other images, but optimized them with respect to the reference in the second phase.

6.3.2.2 Initializing Correspondence

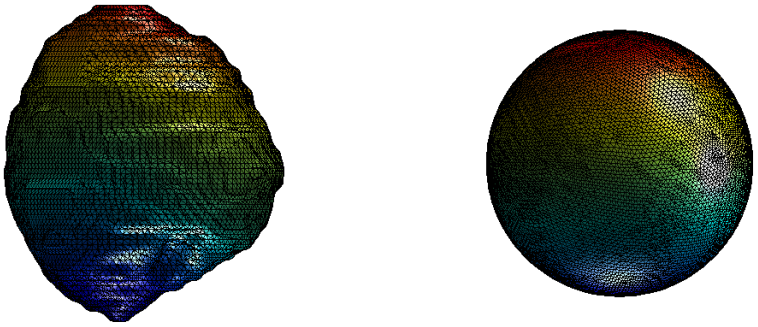
We firstly sampled the reference image parameterization with uniform sphere nodes derived from a 4 time subdivisions of an icosahedron. There were 2562 landmarks and correspondingly, the reference shape surface was also sampled as discussed in section 4.3.2.1. The results of sampled reference shape surface and sampled parameterization are shown in Figure 4.9

Next, we used the consistent parameterization to initialize the correspondence and propagate the sampling and correspondence afterwards. We found the vertices in all other un-sampled shape surfaces in the training set, which are closest to the sampled reference shape vertices. This formed the initial correspondence. The correspondence vertex indices were stored and the corresponding parameterization nodes with the same indices were extracted to form the objective function, as represented in section 4.3.2.2. As for the re-parameterization for manipulating the parameterizations, we chose a rotation adjustment in the longitude values, then a sigmoid transform and then 50 iterations of symmetric theta transformations with a fixed width of 0.2 and uniform sphere kernel centers derived from 1 time subdivision of an icosahedron. The total 42 kernel centers were visited one by one within one iteration. The whole procedure was processed pairwise, namely we adjusted the image one by one with respect to the reference. Two examples of the re-parameterized initial parameterizations are shown in Figure 6.10, which are still image 7 and 11. The same color labeled the same corresponding vertices between shape surfaces and parameterizations.

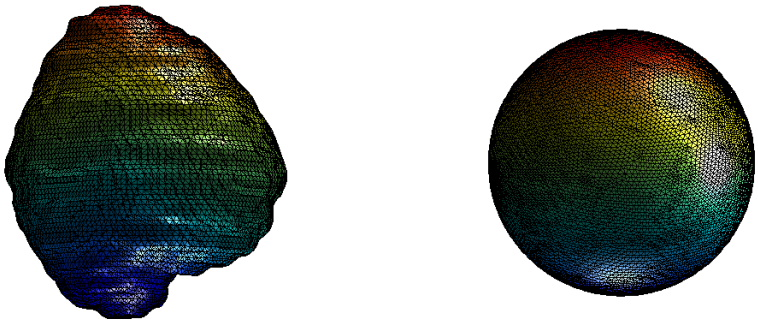
From Figure 6.10 we can see that these re-parameterization are effective to manipulate the parameterizations and the results are more area preserved, which are good for propagating the correspondence next. We used the same sampling as in the reference to propagate the correspondence between different shape vertices. After all the shape surfaces were sampled, the initial the correspondence was established and maintained by the same sampled parameterizations. The sampled shape surfaces and parameterizations of two examples are shown in Figure 6.11. This result shows the established initial correspondence with the same parameterizations. The shape details were not lost much and the number of landmarks was largely reduced and consistent.

6.3.2.3 Optimizing Correspondence

Once we obtained the initial correspondence, we started the correspondence optimization using the MDL objective function. We defined our MDL objective function with the full description length, as equation 4.31 represents. We followed



(a) Image 7 (caudal central to left).

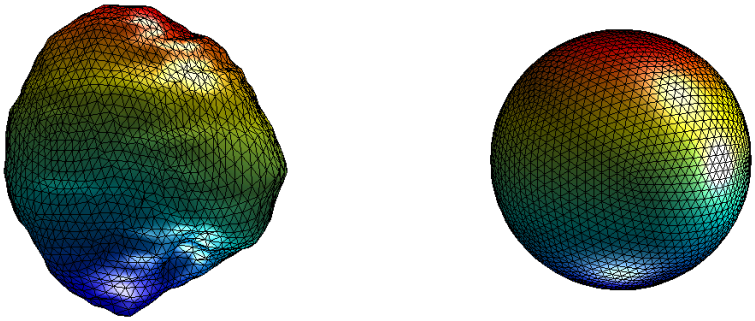


(b) Image 11 (caudal left to central).

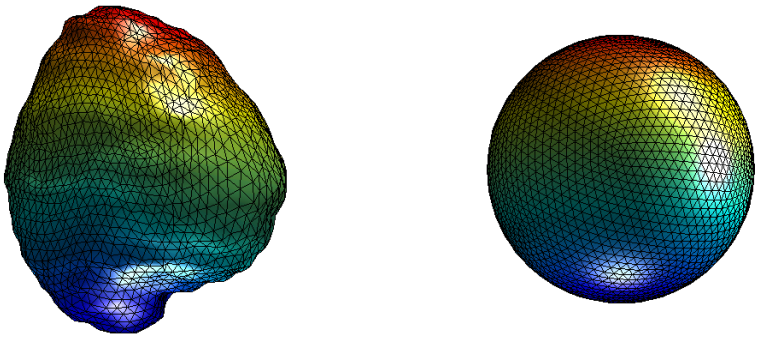
Figure 6.10: Original shape surfaces and re-parameterized initial parameterizations using consistent parameterization.

the suggestions in [86] and had 30 iterations for optimization totally. In the first 15 iterations, we selected the symmetric theta transformations for re-parameterizations with a fixed width of 0.25 and 18 uniformly distributed sphere kernel centers derived from a 1-time subdivision of an octahedron . The kernel centers were generated as Figure 6.12 shows. For the latter 15 iterations, the width started at 0.5 and was halved every 5 iterations (i.e. 0.5, 0.25, 0.125) and the kernels were increased from original octahedron vertices every 5 iterations by one more face subdivision (i.e. 8, 18, 66). In each iteration, the ten shape vertices (except the reference) were optimized successively. For one shape, the kernel centers were selected in turn for one group-wise evaluation of description length. By optimizing the amplitude in the symmetric theta transformation with respect to minimize the full description length, the correspondence was optimized.

After the correspondence optimization, we had the well labelled shape landmarks in each image for modeling the AAM. The correspondence optimized shape vertices and parameterizations of two examples (image 7 and 11) are shown in Figure 6.13. The correspondence optimized shapes show tidier distributions of the vertices mesh lines, compared with the ones after initial correspondence in Figure 6.11.



(a) Image 7 (caudal central to left).



(b) Image 11 (caudal left to central).

Figure 6.11: Sampled shape surfaces and sampled uniform parameterizations.

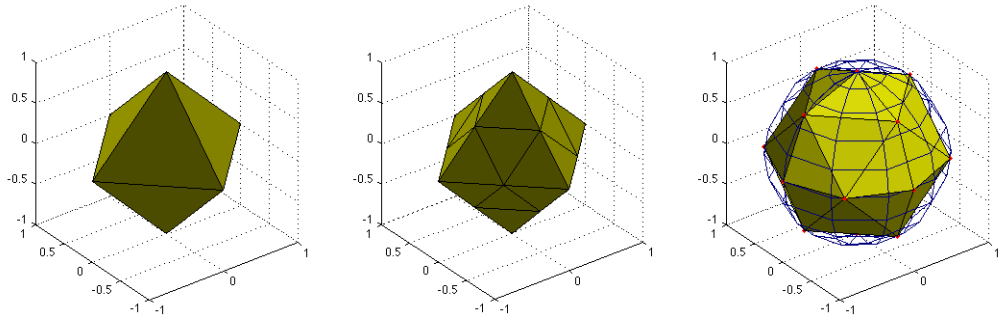
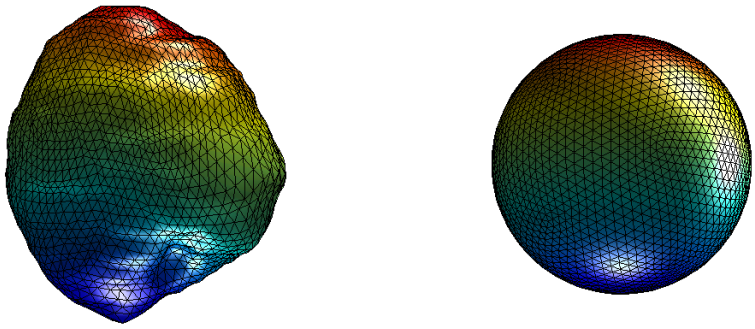
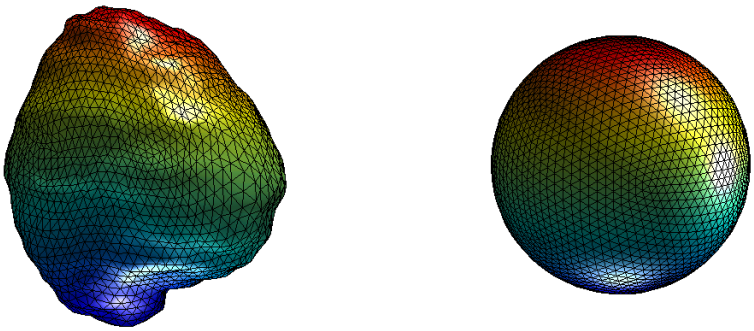


Figure 6.12: The derivation of uniform kernel center points from an octahedron



(a) Image 7 (caudal central to left).



(b) Image 11 (caudal left to central).

Figure 6.13: Optimized shape landmarks and parameterizations.

6.3.3 Building the Shape Model

Once we obtained the shape vertices with good correspondence we started to model the shape variance[1]. It includes the alignment of shape vertices and building the PDM with respect to the aligned shape landmarks.

6.3.3.1 Aligning the Shapes

We chose the unit quaternions based Procrustes alignment and set the reference as the mean shape at first. We followed the steps interpreted in section 3.2.1.1, and implemented 2 iterations for getting the mean shape and then obtained the aligned shape points set. The shape vertices before alignment and after alignment are shown in Figure 6.14. Each shape is assigned with one color, and we display them in a common coordinate space. It can be seen that the aligned shapes have more overlaps (darker colors), which means the shape variations were minimized by dismissing the influence from scaling, rotation and translation.

[1]D. Kroon, ASM and AAM. URL: http://www.mathworks.com/matlabcentral/fileexchange/26706-active-shape-model-asm-and-active-appearance-model-aam

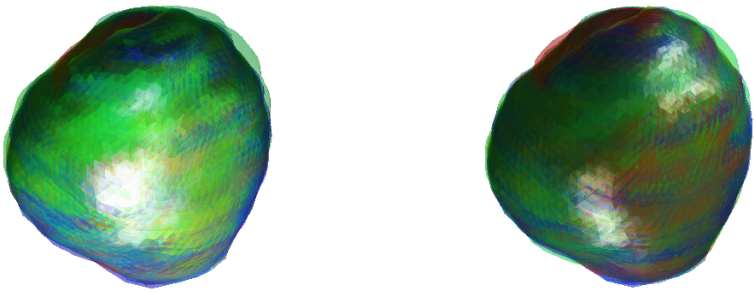


Figure 6.14: Left: 11 Shape surfaces before alignment. Right: 11 Shape surfaces after alignment.

6.3.3.2 Modeling Shape Variances

We calculated the eigenvectors using the SVD approach [102] instead of calculating the covariance matrix first as stated in section 3.2.1.2. Specifically, we stored the centered shape vectors (i.e., subtract each the aligned shape vector with the mean shape vector of all aligned shape vectors) in a 7686×11 matrix $\mathbf{X}$, where $7686 = 2562 \times 3$ is 3 times the landmark number, and 11 represents 11 shapes. Neither calculating $\mathbf{X}^T\mathbf{X}$ nor $\mathbf{X}\mathbf{X}^T$, we directly used SVD which held the equation $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$. Then by mathematical derivation, it has been proven that the column vectors of $\mathbf{U}$ are eigenvectors of the covariance matrix $\mathbf{X}\mathbf{X}^T$ and the square of diagonal elements of $\mathbf{S}$ are eigenvalues. The eigenvectors were in a descending order. We picked the 99% variance significance resulting in 10 modes. Figure 6.15 shows the first 3 modes as the shape parameter varies from $-3\sqrt{\lambda_s}$ to $3\sqrt{\lambda_s}$, where λ_s is the corresponding eigenvalue. From the variations we can see the different modes control different deformations. After the PDM was built, we store the mean shape vectors, the aligned shape vectors, eigenvectors and eigenvalues, mean transformation factors (scaling, rotation and translation).

6.3.4 Building Texture Model

We set the normalized mean aligned shape in 3 dimensions to *mapped base points*, where the texture size was set to $60 \times 60 \times 60$ and a margin of 5 voxels around the edge to give the voxel template. The voxel template was from the mean shape vector. Figure 6.16 shows the representation of mean shape vector, the voxel template, and three example slices of voxel template in three planes.

At the same time, we divided the mean shape mesh into the tetrahedrons for interpolating the corresponding voxels from each image. The tetrahedrons are shown in Figure 6.17

From the figure it is clear that some parts of the background were also included in the tetrahedrons. As a result, in the image warping from each shape to the *mapped base points*, the included background voxels were removed by masking

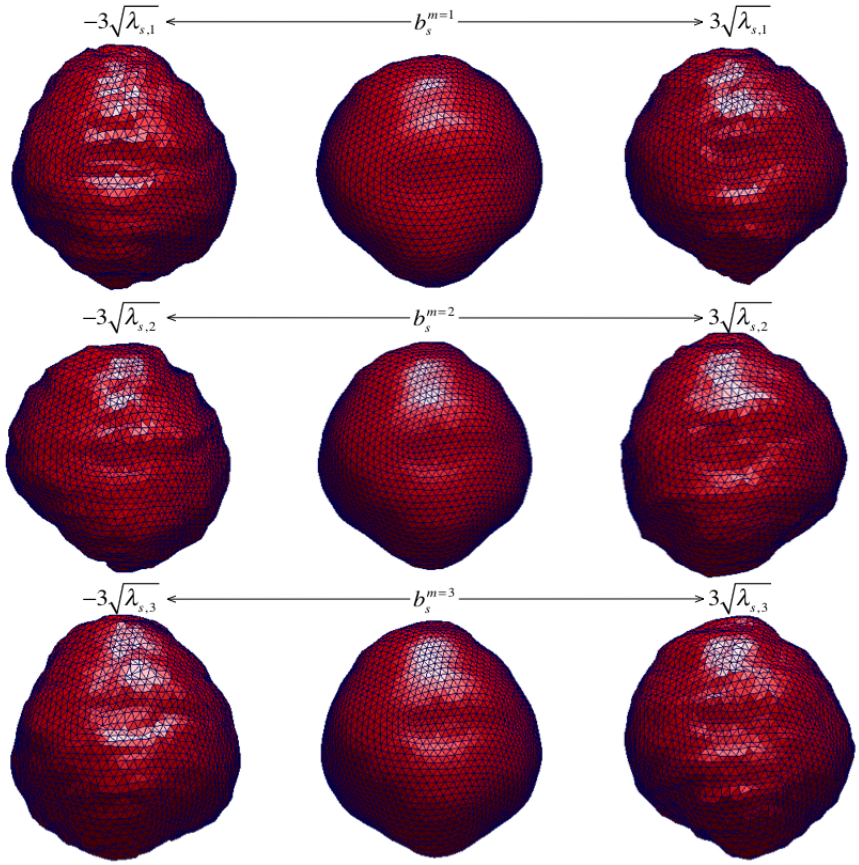
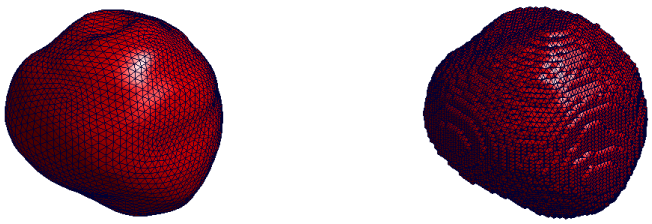
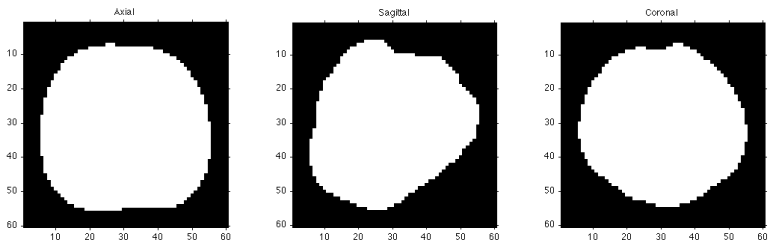


Figure 6.15: The first 3 modes of variation of $\pm 3\sqrt{\lambda_s}$ of the produced shape model



(a) Mean shape and Voxel template



(b) Voxel template slices

Figure 6.16: Voxel Template

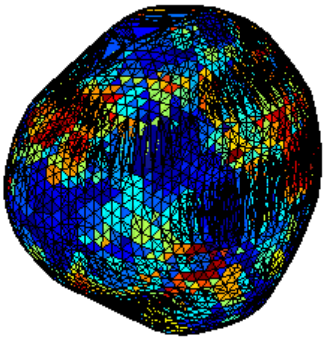


Figure 6.17: The tetrahedrons constructed from the surface mesh of mean shape

using the voxel template. Three examples slices in three planes of a warped image are shown in Figure 6.18, which in turn are image 1(reference), 7 and 11 respectively.

Only the regions of the voxel template were extracted and vectorized. Then they were normalized using the simple approach introduced in section 3.2.2.2. Similarly, the PDM was employed as well in texture model.

When both shape and texture models were built, we combined them together to form the appearance model, as introduced in section 3.2.3. In calculating the weight matrix, we chose to give a displacement in shape model parameter to find out the displacement of intensity, and use the mean of accumulated RMS to form the weight matrix diagonal element.

6.4 AAM Search Testing

The appearance model was built for AAM search. Before we implemented a test image, we built the parameter update model to give the linear relationship between parameter update and intensity displacement. Following the context in section 3.3.1, we used the numeric differentiation method to find the linear transform matrix **A**. Then we implemented the model in AAM fitting. For a same testing image, we use a 50 iterations full 3D search and a 50 iterations regional slices based search respectively. There were only 3 orthogonal slices extracted from the original image for testing, which were parallel to three MR planes and located in the place where the needle was inserted. In our implementation, we not only placed the three slices in their correct position, but also replicated the each slice twice and then put the two copies at two sides parallel to the original one with only one voxel distance. An illustration is shown in 6.19. [1]

From the illustration it can be seen that the intersection point of these slices was located in the position where the needle was inserted. It is reasonable to im-

[1]The left one states the actual appearance, and the right one increases the intervals between the parallel placed replicated slices only for annotating the relative position.

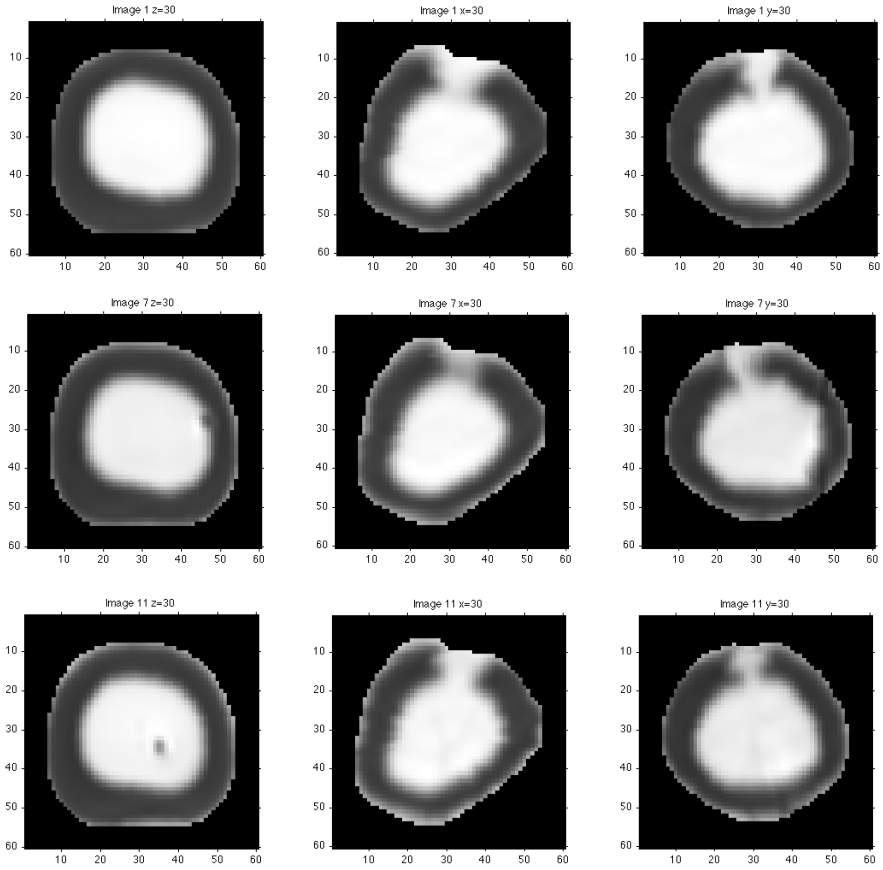


Figure 6.18: The warped image texture after voxel template casting. Left: Axial. Middle: Sagittal. Right: Coronal. First row: Image 1. Second row: Image 7. Third row: Image 11

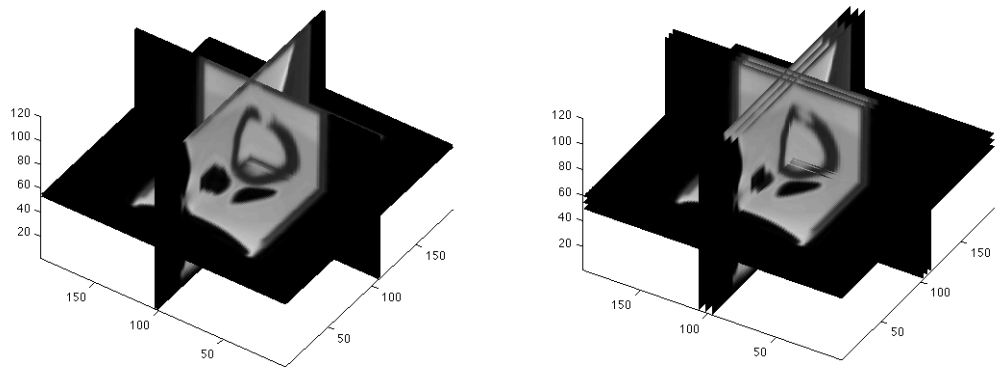


Figure 6.19: The example of 3 slices placement for 2D searching. Left: Actual placement. Right: Aggrandized slices intervals for illustration

plement the three slices like this since the slices are always for visualizing the needle.

The testing was validated using 2 schemes, leave-all-in and leave-one-out scheme. In the leave-all-in scheme, we use all the images for training and use the two approaches for segmenting each of them individually. This scheme is mainly for validating the algorithm. The leave-one-out scheme is mainly for testing the performance of these two approaches, where one image was left for testing and all others were for training. Therefore, we can perform both of the schemes for 11 folds. Figure 6.20 shows two examples of the segmented results using full 3D searching and 2D fitting by leave-all-in scheme. Figure 6.21 shows the segmented result using both approaches by leave-one-out scheme. The testing image in the example is image 7. From the geometric result we can see the leave-all-in validation gave a more similar segmentation compared with the original shape.

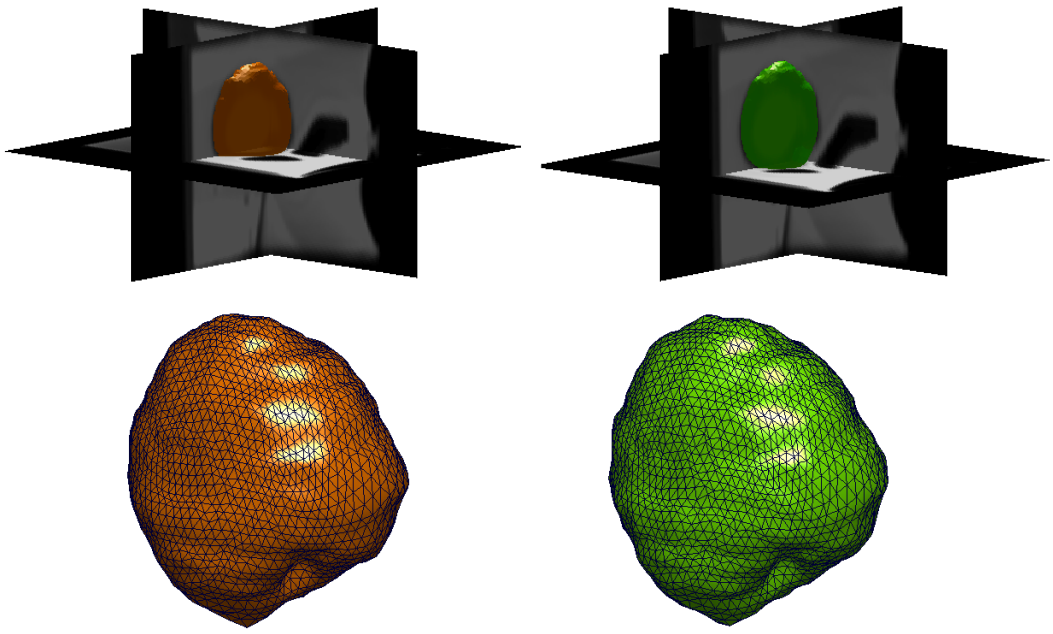


Figure 6.20: The segmentation of image 7 by leave all in validation. Left: Using full 3D AAM search. Right: Using slices based 2D regional AAM search

We also calculated the Dice Similarity Coefficient (DSC), the Absolute Distance (AD), its mean (MAD), and the image distance map to evaluate the performance of model based segmentation with 3D standard AAM search and 2D slices based AAM search. The DSC equals to two times the amount of overlapping voxels between the manual segmentation and the results from AAM search, divided by the sum of the amount of voxels in both the manual segmentation and the searching result. The AD, MAD and distance map are concerned with the Euclidean distance between the segmented shape surfaces and the original the surfaces. We assume the correspondence have been established then the surface distance can then be

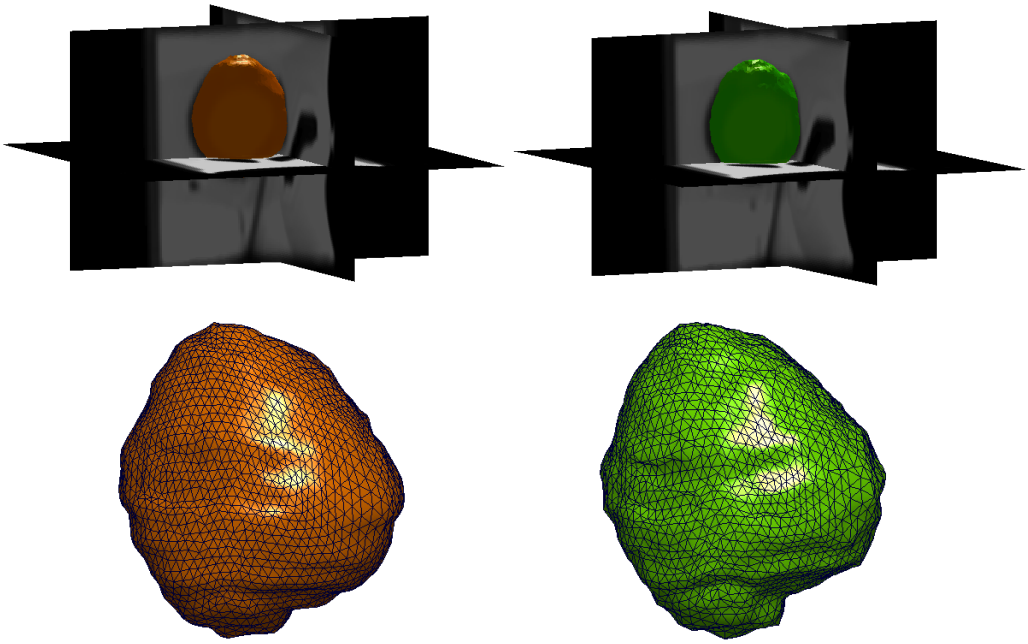
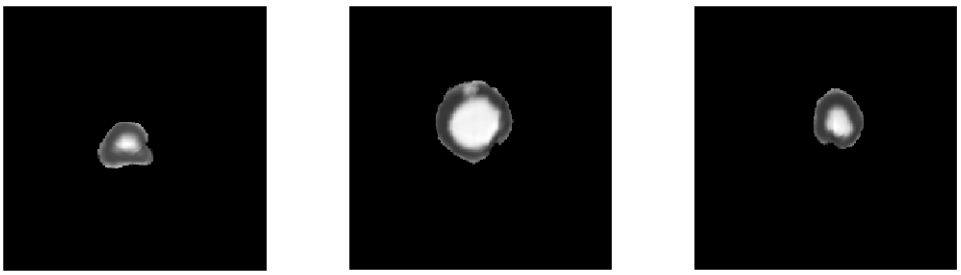


Figure 6.21: The segmentation of image 7 by leave one out validation. Left: Using full 3D AAM search. Right: Using slices based 2D regional AAM search

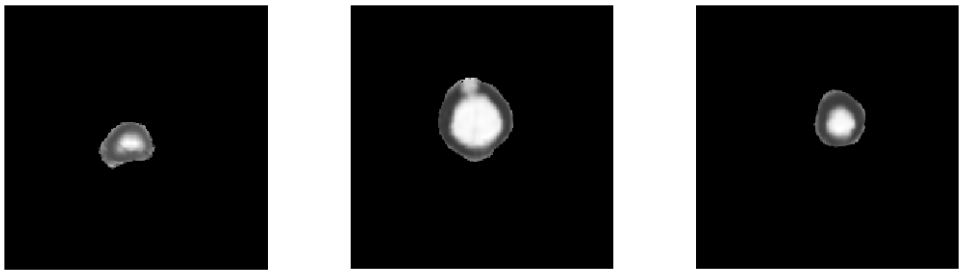
measured using corresponded vertices. These metrics would be discussed in next chapter for evaluation.

After the segmentation using 2D slides AAM search, we can reconstruct the 3D prostate representation using the segmented surface and the texture statistics. Normally if the full 3D testing image is available, the texture information can be segmented using the obtained shape surface. As a result, there is no point reconstructing the texture information from the model. However, if we only have 3 slices available, then the prostate tracking should all be based on the model, and the texture can be rebuilt. Two examples of the reconstructed texture slices are shown in Figure 6.22, which are from the reconstructed texture model of image 7 in leave-all-in scheme and leave-one-out scheme respectively using 2D slices based AAM search. We can construct the 3D representation by stacking the slices together. Figure 6.23 shows three arbitrary slices constructed in 3D space.

We calculated the mean texture difference for evaluation of reconstructed textures. The texture distance is the absolute Euclidian distance between two texture maps, which are both segmented by a same surface. The surfaces we used is from the 2D slices based AAM search in both leave-all-in scheme and leave-one-out scheme. We masked the original 3D texture with these surfaces to get the standard texture, and calculate the absolute texture difference between this standard one and the reconstructed one from AAM. The evaluation would be discussed in next chapter.

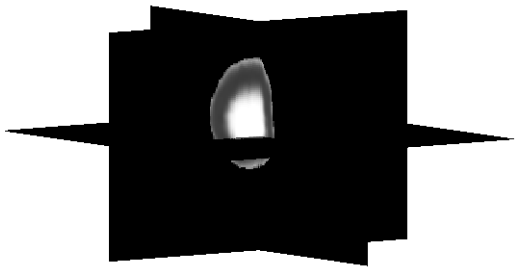


(a) Texture reconstruction in leave-all-in scheme.

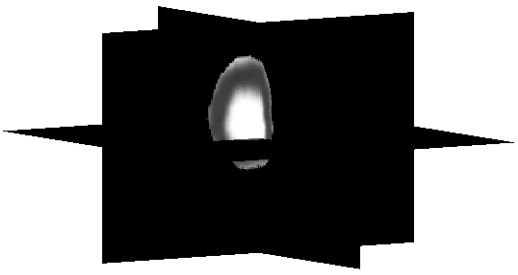


(b) Texture reconstruction in leave-one-out scheme

Figure 6.22: The reconstruction of image 7 by slices based AAM search. Left: Axial plane. Middle: Coronal plane. RIght: Sagittal plane



(a) 3D reconstruction in leave-all-in scheme.



(b) 3D reconstruction in leave-one-out scheme.

Figure 6.23: The 3D reconstruction of image 7 by slices based AAM search.

Chapter 7

Results Evaluation and Discussion

In this chapter we will give the results with evaluations of the implementation from the last chapter. Considering the objectives, we mainly evaluate three issues: the performance of the phantom to mimic the biopsy; the performance of AAM to represent the geometric changes; the segmentation accuracy for both full 3D model search and the 2D slices based model search.

7.1 Phantom Property

In the experiments, we have scanned the 3D images. From the results, we can evaluate the phantom bio-mechanical property in the needle insertion and the visibility and compatibility in MRI, From the T2-weighted scans, such as Figure 6.5, we can see the structure of the phantom clearly, where the darker parts are the frame, rectum, pubis and prostate capsule that have a higher stiffness. The Lighter parts are the soft tissues and the inner prostate. Besides, we did not see obvious artifacts so that the visibility is good and compatible with MRI. There are some parts with much lighter part in the surrounding soft tissues resulting from some inappropriate proportion of ingredients such as a higher percentage of softener. However, it would not affect much so that the experiments were all based on this phantom.

We have also examined the bio-mechanical property, namely checking whether there are some visible deformation and motions and if any whether they are comparable with the practical observations. For viewing a real time prostate mobility, we used an interactive, real-time, multi-slice TrueFISP sequences (BEAT_IRTTT) with an Interactive Front End (IFE) prototype, which is a dedicated interactive navigation system in interventional MRI. Figure 7.1 shows two instances before and after the needle was inserted into the prostate. The biopsy needle was inserted perpendicular to the axial plane, approximately.

Because we used a faster sequence, though it was T2-weighted scan, the artifacts

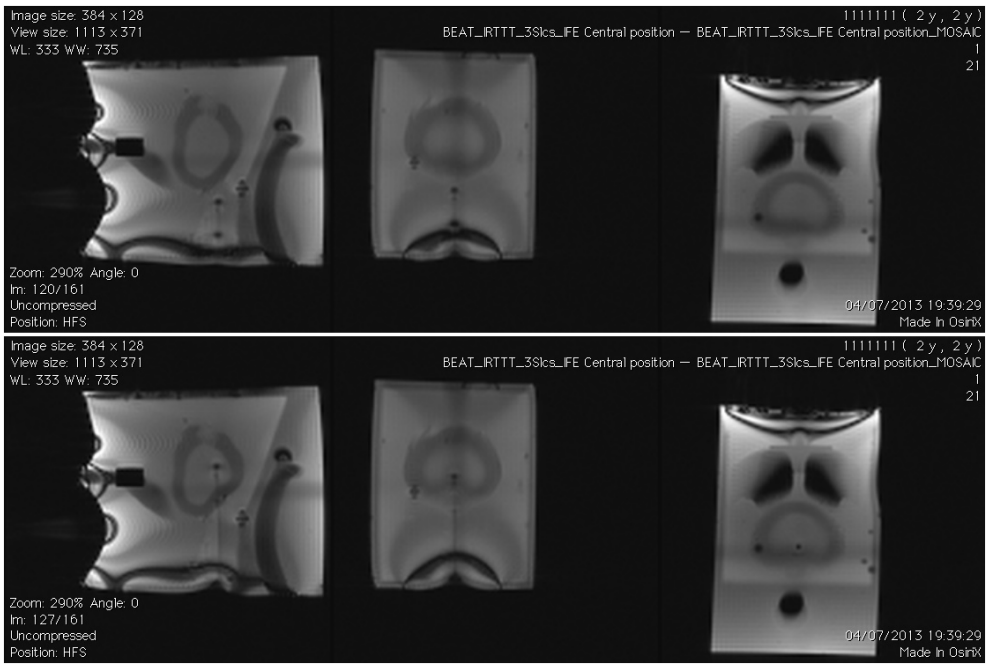


Figure 7.1: The real time slices in sagittal (Left), coronal (Middle), and axial (Right) planes before (Upper) the insertion and after (Lower) the insertion

from the air bubbles in the phantom, the needle, the outer plexiglass and probably the pubic bone are clear in these slices. Actually, in the biopsy, this sequence would be implemented, but for the 3D image acquisition, the previous one will be employed.

A clear deformation in the position can be seen where the needle was inserted. Also, in the opposite side (In this example near the prostate base, both anterior and posterior parts) there was a slight deformation. The total translation is not obvious, but the displacement in the insertion point is obvious. We quantitively measured the displacement using the segmented shape vertices, the displacement statistic curve, the mean and maximum displacements are shown in Figure 7.2 and Tabel 7.1 respectively. The average displacements are a little bit less than the proposed $3 - 10$ mm in both the real time patient radiotherapy [101] and Hungr's phantom [39], but comparable. Besides, it fits the maximum displacement, which makes the phantom sufficient. This may result from the surrounding material is not soft enough. So we suggest a softer surrounding part to fill the phantom.

Table 7.1: Maximum displacement

Image	3	4	5	6	7	8	9	10	11
Maximum(mm)	4.97	5.01	6.67	6.67	7.78	4.96	5.90	6.89	5.84
Mean(mm)	1.80	1.51	2.01	2.93	3.25	1.89	1.57	1.66	1.65

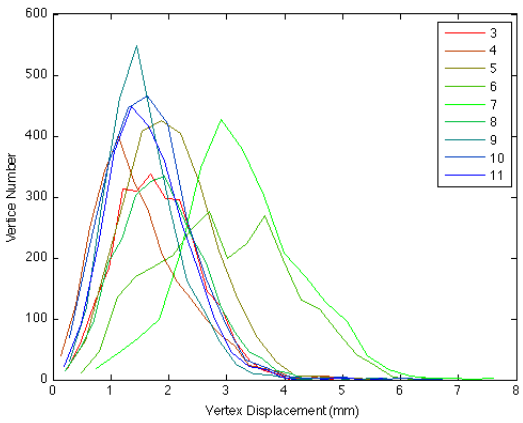


Figure 7.2: The displacement between the cases with and without intervention (reference shape).

7.2 Correspondence

In chapter 6 we have learnt the correspondence problem is important to AAM performance. We used the parameterization based approach for establishing the correspondence. From the principles in chapter 6, it can be seen that the correspondence is determined by the topology of parameterization in the sampling sphere. A significant procedure is to manipulate the mesh in the original parameterization more area preserving and similar with each other. We chose a simple rotation followed by sigmoid transformation and an iterative symmetric theta transformation to achieve this. The objective function measuring the sum squared Euclidean distances of corresponding vertices can be rapidly minimized from over 600 to 50 and the iterative procedure can further minimize it. If we only apply the Cauchy kernel based re-parameterization, the process would take long and hard to get minimized once it is under 100. The examples of the resulting sampled shape and its original parameterization in this case are shown in Figure 7.3.

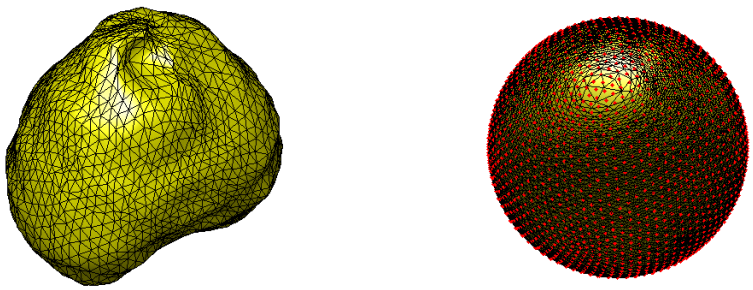


Figure 7.3: Sampled shape vertices and initial parameterizations of image 11 with labelled sampling points, when sigmoid transformation is not applied in consistent parameterization.

From the example we can see that the sampled shape vertices is clustered on the top (and bottom), which is not the case in the reference. In this condition, the correspondence is not well built, and the there would be illegal shapes as varying the shape parameters. Figure 7.4 shows the correspondence comparison between the cases of single cauchy kernel re-parameterization and multiple re-parameterizations in the consistent parameterizations. This correspondence is between reference and image 11.

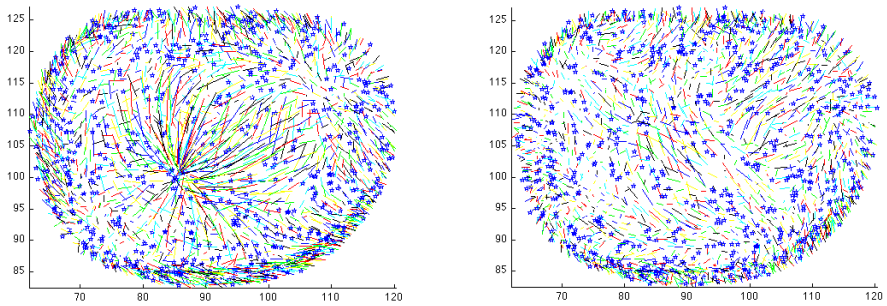


Figure 7.4: Left: Correspondence between reference image and image 11 using single re-parameterizaiton. Right: Correspondence between reference image and image 11 using single re-parameterizaitons.

In Figure 7.4, the short lines connect the corresponding points. In the left figure it clearly shows the bad correspondence concentrating in the center, namely the top and bottom of the prostate shape. In the right one, there is no such a phenomenon.

The correspondence was optimized using the MDL objective functions. We used a uniform kernel centers and a fixed width for each iteration. The description length was minimized gradually, and the correspondence was optimized as well. Figure 7.5 shows the MDL for each iterations. We totally implemented 30 iterations.

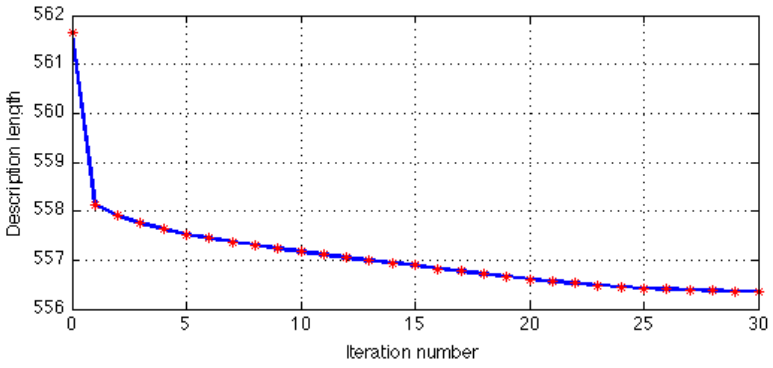


Figure 7.5: The Minimum description length in each iteration.

Figure 7.6 shows the correspondence among the training set. The reference shape and image 7 and 11 are taken as examples and the corresponding points are labeled with same colors.

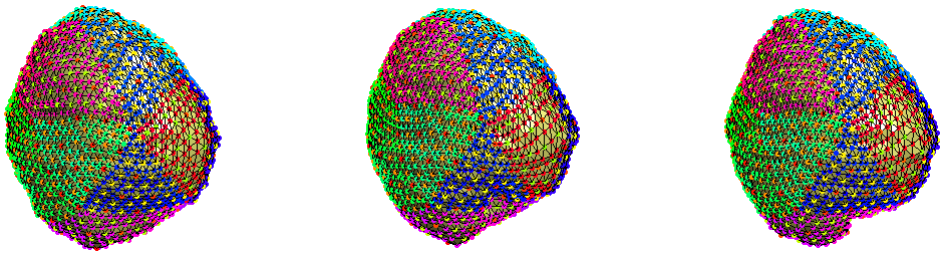


Figure 7.6: The shape vertices correspondence across the data set. Left: Image 1(reference shape). Middle: Image 7. Right: Image 11

As proposed by Davies *et al.* [86], it is more effective by optimizing one kernel in turn since the optimization by controlling a bunch of kernel re-parameterizaiton is difficult and inefficient. This optimization approach has a big drawback that running time is long. The optimization for 11 cases with 30 iterations would take over days and nights. This is also the reason why some researchers preferred a covariance matrix objective function or an approximate MDL objective function. The recommendations in [79] suggest a gradient descent optimization that is computational readily. Considering the process is offline so that the computational problem would not be the biggest bottleneck. However if the approach would considerably speed up the optimization with respect to the same or better result, it would be quite preferred.

Another problem we have found is that the MDL based optimization performs worse when some vertices are gathered as clusters. As we have discussed, when the initialization of correspondence fails to manipulate parameterization well distributing on the sphere, there would be some parts, such as the top and bottom, clustering with shape vertices. Once this happens, it will not be a unique case but spread to other shapes. In this condition the MDL optimization tends to gather the points much more to make the description length become shorter. Obviously this is not what we want. This happens more often when the shape is not spherically topological. So we can conclude that the MDL based optimization is very sensitive to the clustering of shape vertices, and it is more suitable for the spherical shaped objects, such as the prostate.

7.3 Shape Segmentation

By implementing AAM trained with these data set, the system can learn the shape and texture variance and segment the similar object given a new image. As stated in the last chapter, we used two approaches for segmentation: the full 3D AAM search

and three slices based AAM search. The leave-all-in and leave-one-out scheme were also applied. The evaluation would be conducted for the two approaches in each scheme in turn.

We used the metrics DSC, AD, MAD and distance map for results evaluation, which have been employed in MICCAI 2012. The DSC should range from zero to one, where zero represents no overlaps between the original segmentation (Manual segmentation) and the model based segmentation. In previous research, the mean DSC ranges from 0.66 to 0.88 [22, 23, 25, 103, 104, 105] using MRI modality. The DSCs in the leave-all-in scheme and leave-one-out scheme are listed in Table 7.2 for each image. The mean and standard deviation were calculated as well.

Table 7.2: DSCs for each image evaluated for segmentation using full 3D AAM fitting and 2D AAM fitting.

	Leave-All-In		**Leave-One-Out**	
Image	DSC using 3D Fitting	DSC using 2D Fitting	DSC using 3D Fitting	DSC using 2D Fitting
1	0.88978	0.8909	0.89031	0.89127
2	0.88842	0.89188	0.85979	0.88027
3	0.88703	0.88598	0.88888	0.88654
4	0.87639	0.84166	0.86736	0.84762
5	0.88531	0.87647	0.86264	0.89505
6	0.886	0.87789	0.87541	0.84915
7	0.88712	0.87331	0.88156	0.86386
8	0.88772	0.89104	0.88446	0.88397
9	0.88688	0.88488	0.88469	0.88225
10	0.88671	0.88563	0.88228	0.87536
11	0.88551	0.88817	0.87738	0.87898
Mean	0.8861	0.8807	0.8777	0.8758
Median	0.8869	0.8856	0.8816	0.8803
SD	0.0035	0.0144	0.0104	0.0158

From the table we can see the leave-all-in scheme generally performs better than the leave-one-out. This is proper since the system has known the testing cases in leave-all-in scheme, while it has not in leave one out scheme. Besides, it is noticeable that the DSC in this data set is high, however, it does not represent that the segmentation is accurate. In this phantom experiment we have noted that the shape deformation and displacement are minor, though visible. In addition, all the images are generated from the same model, which means they do not differ much in shape. As a result, even if the AAM fails and as such as a mean shape is given, the DSC could also be high enough. Therefore it can be concluded that DSC cannot be a significant factor of evaluating the result.

The absolute distances were calculated based on the corresponding shape vertices pairwise. We averaged the Euclidian distances over all the vertices for each

image and obtained each absolute distance. We calculated the mean over the 11 images and got the MAD. Table 7.3 shows the the ADs and MADs for both the 3D fitting and 2D fitting in each scheme.

Table 7.3: ADs and MADs for each image evaluated for segmentation using full 3D AAM fitting and 2D AAM fitting. (Unit:mm)

	Leave-All-In		**Leave-One-Out**	
Image	AD using 3D Fitting	AD using 2D Fitting	AD using 3D Fitting	AD using 2D Fitting
1	1.53	1.56	1.56	1.62
2	5.78×10^{-6}	0.46	1.41	1.41
3	6.16×10^{-8}	0.408	0.876	1.21
4	3.98×10^{-5}	1.29	1.67	2.02
5	2.28×10^{-5}	0.415	2.22	1.41
6	3.26×10^{-6}	0.55	1.46	2.19
7	4.16×10^{-5}	0.683	2.74	5
8	2.38×10^{-7}	0.514	1.06	1.44
9	6.44×10^{-5}	0.247	0.952	1.16
10	2.71×10^{-6}	0.579	0.834	0.8
11	1.05×10^{-6}	0.446	0.865	1.38
Mean	0.139	0.65	1.42	1.79
SD	0.461	0.403	0.613	1.13

From this table we can see more clearly that the 3D AAM search was valid on most condition, and the 2D slices based AAM fitting provided an acceptable result in leave-all-in scheme, which shows the feasibility of application in model searching. In leave-one-out scheme, both the two model searching approaches performed equivalent results, which are acceptable but worse than the results in leave-all-in validation. In [23] and [103], the MAD are $1.44(\pm0.48)mm$ and $1.32(\pm1.53)mm$; wheres in [104] and [106], the MAD are $3.97(\pm2.74)mm$ and $2.41mm$, and even in [104] it is indicated that the traditional statistical shape model based method gave a $4.05(\pm2.74)mm$ of MAD. In this case both of 3D fitting and slices based 2D fitting have shown satisfying results.

Nevertheless, we care the distances in different regions in the prostate much more, since we aim at representing the 3D deformation by model searching. Hence we cannot only examine the AD and MAD, but should also examine the surface error, namely the vertex distance for each case. We have examined the distance map in the leave one out scheme for both approaches (3D and 2D fittings). Figure 7.7 shows the distance map of image 6 in leave-all-in scheme (first row) and in leave-one-out scheme (second row). Figure 7.8 shows the same but for image 7. We can see that the results in leave-all-in scheme performed much better with respect to the vertex distance map, from which the deformations in the position of needle insertion were detected and the errors were low. Whereas for the leave-one-out scheme, with respect to image 6 and 7, the deformation has not been detected

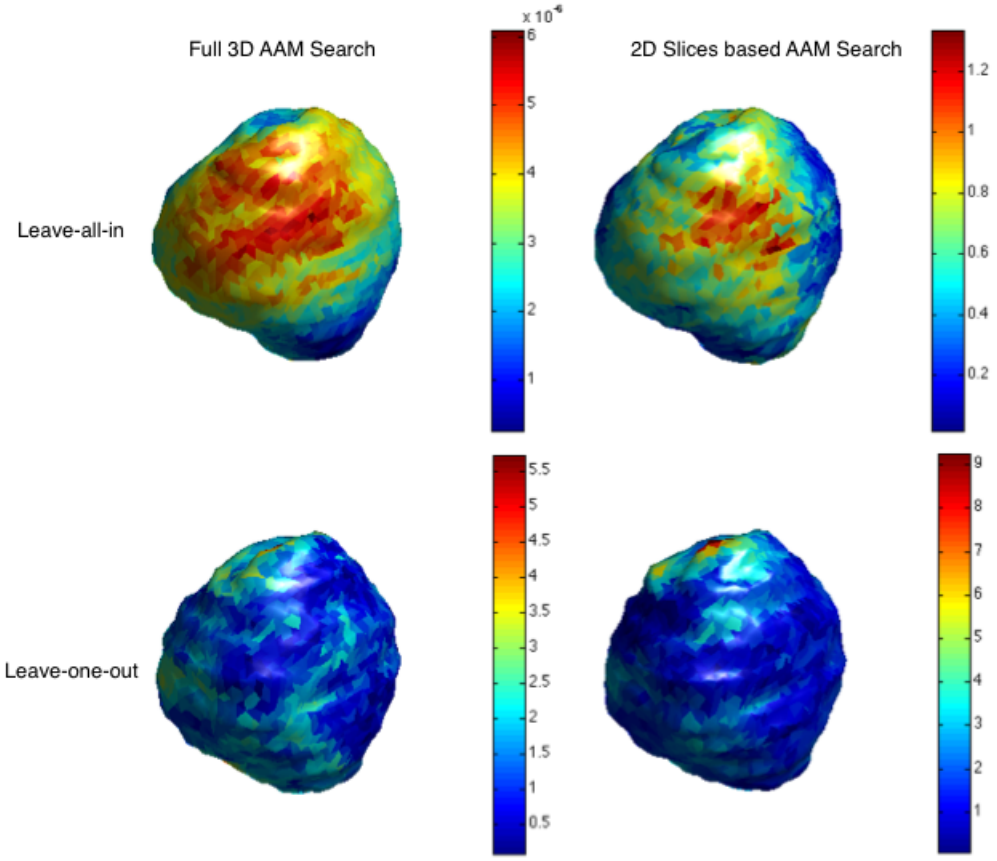


Figure 7.7: The distance map of image 6.

well. An interesting phenomenon is that the segmentations of image 7 using both approaches resemble the case image 11 which has shown many times as example in previous chapters. We infer this happened because the image 7 and 11 are similar. As it is known, AAM is a statistical model so it always fits the object using the learnt information. When image 7 was for testing in leave-one-out scheme, image 11 was used for training, therefore, an image 11-like result was produced. As for the image 6, the shape is more unique, so the learnt information cannot give the correct model parameters to construct the model. Also this result does not resemble any one in the training set. We have looked through all of our results in both schemes, this inference can be validated. In summary, to avoid the problem happened in leave-one-out scheme, the training set should be large enough to cover most of the situations. A major problem in our experiment is the training set is too small, which is insufficient for AAM to learn the deformation.

Another abnormal result is the segmentation from image 1. Normally the leave-all-in scheme, at least for full 3D AAM search, can give a good segmentation. But

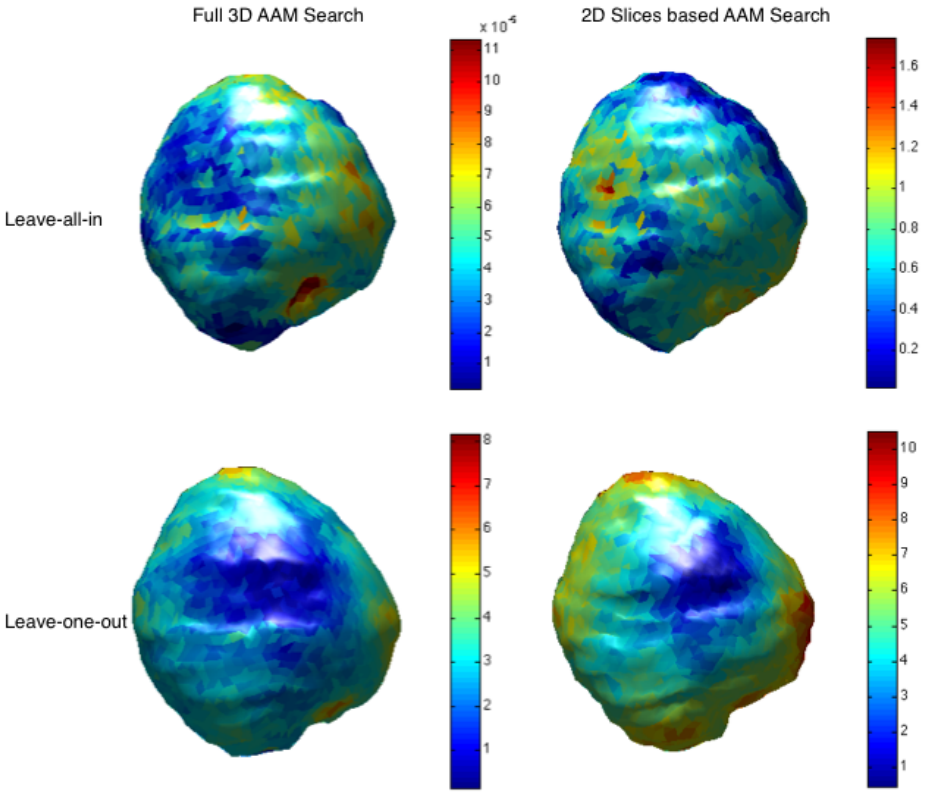


Figure 7.8: The distance map of image 7.

here the AD is higher than other results. We think it is because of the problem in correspondence. We have assumed that the correspondence was well established. However, we find that the correspondence between the shapes except the reference is much better than between the reference shape and other shapes. Due to our optimization on the shapes except the reference, the correspondence would be better for those shapes of which the vertices can be freely manipulated. However the vertices and parameters of reference shape were fixed during the initialization and optimizaiton, the correspondence would be worse between other shapes and the reference. Thus the correspondence problem still existed. Although, this would not be a big problem since the segmentation can be accurate which can be seen from the high DSC of reference shape. This problem can be minimized by enlarging the data set and optimizing the correspondence further.

We have mentioned the positions of the slices taken for testing are at the the needle insertion point. Although it satisfies the real condition in biopsy, we have also verified that this positions are optimum since the most shape and texture difference information can be included, which is of significance for 2D fitting. We tested the other positions such as in the middle of each axis and so forth, even the leave-all-in validation failed to get the appropriate segmentation, which led to a

more general result, like a mean shape. In fact, the slices are not always parallel to the MRI planes. Mostly they will be orthogonal to each other and the needle lies in the intersectional line. Actually, the slice which is perpendicular to the needle will not contribute much to the deformation information, so that it can be placed in the middle or near the needle. No matter which slices are used, the essence is to include the most deformed part.

One more notice is the replication of these three slides and the mean texture background were for getting more realistic intensities at the positions of the given slices. From chapter 3, it is known that the intensity in the corresponding position is obtained from interpolation. If there are only three slides and the background is black, the actual intensities extracted for comparison with the modeled ones would be deeply influenced by the background. Therefore we chose to use a mean intensity map for filling the background of the testing image. Apart from that, only three slides are insufficient to give 'real' information, so we supposed the texture in a very short range is similar. Thus the copies of the original slides can play the role of neighbor slides. It was proven that this implementation worked better than either using only black background or using only three slices in the 2D fitting.

To summarize, the 2D slices based AAM search is feasible for the segmentation. Although it is worse than the full 3D fitting, the results are acceptable. As the training set becomes bigger, more possible cases can be included, which can make the results in leave-one-out scheme tend to the leave-all-in validation.

7.4 Texture Reconstruction

As the segmentations were taken, the corresponding texture information can also be obtained. It is meaningless to evaluate the texture obtained from the full 3D AAM search, since the texture information in the testing data must be available if the image can be searched in 3D space. Therefore we only discuss the reconstructed textures using the slides based AAM fitting, in terms of both leave-all-in scheme and leave-one-out scheme. We calculate the mean absolute offset between the textures masked with the same shape segmentation to evaluate the gray level difference. Table 7.4 shows the mean difference and standard deviation for each testing image in each scheme.

From the result we see that all the mean texture differences are small and the leave-one-out scheme shows higher mean differences in most cases. Since the segmentations in the leave-one-out scheme are not as good as the leave-all-in validation, there are more texture variation. The intensity differences between the results validate the remarks on the segmentation performance. To be more specific for the evaluation of reconstructed texture, we examined the texture difference map of both the slice position where the testing slices were given and the slices in other positions. Figure 7.9 shows an comparison with the texture map in different schemes at different positions. We picked image 7 as example, and used the 2D

Table 7.4: Texture difference for each image evaluated for segmentation by leave-all-in validation and leave-one-out validation.

	Leave-All-In		**Leave-One-Out**	
Image	Mean	SD	Mean	SD
1	0.0013027	0.024487	0.0074165	0.065187
2	0.0013041	0.025069	0.0073844	0.065339
3	0.0012416	0.024215	0.0074348	0.064994
4	0.0012249	0.01972	0.0068594	0.062985
5	0.0012696	0.022429	0.0074653	0.065527
6	0.001338	0.023468	0.0071104	0.064822
7	0.0013806	0.023875	0.0070439	0.063589
8	0.0013267	0.025	0.0075215	0.06588
9	0.0013628	0.024846	0.0073075	0.064168
10	0.00148	0.025602	0.007319	0.064559
11	0.0013803	0.025925	0.0074865	0.065672
Mean	0.0013	0.0241	0.0073	0.0648
SD	0.0001	0.0017	0.0002	0.0009

slices based fitting. It shows the reconstructed texture slices and the original texture using the masks from shape segmentations at axial 54^{th} slice and 70^{th} slice respectively. The texture difference maps are shown as well.

It can be noted that the reconstructed textures have a better resemblance for leave-all-in scheme and the most different parts are always near the contours. The slices in the position of the given slices have a lower texture difference than the slices in other places. However, most of the results show the reconstruction of texture using the 2D fitting is feasible and comparable to the results of the 3D searching.

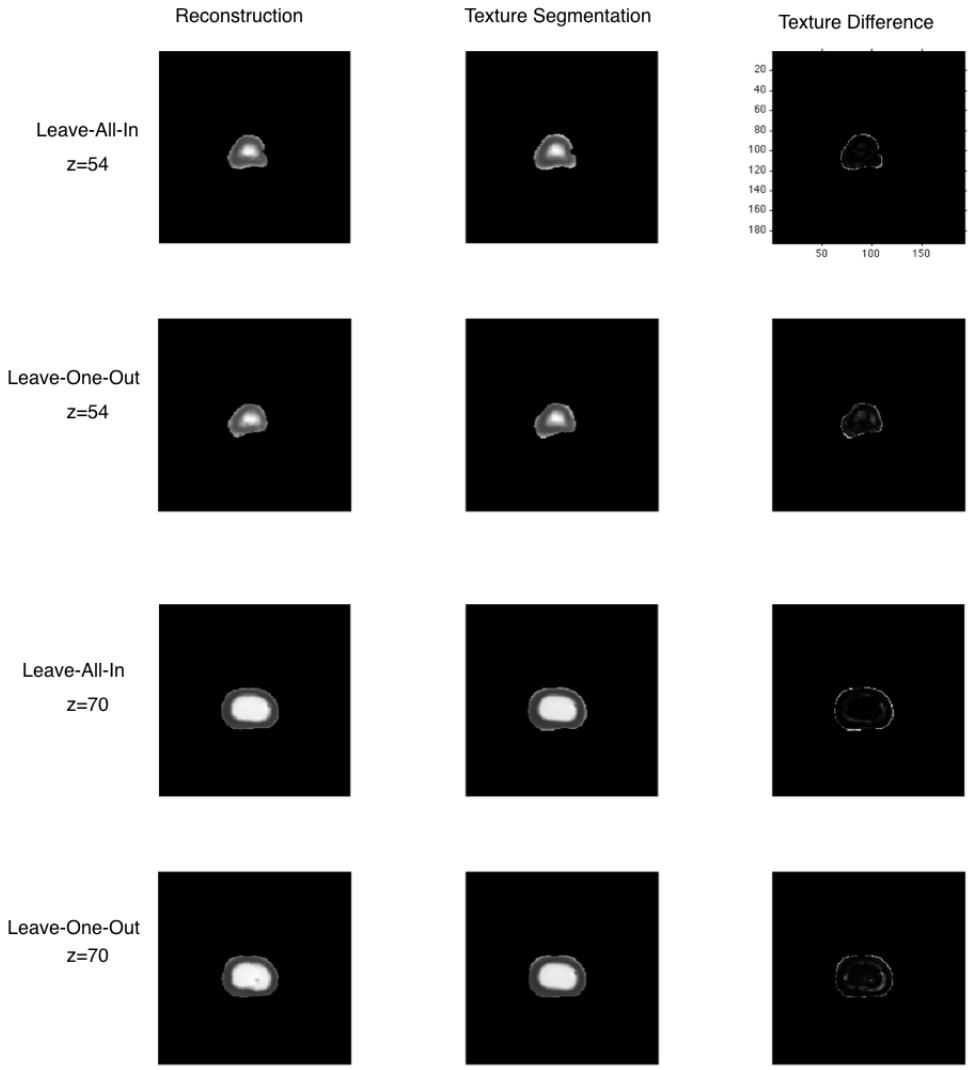


Figure 7.9: The comparison between the reconstructed textures and original ones. z=54 is the position of given image slices for segmentation, and z=70 has not been given. In the first two columns, black is 0 and white is 1, and in the last column 0 is grey

Chapter 8

Conclusion and Recommendations

8.1 Summary

In the previous chapters we have introduced the project background, problems and goals. By summarizing the previous works we put forward the problems and objectives in the project. We finally decided to use an anthropomorphic deformable prostate phantom to simulate the biopsy for acquiring the 3D images for training, and use 3D AAM for tracking the prostate in the biopsy only using the 2D slices based AAM search. For AAM specifications, we reviewed the basics about alignments, shape and texture variance statistics. Then, we explained the standard AAM search in detail. Based on the standard approach and the previous knowledge we proposed the approach for AAM search relying on 3 slices. We have implemented all of these, and obtained a satisfying prostate phantom and a reliable AAM training system. We have applied the leave-all-in and leave-one-out schemes for validation of the AAM system and evaluation of our proposed approach. By showing the segmentation and reconstruction results with quantitive evaluations we gave our experimental conclusions. Also, for some abnormal observations and unsatisfying results we have given the possible explanations and suggestions.

8.2 Remarks

As in chapter 1, we have stated the problems and given the research questions and objectives. Now we examine whether we have achieved the objectives in our experiments and implementations. Firstly, we made the anthropomorphic prostate phantom with soft PVC, which could give the phantom sufficient elastic deformations during the intervention. With different concentrations, the phantom has different bio-mechanical properties in different parts. The prostate model and pubis model were made based on real patient MR segmentations and CT scans and the relative positions were determined under the doctor's suggestions. By examining the MR images, we concluded that the T2-weight scans showed the good

visibility and MR compatibility. In both the BEAT_IRTTT sequence and full 3D sequence the deformations were obvious and the displacements were minor. The motion under intervention was clear in the real time sequence. The data resolution and acquisition time were acceptable. The collected MR images can be easily implemented for 3D image processing. By doing all of these, the objective '*Make an anthropomorphic deformable prostate phantom to simulate prostate motion in needle intervention and collect MR image data with respect to prostate deformations.*' is accomplished.

Secondly, in terms of the other objective, we built the statistic data set using multi-direction needle insertions. For each image, shape and texture information was collected by a manual shape segmentation. By means of a spherical parameterization based MDL group-wise correspondence optimization, we built the shape data variance statistic model using PCA. With a 3D image warping method the texture of different scans can be associated and modeled statistically. The active appearance model can be established by combining the shape model and the texture model. We improved the standard model search approach which conventionally uses 3D image for shape and texture segmentation, with respect to three orthogonal slices that contain more information of shape deformations. By means of searching the intensities on these 2D slices, the parameters for adjusting the shape, transformation, and modeled intensities were updated, until the difference between modeled intensity and the searched intensities were minimized. We used the leave-all-in method for validation and it proved the 2D slices based AAM fitting is convergent and the expected shape and texture can be generated with a small acceptable error distance. Though in the leave-one-out verification, the performance was not as good as that in leave-all-in scheme. Therefore, the objective '*build active appearance models using the collected data and segment the prostate by means of three orthogonal 3D slices based model search*' is not completely achieved. However, the main problem is that the data set is too small for the AAM system to learn most of the shape and texture variations. The verification can be developed using more data set such as building a new phantom and choosing more practical insertion points to collect the data as many as possible. We can anticipate the segmentation results would be better with more training images presence.

Now we will see with these mostly achieved objectives, whether the research questions can be answered.

1 *Is it feasible to learn the prostate deformation and motion knowledge in a 3D space?*

  We planned to use the prostate phantom to simulate the possible motions in the MR guided intervention and indeed we have seen a satisfying dynamic property in the MRI based intervention. This observation stimulated us to aim at learning the rules of motion and deformation of a general prostate. For the sake of making the shape more representative, we designed the prostate shape from a real case. The multi-direction needle insertions were designed for including most of the cases. To learn the prostate deformation and the motion in the

biopsy, we want to generalize the situation from the phantom to the real case. However, what we have is the AAM trained from only a phantom and the limited experimental insertions. In consequence, the obtained statistical models worked well with the phantom, but could not be applicable in the real case. There are several reasons for this. First, PCA extracts the principal axes that represents the most shape point variance. But the shape, though, is almost the same (because all the shapes are from the same phantom), the correspondence cannot be perfect since the image registration approach could be problematic and the segmentation for each image was done manually which always has a huge amount of deviated labeling. When the shape points difference in the deformed part has an equivalent magnitude with the variation in other points caused by false manual segmentation, the principal components for the deformation would be sinked in the false shape variations. In this case, if we apply the shape principal components to the real case, the generated shape would be problematic. Another problem is that the texture pattern of the phantom is way too much different with the MRI scan of male pelvis. In conclusion, the motion and deformation were simulated, and the shape variance of phantom could be learnt if the noise can be neglected.

2 *Is it feasible to segment the prostate based on three orthogonal slices in a 3D MR environment?*

As we discussed above, we have implemented the 2D slices based AAM search for segmenting the shape and texture in 3D space. The leave-all-in validation has proven the feasibility of prostate segmentation depending on the 3D AAM training and 2D fitting. However, the reliability of this approach has not been verified for the leave-one-out scheme. As a result, we need to collect sufficient training data set that has a precise manual segmentation and well establish the shape points correspondence. Another problem is the slices for testing were orthogonal and parallel to the three MR planes (axial, sagittal and coronal). While in the biopsy implementation, it could not be absolute orthogonal and also not parallel to the three planes. Theoretically this problem can be solved as long as the slices include most of the deformable part and the AAM system is precisely employed, which requires the phantom to be simulated the deformation correctly. Moreover, when we compare the 3D T2-weighted MR images with the image slices from BEAT_IRTTT sequence, we can see a lot of differences mainly in intensities. Because the real time sequence is very fast, more artifacts could be observed which can significantly influence the phantom image quality. This would lead to a bad model search with problematic intensities. In fact, in human tissue there would only be the texture difference between the intensities. But for the phantom, since it's hard to get rid off all the air bubbles so the artifact could be a problem in a faster sequence. In summary, prostate segmentation depending on three orthogonal slices is feasible, but for the practical verifications it requires more work.

In short, we have used the phantom to mimic the motion of prostate in biopsy, and tested the deformation by means of using AAM modeling. We successfully

reconstructed the 3D representation with the aid of three orthogonal slices. For a generalization from the phantom to the real case, the current work cannot guarantee a good performance.

8.3 Recommendations

According to the possible problems, we give our suggestions for approaching to our objectives further and answering the research questions thoroughly. Also, these could be the future works.

With respect to the intensity differences in the 3D data and the real time slices, a bubble-less phantom is required. This could be achieved by either advancing the fabrication art, or developing a new material. In addition, the gray level difference can be found by some intensity transformation or normalization. Moreover, the surrounding part should be a little bit softer to ensure a range of displacement.

As for the generalization from the phantom case to the real case, we should figure out the general principal axes for only shape deformations, which, in other words, aims at minimizing the influence from the shape correspondence problem on the AAM. There are mainly two aspects considering this problem as stated in last section. Firstly, due to a high gray level contrast between the capsule and the surrounding tissue in the phantom, we could find out an automatic way of segmenting the contour of prostate instead of manual slices based segmentation, such as using canny edge detector. This application would highly decrease the error in some blurred part, where it is likely to cause a subjective error in placing the contour points. Secondly, the descending gradient MDL correspondence optimization can be employed for a faster convergence, or use an alternative approach for correspondence establishment such as shape context registration. No matter which approach is used, the shape variance in undeformed parts must be minimized extremely.

Furthermore, a more comprehensive experiment should be setup. Developing more directions for needle insertion could lead to an expectable improvement. Not only differing in the needle positions, but also set up the multiple force experiments. Then the training set can become larger for leave-one-out validation. On the basis of some researches (e.g.[107]), the mechanical property and the prostate displacement or deformation have been related for prediction. Therefore it is possible to measure the relationship between the deformation and the force in the needle, and the Young's modulus of the phantom quantitively. Initially, we planed to build the dynamic AAM for one step forward predicting the deformed prostate shape and texture. However, as it was discussed, the motion of prostate during the 3D data acquisition is difficult. It is preferable to figure out a way of capturing the motion in 3D scans, since it will bring a lot of advantages. For instance, firstly it is possible to implement the prediction such as using Kalman filter which has been employed by statistical shape models (e.g. [108]) or using the active appearance motion models

(e.g.[109]). Secondly, the prediction can also be helpful when combined with the model search methods. The proposed 2D fitting is still based on the standard AAM search. But we can imagine that the insertion points are always near the apex of prostate, which allows us to build a deformation position probability model. These probabilities could help with the initialization of model fitting, such as by means of a random forest regression voting mechanism [110], which could be more accurate than the standard approach.

Moreover, the suitability of the PCA in AAM for our task should be examined further. As we all know, the deformations are caused by the needle insertion, namely the external forces. As a result, the deformations could not suffice a Gaussian distribution, which means PCA could not give an optimum modeling. Former studies have shown some statistics of the prostate displacement but they are rarely summarized into some statistical models. From some results we sometimes can see an approximate Rice distribution instead of the Gaussian. In this case, we could also try to implemented the ICA instead of PCA in AAM modeling.

Finally, it is always a good option to verify the system using the real cases, namely the image slices from patients. Considering the lack of prostate textures in the phantom, the implementation of real cases would be more desirable.

Due to the limitation in time, we cannot implement these for testing. For the further research, reviewing and following these recommendations would be a good start.

Bibliography

[1] F. Rabbani, N. Stroumbakis, B. R. Kava, M. S. Cookson, and W. R. Fair, "Incidence and clinical significance of false-negative sextant prostate biopsies", The Journal of urology, vol. 159, no. 4, pp. 1247-50, Apr. 1998.

[2] http://www.cmi-nen.nl/research/projects/DONTmiriam/

[3] P. C. Vos, "Computer aided diagnosis of prostate cancer with magnetic resonance imaging", Radboud University Nijmegen, The Netherlands, Sept. 2011.

[4] J. J. Fütterer, S. W. T. P. J. Heijmink, T. W. J. Scheenen, J. Veltman, H. J. Huisman, P. Vos, C. A. Hulsbergen-Van de Kaa, J. A. Witjes, P. F. M. Krabbe, A. Heerschap, and J. O. Barentsz, "Prostate cancer localization with dynamic contrast-enhanced MR imaging and proton MR spectroscopic imaging", Radiology, vol. 241, no. 2, pp. 449-58, Nov. 2006.

[5] S. Ghose, A. Oliver, R. Mart, X. Llad, J. C. Vilanova, J. Freixenet, J. Mitra, D. Sidib, and F. Meriaudeau, "A survey of prostate segmentation methodologies in ultrasound, magnetic resonance and computed tomography images", Computer Methods and Programs in Biomedicine, vol. 108, no. 1, pp. 1-44, Oct. 2012.

[6] Y. J. Liu, W. S. Ng, M. Y. Teo, and H. C. Lim, "Computerised prostate boundary estimation of ultrasound images using radial bas-relief method", Medical & biological engineering & computing, vol. 35, no. 5, pp. 445-54, Sep. 1997.

[7] C. K. Kwoh, M. Y. Teo, W. S. Ng, S. N. Tan, and L. M. Jones, "Outlining the prostate boundary using the harmonics method". Medical & biological engineering & computing, vol. 36, no. 6, pp. 768-71, Nov. 1998.

[8] R. G. Aarnink, S. D. Pathak, J. J. de la Rosette, F. M. Debruyne, Y. Kim, and H. Wijkstra, "Edge detection in prostatic ultrasound images using integrated edge maps", Ultrasonics, vol. 36, no. 1-5, pp. 635-42, Feb. 1998.

[9] S. D. Pathak, V. Chalana, D. R. Haynor, and Y. Kim, "Edge-guided boundary delineation in prostate ultrasound images", IEEE transactions on medical imaging, vol. 19, no. 12, pp. 1211-9, Dec. 2000.

[10] P. Abolmaesumi and M. R. Sirouspour, "Segmentation of prostate contours from ultrasound images", in Proceedings of IEEE International Conference

on Acoustics, Speech, and Signal Processing, IEEE Computer Society Press, 2004, pp. 517-520.

[11] C. Knoll, M. Alcan, V. Grau, C. Monserrat, and M. C. Juan, "Outlining of the prostate using snakes with shape restrictions based on the wavelet transform", Pattern Recognition and Image Analysis, vol. 32, pp. 1767-1781, 1999.

[12] H. M. Ladak, F. Mao, Y. Wang, D. B. Downey, D. A. Steinman, and A. Fenster, "Prostate segmentation from 2d ultrasound images", in Proceedings of the 22nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE Computer Society Press, 2000, pp. 3188-3191.

[13] M. Ding, C. Chen, Y. Wang, I. Gyacskov, and A. Fenster, "Prostate segmentation in 3D US images using the cardinal-spline basd discrete dynamic contour", in Proceedings of SPIE Medical Imaging: Visualization, Image-Guided Procedures, and Display, SPIE, 2004, vol. 5029, pp. 2568-2571.

[14] A. Jendoubi, J. Zeng, and M. F. Chouikha, "Segmentation of Prostate ultrasound images using an improved snakes model", in Signal Signal Processing, 2004. Proceedings. ICSP04. 2004 7th International Conference, 2004, vol. 3, pp. 2568-2571.

[15] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models - their training and application", Computer Vision and Image Understanding, vol. 61, no. 1, pp. 38-59, Jan. 1995.

[16] T. F. Cootes and C. J. Taylor, "Active shape models – 'smart snakes'", in Proceedings of British Machine Vision Conference, 1992, pp. 639-648.

[17] T. F. Cootes and C. J. Taylor, "Active shape model search using local grey-level models: a quantitative evaluation", Proceedings of the British Machine Vision Conference 1993, pp. 64.1-64.10, 1993.

[18] T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam, "Use of active shape models for locating structures in medical images", Image and vision computing, vol. 12, pp. 355-366, 1994.

[19] Y. Zhu, S. Williams, and R. Zwiggelaar, "A hybrid ASM approach for sparse volumetric data segmentation", Pattern Recognition and Image Analysis, vol. 17, no. 2, pp. 252-258, Jun. 2007.

[20] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 6, pp. 681-685, Jun. 2001.

[21] M. Kirschner, F. Jung, and S. Wesarg, "Automatic Prostate Segmentation in MR Images with a Probabilistic Active Shape Model". in PROMISE 12: MICCAI 2012 Grand Challenge on Prostate MR Image Segmetation, 2012.

[22] B. Maan and F. van der Heijden, "Prostate MR image segmentation using 3D active appearance models", in PROMISE 12: MICCAI 2012 Grand Challenge on Prostate MR Image Segmetation, 2012.

[23] G. Vincent, G. Guillard, and M. Bowes, "Fully Automatic Segmentation of the Prostate using Active Appearance Models", in PROMISE 12: MICCAI 2012 Grand Challenge on Prostate MR Image Segmetation, 2012.

[24] S. Klein, U. a. van der Heide, I. M. Lips, M. van Vulpen, M. Staring, and J. P. W. Pluim, "Automatic segmentation of the prostate in 3D MR images by atlas matching using localized mutual information", Medical Physics, vol. 35, no. 4, pp. 1407-1417, 2008.

[25] J. Dowling, J. Fripp, S. Chandra, J. P. W. Pluim, J. Lambert, J. Parker, J. Denham, P. B. Greer, and O. Salvado, "Fast automatic multi-atlas segmentation of the prostate from 3d mr images", in Prostate Cancer Imaging. Image Analysis and Image-Guided Interventions, vol. 6963, A. Madabhushi, J. Dowling, H. Huisman, and D. Barratt, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 10-21.

[26] B. Maan, F. van der Heijden, and J. J. Fütterer, "A new prostate segmentation approach using multispectral magnetic resonance imaging and a statistical pattern classifier", Medical Imaging 2012: Image Processing, vol. 8314, p. 83142Q-83142Q-9, Feb. 2012.

[27] S. Martin, V. Daanen, and J. Troccaz, "Atlas-based prostate segmentation using an hybrid registration", International Journal of Computer Assisted Radiology and Surgery, vol. 3, no. 6, pp. 485492, Sep. 2008.

[28] A. Tsai, A. Yezzi, W. Wells, C. Tempany, D. Tucker, A. Fan, W. E. Grimson, and A. Willsky, "A shape-based approach to the segmentation of medical imagery using level sets", IEEE transactions on medical imaging, vol. 22, no. 2, pp. 137-54, Feb. 2003.

[29] R. Toth, P. Tiwari, M. Rosen, G. Reed, J. Kurhanewicz, A. Kalyanpur, S. Pungavkar, and A. Madabhushi, "A magnetic resonance spectroscopy driven initialization scheme for active shape model based prostate segmentation", Medical image analysis, vol. 15, no. 2, pp. 214-25, Apr. 2011.

[30] A. Firjani, A. Elnakib, F. Khalifa, G. L. Gieml, M. A. El-Ghar, J. Suri, A. Elmaghraby, and A. El-Baz, "A new 3D automatic segmentation framework for accurate segmentation of prostate from DCE-MRI", Internetional Symposium on Biomedical Imaging: From Nano to Macro, IEEE, pp. 14761479, 2011.

[31] H. S. S. Ho, P. Mohan, E. D. Lim, D. L. Li, W. S. Ng, W. K. O. Lau, and C. W. S. Cheng, "Robotic ultrasound-guided prostate intervention device: system description and results from phantom studies", The international Journal of Medical robotics and computer assisted surgery, vol. 5, pp. 51-58, 2009.

[32] R. Wilkin and R. Hamm, "How to make a cheap and simple prostate phantom", Journal of Ultrasound in Medicine, vol. 29, pp. 1151-1152, 2010.

[33] V. P. Heikkilä and N. Suorsa, "A technique for simultaneous needle insertion in prostate seed implantation", Physics in medicine and biology, vol. 53, no. 4, pp. N35-9, Feb. 2008.

[34] B. C. Porter, D. J. Rubens, J. G. Strang, J. Smith, S. Totterman, and K. J. Parker, "Three-dimensional registration and fusion of ultrasound and MRI using major vessels as fiducial markers", IEEE transactions on medical imaging, vol. 20, no. 4, pp. 354-9, Apr. 2001.

[35] S. Xu, J. Kruecker, B. Turkbey, N. Glossop, A. K. Singh, P. Choyke, P. Pinto, and B. J. Wood, "Real-time MRI-TRUS fusion for guidance of targeted prostate biopsies", Computer aided surgery: official journal of the International Society for Computer Aided Surgery, vol. 13, no. 5, pp. 255-64, Sept. 2008.

[36] E. Dehghan, X. Wen, R. Zahiri-Azar, M. Marchal, and S. E. Salcudean, "Modeling of needle-tissue interaction using ultrasound-based motion estimation", Medical image computing and computer-assisted intervention : MICCAI International Conference on Medical Image Computing and Computer-Assisted Intervention, vol. 10, no. Pt 1, pp. 709-16, Jan. 2007.

[37] J. Sherman, T. K. Podder, V. Misic, L. Fu, D. Fuller, B. Winey, E. M. Messing, D. J. Rubens, J. G. Strang, R. Brasacchio, and Y. Yu, "Efficacy of prostate stabilizing techniques during brachytherapy procedure", Conference proceedings : Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference, vol. 1, pp. 563-6, Jan. 2006.

[38] T. Podder, D. Clark, J. Sherman, D. Fuller, E. Messing, D. Rubens, J. Strang, R. Brasacchio, L. Liao, W.-S. Ng, and Y. Yu, "*In vivo* motion and force measurement of surgical needle intervention during prostate brachytherapy", Medical Physics, vol. 33, no. 8, pp. 2915-2922, 2006.

[39] N. Hungr, J.-A. Long, V. Beix, and J. Troccaz, "A realistic deformable prostate phantom for multimodal imaging and needle-insertion procedures", Medical physics, vol. 39, no. 4, pp. 2031-41, Apr. 2012.

[40] B. Maurin, L. Barbe, B. Bayle, P. Zanne, J. Gangloff, M. De Mathelin, A. Gangi, L. Soler, and A. Forgione, "In vivo study of forces during needle insertion", In Preceedings of the Medical Robotics, Navigation and Visualization Scientific Workshop, 2004, pp. 1-8.

[41] H. Kataoka, T. Washio, and K. Chinzei, "Measurement of the tip and friction force acting on a needle during penetration", in Medical image computing and computer-assisted intervention : MICCAI, 2002, pp. 216-223.

[42] V. Lagerburg, M. a Moerland, J. J. W. Lagendijk, and J. J. Battermann, "Measurement of prostate rotation during insertion of needles for brachytherapy", Radiotherapy and oncology : journal of the European Society for Therapeutic Radiology and Oncology, vol. 77, no. 3, pp. 318-23, Dec. 2005.

[43] T. A. Krouskop, T. M. Wheeler, F. Kallel, B. S. Garra, and T. Hall, "Elastic Moduli of Breast and Prostate Tissues under Compression", Ultrasonic Imaging, vol. 20, no. 4, pp. 260-274, Oct. 1998.

[44] S. Phipps, T. H. J. Yang, F. K. Habib, R. L. Reuben, and S. a McNeill, "Measurement of tissue mechanical characteristics to distinguish between benign and malignant prostatic disease", Urology, vol. 66, no. 2, pp. 447-50, Aug. 2005.

[45] M. Zhang, P. Nigwekar, B. Castaneda, K. Hoyt, J. V Joseph, A. di Sant' Agnese, E. M. Messing, J. G. Strang, D. J. Rubens, and K. J. Parker, "Quantitative characterization of viscoelastic properties of human prostate correlated with histology", Ultrasound in medicine & biology, vol. 34, no. 7, pp. 1033-42, Jul. 2008.

[46] J. Kemper, R. Sinkus, J. Lorenzen, C. Nolte-Ernsting, A. Stork, and G. Adam, "MR elastography of the prostate: Initial in-vivo application", Rofo. Fortschritte auf dem Gebiet der Rontgenstrahlen und der bildgebenden Verfahren, vol. 176, no. 8, pp. 1094-1099, 2004.

[47] T. H. J. Yang, S. K. W. Leung, S. Phipps, R. L. Reuben, S. a McNeill, F. K. Habib, a Schnieder, and R. Stevens, "In-vitro dynamic micro-probing and the mechanical properties of human prostate tissues", Technology and health care : official journal of the European Society for Engineering and Medicine, vol. 14, no. 4-5, pp. 281-96, Jan. 2006.

[48] E. J. Chen, J. Novakofski, W. K. Jenkins, and W. D. O' Brien, "Young's modulus measurements of soft tissues with application to elasticity imaging", IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control, vol. 43, no. 1, pp. 191-194, Jan. 1996.

[49] K. J. Parker, S. R. Huang, R. M. Lerner, F. Lee, D. Rubens, and D. Roach, "Elastic and ultrasonic properties of the prostate", in Proceedings of IEEE Ultrasonics Symposium, 1993, pp. 1035-1038.

[50] T. F. Cootes and C. J. Taylor, "Statistical models of appearance for computer vision", Imaging Science and Biomedical Engineering, University of Manchester, 2004.

[51] T. Cootes and C. Taylor, "Combining elastic and statistical models of appearance variation", in Proceedings of European Conference on Computer Vision, 2000, pp. 149-163.

[52] G. J. Edwards, T. F. Cootes, and C. J. Taylor, "Advances in active appearance models", in Conference on Computer Vision, 1999, pp. 137-142.

[53] S. C. Mitchell, J. G. Bosch, B. P. F. Lelieveldt, R. J. van der Geest, J. H. C. Reiber, and M. Sonka, "3-D active appearance models: segmentation of cardiac MR and ultrasound images", IEEE transactions on medical imaging, vol. 21, no. 9, pp. 1167-78, Sep. 2002.

[54] G. Vincent and C. Wolstenholme, "Fully automatic segmentation of the knee joint using active appearance models", Medical Image Analysis for the Clinic: A Grand Challenge, pp. 224 -230, 2010.

[55] D.-J. Kroon, P. Kowalski, W. Tekieli, E. Reeuwijk, D. Saris, and C. H. Slump, "MRI based knee cartilage assessment", in Proc. SPIE 8315, Medical Imaging 2012: Computer-Aided Diagnosis, 2012, vol. 8315, pp. 83151V-83151V-10.

[56] D. J. Kroon, "Segmentation of the mandibular canal in cone-beam CT data", Doctoral dissertation, in Signals and Systems group, EEMCS Faculty, University of Twente, 2011.

[57] T. F. Cootes, G. V. Wheeler, K. N. Walker, and C. J. Taylor, "View-based active appearance models", Image and Vision Computing, vol. 20, no. 9-10, pp. 657-664, Aug. 2002.

[58] T. Cootes, G. Wheeler, K. Walker, and C. Taylor, "Coupled-View Active Appearance Models", British Machine Vision Conference, vol. 1, pp. 52-61, 2000.

[59] B. P. F. Lelieveldt, M. Üzümcü, R. J. van der Geest, J. H. C. Reiber, and M. Sonka, "Multi-view active appearance models for consistent segmentation of multiple standard views: application to long- and short-axis cardiac MR images", International Congress Series, vol. 1256, pp. 1141-1146, Jun. 2003.

[60] M. Üzümcü, R. J. van der Geest, M. Sonka," Multiview active appearance models for simultaneous segmentation of cardiac 2-and 4-chamber long-axis magnetic resonance images", Investigative radiology, vol 40, no.4, pp. 195-203, 2005.

[61] E. Oost, G. Koning, M. Sonka, P. V Oemrawsingh, J. H. C. Reiber, and B. P. F. Lelieveldt, "Automated contour detection in X-ray left ventricular angiograms using multiview active appearance models and dynamic programming", IEEE transactions on medical imaging, vol. 25, no. 9, pp. 1158-71, Sep. 2006.

[62] J. Hansegård, S. Urheim, K. Lunde, and S. I. Rabben, "Constrained active appearance models for segmentation of triplane echocardiograms", IEEE transactions on medical imaging, vol. 26, no. 10, pp. 1391-400, Oct. 2007.

[63] K. Y. E. Leung, M. van Stralen, G. van Burken, N. de Jong, and J. G. Bosch, "Automatic active appearance model segmentation of 3D echocardiograms", 2010 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, pp. 320-323, 2010.

[64] P. Schönemann and R. Carroll, "Fitting one matrix to another under choice of a central dilation and a rigid motion". Psychometrika, vol. 35, no. 2, pp. 245 - 255, 1970.

[65] M. D. Akca, "Generalized procrustes analysis and its applications in photogrammetry", Swiss Federal Institute of Technology, 2003.

[66] B. Horn, "Closed-form solution of absolute orientation using unit quaternions", Journal of the Optical Society of America A, vol. 4, no. April, 1987.

[67] B. Horn, H. Hilden, and S. Negahdaripour, "Closed-form solution of absolute orientation using orthonormal matrices", Journal of the Optical Society of America A, vol. 5, no. 7, pp. 1127-1135, 1988.

[68] M. B. Stegmann, "Active appearance models: Theory, extensions and cases". Master's Thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2000.

[69] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models", Computer Vision – ECCV'98. Springer Berlin Heidelberg, vol. 2, pp. 484-498, 1998.

[70] W. Lorensen and H. Cline, "Marching cubes: A high resolution 3D surface construction algorithm", ACM Siggraph Computer Graphics, vol. 21, no. 4, pp. 163-169, 1987.

[71] P. J. Besl and H. D. McKay, "A method for registration of 3-D shapes", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 239-256, 1992.

[72] C. M. Bastuscheck, E. Schonberg, J. T. Schwartz, and M. Sharir, "Object recognition by three-dimensional curve matching", International Journal of Intelligent Systems, vol. 1, no. 2, pp. 105-132, 1986.

[73] G. Subsol, J. P. Thirion, and N. Ayache, "A scheme for automatically building three-dimensional morphometric anatomical atlases: application to a skull atlas", Medical image analysis, vol. 2, no. 1, pp. 37-60, Mar. 1998.

[74] J. Declerck, G. Subsol, J. Thirion, and N. Ayache, "Automatic retrieval of anatomical structures in 3D medical images", in Computer Vision, Virtual Reality and Robotics in Medicine, Springer Berlin/Heidelberg, pp. 151-162, 1995.

[75] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. Hill, M. O. Leach, and D. J. Hawkes, "Nonrigid registration using free-form deformations: application to breast MR images", IEEE transactions on medical imaging, vol. 18, no. 8, pp. 712-21, Aug. 1999.

[76] F. L. Bookstein, "Principal warps: thin-plate splines and the decomposition of deformations", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 6, pp. 567-585, Jun. 1989.

[77] F. L. Bookstein, “Landmark methods for forms without landmarks: morphometrics of group differences in outline shape”, Medical image analysis, vol. 1, no. 3, pp. 225-43, Apr. 1997.

[78] T. Richardson and S. Wang, “Nonrigid shape correspondence using landmark sliding insertion and deletion”, Medical Image Computing and Computer-Assisted Intervention – MICCAI 2005, vol. 3750, pp. 435-442, 2005.

[79] R. H. Davies, C. J. Twining, and C. J. Taylor, “Statistical Models of Shape - Optimisation and Evaluation”. London: Springer-Verlag London, 2008.

[80] T. Heimann and H.-P. Meinzer, “Statistical shape models for 3D medical image segmentation: a review”, Medical image analysis, vol. 13, no. 4, pp. 543-63, Aug. 2009.

[81] S. Belongie, J. Malik, and J. Puzicha, “Shape matching and object recognition using shape contexts”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 4, pp. 509-522, Apr. 2002.

[82] a Kelemen, G. Székely, and G. Gerig, “Elastic model-based segmentation of 3-D neuroradiological data sets”, IEEE transactions on medical imaging, vol. 18, no. 10, pp. 828-39, Oct. 1999.

[83] C. Brechbühler, G. Gerig, and O. Kübler, “Parametrization of closed surfaces for 3-D shape description”, Computer Vision and Image Understanding, vol. 61, no. 2, pp. 154 -170, 1995.

[84] M. Quicken, C. Brechbühler, J. Hug, H. Blattmann, and G. Szekely, “Parameterization of closed surfaces for parametric surface description”, in Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662), vol. 1, pp. 354-360, 2000.

[85] C. Twining, S. Marsland, and C. Taylor, “Measuring Geodesic Distances on the Space of Bounded Diffeomorphisms”, British Machine Vision Conference, vol. 2, pp. 847-856, 2002.

[86] R. Davies, C. Twining, and T. Cootes, “3D statistical shape models using direct optimisation of description length”, Computer Vision – ECCV 2002. Springer Berlin Heidelberg, pp. 3-20, 2002.

[87] S. Marsland, C. J. Twining, and C. J. Taylor, “Groupwise non-rigid registration using polyharmonic clamped-plate splines”, Medical Image Computing and Computer-Assisted Intervention - MICCAI 2003, Springer Berlin Heidelberg, pp. 771-779, 2003.

[88] T. Heimann, I. Wolf, T. Williams, and H.-P. Meinzer, “3D active shape models using gradient descent optimization of description length”, In proceedings of Information processing in medical imaging : , vol. 19, pp. 566-77, Jan. 2005.

[89] R. Davies, C. J. Twining, and C. J. Taylor, "Consistent Spherical Parameterisation for Statistical Shape Modelling", in 3rd IEEE International Symposium on Biomedical Imaging: Macro to Nano, 2006., 2006, no. 1, pp. 1388-1391.

[90] M. Kirschner and S. Wesarg, "Propagation of shape parameterisation for the construction of a statistical shape model of the left ventricle", Vision, Modeling, and Visualization 2009, pp. 313-322, 2009.

[91] M. Kirschner and S. Wesarg, "Area preserving parameterisation of shapes with spherical topology", Informatik 2009: Workshop: Jahrestagung der Gesellschaft Informatik, pp. 1257-1266, 2009.

[92] R. H. Davies, C. J. Twining, T. F. Cootes, J. C. Waterton, and C. J. Taylor, "A minimum description length approach to statistical shape modeling", IEEE transactions on medical imaging, vol. 21, no. 5, pp. 525-37, May 2002.

[93] H. Thodberg, "Minimum Description Length Shape and Appearance Models", Information Processing in Medical Imaging, vol. 2732, pp. 51-62, 2003.

[94] T. Cootes, C. Taylor, and A. Lanitis, "Multi-resolution search with active shape models", in Proceedings of 12th International Conference on Pattern Recognition, 1994, vol. 1, pp. 610-612.

[95] T. F. Cootes, G. Edwards, and C. J. Taylor, "A Comparative Evaluation of Active Appearance Model Algorithms", Procedings of the British Machine Vision Conference 1998, pp. 68.1-68.10, 1998.

[96] I. Matthews and S. Baker, "Active Appearance Models Revisited" International Journal of Computer Vision, vol. 60, no. 2, pp. 135-164, Nov. 2004.

[97] A. Andreopoulos and J. K. Tsotsos, "Efficient and generalizable statistical models of shape and appearance for analysis of cardiac MRI", Medical image analysis, vol. 12, no. 3, pp. 335-57, Jun. 2008.

[98] R. Beichel, H. Bischof, F. Leberl, and M. Sonka, "Robust active appearance models and their application to medical image analysis", IEEE transactions on medical imaging, vol. 24, no. 9, pp. 1151-69, Sep. 2005.

[99] R. Donner, M. Reiter, G. Langs, P. Peloschek, and H. Bischof, "Fast active appearance model search using canonical correlation analysis", IEEE transactions on pattern analysis and machine intelligence, vol. 28, no. 10, pp. 1690-4, Oct. 2006.

[100] P. Tresadern, P. Sauer, and T. Cootes, "Additive Update Predictors in Active Appearance Models", Procedings of the British Machine Vision Conference 2010, pp. 91.1-91.12, 2010.

[101] K. M. Langen, T. R. Willoughby, S. L. Meeks, A. Santhanam, A. Cunningham, L. Levine, and P. a Kupelian, "Observations on real-time prostate gland motion using electromagnetic tracking", International journal of radiation oncology, biology, physics, vol. 71, no. 4, pp. 1084-90, Jul. 2008.

[102] J. Shlens, "A tutorial on principal component analysis: derivation, discussion, and singular value decomposition", Systems Neurobiology Laboratory, University of California at San Diego, 2005.

[103] S. Ghose, A. Oliver, R. Mart, X. Llad, J. Freixenet, J. Mitra, J. C. Vilanova, and F. Meriaudeau, "A hybrid framework of multiple active appearance models and gloal registration for 3D prostate segmentation in MRI", Proc. SPIE 8314, Medical Imaging 2012: Image Processing, vol. 8314, pp. 83140S-83140S-9, Feb. 2012.

[104] S. Ghose, a Oliver, R. Marti, X. Llado, J. Freixenet, J. C. Vilanova, and F. Meriaudeau, "Prostate Segmentation with Texture Enhanced Active Appearance Model", 2010 Sixth International Conference on Signal-Image Technology and Internet Based Systems, pp. 18-22, Dec. 2010.

[105] R. Toth, J. Bulman, A. D. Patel, B. N. Bloch, E. M. Genega, N. M. Rofsky, R. E. Lenkinski, and A. Madabhushi, "Integrating an adaptive region-based appearance model with a landmark-free statistical shape model: application to prostate MRI segmentation", in Proc. SPIE 7962, Medical Imaging 2011: Image Processing, 2011, vol. 7962, pp. 79622V-79622V-12.

[106] S. Martin, J. Troccaz, and V. Daanenc, "Automated segmentation of the prostate in 3D MR images using a probabilistic atlas and a spatially constrained deformable model", Medical physics, vol. 37, no. 4, pp. 1579-90, Apr. 2010.

[107] S. Mcanearney, A. Fedorov, G. R. Joldes, N. Hata, C. Tempany, K. Miller, and A. Wittek, "The Effects of Youngs Modulus on Predicting Prostate Deformation for MRI-Guided Interventions", in in Computational Biomechanics for Medicine, A. Wittek, P. M. F. Nielsen, and K. Miller, Eds. New York, NY: Springer New York, 2011, pp. 39-49.

[108] J. Kang, J. Shin, S. Lee, and J. Paik, "Hierarchical active shape model with motion prediction for real-time tracking of non-rigid objects", IET Computer Vision, vol. 1, no. 1, pp. 17-24, Mar. 2007.

[109] J. G. Bosch, S. C. Mitchell, B. P. F. Lelieveldt, F. Nijland, O. Kamp, M. Sonka, and J. H. C. Reiber, "Automatic segmentation of echocardiographic sequences by active appearance motion models", IEEE transactions on medical imaging, vol. 21, no. 11, pp. 1374-83, Nov. 2002.

[110] T. Cootes, M. Ionita, C. Lindner, and P. Sauer, "Robust and accurate shape model fitting using random forest regression voting", Computer Vision-ECCV 2012, pp. 1-14, 2012.

Appendix A

Phantom Information

A.1 Image Data for Prostate and Pubis Making

Table A.1: Segmented prostate image information

Specimen label	0	1	2	3	4	5	6	7	8	9
Pixel spacing ($\times 0.1mm$)	6	6	6.25	6.25	6.25	6.25	6	6.25	6.25	6.25
Spacing between slices (mm)	4	4	3.6	3.6	3.6	3.6	4	3.6	3.6	3.6
Slice number for segmentation	15	15	20	20	20	20	15	20	24	20
Slice number finally used	14	13	9	13	17	14	14	14	18	16
Volume (cc)	48.9	52.2	16.6	45.5	62.9	47.1	109.7	80.3	73.2	73.6

Table A.2: Abdominal tomography image information

Specimen label	q	x	xx	xxx	y	yy	z	zz	zzz
Pixel spacing ($\times 0.1mm$)	7.23	7.81	7.82	7.422	7.422	7.82	9.10	7.813	7.227
Spacing between slices (mm)	0.8	0.6	0.8	0.7	0.8	0.8	3	0.8	0.7
Slice number for segmentation	82	82	82	82	82	82	20	82	82
Slice number finally used	42	67	42	59	42	42	14	42	59

A.2 Phantom Components Position Statistics

Table A.3: Prostate dimension data statistics*

Dimension	Vol. (cc)	Anterior -Posterior (mm)	Apex -Base (mm)	Left -Right (mm)	Anus -Apex (mm)	Anus -Base (mm)	Table -Posterior (mm)	Rotate Angle ($^\circ$)
Mean	50.8	36.2	52.4	52.8	44.8	94.7	107.7	99.1
Maximum	60	50.8	70.7	66	63.4	118	146.2	130
Minimum	45	25	37.1	41.8	20.7	55.5	89.1	74.1

*These data are provided by UMC St. Radboud .

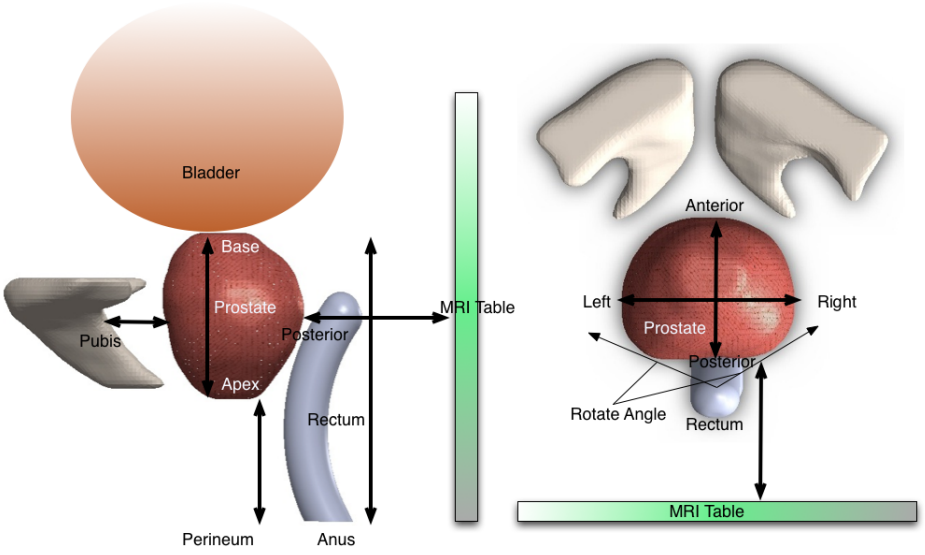


Figure A.1: Annotation of dimensions

A.3 Phantom Dimension Specifications

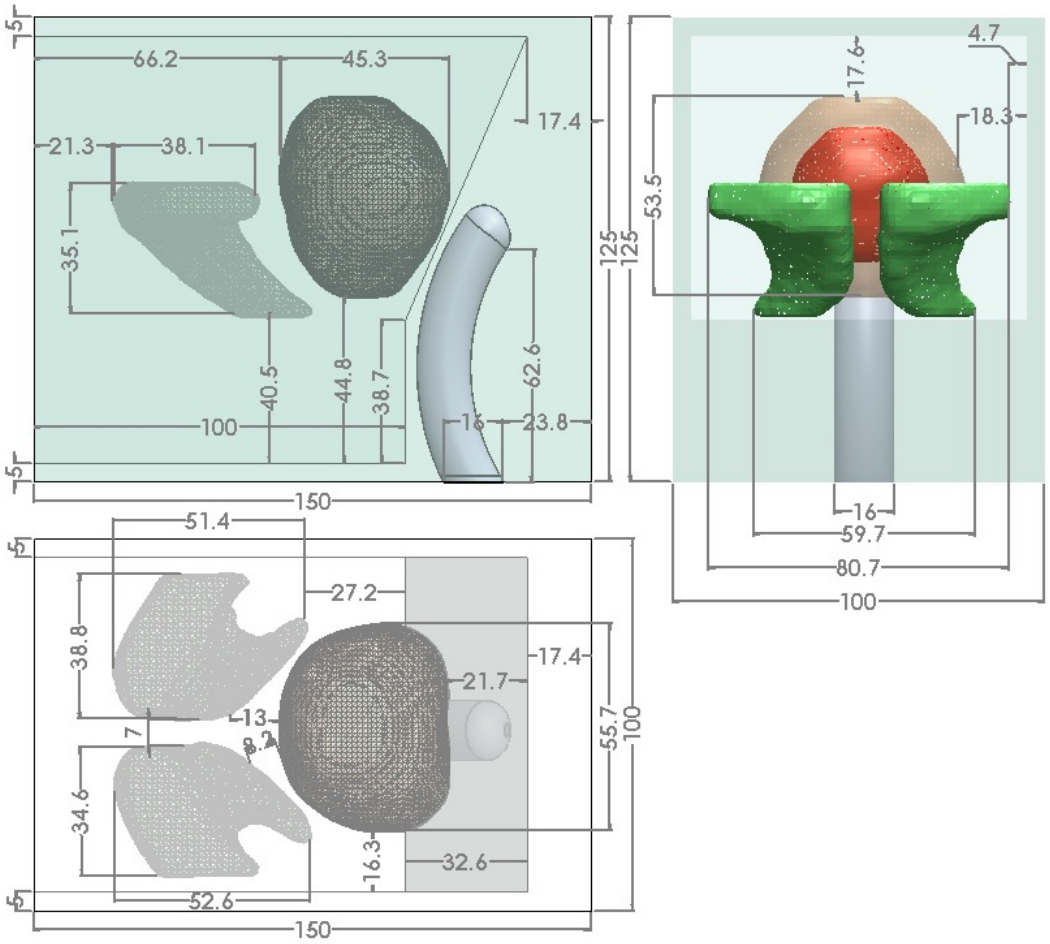


Figure A.2: Phantom dimensions in 3 views (Unit: *mm*)

A.4 Phantom Fabrication Remarks

There are some problems during the fabrication. Firstly, the polymer would turn yellow after a long heating. If the temperature is too high at the beginning, although the polymerization would be faster, it also turn yellow faster. Actually the yellow material does not vary much but it seems to be a little harder than the normal one and the more yellow it is, the harder it would be. As a result, the polymer should be avoid heating for a long time. Consequently, the temperature of heating plate should not be as high as 450 ℃ suggested by Hungr *et al.*. We suggest 250-350 ℃ is appropriate.

Secondly, the material would shrink after cooling down. It happens especially to the frame wall. If it happens, we can fill the space again with same material. Moreover, from the empirical experiments, a cavity would be easily formed in the base of frame because of the inner block which makes the path for air escaping so thin. As a result, it can be fixed after the frame is cooled down by filling the cavity.

Thirdly, air bubbles can be easily attached on the surface of pubis and prostate. It is not wise to fill the surroundings in two steps, where the first one could be to fill the soft material to half of the prostate for fixing it. This is because there would be more bubbles gathering on the surface after the first step during the vacuum, which are more hard to remove. One attempt is to fill the surroundings up once the pubis and prostate placed then vacuum the phantom. But the time for vacuuming should be experimented further and we did not find an optimum one so far.

Appendix B

Orthogonal Procrustes Analysis

B.1 Derivations of Extended Orthogonal Procrustes Analysis

An extended orthogonal Procrustes problem holds the residual matrix

$$\mathbf{L} = s\mathbf{X}\mathbf{R} + \mathbf{j}\mathbf{t}^T - \mathbf{Y} \tag{B.1}$$

where $\mathbf{X}$ and $\mathbf{Y}$ are two $n_p \times 3$ point matrices, R is a 3×3 rotation matrix, $\mathbf{j}^T = [1, 1, \cdots, 1]$ is $1 \times n_p$ unit vector, $\mathbf{t}$ is a 3×1 translation vector and s is a scale factor. Together with the orthogonal rotation matrix $\mathbf{R}^T\mathbf{R} = \mathbf{I}$ constrain, we can write the Lagrangean function

$$\begin{aligned} F &= tr\{\mathbf{L}^T\mathbf{L}\} + tr\{\mathbf{L}(\mathbf{R}^T\mathbf{R} - \mathbf{I})\} && \text{(B.2)} \\ &= tr\{(s\mathbf{X}\mathbf{R} + \mathbf{j}\mathbf{t}^T - \mathbf{Y})^T(s\mathbf{X}\mathbf{R} + \mathbf{j}\mathbf{t}^T - \mathbf{Y})\} + tr\{\mathbf{L}(\mathbf{R}^T\mathbf{R} - \mathbf{I})\} && \text{(B.3)} \\ &= tr\{\mathbf{Y}^T\mathbf{Y}\} + s^2\ tr\{\mathbf{R}^T\mathbf{X}^T\mathbf{X}\mathbf{R}\} + n_p\mathbf{t}^T\mathbf{t} - 2s\ tr\{\mathbf{Y}^T\mathbf{X}\mathbf{R}\} - 2\ tr\{\mathbf{Y}^T\mathbf{j}\mathbf{t}^T\} \\ &\quad + 2s\ tr\{\mathbf{R}^T\mathbf{X}^T\mathbf{j}\mathbf{t}^T\} + tr\{\mathbf{L}(\mathbf{R}^T\mathbf{R} - \mathbf{I})\} && \text{(B.4)} \end{aligned}$$

where $tr\{\cdot\}$ is short for $trace\{\cdot\}$ to calculate the sum of the diagonal elements of the matrix.

In order to obtain a least square estimation, the partial derivatives of the Lagrangean function with respect to $s, \mathbf{R}, \mathbf{t}$ must be zeros, namely,

$$\begin{aligned} \frac{\partial F}{\partial \mathbf{R}} &= 2s^2\mathbf{X}^T\mathbf{X}\mathbf{R} - 2s\mathbf{X}^T\mathbf{Y} + 2s\mathbf{X}^T\mathbf{j}\mathbf{t}^T + \mathbf{R}(\mathbf{L} + \mathbf{L}^T) = 0 && \text{(B.5)} \\ \frac{\partial F}{\partial \mathbf{t}} &= 2n_p\mathbf{t} - 2\mathbf{Y}^T\mathbf{j} + 2s\mathbf{R}^T\mathbf{X}^T\mathbf{j} = 0 && \text{(B.6)} \\ \frac{\partial F}{\partial s} &= 2s\ tr\{\mathbf{R}^T\mathbf{X}^T\mathbf{X}\mathbf{R}\} - 2\ tr\{\mathbf{Y}^T\mathbf{X}\mathbf{R}\} + 2\ tr\{\mathbf{R}^T\mathbf{X}^T\mathbf{j}\mathbf{t}^T\} = 0 && \text{(B.7)} \end{aligned}$$

The translation vector can be obtained directly from equation B.6,

$$\mathbf{t} = (\mathbf{Y} - s\mathbf{X}\mathbf{R})^T\mathbf{j}/n_p \tag{B.8}$$

Then we multiply equation B.5 on the left side by $\mathbf{R}^T$,

$$s^2\mathbf{R}^T\mathbf{X}^T\mathbf{X}\mathbf{R} - s\mathbf{R}^T\mathbf{X}^T\mathbf{Y} + s\mathbf{R}^T\mathbf{X}^T\mathbf{j}\mathbf{t}^T + \frac{\mathbf{L}+\mathbf{L}^T}{2} = 0 \tag{B.9}$$

$$-s^2\mathbf{R}^T\mathbf{X}^T\mathbf{X}\mathbf{R} + s\mathbf{R}^T\mathbf{X}^T\mathbf{Y} - s\mathbf{R}^T\mathbf{X}^T\mathbf{j}\mathbf{t}^T = \frac{\mathbf{L}+\mathbf{L}^T}{2} \tag{B.10}$$

Because $mathbf{R}^T\mathbf{X}^T\mathbf{X}\mathbf{R}$ and $\mathbf{L}+\mathbf{L}^T$ are symmetric matrices, $\mathbf{R}^T\mathbf{X}^T\mathbf{Y}-\mathbf{R}^T\mathbf{X}^T\mathbf{j}\mathbf{t}^T$ should also be symmetric. According to equation B.8, we have

$$\mathbf{R}^T\mathbf{X}^T\mathbf{Y} - \mathbf{R}^T\mathbf{X}^T\frac{\mathbf{j}\mathbf{j}^T}{n_p}(\mathbf{Y} - s\mathbf{X}\mathbf{R}) \tag{B.11}$$

$$=\mathbf{R}^T\mathbf{X}^T\mathbf{Y} - \mathbf{R}^T\mathbf{X}^T\frac{\mathbf{j}\mathbf{j}^T}{n_p}\mathbf{Y} + s\mathbf{R}^T\mathbf{X}^T\frac{\mathbf{j}\mathbf{j}^T}{n_p}\mathbf{X}\mathbf{R} \tag{B.12}$$

is symmetric. It is obvious that $\mathbf{R}^T\mathbf{X}^T\frac{\mathbf{j}\mathbf{j}^T}{n_p}\mathbf{X}\mathbf{R}$ is symmetric, so the rest of the equation must also be symmetric, namely,

$$\mathbf{R}^T\mathbf{X}^T\mathbf{Y} - \mathbf{R}^T\mathbf{X}^T\frac{\mathbf{j}\mathbf{j}^T}{n_p}\mathbf{Y} \tag{B.13}$$

$$=\mathbf{R}^T\mathbf{X}^T\left[\mathbf{Y} - \frac{\mathbf{j}\mathbf{j}^T}{n_p}\mathbf{Y}\right] \tag{B.14}$$

$$=\mathbf{R}^T\mathbf{X}^T\left(\mathbf{I} - \frac{\mathbf{j}\mathbf{j}^T}{n_p}\right)\mathbf{Y} \tag{B.15}$$

is symmetric. We define

$$\mathbf{S} = \mathbf{X}^T\left(\mathbf{I} - \frac{\mathbf{j}\mathbf{j}^T}{n_p}\right)\mathbf{Y} \tag{B.16}$$

If equation B.15 is satisfied, then we have

$$\mathbf{R}^T\mathbf{S} = \mathbf{S}^T\mathbf{R} \Rightarrow \mathbf{S} = \mathbf{R}\mathbf{S}^T\mathbf{R} \tag{B.17}$$

If we use SVD for $\mathbf{S}$, then $\mathbf{S} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices. So we can get

$$\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{R}\mathbf{V}\mathbf{\Sigma}\mathbf{U}^T\mathbf{R} \tag{B.18}$$

Then $\mathbf{R}$ can be

$$\mathbf{R} = \mathbf{U}\mathbf{V}^T \tag{B.19}$$

Then by synthesizing these results, we can also get the scale s as

$$s = \frac{trace\left\{\mathbf{R}^T\mathbf{X}^T\left(\mathbf{I} - \frac{\mathbf{j}\mathbf{j}^T}{n_p}\right)\mathbf{Y}\right\}}{trace\left\{\mathbf{X}^T\left(\mathbf{I} - \frac{\mathbf{j}\mathbf{j}^T}{n_p}\right)\mathbf{X}\right\}} \tag{B.20}$$

B.2 General Algorithms for Calculation

We usually use the following steps to calculate the transformation parameters $s, \mathbf{R}, \mathbf{t}$.

1 Center the points $\mathbf{X}$ and $\mathbf{Y}$ to the coordinate origin by subtracting the centroids $\mathbf{x}_c$ and $\mathbf{y}_c$. The centered points are denoted as $\mathbf{X}_o$ and $\mathbf{Y}_o$.

2 Normalize $\mathbf{X}_o$ and $\mathbf{Y}_o$ using Frobenius norms.

3 Calculate the SVD of normalized $\mathbf{X}_o^T\mathbf{Y}_o$, and obtain the rotation matrix $\mathbf{R}$.

4 Calculate the optimum scaling s and translation $\mathbf{t}$.

B.3 Weighted Extended Orthogonal Procrustes Analysis

We think of adding a weight matrix to minimize the function

$$F = tr\{\left(s\mathbf{X}\mathbf{R} + \mathbf{j}\mathbf{t}^T - \mathbf{Y}\right)^T\mathbf{W}\left(s\mathbf{X}\mathbf{R} + \mathbf{j}\mathbf{t}^T - \mathbf{Y}\right)\} \tag{B.21}$$

where $\mathbf{W}$ is a $n_p \times n_p$ weight matrix. It can be decomposed into a lower and an upper triangle matrix by Cholesky decomposition $\mathbf{W} = \mathbf{Q}^T\mathbf{Q}$. So the equation can be written as

$$F = tr\{\left(s\mathbf{X}\mathbf{R} + \mathbf{j}\mathbf{t}^T - \mathbf{Y}\right)^T\mathbf{Q}^T\mathbf{Q}\left(s\mathbf{X}\mathbf{R} + \mathbf{j}\mathbf{t}^T - \mathbf{Y}\right)\} \tag{B.22}$$

$$= tr\{\left(s\mathbf{Q}\mathbf{X}\mathbf{R} + \mathbf{Q}\mathbf{j}\mathbf{t}^T - \mathbf{Q}\mathbf{Y}\right)^T\left(s\mathbf{Q}\mathbf{X}\mathbf{R} + \mathbf{Q}\mathbf{j}\mathbf{t}^T - \mathbf{Q}\mathbf{Y}\right)\} \tag{B.23}$$

We substitute $\mathbf{X}_w = \mathbf{Q}\mathbf{X}, \quad \mathbf{Y}_w = \mathbf{Q}\mathbf{Y}, \mathbf{j}_w = \mathbf{Q}\mathbf{j}$, then we get

$$F = tr\{\left(s\mathbf{X}_w\mathbf{R} + \mathbf{j}_w\mathbf{t}^T - \mathbf{Y}_w\right)^T\left(s\mathbf{X}_w\mathbf{R} + \mathbf{j}_w\mathbf{t}^T - \mathbf{Y}_w\right)\} \tag{B.24}$$

Therefore it turns into a normal extended orthogonal Procrustes problem. We construct the matrix $\mathbf{S}_w$ which equals to

$$\mathbf{S} = \mathbf{X}_w^T\left(\mathbf{I} - \frac{\mathbf{j}_w\mathbf{j}_w^T}{\mathbf{j}_w^T\mathbf{j}_w}\right)\mathbf{Y}_w \tag{B.25}$$

and we get the two orthogonal metrics $\mathbf{U}$ and $\mathbf{V}$ using its SVD, then the transformation parameters are

$$\mathbf{R} = \mathbf{U}\mathbf{V}^T \tag{B.26}$$

$$s = \frac{trace\left\{\mathbf{R}^T\mathbf{X}_w^T\left(\mathbf{I} - \frac{\mathbf{j}_w\mathbf{j}_w^T}{\mathbf{j}_w^T\mathbf{j}_w}\right)\mathbf{Y}_w\right\}}{trace\left\{\mathbf{X}_w^T\left(\mathbf{I} - \frac{\mathbf{j}_w\mathbf{j}_w^T}{\mathbf{j}_w^T\mathbf{j}_w}\right)\mathbf{X}_w\right\}} \tag{B.27}$$

$$\mathbf{t} = (\mathbf{Y}_w - s\mathbf{X}_w\mathbf{R})^T\frac{\mathbf{j}_w}{\mathbf{j}_w^T\mathbf{j}_w} \tag{B.28}$$