TUM Data Innovation Lab

Munich Data Science Institute

Technical University of Munich

&

TUM Chair for Computer Aided Medical Procedures & Augmented Reality (CAMP)

Final report of project:

Partial 3D Thyroid Registration

Authors	Alexander Baumann, Vincent Bürgin, Emily Hoppe, Ha Young Kim
Mentor(s)	M.Sc. Lennart Bastian and M.Sc. Mahdi Saleh of the TUM Chair for Computer Aided Medical Procedures & Augmented Reality (CAMP)
Project Lead	Dr. Ricardo Acevedo Cabra
Supervisor	Prof. Dr. Massimo Fornasier

Aug 2022

Abstract

Statistical shape models and 2D/3D registration are both important topics for many tasks in medical imaging. In this report, we discuss these two topics: the construction of a statistical shape model and registration of thyroid 2D US scans to a 3D thyroid model. To create a statistical shape model, one needs to find correspondences between different shapes. For this, we introduce a learning-based approach that uses functional maps. The accuracy of the 3D statistical shape model is analyzed and its variation is examined. For 2D/3D registration, we introduce an approach that uses a U-Net-based encoder and Procrustes alignment. To combine the results of the statistical shape model and the 2D/3D registration, we experiment with registering 2D thyroid scans to a 3D statistical shape model of the thyroid. In particular, we register slices from one thyroid to a mesh of another thyroid.

Contents

1 Introduction

1.1 Registration of Medical Images

Many different medical imaging devices have been developed over the years to diagnose and treat patients. Ranging from ultrasound (US) over computed tomography (CT) to magnetic resonance imaging (MRI) and much more, each device has its unique characteristics of use. For example, CT provides fast scanning of skeletal structure and organs, but the patient gets exposed to radiation. MRI, on the other hand, does not have radiation risk and provides high-resolution soft tissue images despite having higher costs and a long scanning time. Compared to CT and MRI, Ultrasound allows real-time scanning at lower costs and is easy to access. However, it is highly dependent on the operator and gives lower resolution images. [1]

To combine the advantages of different medical imaging modalities, registration has been a popular research topic in the field. By registering images taken with different settings of an organ, one can establish the correspondence between the images. This allows the physicians to understand the patient data better and combine the information. Registration has improved the clinical field by allowing interventional procedures such as navigating through a needle biopsy or even developing devices with combined modalities such as MRT, US/CT, and so on.

Registration can also be used with combining data in different dimensions. Several works have been done in the field of computer vision to register 3D views with 2D images [2] [3] [4]. We further extend this to the medical domain by registering 2D scans to a 3D atlas. Throughout the project, the goal is to locate 2D US slice images on a 3D atlas of a thyroid.

This research will have the potential to assist physicians to locate and orient better in which angle and area of the organ they are scanning. It will ease the scanning, and help figure out if there are any missing details of the scan. Moreover, it could be used for the education and training of potential physicians using US. Furthermore, it could help improve robot-guided US technology to capture the entire organ of interest without missing some parts.

1.2 Understanding Mean Shapes and Variation in Anatomy

What does it imply for clinicians to declare that a structure, such as an organ, is “normal”? Does the word “normal” signify it is “common” or “average”? Furthermore the word “abnormal” denotes infrequently observed characteristics. This terminology, which heavily relays on the observations, indicates a statistical background of the term “normality”. Identifying patterns in size, form, and relative position is crucial to recognizing anatomical variances. Such patterns can fluctuate in a range that is seen as normal variation. [5]

A statistical shape model is a geometric model which describes a group of semantically similar objects. It represents an average shape derived from a cohort as well as the variation in shape [6]. Moreover, it is important to note that a shape is defined as a property that does not change under similarity transformations. This means it is invariant to translation, rotation, and scaling [7].

The statistical shape model can be applied in different methods. For example, gained knowledge of the shape can be used to segment images. Additionally, variations of the

shape can be differentiated and used for clustering and classification. The variations can also be adjusted to create realistic shapes for phantom generation.

1.3 Project Goals

The project consists of two main goals. The first goal is the creation of a 3D statistical shape model (SSM) of the thyroid. At first the correspondences between several thyroids are found. From these correspondences, the mean shape and eigenshapes are calculated to create the SSM of the thyroid.
Second, 2D US scans are registered to a 3D thyroid model. With the 3D thyroid shape obtained from 3D US scan data, 2D US scans are localized within the 3D shape.
Furthermore, these two main tasks are combined by taking first steps towards registering 2D scans to a 3D SSM of the thyroid: namely, by registering 2D scans of one thyroid to a 3D representation of a different thyroid.

1.4 Thyroid Anatomy, Physiology and Pathology

The thyroid gland is an organ located in the neck that has the shape of a butterfly. It consists of left and right lobes that are connected in the middle by a narrow structure known as the isthmus. The anatomy of the thyroid can be seen in figure 1. The thyroid works as an important regulatory organ that controls important body functions, such as metabolism or growth, by producing hormones. [8]
One of the common disorders of the thyroid is hyper- and hypothyroidism. Around 4-5% of the population of the United States is affected. Hypothyroidism is when a thyroid does not generate and release enough thyroid hormones leading to a slow metabolism. One cause of hypothyroidism is thyroiditis, which is the inflammation of the thyroid. On the other hand, hyperthyroidism is when the thyroid produces more hormones than needed and the metabolism is sped up. The autoimmune Graves disease is the most common cause of this [8]. Another disease of the thyroid is Golter, which occurs when the thyroid enlarges. Moreover, thyroid cancer is the 20th most common cancer in the UK, and the 5-year survival rate is 87% [9].

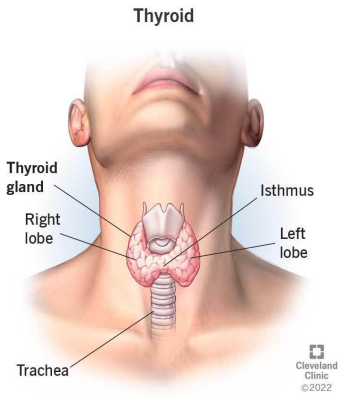


Figure 1: Anatomy of the thyroid gland [8]

2 Data Acquisition

Ultrasound imaging techniques have been developed over the years allowing us to acquire not only 2D but also 3D US images. Compared to 2D ultrasound, 3D ultrasound en-

hance the understanding of physicians of the scanned volume region of interest. It gives more accurate and less operator-dependent results. Due to its advantages, 3D US has been frequently used in clinics for diagnostics and image guidance. There are two main approaches to 3D US scanning: Fixed probe and freehand 3D US. Fixed probe 3D US acquires 3D volume images while the probe is held still giving high-quality images with a fast reconstruction time. On the other hand, freehand 3D US scans the 3D volume by moving a conventional 2D US probe over a volume of interest. It records a sequence of US scans and probe positions to reconstruct a 3D volume. The freehand US has the advantage that it can scan an arbitrary large volume, but it requires a longer scanning time.
The dataset "SegThy" used in this project was acquired using freehand 3D US obtained from the Department of Nuclear Medicine at Klinikum Rechts der Isar, Munich. Thyroid US scans of 16 healthy volunteers between the age of 24 - 39 years were scanned. Since the thyroid consists of two lobes, a total of 32 thyroid US scans were acquired. All volunteers were informed about the scanning procedures, data storage and management, any contraindication for participating in the study, and implications in case of clinical findings. [10] [11]
For each freehand 3D US scan, the 3D US volume was generated by compounding each US sweeps to volumes with a spacing of 0.12 x 0.12 x 0.12. Additionally, the thyroid was labeled manually by experts using ImFusion Labels for Academia program (version 2.25.1) (ImFusionGmbH, Munich, Germany). The processed 3D US data were provided in NIFTI file format. Units of length in this report refer to voxel units with respect to this grid.
Furthermore, the current project works on mesh data of the thyroid. Therefore the labeled US scans are converted to mesh format (.ply). However, the meshes had more than 200,000 vertices and 500,000 faces which are too large to train models. Therefore, the meshes were downsampled to 10,000 faces using the method from [12].

3 Related Work

3.1 Statistical Shape Model

Statistical shape models have been studied for a long time, as they are a promising approach for many tasks in medical imaging. The two main steps are finding correspondences between two 3D shapes and estimating the variation. In the following section, different approaches for these challenges are discussed.

A Statistical Shape Model for the Liver The 2002 paper [13] proposes a geometric approach to build a SSM for the liver. The data consisted of 20 CT scans. At first, a user has to manually decompose the surface into patches, which are topologically equivalent to disks. The goal is to map a patch of one liver onto a corresponding patch on another surface while minimizing distortion. Using the computed correspondences, the SSM is built. Before computing the mean, the two livers are aligned by the "center of gravity" method and a mean least squares fit of the displacements. To obtain the variability, principal component analysis is applied. The compactness of the model is measured and does not yet reach convergence. This indicates a larger needed training set or improved correspondences.

Gaussian Process Morphable Models This paper [14] uses a class of shape models, the point distribution models (PDMs). The shapes are represented as a normal distribution of point variations, the parameters are estimated from example shapes. To get a low-dimensional representation of the shape variation in terms of the leading principal components, PCA is applied. The required input are points which are in correspondence and all components have to have the same number of points. The x, y, z components of each point are stacked into vector s. It is assumed that the shape variations can be modeled using a normal distribution $S \sim \mathcal{N}(\mu, \Sigma)$, with mean $\mu = \frac{1}{n}\sum_{i=1}^{n} s_i$ and covariance matrix $\Sigma = \frac{1}{N-1}\sum_{i=1}^{n}(s_i - \mu)(s_i - \mu)^T$. As the total number of points is usually a large number, the covariance matrix Σ cannot be explicitly represented. However, as is it determined by the n example data sets, the rank is at most n and can be represented by using n basis vectors. This is done by performing PCA with the model $s = \mu + \sum_{i=1}^{n} \alpha_i \sqrt{d_i} u_i$, where u_i are the eigenvectors, d_i the eigenvalues and $\alpha_i \sim \mathcal{N}(0, 1)$. These eigenvectors represent variations of the statistical shape model.
However, this paper does not elaborate on point-to-point correspondence which is a priori a very strong assumption. Furthermore, it is not explained how to convert the PDM to a mesh surface.

Geodesic distances to landmarks for dense correspondence on ensembles of complex shapes This article [15] tackles the correspondence problem on biomedical shapes. After manually selecting some landmarks on the shapes, the geodesic distances between a landmark and any other vertex on the mesh can be efficiently calculated using the fast iterative method (FIM) method [16]. Afterwards, one can continuously interpolate the geodesic distances onto the faces of the mesh. With these values one aims to optimize the entropy of the particle distribution of the ensemble of shapes. This entropy is approximated by the covariance of the geodesic distance features, whose gradient can be calculated. In this way, the position of the particles are optimized. A downside of this method is that one requires an expertise to annotate some landmarks. In this work, 6 landmarks on MRI scans of brains are marked, which is only feasible for experts and time-consuming.

Functional Maps: A Flexible Representation of Maps Between Shapes For rigid shapes, the deformation of two shapes can be represented as a rotation and translation. The correspondence task becomes difficult when trying to match non-rigid shapes. Here the matchings are commonly represented as pairings of points on the two shapes. These pairings are also called correspondences. This paper [17] approaches the problem not by observing correspondence points on the shapes, but by looking at mappings between functions defined on the shapes. The functional representation has the important property that many natural constraints on a map become linear. Using the Laplace-Beltrami eigenfuctions as the basis for the functional representations lets the map be compact. This means that the functions are all approximated by using a small number of basis elements. An algorithm to match the shapes is presented. Given two meshes and the Laplace-Beltrami eigen-decomposition, the functional constraints that correspond to descriptor and segment preservation constraints are computed. Together with the operator commutativity, a linear system of equations is formed and solved by least squares.

Deep Functional Maps: Structured Prediction for Dense Shape Correspondence This approach [18] is learning-based and proposes a structured prediction model

in the space of functional maps and linear operators. The learning process is modelled through a deep residual network. The input is a dense descriptor fields, defined on two shapes, the output is a soft map between the two given objects. The computation of the correspondence is part of the learning procedure.
For the input feature the fast and robust SHOT descriptor is used, which is defined in paper [19]. Non-parametric layers are added to the network to implement the least-squares solve.The computation of the soft correspondences follows. The loss can be interpreted as a soft error, which weights the probabilities of the geodesic distance from the ground truth.

3.2 Partial Registration

For the partial registration, we are interested in approaches that can correspond 2D and 3D data in order to find an alignment. Matching between 2D and 3D modalities has been predominantly used for SLAM (simultaneous localization and mapping) applications in the area of autonomous driving. Our main inspiration is 2D-3D MatchNet [3] which jointly trains 2D and 3D embeddings and uses them to localize camera images in a 3D scene represented by a point cloud. A similar approach is taken by [4] which computes joint 2D/3D features and obtains similarities via a metric network. In 2D3D-GAN-Net [20], a GAN architecture is used to learn similar embeddings for 2D/3D modalities. Earlier work is [2] which uses SIFT features and a trained "visual vocabulary" to align images to point clouds. Another approach that computes descriptors to match between 2D and 3D modalities is LCD [21], which computes joint 2D/3D features using auto-encoders.
Very recent work from the medical domain is [22], where US images are registered to MRI scans of the abdominal area. The method uses U-Net-based dense feature extractors, matches the jointly trained 2D/3D features, and computes an alignment.

2D3D - MatchNet : Learning to Match Keypoints Across 2D Image and 3D Point Cloud The 2D-3D MatchNet [3] is a deep network model that registers a 2D camera image with a 3D point cloud by jointly learning the similarity of an image patches and point cloud patches. They extract keypoints from both modalities and compute embedding vectors. Then they identify matching pairs of patches based on a distance threshold in the embedding space and run a perspective projection algorithm to estimate the camera position. The encoder uses a VGG and a PointNet network and is trained via triplet learning

Global Multi-modal 2D/3D Registration via Local Descriptors Learning The objective of [22] is closely related to our goal: registration of ultrasound slices to MRI images. It uses 2D and 3D U-Net architectures to compute dense feature maps. Similarities between feature vectors are computed using a dot-product similarity, which gives a soft assignment matrix. Similar encodings are detected as matches, based on which the ultrasound slice is located relative to the MRI image. A translation and rotation are computed using a RANSAC-based algorithm.

4 Statistical Shape Model

The task of creating a SSM is divided into two sub-tasks. First, one needs to establish correspondences between the vertices of the thyroid meshes. An approach to solve the correspondence problem is discussed in section 4.1. Then in section 4.2, we compute the mean and the variation of the corresponded points to obtain a model for the shapes.

4.1 Correspondence Problem

Since we do not have ground-truth labels for the corresponding points, an unsupervised or weakly-supervised method is needed to tackle this problem. For a (weakly-) supervised approach, one can find landmark points, as in paper [15]. Especially when there are outliers in the data set, it can be challenging to determine these landmarks without strong medical expertise. Using an unsupervised approach makes our model fully automatized, scalable, and applicable to everyone. At future time points when more data has been acquired, the model can easily be extended.
We chose the unsupervised model SURFMNet [23] to learn correspondences between the thyroids. This method is based on functional maps [17].

4.1.1 Background

Here, a mesh is modelled as a two-dimensional Riemannian manifold $\mathcal{X}$ with the standard measure $d\mu$ induced by the volume form. For such a manifold $\mathcal{X}$, one defines the function space $\mathcal{L}^2(\mathcal{X}) = \{f : \mathcal{X} \longrightarrow \mathbb{R} \quad | \quad \langle f, f\rangle_{\mathcal{X}} < \infty\}$, where $\langle f, g\rangle_{\mathcal{X}} = \int_{\mathcal{X}} f \cdot g \, d\mu$. A bijective map $T : \mathcal{X} \longrightarrow \mathcal{Y}$ between two shapes $\mathcal{X}$ and $\mathcal{Y}$ induces a map $T_F : \mathcal{L}^2(\mathcal{X}) \longrightarrow \mathcal{L}^2(\mathcal{X})$ by mapping $f \in \mathcal{L}^2(\mathcal{X})$ to $g = f \circ T^{-1} \in \mathcal{L}^2(\mathcal{Y})$. In fact, there is an one-to-one correspondence:

$$\{T : \mathcal{X} \longrightarrow \mathcal{Y} \text{ bijective}\} \longleftrightarrow \left\{T_F : \mathcal{L}^2(\mathcal{X}) \longrightarrow \mathcal{L}^{\mathcal{Y}})\right\} \tag{1}$$

For the other direction, one can retrieve $T(x)$ with $x \in \mathcal{X}$ by considering the indicator function $\delta_x \in \mathcal{L}^2(\mathcal{X})$. Hence, such maps T_F are a generalization of point-to-point correspondences. These functional maps have some useful properties, which is the reason to consider them. For example, they are linear and can be represented as a (possibly infinite) matrix with respect to bases on $\mathcal{L}^2(\mathcal{X})$ and $\mathcal{L}^2(\mathcal{Y})$ [17]. As a basis, one uses the eigenfunctions of the Laplace-Beltrami operator on the given shape. This choice is proven to be optimal [24]. We will compute a certain number of eigenfunctions of the discrete Laplace-Beltrami operator for each shape in the preprocessing step. We reference [25] as good introduction to this topic.

4.1.2 Preprocessing

As a preprocessing method, mesh augmentation is done by deforming the thyroid mesh. This gives variations of thyroid structure and improves the generalization during training. The thyroid is deformed locally, deforming only a part of the thyroid mesh. First, a vertex is selected to define the center of augmentation. Then the surrounding vertices are obtained within a given radius of augmentation. These vertices are multiplied with a 3D Gaussian density centered at the center vertex in the direction of the norm of each vertex. Moreover, the mesh can be augmented in several parts as shown in the right image of figure 14 in the appendix.

4.1.3 Model

As we do not have ground-truth labels, we cannot use the classical supervised deep functional map approach [18]. Therefore, we use an unsupervised version called Spectral Unsupervised Functional Map Network (SURFMNet) [23]. This model has the same architecture as [18] (see figure 2). However, it uses an unsupervised loss, which forces the resulting functional map to have some good properties such as :

- **Bijectivity** of the functional map

- **Orthogonality** of the functional map
- **Commutativity** of the functional map with the diagonal matrix containing the eigenvalues of the Laplace-Beltrami operator
- **Preservation of shape descriptors** via commutativity with the functional map

For this, one computes the functional map from shape X to shape Y and from shape Y to X. Of course, one expects that these maps are inverse, as stated before. Furthermore, if the point-to-point correspondences are volume-preserving, then the associated functional map must be orthonormal [17], which explains the second part of the loss. The same article shows that if the point map is an isometry, then the corresponding functional map commutes with the Laplace-Beltrami operator. This is motivated by the fact that a non-trivial map $\mathcal{T}$ on function spaces corresponds to a point-to-point map if and only if $\mathcal{T}$ preserves the point-wise product between functions [26]. Using this fact and taking the fine-tuned descriptors as functions, [27] deduces the fourth part of the loss. These parts are summed up with associating weights, which are taken from the original article [23]. In this way, the fourth summand gets the largest weight and the third one the smallest.

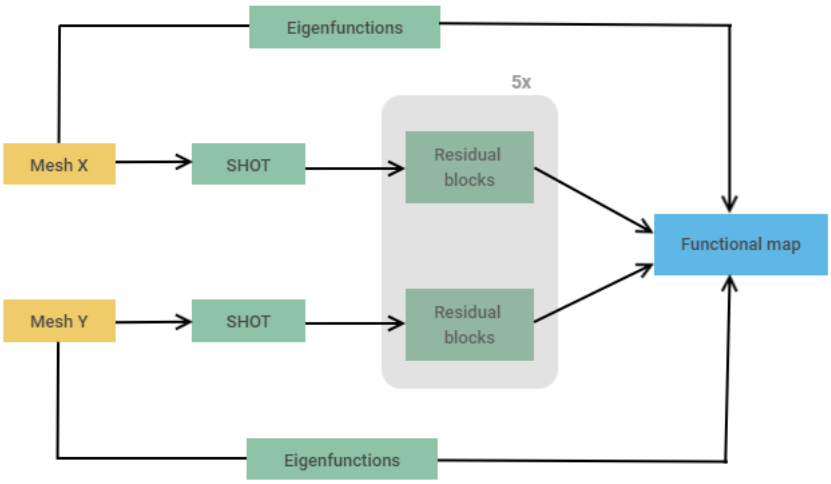


Figure 2: Model architecture of SURFMNet [23]. From a pair of shapes X and Y, one extracts the hand-crafted SHOT descriptors [19] and refines them in five residual blocks. The refined features and the Laplace eigenfunctions of both shapes are then used to calculate the functional map. Note that the only learnable parameters are in the residual blocks.

4.1.4 Implementation Details

We used an existing implementation of the SHOT descriptor, which gives a slightly modified feature dimension than in the original paper [19], namely 336 instead of 352. Having these features, we refined them in five residual blocks of hidden and output dimension 336. Overall, we trained two models of SURFMNet, one for the left lobes and one for the right lobes. In order to increase our dataset and add some more variation, we applied the augmentation technique from section 4.1.2 and random rotations. All thyroids are taken into the training set since we only have a small number of thyroids available and our SSM should include all thyroids in the end. After 50 epochs of training with a learning rate of 10^{-3}, we obtained our results which are discussed in section 6.1.

4.2 Construction of Statistical Shape Model

To generate the model from the corresponded thyroid pairs, a reference thyroid X has to be chosen. In section 6.1.3 the 16 different options for the left and right side are evaluated. After choosing a reference thyroid X, the correspondences of every thyroid Y to the reference thyroid are calculated with the method from section 4.1.3.
Now, it is possible to compute the mean shape. For each vertex i of the reference thyroid X, we form the set of vertices from all other thyroids which correspond to this vertex i. Note that these correspondence sets can have a large variance in the number of elements. After calculating the mean and the standard deviation on this set, we apply an outlier detection to exclude all points from this set that are more than 2 standard deviations away from the mean. Finally, we recompute the mean and standard deviation on this filtered set.
The calculation of the variance on a per-point basis differs from other SSM approaches. Other works like [13, 14, 7] represent a variation of the SSM with eigenshapes. For this, one needs an equal-sized set of points for each shape whose points are in 1:1 correspondence to each other. Otherwise, one cannot compute the covariance matrix of the point coordinates that leads eventually to the eigenshapes. In our approach, this matrix can not be derived. While the model learns to minimize the bijectivity loss, not all points have 1:1 correspondence (see figure 15). So, every point on the reference thyroid can have a different number of corresponding points (or there is even no corresponding point). Hence, it is not possible to form the covariance matrix.
In contrast to our correspondence approach, the authors from [13, 14] use landmarks. This makes it possible to create eigenshapes. In future work, we will tackle this problem to encapsulate higher dimensional variance.
Now, the SSM built from these mean points and standard deviations is the form of a point cloud. Since we want to end up with a mesh it is useful to apply another outlier detection on the point cloud before generating the mesh. This is a radius outlier removal, where points that only have a few neighbors in a given sphere around them. We set the number of points to be 20 and the radius to be 0.05. The mesh is now constructed from the point cloud by applying a Poisson reconstruction [28]. After having a mesh, we apply smoothing methods from [29, 30] to improve the quality of the meshes.
The full pipeline can be seen in 3.

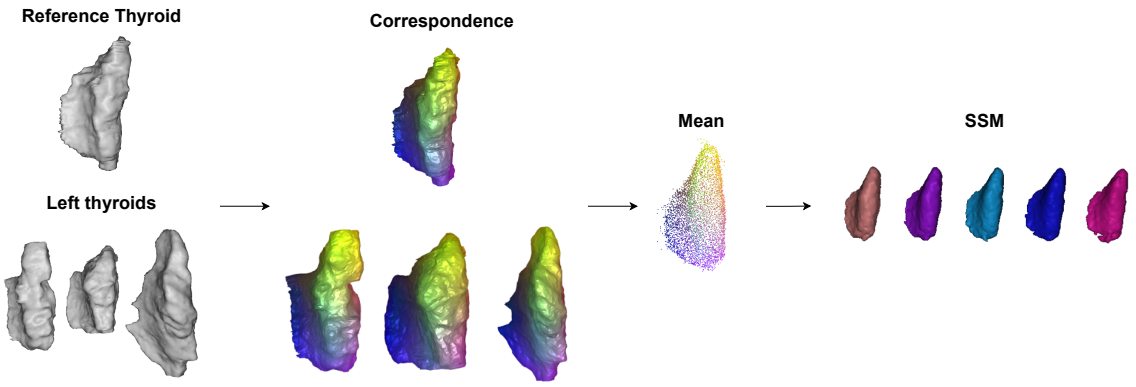


Figure 3: Statistical shape model pipeline

5 Partial Registration

5.1 Approach

2D US scans give a scan of a thin plane at some arbitrary angle in the body. However, it is challenging to localize the plane angle and direction according to the organ. Therefore, in this partial registration section, we aim to localize the 2D US slice in a 3D model of the thyroid by 2D–3D registration of the US scan.

Using the 3D US scan data, 2D images of US intensity are extracted as a slice. To locate the 2D slice in a mesh representation of the thyroid, we employ a two-stage approach which is inspired by 2D-3D MatchNet [3].

In the first stage, a large amount of small patches is extracted from both the 2D slice data and the 3D thyroid representation. The patches are fed through a neural network encoder which produces high-dimensional embeddings of the data. The network is trained to jointly embed patches from the two data modalities such that patches that come from the same region are mapped to nearby embeddings. Among the given 32 scans of thyroid lobes, 24 are used for training, 4 for validation, and 4 for the test.

In the second stage, matching pairs between the 2D slice patch embeddings and the 3D model patch embeddings are determined. Our algorithm assumes that if such a pair of embeddings is close in the embeddings space, the corresponding patches are close in the input data. Using a number of such matches, a classical algorithm is run to register the slice to the 3D model. We use a Procrustes alignment in the simplest case, and introduce several tweaks to deal with false-positive matches and increase the performance.

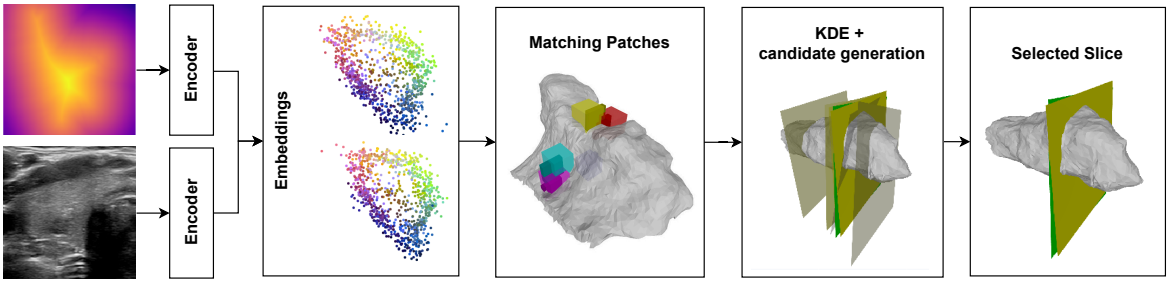


Figure 4: 2D/3D Registration pipeline

5.1.1 Patches from Ultrasound and Mesh Representation

Two types of patches are extracted from the US scans: SDF patches and US slice patches. To represent the 3D mesh generated from the segmentation labels as a voxel grid, we use a discretized signed distance field (SDF) representation. We sample box-shaped patches of size $(N \times N \times N)$ from the SDF grid and patches of size $(M \times M \times D)$ from the US grid (see figure 19b in the appendix). In particular, we use ultrasound patches of a certain depth D instead of a flat 2D image: We conjecture that thicker patches are easier to be matched to a mesh because they intersect it not only at a line, but at a surface. One goal is to experiment with relaxing this simplification.

5.2 Network

5.2.1 Encoder Structure

Our network uses two encoder networks with identical architecture (but different weights): one for US patches, and one for SDF patches. For the encoders, the 3D U-Net encoder is used [31] followed by two fully-connected linear layers that map the features to the embedding space of dimension $C_{\text{embedding}}$ (128 in our experiments).
The 3D U-Net encoder is a convolutional architecture that consists of several layers of 3D convolutions with batch normalization and ReLU, interleaved with spatial max pool operations. Originally, in the 3D U-Net, this architecture was used together with a mirrored decoder architecture and skip connections to segment medical images. Here, the 3D U-Net is selected because it is a standard architecture known to work well with medical data. Additionally, it can segment the thyroid data in our dataset [11] and has the potential to be pre-trained on a segmentation task in future extensions to this project. [31]

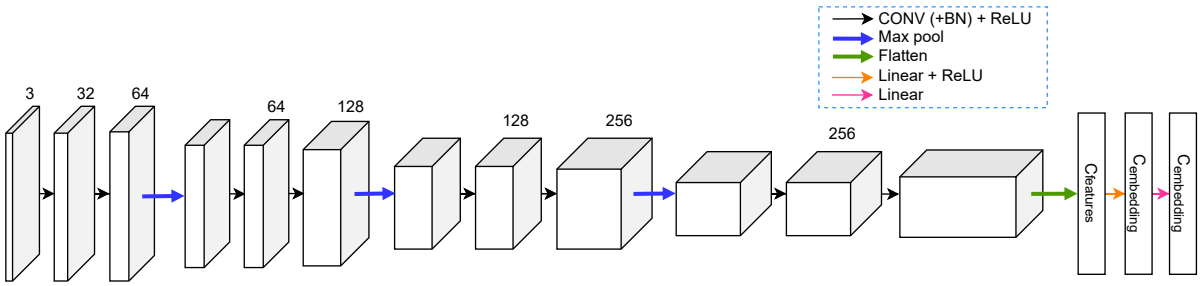


Figure 5: Encoder architecture (up to the green arrow, equal to the 3D U-Net encoder)

5.2.2 Training and Loss Functions

To train our network, we use a triplet learning procedure (see e.g. [32], [33]). The network gets a triplet as input: an ultrasound anchor patch p_0, a positive SDF patch p_+, and a negative SDF patch p_-. These three patches are then passed through the respective encoder networks to get the embeddings. Then the loss is computed and the weights of the network are updated using the Adam [34] algorithm.
Two different loss functions are considered for training the network.
The first loss function is the *weighted soft-margin loss* which was proposed by [35]. The motivation behind this loss is to perform triplet learning without having to fine-tune the distance parameter in the classical triplet loss [36]. The weighted soft-margin loss is defined by:

$$\mathcal{L}_{\text{wsm}}(e_0, e_+, e_-) = \log(1 + e^{\alpha\, d_{+-}}) \tag{2}$$

with e_0, e_+, e_- as embedding vectors corresponding to p_0, p_+, p_- and $d_{+-} = ||e_0 - e_+||_2 - ||e_0 - e_-||_2$ as the difference between the positive and negative distance. We set $\alpha = 5.0$ based on the 2D-3D MatchNet recommendation [3].
Furthermore, when using L_{wsm}, we normalized the embedding vectors to bring them to the unit hypersphere at the end of the encoder.
The second loss function, which we call the *distance matching loss*, is considered as an alternative loss. Here, we aim to directly match the Euclidean distance in the 3D space our

patches come from in the high-dimensional embedding space. Similar ideas are pursued e.g. in [37]. Given the assumption that the patches p_0, p_+, p_- are centered at locations x_0, x_+, x_-, the distance matching loss is defined as:

$$\mathcal{L}_{\text{dm}}(e_0, e_+, e_-) = \left(||x_0 - x_+|| - ||e_0 - e_+||\right)^2 + \left(||x_0 - x_-|| - ||e_0 - e_-||\right)^2 \tag{3}$$

The training is further tweaked using perturbation techniques that are described in detail in section 5.4.

5.3 Slice Identification

After training our model, the location of the US slice in the thyroid mesh is estimated using the trained model. For this, the idea from 2D-3D MatchNet [3] that registers images taken by a camera to a 3D scene is adopted.
First, matching pairs of patches are identified using our encoder network. Afterwards, using a classical (non-trained) algorithm, slice parameters are computed such that the matching patches on the slice are brought close to their counterparts in 3D space. For this last step, we use a combination of Procrustes alignment and a hypothesis generation step based on kernel density estimation (KDE) which helps with outlier detection.

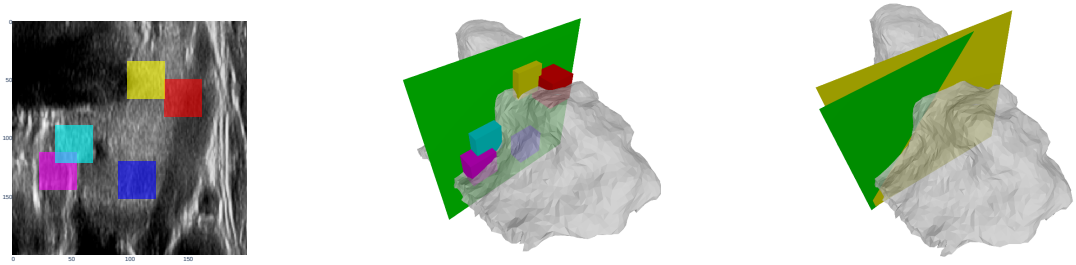
Finding Matching Patches To find pairs of matching patches, a set of k_{slice} 2D patches P_{slice} are sampled from the ultrasound data given as a 2D slice. Also, a set of k_{thyr} patches P_{thyr} from the 3D space are sampled. These 2D and 3D patches are sampled near the surface of the thyroid mesh. The US and SDF patches are encoded into the embedding space and the nearby embeddings are identified. To this end, the closest SDF embedding to each of the US embeddings is found. Of the k_{slice} pairs constructed in this manner, we pick the k_{match} pairs with closest distances.

Procrustes Approach Given the 2D coordinates $C_{\text{slice}} = \{(x_i, y_i)^\top \mid i \in [k]\}$ of the ultrasound matches with respect to the slice (see figure 6a), and the 3D coordinates $C_{\text{thyr}} = \{\tilde{c}_i = (\tilde{x}_i, \tilde{y}_i, \tilde{z}_i)^\top \mid i \in [k]\}$ of the SDF matches (see figure 6b), we map the slice coordinates to 3D space to obtain coordinates $C_{\text{slice-3D}} = \{c_i = (x_i, y_i, 0)^\top \mid i \in [k]\}$ [1]. Then the Procrustes algorithm [38] is used to find an affine transformation $A \in \text{SE}(3)$ such that the *Procrustes loss* $\sum_{i\in[k]} ||Ac_i - \tilde{c}_i||_2^2$ is minimized (where c_i is assumed to be in a homogeneous coordinate representation). In this particular flavor of Procrustes analysis, A may translate and rotate, but not mirror or scale the point clouds (see figure 6c).

Hypothesis Generation with Kernel Density Estimation The Procrustes alignment was observed to work well if the matching patch pairs correspond to the same region. However, it is not robust to noise and can be significantly perturbed in the presence of outliers (i.e. false-positive matches). On the other hand, if the matches lie roughly in the correct region, the Procrustes algorithm also predicts the slice well (see 6c).
For this reason, we adopt an approach that first generates hypotheses about potential slice regions. Then the matching algorithm is run for each region with SDF patches sampled only from the specific region. For this step, the slices are assumed to be roughly axis-aligned to the x-y plane (see 6b). In a clinical setting, one would assume the slices to be rotated by a small angle only, which justifies this simplification,

[1] The location is arbitrary, as the result of Procrustes registration is invariant under translation.

Concretely, we first generate matching patch pairs on the whole thyroid, and project the SDF match coordinates C_{thyr} to their z coordinates $C_{\text{thyr},z} \in \mathbb{R}$. On this set of projections, we run *kernel density estimation* (KDE) [39] that fits a Gaussian mixture density to $C_{\text{thyr},z} \in \mathbb{R}$: The resulting density ϕ is a convex combination of Gaussian densities corresponding to $\mathcal{N}(\tilde{c}_i, \sigma^2)$ for $\tilde{c}_i \in C_{\text{thyr}}$ and a bandwidth parameter σ^2. We then use the m_{KDE} biggest local maxima $\{z_j \mid j \in [m_{\text{KDE}}]\}$ of the KDE density as our hypotheses z coordinates. We go on to run m_{KDE} more iterations of the matching algorithm, and in each iteration j sample SDF patches only within the z coordinate region $[z_j - w_{\text{restr-z}}, z_j + w_{\text{restr-z}}]$, where the z restriction width $w_{\text{restr-z}}$ is a hyperparameter (typically chosen to be 10.0). This generates m_{KDE} candidate slices, of which we pick the one with the lowest Procrustes loss as our prediction.



(a) Ultrasound patches (b) SDF Patches/True slice (c) Procrustes result

Figure 6: Example of approximately matching 2D/3D patches with Procrustes

5.4 Refinements and Implementation

5.5 Patch Generation and Refinements

We generate coordinates for the US anchor patches and corresponding positive SDF patches by sampling patch centers uniformly from the mesh surface which are taken as centers. We assume that patches close to the mesh are more useful for registering to the mesh. Furthermore, we assume that at test time such patches can be generated as there are accurate thyroid segmentation methods available (see e.g. [11]). We generate negative samples by sampling centers uniformly from the mesh surface, and ensuring they are at least a certain threshold t_{neg} (typically 40.0) way from their anchor patch. To prevent overfitting, we reshuffle negative samples after each epoch.
During registration at test time, we cannot assume to have US and SDF patches in exactly the same locations. To allow for patches in slightly different locations to match, we add random perturbations of up to ϵ_{pert} (typically 10.0 or 25.0) voxel units to the positive patch coordinates.

5.6 Cross-Thyroid Registration

Towards combining the partial registration with the SSM, we want to be able to perform *cross-thyroid slice registration*. That is, given an ultrasound slice from one thyroid, we want to be able to determine its position with respect to another thyroid mesh. This can be useful if the full shape of the currently scanned thyroid is unknown, but another reference thyroid shape – for example the mean shape constructed in section 4.2 – is available.

5.6.1 Cross-Thyroid Training

To achieve this, we perform cross-thyroid training using the functional map correspondences from section 4.2. We keep the triplet learning approach, but within a triplet use samples from different thyroids: If the anchor patch comes from thyroid lobe L_i, we transport its positive and negative sample patches to another thyroid lobe L_j with which correspondences exist (i.e. it is also a left or also a right lobe, respectively). We transport the patches by finding the nearest mesh vertex v in L_i, selecting the corresponding vertex $\tilde{v}$ in L_j, and placing the patch at the same offset to $\tilde{v}$ as the original patch was to v. For each anchor patch, the partner lobe that its positive and negative sample patches are sent to is randomly selected in every epoch.

5.6.2 Cross-Thyroid Slice Matching

Using the model trained on cross-thyroid patches, we perform the slice identification algorithm described in section 5.3 in a modified form, where SDF patches are used from the lobe we want to match to, while US patches come from another thyroid.

6 Results

6.1 Statistical Shape Model

6.1.1 Evaluation of Correspondences

It is not obvious how to evaluate the correspondences without having ground-truth values. Besides the visual evaluation (figures 16 a 17), we found out that it is important for the SSM "how bijective" the correspondences are. That is why we computed the bijectivity rate, which is the ratio of points that are mapped to the reference thyroid and via the inverse functional map back to the starting point. This ratio is calculated for one fixed thyroid to all other thyroids and the mean of these rates is taken. In figure 15 in the appendix, we visualized these results for all fixed thyroids. One can see that for both lobes there are some significant outliers like thyroid 16 and 21. In figures 16 and 17, this fact is confirmed visually. There, one sees that these thyroids have an atypical shape and therefore it is more difficult for the model to find the correct correspondences.
Furthermore, the left lobes have slightly better correspondences than the right lobes with respect to this evaluation matrix. Additionally, it also has a lower standard deviation. This is observed multiple times during the development of the SSM (see table 1 and 2).

6.1.2 Evaluation Metric of Statistical Shape Model

We now want to evaluate the performance of the correspondences used for the statistical shape analysis. For this, we use the evaluation metrics defined by Davies [40]. As it is proposed for SSM including eigenshapes, we have to adapt some measures, while preserving the basic characteristics. In the following sections, we will describe how each of these is measured. The results in each of the three metrics are ranked and finally, we take the sum of ranks (the lower the better).

Generalisation Ability A SSM should be able to represent any instance of the class. To measure this, we look at the performance of the model when it has to represent unseen instances. We use the leave-out method, meaning we build the SSM including all thyroids except $N = 3$. These are chosen randomly in each iteration. We then look at the average Chamfer Distance of the unseen thyroids to the mean shape of the resulting SSM. This is repeated ten times to reduce the impact of randomness. Overall, for a SSM based on

reference thyroid X_j with a left-out set M, we have

$$G(S_j) = \frac{1}{N} \sum_{X_i \in M} \text{ChamferDistance}(X_i, S_{j \setminus M})$$

Specificity For a model to be specific, it should only generate instances of the object which are similar to those in the training set. We assess this qualitatively by generation instances using the SSM and then compare it to raw examples from the training set.
We take $N = 10$ shapes R_{ij} we randomly generated from the SSM S_j and find the nearest of the 16 training set thyroids X_i.

$$S(S_j) = \frac{1}{N} \sum_{i \leq N} \min_{k \in train} (\text{ChamferDistance}(R_i, X_k))$$

Compactness A SSM is compact if the variance is as little as possible and requires as few parameters as possible. For this, we look at the standard deviations λ_i over all N points i.

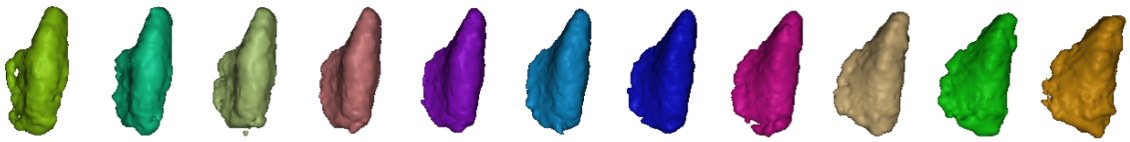
$$C(S) = \frac{1}{N} \sum_{I=1}^{N} \lambda_i$$

6.1.3 Choice of Reference Thyroid

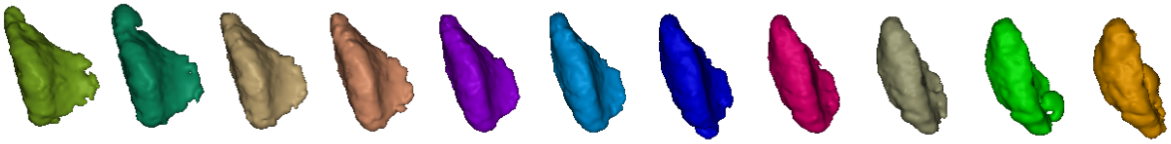
For building the SSM we need to find a reference thyroid. To this thyroid all correspondences for each thyroid are computed. We evaluate all 16 SSMs for each side and compare the results, which can be found in tables 1 and 2 in the appendix.
For the left thyroid lobe using thyroid 10 gives the best results. For the right lobe we use the thyroid 17 as the reference thyroid. The results are consequently better for the left lobes than for the right lobes.

6.1.4 Analysis of the Statistical Shape Model

The computed correspondences for the chosen reference thyroids can be found in figures 16 and 17 in the appendix. As described in section 4.2, we then create a SSM for each side of the thyroid, which can be seen in figure 7.
We also compared our proposed model with some modified versions. As can be seen in figure 15, the bijectivity measure of thyroids 19 and 21 are significantly worse in comparison to the rest. Therefore, we created a SSM with the same reference thyroid without these two outliers. Table 3 shows that this clearly improves the compactness and generality of the model. One can conclude that these shapes bring much variation to the model, which can also be seen in figure 17. Instead of reducing our dataset by the two shapes, we decided to take all thyroids into account.
Furthermore, one can deduce from 3 that smoothing steps in the mesh reconstruction improves the generality score. A figure comparing the mean shape with and without smoothing can be found in figure 18. At last, the results show the great impact of our outlier detection methods, which are crucial to obtain reasonable meshes in the end.



(a) SSM for left lobes with reference thyroid 10



(b) SSM for right lobes with reference thyroid 17

Figure 7: The mean shape of the model is located in the middle, i.e. at the 6^{th} place. Going from the mean to right, more standard deviation is added to the mean. When going to the left, standard deviation is subtracted.

6.2 Partial Registration

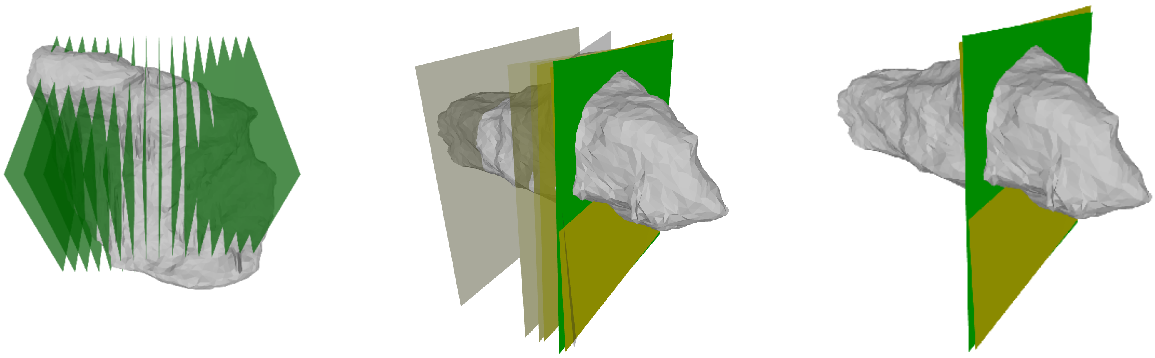
6.2.1 Evaluation Metrics

We employ several evaluation metrics for the two stages of our approach.

- For the neural network encoder, different network architectures with different training hyperparameters are compared based on the **training/validation loss**. Additionally, the runs are compared on both loss functions using the metric.
- We use **positive/negative distance histograms** to evaluate how well the two classes of triplet learning samples – positive and negative samples – are separated. For each 2D anchor patch, the distance between its embedding vector and the embedding of the positive sample and of the negative sample is measured. The results of these distances are displayed in histograms.
- To determine the best-matching slice at test time, we use the candidate slice which minimizes the **Procrustes loss** described in Section 5.3.
- To evaluate slice matching, we introduce the **slice mean distance** metric which indicates how similarly two slices are located and oriented. Let S be a ground-truth slice and $\hat{S}$ a predicted slice. Let x be uniformly distributed on S, and $\hat{x}$ be the point corresponding to x on $\hat{S}$. We define the slice mean distance as $d_{\text{mean}}(S, \hat{S}) = \mathbb{E}(||x - \hat{x}||)$. We estimate this integral via discretization. For examples that help interpret this metric, refer to figure 19a. All the results of slice mean distance are shown in section A.2.

6.2.2 Encoder Evaluation

In the following, the effect of different hyperparameter choices on the encoder training and validation loss is described. The plots of the loss curves are found in the appendix (section A.2).
For different experiments, we use between 40 - 60 epochs of training and generally observe a convergence of the validation loss within that period.



(a) Subset of 50 slices (b) True/Candidate slices (c) Selected candidate slice

Figure 8: Examples of slices used in slice identification experiments

6.2.2.1 Effect of Patch Sizes and Slice Depth When the patch size is increased in the x and y direction, there are slight improvements in the loss curves. When the slice depth is decreased, there are slight deteriorations in the loss curves. This is an expected effect, and presents a tradeoff between embedding quality on one side and computational cost and depth of available ultrasound data on the other side. One can also observe that pairs of patch sizes that represent a voxel grid of the same volume but with thinner slices, such as $(32 \times 32 \times 32)$ vs. $(64 \times 64 \times 8)$ vs. $(128 \times 128 \times 2)$, the thinner but wider patches tend to give better results. This can be caused by the fact that only the US patches, not the SDF patches, are made thinner. The absence of sufficient depth in the US data is covered by the bigger size of the patches, although this comes with a higher computational cost (figure 20).

6.2.2.2 Effect of Shuffling and Perturbation As described in section 5.5, the negative samples are reshuffled to prevent overfitting. This effect is achieved as shown in figure 21: without reshuffling the validation curves increases after 10-20 epochs, while with shuffling there is no overfitting.

The goal of positive sample perturbation is to make the identification outputs more robust, which cannot be evaluated on the loss curves. Nonetheless, the losses increase with higher pertubation, showing that perturbation makes the training harder for the networks.(figure 22)

6.2.3 Slice Identification Experiments

To evaluate the slice identification method, experiments are run on the train and validation data. For each thyroid lobe, 50 axis-aligned slices evenly spaced along the z axis (figure 8a) are used. Then the slice identification algorithm (section 5.3) extracts 6 candidate slices for each ground-truth slice (figure 8b). The candidate slices are ordered by their Procrustes loss with candidate 1 having the lowest loss, which is the selected candidate (figure 8c). For each candidate, slice-mean distance loss is computed. Furthermore, the influence of different hyperparameters and the results of the configurations are investigated.

Effect of Patch Size The effect of different patch sizes is investigated. Patch sizes are varied in x-y by increasing the M, N in figure 19b. If patch sizes are increased, we conjecture that the model can "see" more of the surrounding thyroid and use this

information to better locate the patch. This can be seen in figure 23, that the bigger the patch size the training loss reduces dramatically. Validation loss also decreases, but this is not always the case. Additionally, one can notice that bigger patches overfit. However, more experiments should be conducted to further explain the reasoning for this. Furthermore, bigger patches are computationally more expensive resulting in longer training time.

Effect of Slice Thickness As mentioned in section 5.1.1, US patches are assumed to have a certain depth. On the other hand, reducing the thickness of slices brings it closer to an actual 2D slice, potentially making the algorithm more useful for practical applications. The performance degrades for thinner patches, but still shows a considerably good result.

Effect of matching patch amount As in section 5.3, k_{slice} US slice patches and k_{thyr} SDF patches are sampled. Among this, k_{match} pairs are considered as matches. We experiment with varying these numbers. It shows that a larger number of sampled patches improves the result with a computational cost. The ratio $\frac{k_{\text{match}}}{k_{\text{slice}}}$ influences the accuracy of the orientation: if too big, US patches are forced to take SDF matches although no close match exists. If too small, fewer matches are found, which causes the false-positive matches to have a greater effect. (figure 24)

Effect of Loss Function As described in section 5.2.2, two types of loss functions were used to train the encoder, the weighted soft-margin triplet loss $\mathcal{L}_{\text{wsm}}$ and the distance matching loss $\mathcal{L}_{\text{dm}}$. The distance matching loss gives better results on the training data, but worse results on the validation data compared to the weighted soft-margin loss (figure 25). However, the violin plot shows that distance matching loss has fewer outliers and has better distribution in data.

Best Model Selection Based on the full evaluation results, a few trends stand out: Using a higher number of samples tends to give better results, but is proportionally more expensive. The distance matching loss leads to considerably better results on the training data, but comparable or slightly worse results on the validation data compared to the triplet loss. However, it tends to have fewer outlier slice predictions and can therefore be seen as more robust.

Based on these observations and the validation errors, we select three model configurations **A**, **B** and **C** for different use cases. **A** is the configuration that gives the best results but at a high computational cost. **B** is the configuration that gives the best results among the less expensive runs (fewer patch samples). **C** is a configuration that has a slightly higher validation error than **B**, but with fewer outlier predictions. The configurations of **A**, **B**, **C** are as follows:

	k_{slice}	k_{thyr}	k_{match}	US patch size	SDF patch size	Loss	Slice mean dist.
A	500	1500	250	$(32 \times 32 \times 32)$	$(32 \times 32 \times 32)$	$\mathcal{L}_{\text{wsm}}$	29.27
B	100	300	50	$(50 \times 50 \times 8)$	$(50 \times 50 \times 32)$	$\mathcal{L}_{\text{wsm}}$	22.28
C	100	300	50	$(64 \times 64 \times 8)$	$(64 \times 64 \times 32)$	$\mathcal{L}_{\text{dm}}$	31.58

These selected models are evaluated on the test set. The errors on the test set are similar to the validation error and are shown in figure 28. They are also reported in the above table.

6.3 Cross-Thyroid Slice Identification

We train a cross-thyroid encoder with patch sizes ($32 \times 32 \times 32$) and $\mathcal{L}_{\text{wsm}}$ loss. The results of the cross-thyroid slice identification between the two left lobes are qualitatively evaluated. We observed examples of slices where the slice transfer works well, but also observe that the task is harder than intra-thyroid slice identification (see figure 9).

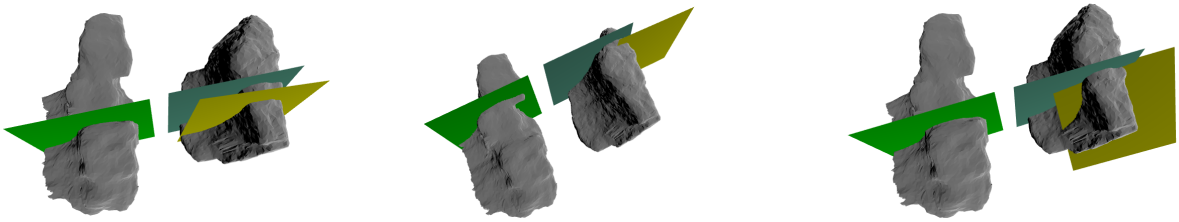
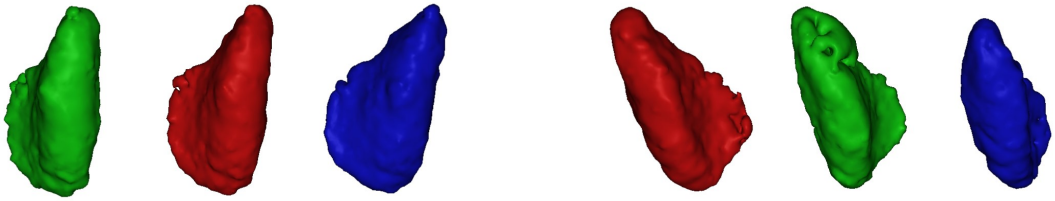


Figure 9: Cross-Thyroid result (blue: surrogate ground truth, by section 7.3 method)

7 Use Cases

7.1 Random Thyroid Generation

When dealing with medical data, the amount of available data is often limited. Also in our project we had to work with the sample size of 16 thyroids. Therefore, it can be of great importance to generate new sampled data. For this one can use the SSM to create realistic representations of the thyroid. We sample uniformly a standard deviation coefficient between -1 and 1. Then, we form the associating SSM shape. To increase the variation, we add a normal-distributed sample to each point of the shape. In figure 10 we can see the different generated thyroid shapes.



(a) Randomly generated left thyroids

(b) Randomly generated right thyroids

Figure 10: Creation of random thyroids

7.2 Partial Correspondence

Using deep functional maps also allows for computing partial correspondences. The framework learns from the categories of shapes that are represented in the training data, hence it does not depend on any specific shape model. Particularly, the objects don't need to be complete shapes. Instead, it is sufficient if different types of partiality are sufficiently represented in the training set. [18]
In the medical setting 3D shapes can show partiality such as missing parts or having holes due to acquisition errors [41].
In figure 11, we have partial shapes with missing parts of the thyroid corresponded to the SSM.

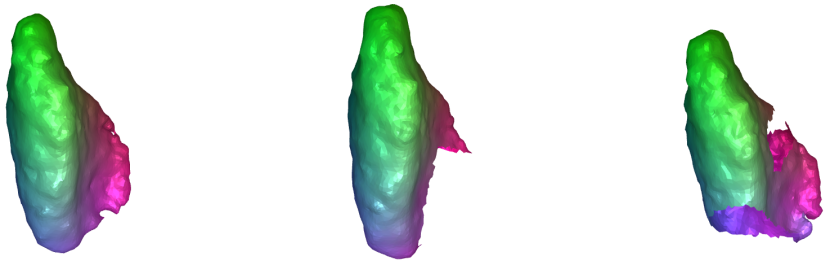


Figure 11: Each partial thyroid is corresponded to to full mean shape on the left

7.3 Slice Transfer

Having a located slice on a thyroid through the method of partial registration, one is capable to transfer the slice to the SSM, for example to the mean shape. Because of the simplifications made in section 5.3, a slice is uniquely determined by the center. Given center of a slice, which intersects the thyroid mesh, one can find its k nearest neighbors within the vertices of the mesh. For this, we use the algorithm from [42]. From the SURFMNet model, we have now correspondences to mean shape which are used to find the k corresponding point on the target mesh. Now, we take the mean of these points to obtain the predicted center of the slice on the target mesh.
In order to evaluate this method, we first predict the slice on the target mesh. Then, we transfer the predicted slice back to the source by the same procedure. Now, we can measure the distance between the center of the original slice and resulting slice after two transfers. X slices were generated on each thyroid and the distances between the centers can be seen in a histogram in figure 12, One can deduce that the right lobes have much more outlier. This can be traced back to the fact that the correspondences are more robust for the left lobes. In figure 13, one can see an example of a slice transfer. Its distance of centers is 13.57. The mean distance is 11.52 for the left lobes and 14.04 for the right lobes. So this example reflects approximately the mean error.

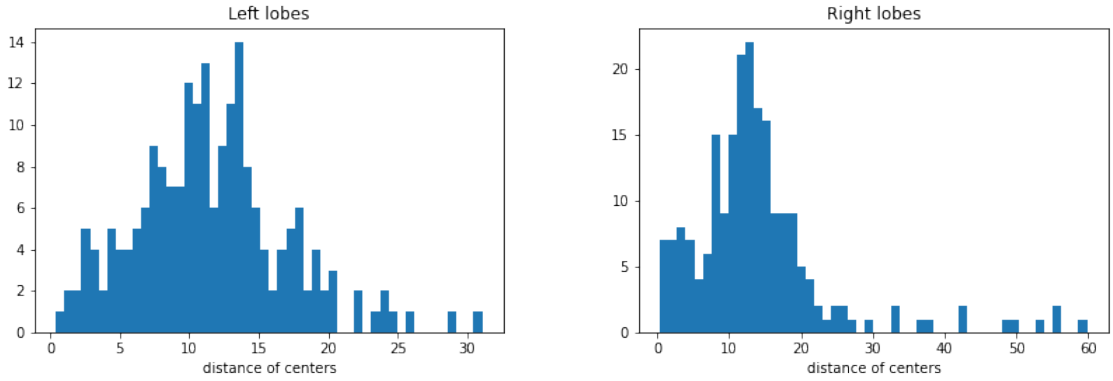


Figure 12: Evaluation of the slice transfer

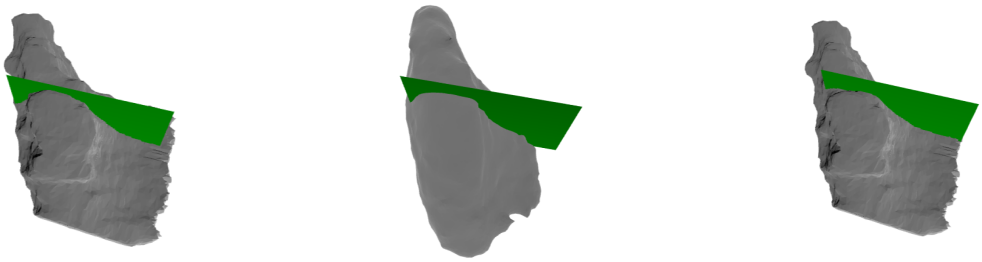


Figure 13: From the original slice on the source (left) to the predicted slice on the mean shape from the SSM (middle) and again back to source with the predicted slice (right).

8 Conclusion

First, we introduced a 3D statistical shape model of the thyroid. One of the biggest steps in creating a reasonable statistical shape is finding the correspondences between the shapes, for which we presented an unsupervised neural network built on the deep functional maps framework. With these correspondences, we were able to find the best reference thyroid and construct a mean shape with variation. We analyzed the variations of the 3D model and compared the left and right lobes of the thyroid gland. Applications can be found in the random generation of realistic representations or partial correspondence findings.

The presented method is an easy, scalable, and efficient approach to creating a 3D statistical shape model using a small data set. In further steps, we plan to include landmarks in the correspondence method to obtain 1:1 point correspondences. This weakly-supervised method could lead to eigenshapes taking higher dimensional variability into account.

Second, we presented a method to locate 2D thyroid US scans in a 3D thyroid model. For this 2D / 3D registration, an encoder network was trained and patches were put through the learned encoder networks to produce embeddings. Then, slices were located using the best of several Procrustes predictions. We selected three model configurations for different use cases and evaluated them on the test set. Our results show good orientation results of 2D US slices on a 3D thyroid model, as seen in section 6.2.

Regarding future directions, there are several promising approaches to improve partial registration. One could pre-train the 3D U-Net on a segmentation task, as was done in [11]. Furthermore, one could try different ways of computing (dis-)similarities of embeddings, such as dot products, or even train a classifier that predicts whether the two samples come from the same region. With such a soft match prediction, one could use matches with high certainty for the orientation task. Another direction would be to test the method on non-axis-aligned slices and train it on rotated patches to improve its performance. Moreover, one could incorporate keypoints for choosing patches to have a higher probability of generating 2D and 3D patches in similar positions. Finally, there is potential to try other methods for hypothesis generation than the KDE-based algorithm we described, and compare what approach performs best.

References

[1] D. Nam, R. Barrack, and Hollis G Potter. What are the advantages and disadvantages of imaging modalities to diagnose wear-related corrosion problems? *Clinical Orthopaedics and Related Research*, 472(12):3665–3673, 2014.

[2] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Fast image-based localization using direct 2D-to-3D matching. In *2011 International Conference on Computer Vision*, pages 667–674. IEEE, 2011.

[3] Mengdan Feng, Sixing Hu, Marcelo H Ang, and Gim Hee Lee. 2D-3D-MatchNet: Learning to match keypoints across 2D image and 3D point cloud. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4790–4796, 2019.

[4] Baiqi Lai, Weiquan Liu, Cheng Wang, Shuting Chen, Xuesheng Bian, Xiuhong Lin, Chenglu Wen, and Jonathan Li. Metric learning for 2D image patch and 3D point cloud volume matching. In *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pages 3416–3419, 2021.

[5] Paul M Rea, editor. *Biomedical Visualisation*. Advances in experimental medicine and biology. Springer Nature, Cham, Switzerland, 1 edition, April 2019.

[6] Felix Ambellan, Hans Lamecker, Christoph von Tycowicz, and Stefan Zachow. Statistical shape models: Understanding and mastering variation in anatomy. In *Advances in Experimental Medicine and Biology*, Advances in experimental medicine and biology, pages 67–84. Springer International Publishing, Cham, 2019.

[7] Tobias Heimann and Hans-Peter Meinzer. Statistical shape models for 3D medical image segmentation: a review. *Med. Image Anal.*, 13(4):543–563, August 2009.

[8] Cleveland Clinic. Anatomy of the thyroid gland. `https://my.clevelandclinic.org/health/body/23188-thyroid`, 2022. Accessed: 2022-07-22.

[9] Cancer Reasearch UK. Thyroid cancer statistics. `https://www.cancerresearchuk.org/health-professional/cancer-statistics/statistics-by-cancer-type/thyroid-cancer#heading-Two`, 2015. Accessed: 2022-07-22.

[10] Markus Krönke, Christine Eilers, Desislava Dimova, Melanie Köhler, Gabriel Buschner, Lilit Mirzojan, Lemonia Konstantinidou, Marcus R. Makowski, James Nagarajah, Nassir Navab, Wolfgang Weber, and Thomas Wendler. Tracked 3D ultrasound and deep neural network-based thyroid segmentation reduce interobserver variability in thyroid volumetry. *CoRR*, abs/2108.10118, 2021.

[11] Lemonia Konstantinidou. 3D ultrasound compounding for volume estimation in thyroid diagnostics. Master's thesis, 09 2021.

[12] Michael Garland and Paul S Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 209–216, 1997.

[13] Hans Lamecker, Thomas Lange, and Martin Seebass. A statistical shape model for the liver. In Takeyoshi Dohi and Ron Kikinis, editors, *Medical Image Computing and Computer-Assisted Intervention — MICCAI 2002*, pages 421–427, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[14] Marcel Lüthi, Thomas Gerig, Christoph Jud, and Thomas Vetter. Gaussian process morphable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(8):1860–1873, 2018.

[15] Manasi Datar, Ilwoo Lyu, Sunhyung Kim, Joshua Cates, Martin A Styner, and Ross Whitaker. Geodesic distances to landmarks for dense correspondence on ensembles of complex shapes. 16(Pt 2):19–26, 2013.

[16] Won-Ki Jeong and Ross Whitaker. A fast iterative method for eikonal equations. *SIAM J. Scientific Computing*, 30:2512–2534, 01 2008.

[17] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (ToG)*, 31(4):1–11, 2012.

[18] Or Litany, Tal Remez, Emanuele Rodolà, Alex M Bronstein, and Michael M Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. 2017.

[19] Federico Tombari, Di Salti, Samuele, and Luigi Stefano. Unique signatures of histograms for local surface description. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, pages 356–369, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[20] Weiquan Liu, Baiqi Lai, Cheng Wang, Xuesheng Bian, Chenglu Wen, Ming Cheng, Yu Zang, Yan Xia, and Jonathan Li. Matching 2D image patches and 3D point cloud volumes by learning local cross-domain feature descriptors. In *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 516–517. IEEE, 2021.

[21] Quang-Hieu Pham, Mikaela Angelina Uy, Binh-Son Hua, Duc Thanh Nguyen, Gemma Roig, and Sai-Kit Yeung. Lcd: Learned cross-domain descriptors for 2D-3D matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11856–11864, 2020.

[22] Viktoria Markova, Matteo Ronchetti, Wolfgang Wein, Oliver Zettinig, and Raphael Prevost. Global multi-modal 2D/3D registration via local descriptors learning. *arXiv preprint arXiv:2205.03439*, 2022.

[23] Jean-Michel Roufosse, Abhishek Sharma, and Maks Ovsjanikov. Unsupervised deep learning for structured shape matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1617–1627, 2019.

[24] Yonathan Aflalo, Haim Brezis, and Ron Kimmel. On the optimality of shape and data representation in the spectral domain. *SIAM Journal on Imaging Sciences*, 8(2):1141–1160, 2015.

[25] Maks Ovsjanikov, Etienne Corman, Michael Bronstein, Emanuele Rodolà, Mirela Ben-Chen, Leonidas Guibas, Frederic Chazal, and Alex Bronstein. Computing and processing correspondences with functional maps. In *SIGGRAPH ASIA 2016 Courses*, SA '16, New York, NY, USA, 2016. Association for Computing Machinery.

[26] Raj Kishor Singh and Jasbir Singh Manhas. *Composition operators on function spaces*. Elsevier, 1993.

[27] Dorian Nogneng and Maks Ovsjanikov. Informative descriptor preservation via commutativity for shape matching. In *Computer Graphics Forum*, volume 36, pages 259–267. Wiley Online Library, 2017.

[28] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32(3), jul 2013.

[29] Jörg Vollmer, Robert Mencl, and Heinrich Mueller. Improved laplacian smoothing of noisy surface meshes. In *Computer graphics forum*, volume 18, pages 131–138, 1999.

[30] Gabriel Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358, 1995.

[31] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3D U-Net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016.

[32] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[33] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2015.

[34] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[35] Sixing Hu, Mengdan Feng, Rang MH Nguyen, and Gim Hee Lee. Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7258–7267, 2018.

[36] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11(36):1109–1135, 2010.

[37] Haijun Liu, Jian Cheng, Wen Wang, and Yanzhou Su. The general pair-based weighting loss for deep metric learning. *arXiv preprint arXiv:1905.12837*, 2019.

[38] Peter H Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966.

[39] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[40] Rhodri Huw Davies. Learning shape: optimal models for analysing natural variability. Technical report, University of Manchester, 2002.

[41] Souhaib Attaiki, Gautam Pai, and Maks Ovsjanikov. DPFM: Deep partial functional maps. 2021.

[42] Songrit Maneewongvatana and David M Mount. Analysis of approximate nearest neighbor searching with clustered point sets. *arXiv preprint cs/9901013*, 1999.

A Appendix

A.1 Statistical Shape Model

A.1.1 Preprocessing

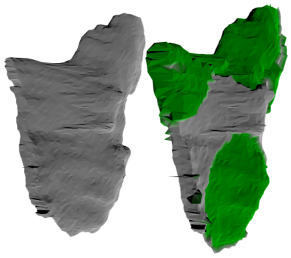
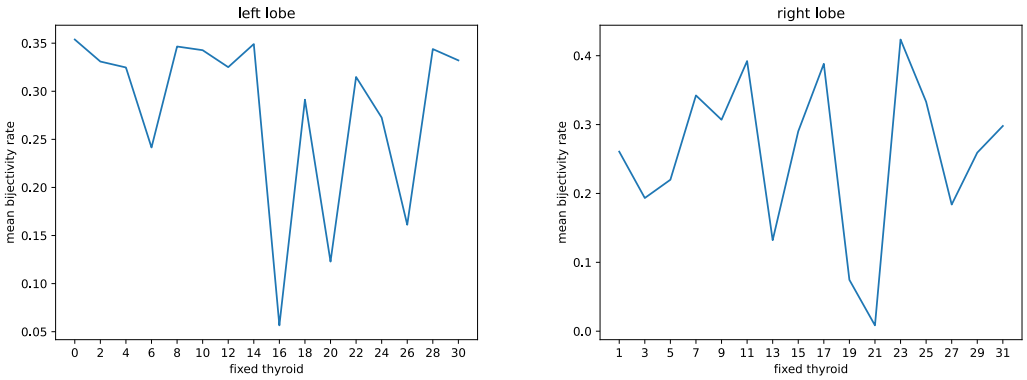


Figure 14: Left: original thyroid; Right: augmented thyroid

A.1.2 Evaluation of Correspondence Problem



(a) (b)

Figure 15: Average bijectivity rate from one thyroid to all other thyroids

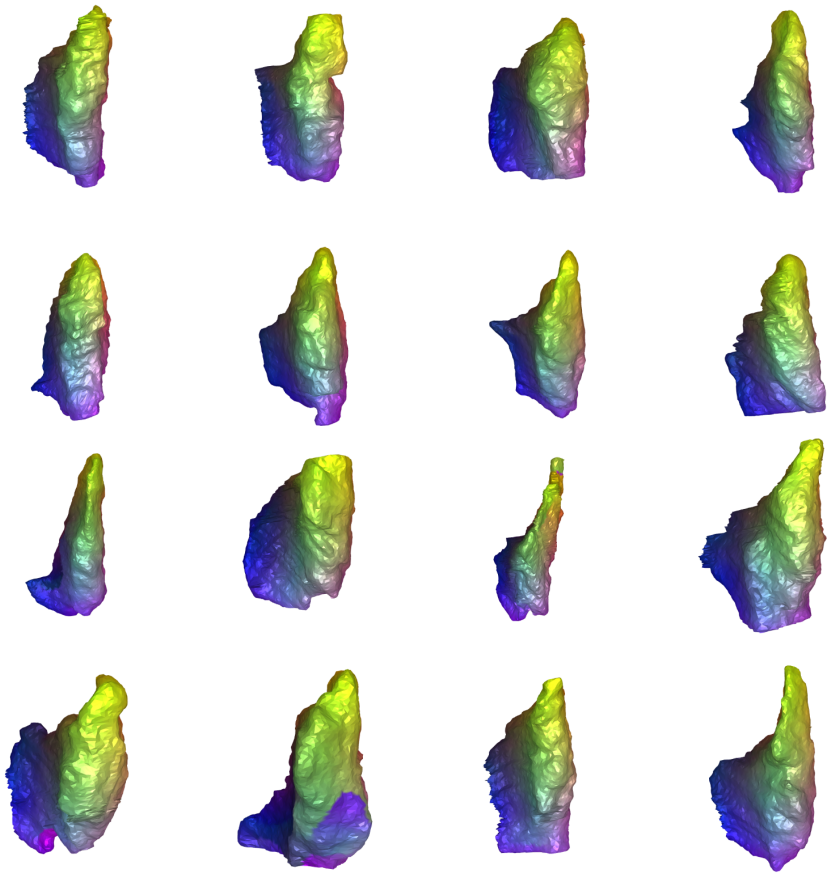


Figure 16: Correspondences of left thyroids to reference thyroid 10 in the lop left corner. Areas of the same color correspond to each other.

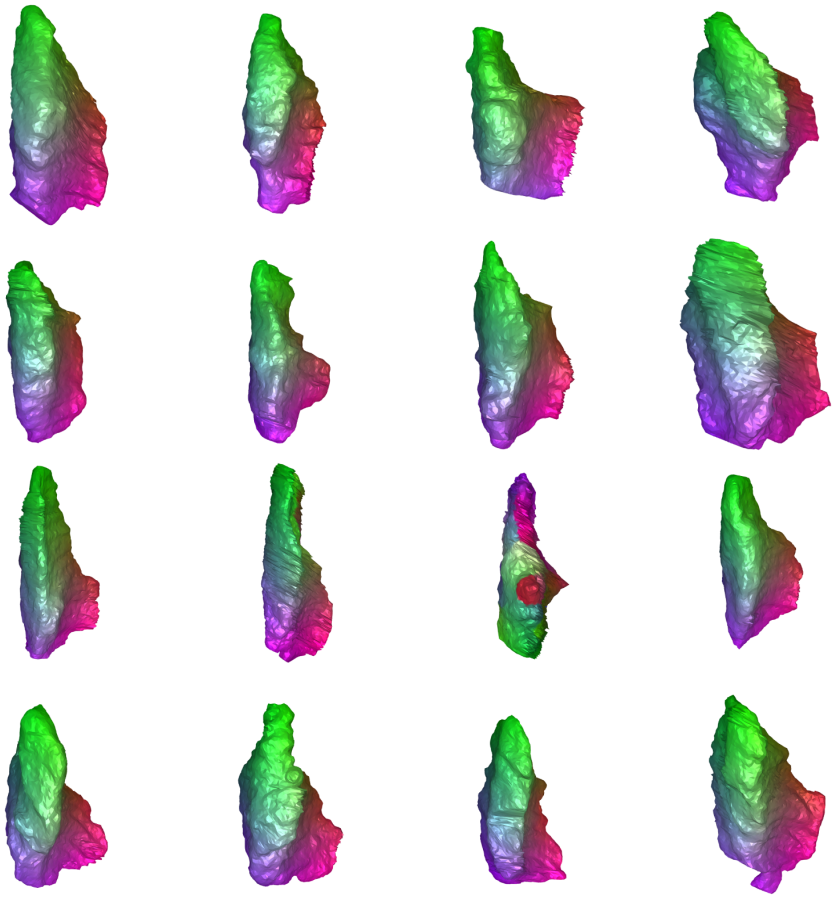


Figure 17: Correspondences of right thyroids to reference thyroid 17 in the lop left corner. Areas of the same color correspond to each other.

A.1.3 Evaluation of Statistical Shape Model

Thyroids	Specificity	Compactness	Generality	sum of ranks
0	2.5667	0.0329	5.9980	13
2	2.5816	0.0362	5.8861	19
4	2.7452	0.0359	6.3516	27
6	2.2326	0.0384	6.0292	22
8	2.8708	0.0341	6.9807	32
10	**2.5309**	**0.0331**	**5.9491**	**12**
12	2.7830	0.0333	5.3484	17
14	2.8130	0.0314	6.8765	23
16	6.6219	0.0471	7.2458	48
18	2.4773	0.0374	7.1914	29
20	4.3107	0.0376	5.3259	29
22	2.6774	0.0350	6.8993	27
24	2.6444	0.0365	5.9159	22
26	2.4864	0.0406	6.3714	27
28	2.6954	0.0335	6.9147	26
30	2.8783	0.0348	7.1255	35
Average	2.9947	0.0361	6.4006	

Table 1: Evaluation results of SSM including the left lobes

Thyroids	Specificity	Compactness	Generality	sum of ranks
1	2.4170	0.0473	7.4411	26
3	2.6173	0.0448	7.2130	23
5	3.0658	0.0446	8.1198	36
7	2.2644	0.0445	7.8554	21
9	2.5451	0.0438	7.0325	17
11	2.4392	0.0415	7.7205	19
13	2.4163	0.0410	7.9260	18
15	2.6310	0.0432	8.0451	29
17	**2.2752**	**0.0411**	**6.5976**	**6**
19	7.4713	0.0536	6.8980	32
21	33.2593	0.0729	7.7591	43
23	2.6213	0.0409	7.7440	20
25	3.6217	0.0449	7.2119	28
27	2.4257	0.0467	8.4313	34
29	5.0830	0.0438	7.3322	26
31	2.7750	0.0461	7.3353	30
Average	4.9955	0.0463	7.5414	

Table 2: Evaluation results of SSM including the right lobes

	Specificity	Compactness	Generality
Proposed model	**2.2752**	**0.0411**	**6.5976**
Without outliers 19 and 21	2.3766	0.0355	6.4479
Remeshing without smoothing	2.1732	0.0411	8.3491
Without any outlier detection	9.7650	0.0629	7.7882

Table 3: Analysis on SSM on right lobes with reference thyroid 17

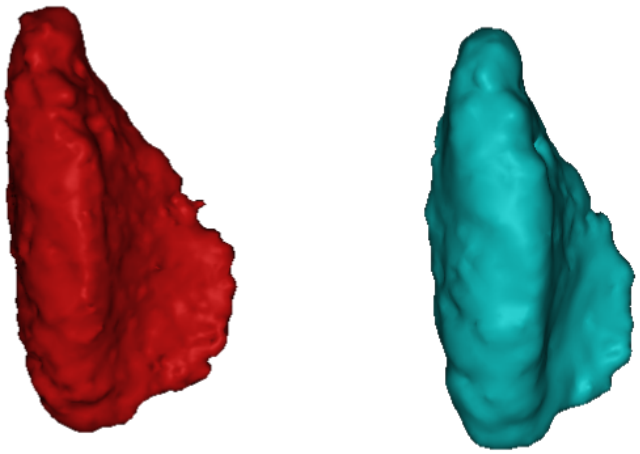


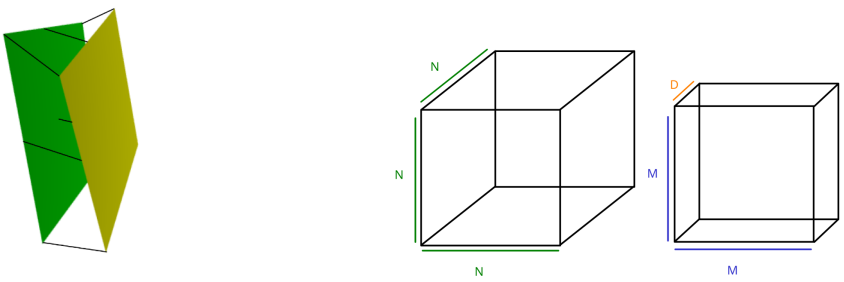
Figure 18: Left: SSM without smoothing; Right: SSM including smoothing

A.2 Partial Registration: Additional Figures

Figure 19a illustrates corresponding points in the definition of slice mean distance. Figure 19b shows the patch dimensions.

Partial Registration: Network training Loss curves

Figure 20 demonstrates the effect of different patch sizes on the loss curves. Figure 21 shows that reshuffling prevents overfitting. Figure 22 illustrates that perturbation increases the loss.



(a) Measurement of slice mean distance (b) SDF patches and US patches

Figure 19: Additional illustrations for partial registration part

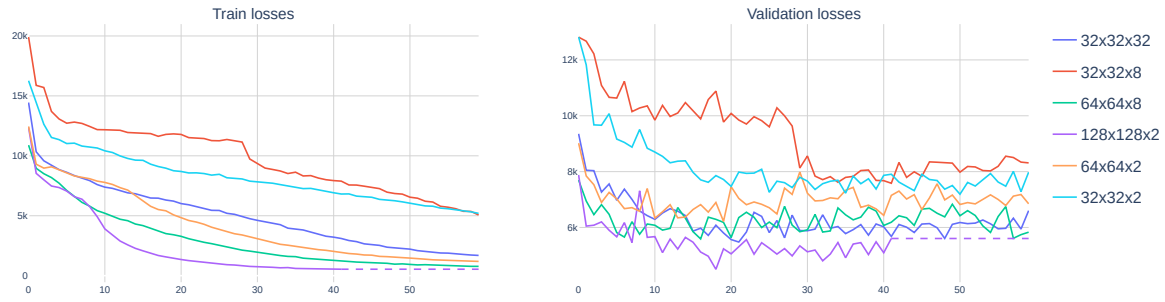


Figure 20: Distance matching loss with different patch dimensions

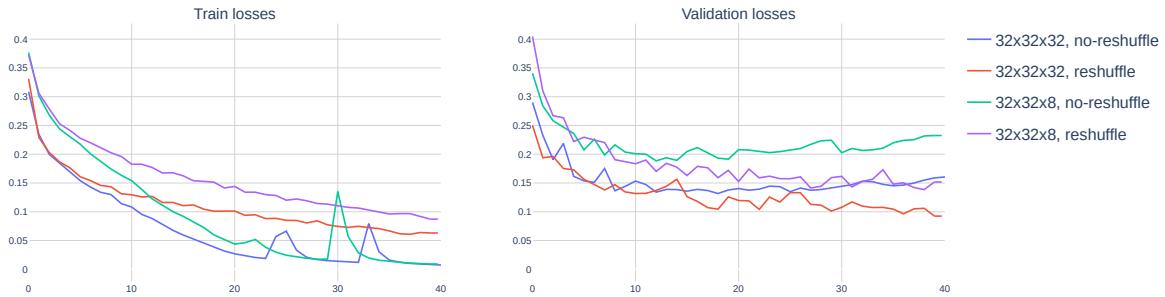


Figure 21: Effect of reshuffling

Partial Registration: Selected Slice Mean Distance Statistics

Figure 23 and figure 24 illustrate the effect of patch sizes and hyperparameters. Figure 25 shows the effects of the two different loss functions and that there are less outliers with $\mathcal{L}_{\text{dm}}$. Figures 26 and 27 show the results of all mentioned runs in one plot. Figure 28 shows the performance of the three selected configurations **A**, **B**, **C** on the test set.

Selected slice mean angle and centroid distance

Figure 29 and 30 show the alternative metrics of slice centroid distance and slice angle for all experiments.

Slice matching results

Figure 31 illustrates how to interpret the slice mean distance metric: we sampled 9 estimations (yellow) for ground truth slices (green) and report their slice mean distances.

Slice distribution graphs

Figures 32, 33 and 34 show the distribution of selected candidate slice mean distances on the train, test and validation set for selected model **A**, **B** and **C**, respectively.

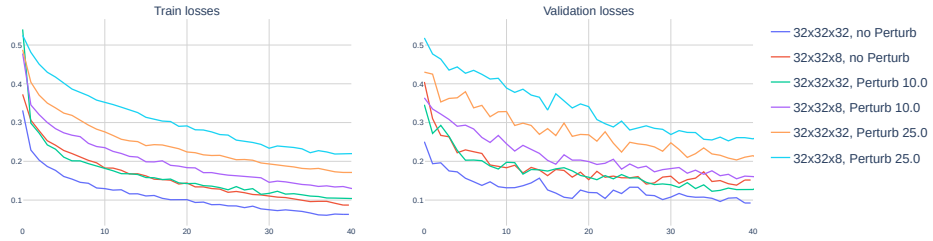


Figure 22: Effect of perturbation

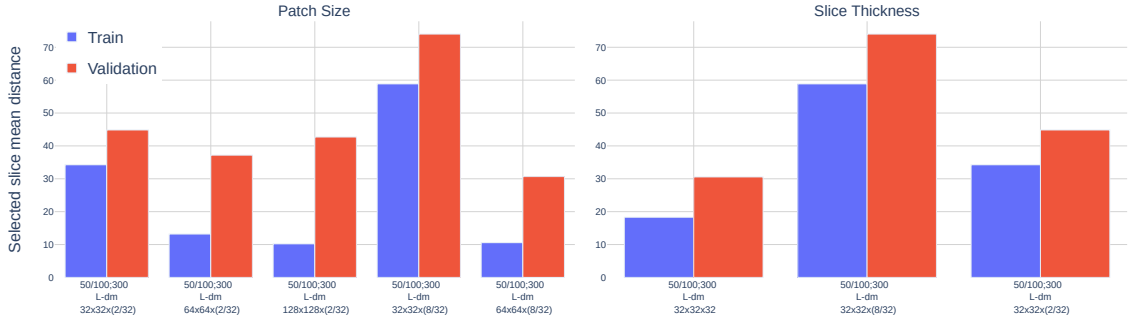


Figure 23: Slice mean distance of different patch size and slice thickness

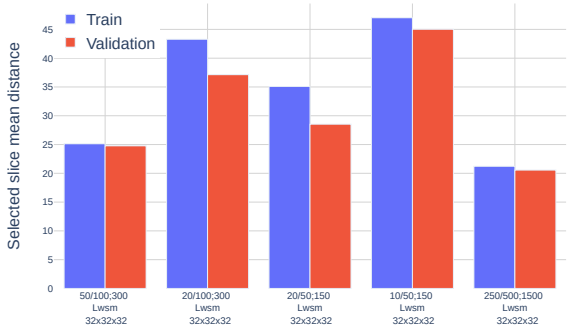


Figure 24: Slice mean distance of different amount of matching patches (notation: "number of taken matches/number of 2D slice patches; number of SDF patches")

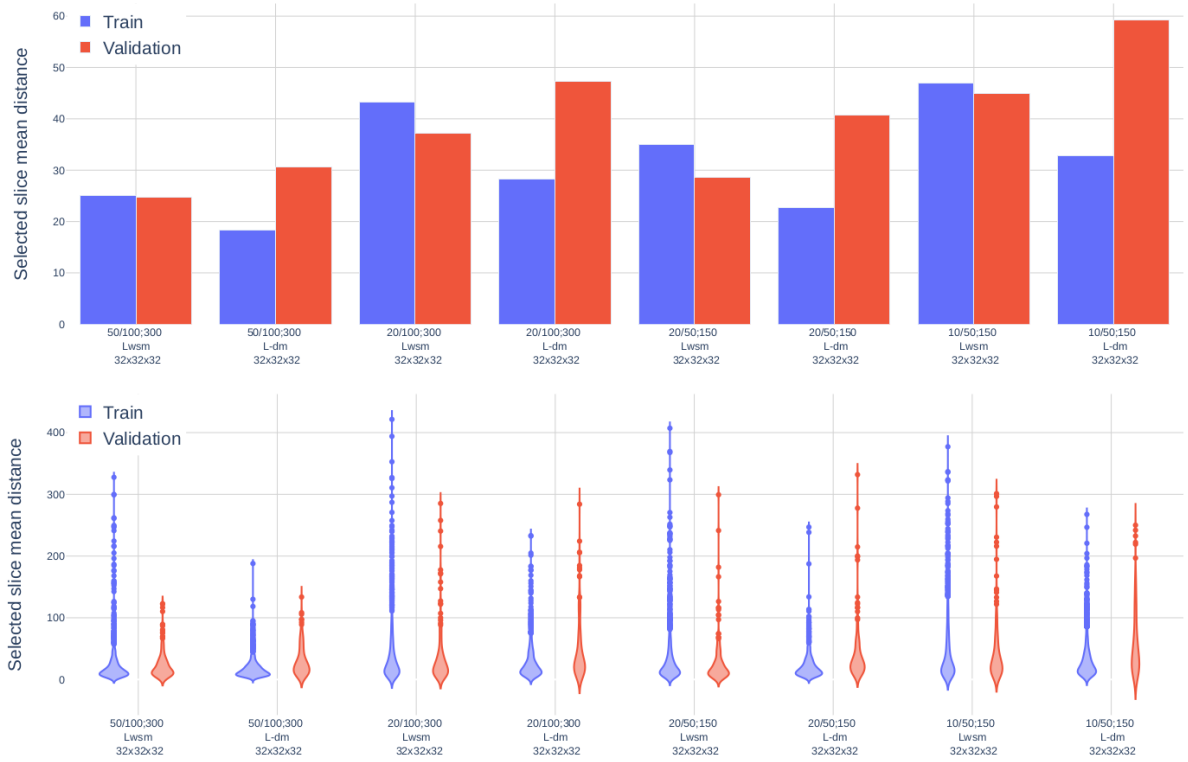


Figure 25: Slice mean distance bar plot and violin plot using different losses

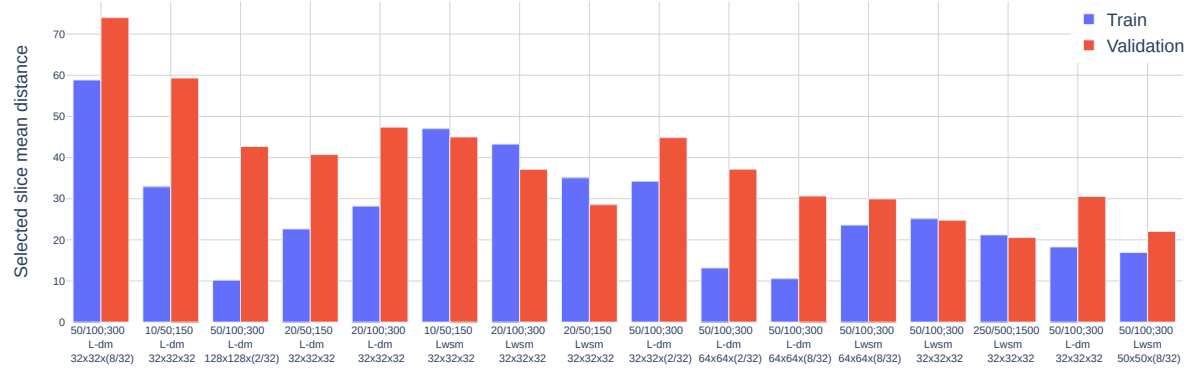


Figure 26: Slice mean distance of all experiments

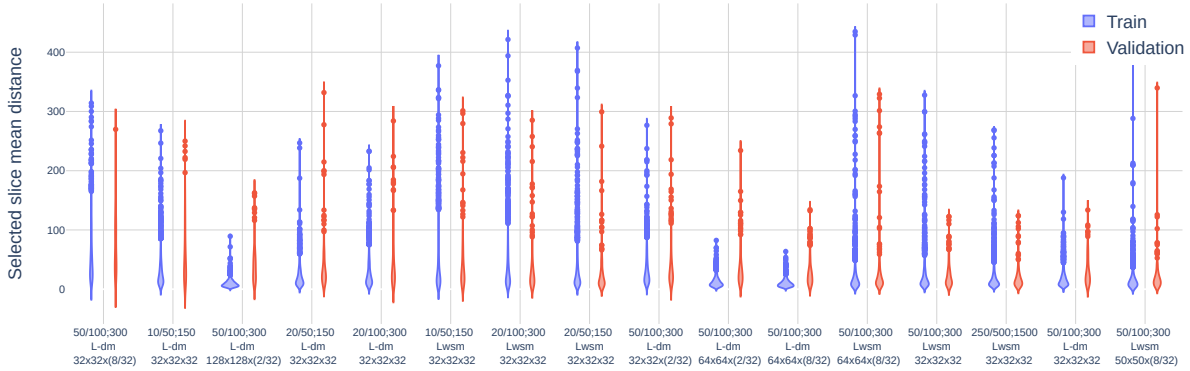


Figure 27: Slice mean distance violin plot of all experiments

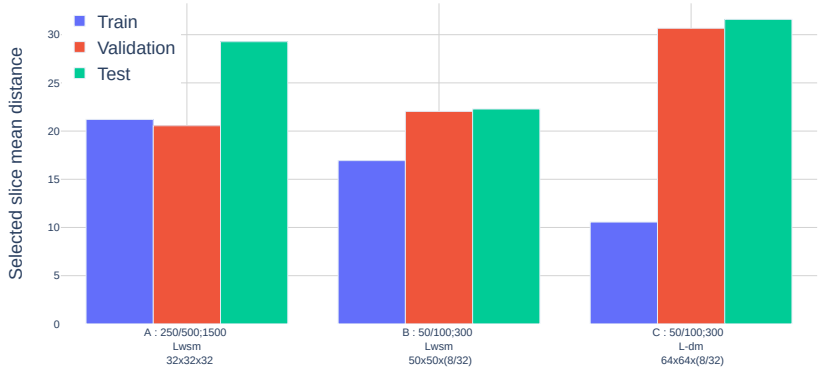


Figure 28: Three models slice mean distance test set performance

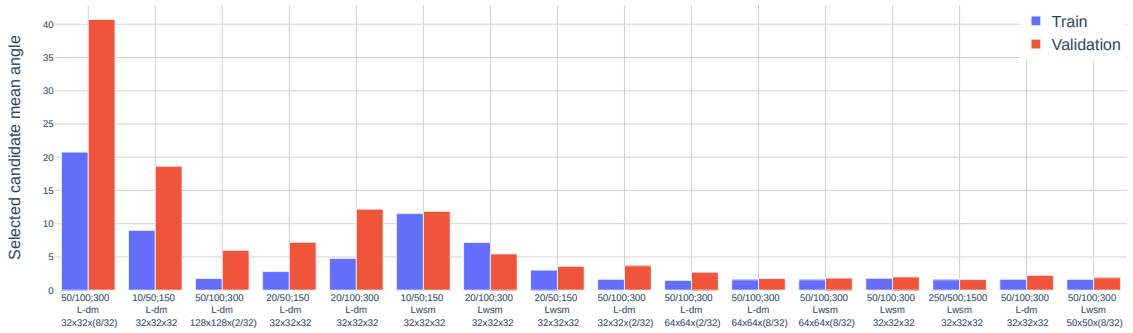


Figure 29: Slice mean angles of all experiments (in degrees)

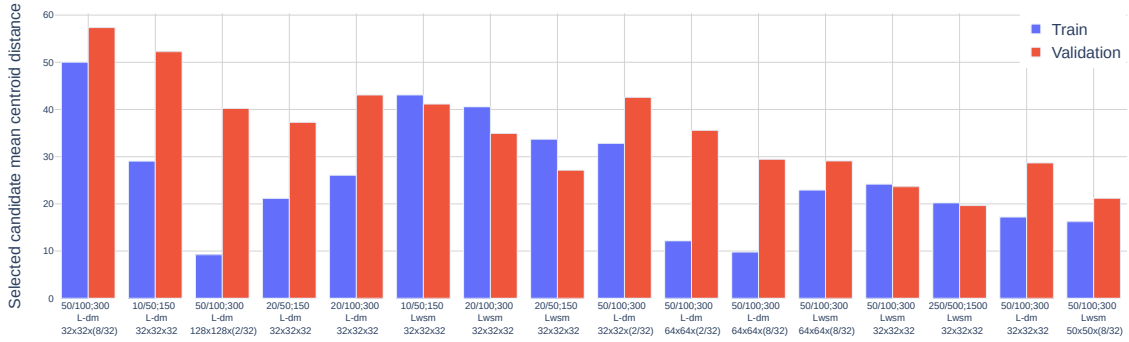


Figure 30: Slice centroid distances of all experiments

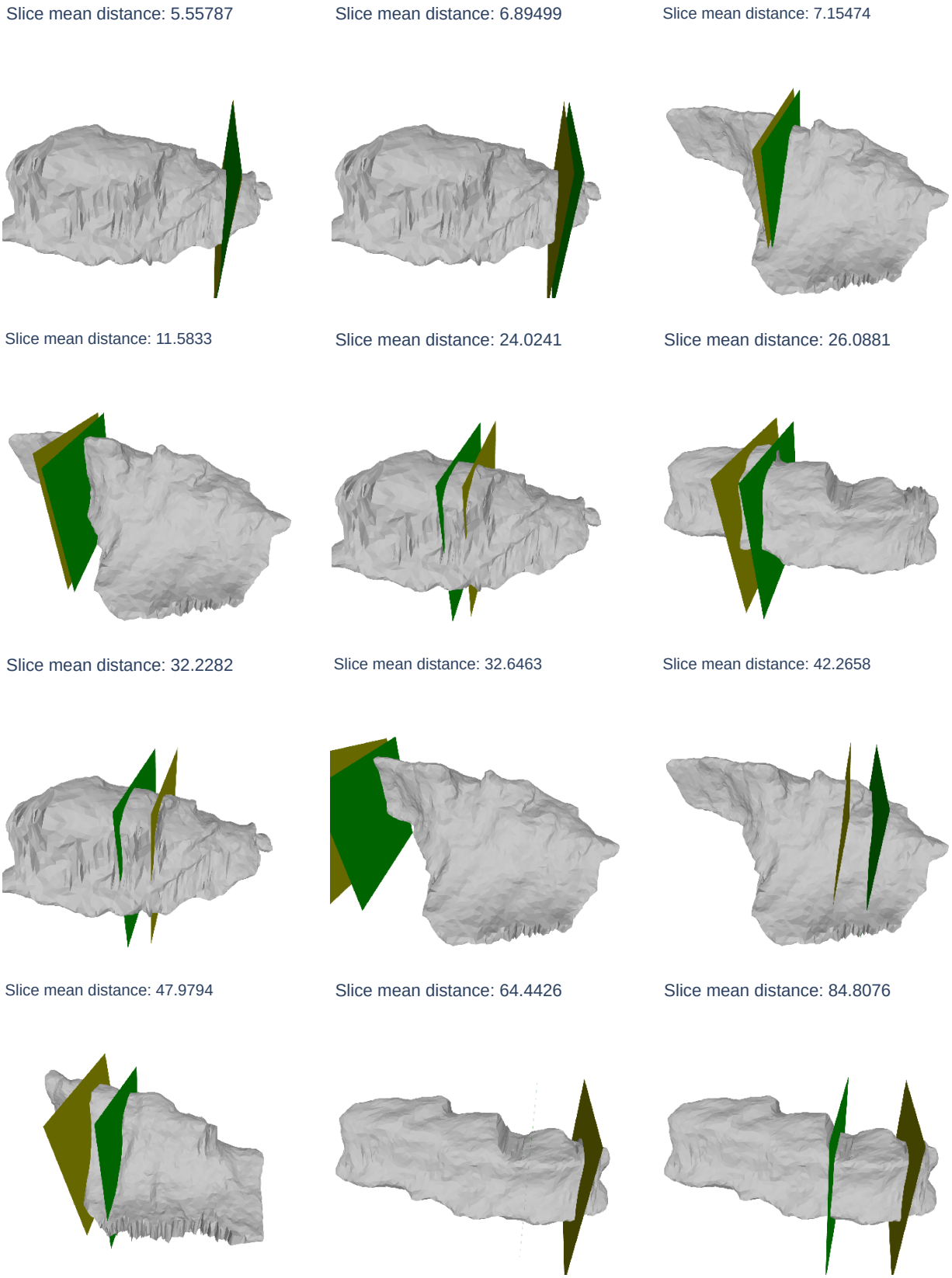


Figure 31: Samples of slice matching results with mean distances

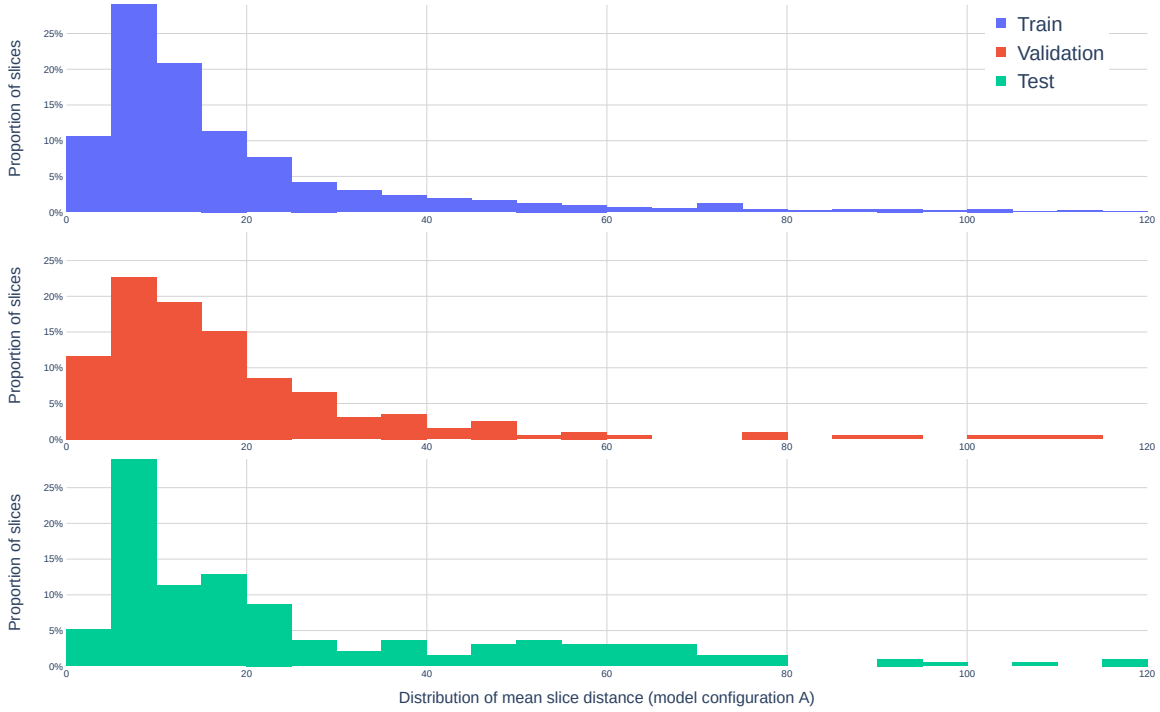


Figure 32: Slice distribution of model A

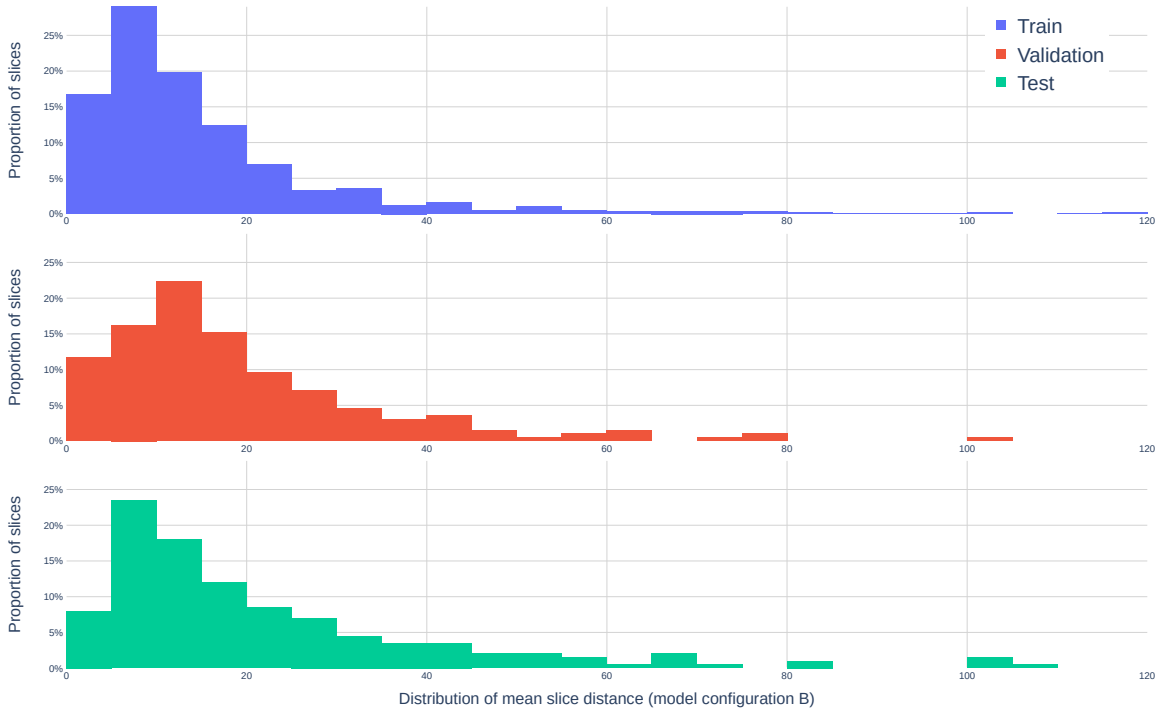


Figure 33: Slice distribution of model B

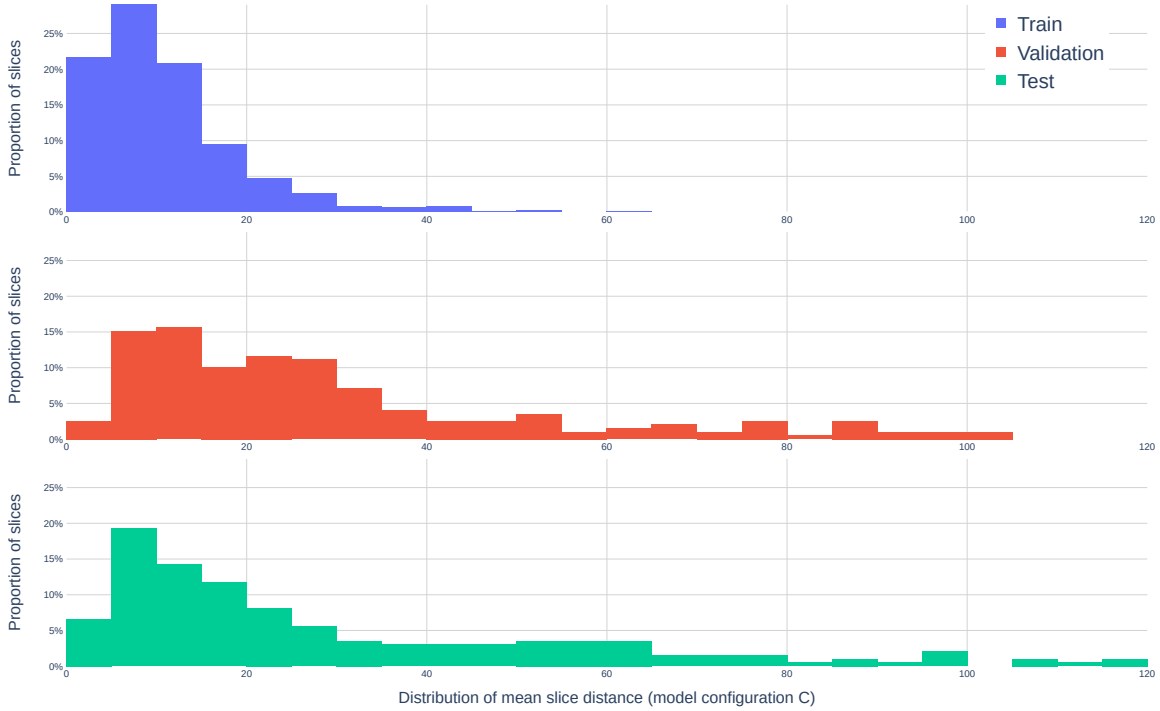


Figure 34: Slice distribution of model C