COSC470 Research Project Report

Contour Splitting for Branching Structures in CT Image Reconstructions

Cameron Stevenson

Supervisor: Ramakrishnan Mukundan

October 21, 2021

Abstract

Medical imaging modalities such as HRCT and MRI produce stacks of images depicting regions of different tissue. These image stacks can be used to reconstruct the 3D structure of internal organs. The renders or models produced are useful in diagnosis, treatment, education, surgical simulation, and robot assisted surgery [1, 2, 3]. Correspondence methods acting on contours segmented from scan images are one such way of 3D reconstruction. These methods have difficulty in reconstructing structures such as branches and bends. There is ambiguity in contour correspondence, and extra steps must be added to point correspondence. This paper builds upon prior correspondence methods, with a new approach to point correspondence, and specifically aiming to improve on reconstructing these problematic cases. Analysis has been performed on synthetic models and real scan data. The proposed method handles low plane count and branching structures better than prior correspondence methods, giving more accurate reconstructions. On average the proposed method improved in reconstruction accuracy compared to the most related prior method by 15.2%. The main contribution is the new techniques and algorithms developed for point correspondence, which may be useful in future applications.

Contents

0 Overview

The field of HRCT and MRI reconstructions will be introduced, followed by a broad background survey of stages in the reconstruction pipeline, with prior work relevant to this paper. The background summary will give a clear overview of what needs improvement. A method, "contour splitting and point angle", will be introduced which aims to satisfy these improvements. Analysis will show that this method does improve upon prior work in the intended way.

The latest version of this paper may be found here[1].

[1] `https://github.com/cstevenson3/cosc470writing/blob/main/report.pdf`

1 Introduction

Medical imaging modalities such as HRCT and MRI produce stacks of images depicting regions of different tissue. These image stacks can be used to reconstruct the 3D structure of internal organs. The renders or models produced are useful in diagnosis, treatment, education, surgical simulation, robot assisted surgery, and virtual reality [1, 2, 3, **?**]. The process of producing renders or reconstructions can be improved by reducing manual user input, improving reconstruction accuracy, or improving computational efficiency.

Prior methods include volumetric rendering (directly rendering the image data) or surface reconstruction (creating a mesh to render). Surface reconstruction methods include marching cubes, and point cloud methods such as screened Poisson. Recent papers [1, 2] have examined contour-based reconstructions, first segmenting images to find contours, then corresponding contours and points across slices. This paper will focus on extending this type of method.

1.1 Motivation

A paper by Mackay [1] investigates the use of dynamic time warping as an algorithm for point correspondence. The analysis shows two problems impacting reconstruction accuracy. Firstly, low slice count impacts the accuracy of all reconstruction methods analysed. Secondly, branches and other such structures can be problematic to reconstruct, affecting both contour and point correspondence.

This project aims to improve the accuracy of reconstructions involving branches and other problematic structures. In particular, point correspondence will be investigated. The lungs are used as an organ of focus, as a good example structure to work on. The bronchi have many branches and the tubular shapes are easy to understand.

2 Background

We begin with a look at typical methods in rendering or reconstructing scan data. Then, correspondence methods are looked at in depth as the approach this report builds upon.

2.1 Overview

Images from scans (also referred to as sections, slices, or planes) can be segmented based on intensity into pixel regions to define structure boundaries. These can be processed further by finding contours to represent these boundaries.

Volumetric rendering treats pixels in images as voxels, and a variety of rasterization and raycasting techniques are available for rendering these.

Surface rendering requires a surface be defined, either implicitly or as a mesh. Point cloud methods generate implicit surfaces, whilst meshes can be constructed via marching cubes or contour-based methods.

2.2 Applications

Xuyi et al. [4] use 3D reconstructions from hip CT scans to make patient-specific surgery plans. From the model they measure the direction and degree of the acetabular fragment, and use this to guide their surgery.

Pan et al. [3] refers to the importance of 3D reconstructions and rendering in robot-assisted surgery. With real time rendering being a priority, surface reconstructions are preferred over volume rendering. Of the four methods analysed, marching cubes was found to be the most suitable due to its speed and render quality.

Lim et al. [5] found that the use of 3D printed cardiac models in education resulted in a statistically significant improvement in test scores of medical students.

Lobachev et al. [**?**] developed a system for virtual reality rendering of reconstructions from immunohistological sections. They state that domain experts are better at analysing details in the original images, but only 3D reconstructions can offer a view of structures spanning across slices, such as blood vessels. They assert that virtual reality is the best way to allow these domain experts to view 3D reconstructions, giving them natural control over their view. High frames per second is important in VR to avoid nausea etc., and volumetric rendering may be unsuitable, with FPS as low as 25 [**?**]. For this reason a mesh is reconstructed with marching cubes to render.

2.3 Scanning and Image Processing

As with most areas involving measurement of the real world, unwanted artefacts such as noise are introduced in the imaging process. There can also be variation from subject to subject and between imaging machines. Therefore standard image processing techniques are used to pick out the parts of images which are relevant to the application. Running signal processing on the 2D data (as opposed to considering the whole image stack in the reconstruction stage) reduces complexity. However techniques considering the image data from all slices have been investigated.

2.3.1 Scanning

The effect of slice thickness on detection has been considered by Fischback et al. [6]. They found that reduced slice thickness improves on small nodule detection. They compared 5mm and 1.25mm reconstruction intervals. From this we infer that higher plane count scanning through an object is better. However for small objects, say 10mm in size, we may only get up to 10 planes passing through, and so reconstruction methods need to cope with smaller plane counts.

Similarly, slice thickness has been considered for volume estimiation by Laakso et al. [7]. They found no significant difference in estimated volume when using different slice thicknesses, although they recommend using smaller slices where possible. It is stated that smaller slice thickness results in longer scan times, so it may still be worth considering larger slice thicknesses and lower plane counts.

2.3.2 Segmentation

Segmentation is the process of picking out the pixels belonging to individual objects in an image. From this the projected geometry of the object onto the image can be inferred, which is then used in reconstruction or rendering. Birkfellner [8] observes that organs are usually composed of multiple tissue types, which show up as different intensities under imaging. This makes segmentation "a rather complex, specialized procedure often requiring considerable manual interaction". Particular organs are often focussed on when developing segmentation methods.

Birkfellner [8] covers some advanced segmentation methods. The watershed transform for example uses the physical idea of water running to the bottom of valleys in a landscape. After taking a gradient transform on an image, edges are peaks in the landscape, and the virtual water will fill up basins representing segments in the image. Various interpretations of the physical behaviour can be used.

Carr [9] refers to various morphological methods used to remove noise. An opening operation acts like a low pass filter whilst still preserving edges. Opening and closing in sequence tends to be better at maintaining the mean intensity.

Mukundan [2] observes that in HRCT lung scans, tissue regions are "characterized by different and easily separable intensity levels". In this case simple thresholding can be used to pick out regions.

2.3.3 Contour Finding

Rather than use pixels/voxels, some reconstruction techniques use contours defining the boundaries of the region objects occupy in an image.

Mukundan [2] starts with a binary image after thresholding. Eroding the image with a 3x3 element then subtracting this from the thresholded image gives one pixel wide edges. Sequential edge following is used to extract contours. Discarding small contours reduces the number of contours significantly.

Pu et al. [10] introduce a border marching algorithm with an adaptive step size to find the outer contours of the lungs. The metric for adjusting the step size for a border segment is based on how far (at most) the segment lies from the true border. This method has the advantage of including small juxtapleural pulmonary modules in the segmentation despite their imaged intensity being dissimilar to the rest of the lung. Mackay [1] uses this method.

2.3.4 Contour Interpolation

Between two slices filled with contours, new slices can be added with contours interpolated from those in the slices above and below them. Some methods are able to do this without a contour correspondence.

Barrett et al. [11] present a contour interpolation algorithm in image space based on morphological operations. An image with both contours present (as different grayscale values) is dilated until the space between contours is filled. The front where the two dilations meet is where the interpolated contour is found. It is noted that this method handles branching cases with no modification necessary.

Chai et al. [12] use partial differential equations to interpolate between contours on a terrain map. Their method produces smooth interpolations, and can handle complex shapes such as two or three branches.

2.4 Typical Methods

Many reconstruction methods exist. Volumetric rendering and marching cubes use the original image stack data, whilst other mesh reconstruction methods may use contours processed from scan data.

2.4.1 Volumetric Rendering

Each image in the image stack is treated as a slice with thickness. Thus the pixels in the image are voxels, and various volume rendering techniques can be used to directly render the data without an intermediate structural representation such as a mesh. Since every voxel may be involved in the final render, naive implementations can be expensive. Techniques to improve the render quality and performance have been investigated.

Rasterization In rasterization, the forward direction of an object's effect on an image is considered. Each voxel may directly affect the final render, or first be projected onto a intermediate object which is then itself drawn.

Splatting takes each voxel's value and "splats" it against the drawn image, contributing to a few pixels, with its contribution fading away as you move outwards. Zwicker et al. [13] use elliptic Gaussian kernels as the basis of the shape of each splat.

Texture-based volume rendering intersects many planes with the volume [14]. On these planes polygons are rendered, with texture mappings from their coordinates to the 3D space of the volume, to pull texture values from the voxels. The planes must not be parallel to the viewing direction.

Raycasting In raycasting, we take each pixel of the render and work backwards to find which objects affect it. Each pixel emits a ray which intersects with many voxels. The weighting of voxels is dependent on the technique.

Maximum intensity projection (MIP) is a raycasting method where rays project the most intense voxel they pass through [8]. The images produced have high contrast detail and are easy to understand. Summed voxel rendering is another raycasting method where rays sum up intensities from every voxel they pass through, giving a blurred image [8].

Fishman et al. [15] make comparisons between maximum intensity projection and other volume rendering. MIP tends to not contrast the background well with the structure of interest. Other volume rendering methods can weight voxels differently and give different tissue types different colours.

Intersecting arbitrary rays with voxels can be computationally expensive. Shear-warp rendering solves this by using projections which make rays orthogonal to the voxel axes [16].

2.4.2 Surface Reconstructions

In each image of the image stack, we can see where the boundaries of tissue are. We can therefore describe the geometry of a structure as it intersects the image plane. By combining all images in the stack, a surface reconstruction of the entire object can be found, provided the relationships between slices are inferred accurately. Surfaces have the advantage of having commonplace rendering techniques, and support on 3D printers.

Marching Cubes Marching cubes [17] converts voxels into surfaces. Each voxel either belongs to a structure or does not, based on imaged intensity. Surface voxels ("inside" voxels bordering "outside" voxels, on some isosurface) are found. Each voxel has its corners defined as inside or outside based on its neighbours, and is then assigned a set of triangles based on those corners, using a lookup table. The triangles are joined

in neighbouring surface voxels to form the overall mesh. The results tend to look jagged, and smoothing is usually applied either to the mesh or during rendering for a more visually accurate output.

Newman et al. [18] have conducted a comprehensive survey of marching cubes. One problem marching cubes has is ambiguity which induces defects. The survey recommends avoiding lookup tables with reflective symmetry, where "outside" and "inside" corners of a voxel are swapped to give the same triangle set but with opposite normals. Instead only rotational symmetry (rotating corners around the voxel) is allowed. Such a lookup table was identified by Nielson et al. [19].

Point Cloud Methods One of the more general ways of defining a surface with incomplete data is by sampling many points from it, and making assumptions about the way these points would be connected. This is common in 3D depth scanning of exteriors of objects, where many points are sampled but the entire surface is not known and must be inferred. If we treat the boundaries of tissue regions in images as sets of points, we can apply point cloud methods to all the points gathered. The techniques can be tuned for specific applications. Approximating methods produce surfaces which lean more heavily on the assumptions made, with the points guiding the end results. This is useful when the sampling of the points is noisy. Methods which interpolate assume the points are perfect and so the surfaces produced must pass through them. It is common to define a function from 3D space to a value so that the surface should be found where the function output is zero, then an isosurface at zero is generated. A drawback of point cloud methods is that the explicit relations between points on the same contour are lost. In addition, when planes are more separated, artefacts tend to appear.

Models Braude et al. [20] employ Multi-level Partition of Unity (MPU) implicit models to generate isosurfaces from. MPU closely approximates Euclidean distance near points. This method requires surface normals.

Guennebaud et al. [21] fit algebraic spheres to point sets to construct surfaces. Their method (APSS) performs better than prior methods on sharp features and sparse data.

Oztireli et al. [22] combine Moving Least Squares (MLS) with local kernel regression to obtain Robust Implicit Moving Least Squares (RIMLS). This method reconstructs sharp (non-smooth) corners more accurately than APSS.

Taubin et al. [23] demonstrate colour maps extrapolated from source points onto a reconstructed surface.

Estimating Normals Some of the methods above require normals for each point. Estimating normals from the points sampled requires some understanding of the structure itself.

Mitra et al. [24] use least squares distance to fit a plane to a neighbourhood of points for each point in the cloud. There is a sweet spot for the radius of the neighbourhood used. Small radius makes noisy points have more impact on the plane found, and large radius allows for surface curvature to introduce error.

Mesh Rendering Mackay [1] states that surface rendering is not as widely implemented as volume rendering in medical contexts. However mesh rendering is abundant in literature from applications such as video games. Should mesh reconstruction become more accurate and faster, more implementations of mesh rendering for medical imaging may become available.

2.5 Testing

The end aim of these reconstructions and renders is to give humans a better visualisation of internal structures. However to compare methods objective testing is required. Requiring human experts to come to a consensus on desired outputs is resource consuming for stages such as image segmentation, and infeasible for complex structures such as meshes. Therefore metrics and repeatable methods have been devised to test algorithms.

2.5.1 Measuring Similarity in Segmentation

Pixel-wise XOR operations are common, simply working out where an ideal segmentation and an output segmentation differ.

Pu et al. [10] use a reference segmentation contour (defined by experts), and evaluate the distribution of distance error along their result contour from this reference.

2.5.2 Generating Models

It is difficult to obtain real models of internal structures. Techniques to generate synthetic models suitable for testing have been developed.

Mackay [1] uses Blender3D to create test models, by creating surface of revolutions about bezier curves. Multiple surfaces are merged for more complex structures such as branches. The main intent here was to create simple models of problem structures. Contours can be sampled from these models by plane intersections.

Mackay [1] also proposes an alternative method of generating test data, where a side view of a branching structure is drawn in black. An inter-slice distance is defined to separate rows. Each row is scanned for black pixels. When a black pixel is encountered, a contour is generated by revolving points about this pixel location. Noise can be added in. Where contours intersect, they are merged. This allows for branching structures to be generated by this method.

Pluta et al. [25] propose a rule-based method of generating lung models, including deformations and noise. Their rules are based on airflow and measured constants, and the distribution of final bronchioles in a lung. Their results are suitably accurate for testing all parts of lung reconstruction methods.

2.5.3 Measuring Similarity in 3D Models

Mackay [1] uses Hausdorff distance (essentially a maximum deviation between point sets) to measure mesh similarity. Points are sampled from a ground truth model and the nearest distance is found to the reconstructed model.

Meshes can be sliced at the same heights and compared slicewise to reduce similarity to a 2D problem.

Shum et al. [26] compare simple 3D objects by comparing the curvature of points based on spherical coordinates. This could be applied locally on more complex medical reconstructions.

2.6 Performance

Recent papers in medical imaging reconstructions tend to have an emphasis on taking advantage of modern hardware. This can be achieved through parallelization or GPU implementations.

Saxena et al. [27] describe parallel techniques for various image processing tasks such as segmentation and noise reduction. The main idea is to tile an image and let different processors work on different tiles, then deal with the borders when recombining the tiles. They found their parallel implementation to be 2.5 times faster than a sequential implementation.

Alsmirat et al. [28] investigate pure GPU and hybrid CPU-GPU implementations of segmentation of medical images. They found that both outperform a CPU only implementation, and the hybrid method performs the best.

2.7 Correspondence Methods

Contour-based mesh reconstruction first gathers contours from image data, then corresponds contours between slices. Matched contours then go through point correspondence and triangulation to produce a final mesh. This approach to mesh reconstructions is the focus of this paper.

2.7.1 Contour Correspondence

Rather than lose the relationship between points, the original contours can be used in creating a surface reconstruction. Working out which contours on which slices represent a connected tissue in the original structure is called contour correspondence, and the resulting correspondences are used in later steps of reconstruction.

Herbert et al. [29] classify correspondence algorithms into four types.

- Manual methods use user input to connect contours. This is time consuming for large datasets.
- Local algorithms takes pairs of slices at a time and considers contour matchings between these.
- Global algorithms look for contour pairings across all sections.
- Growing algorithms create a hierarchy of components, attempting to join unmatched contours onto existing components if suitable.

Contour centroid position comparison is common in local algorithms.

Herbert et al. [29] suggest growing objects one contour at a time instead of considering pairings of contours globally. In preprocessing spacial information is gathered such as contour characteristics (position, shape, size), intra-sectional relationships between contours (to validate complex structures later), and inter-sectional relationships. Contour relationship metrics include distance between centroids, distance between major/minor axes, minimum bounding rectangle overlap, shape comparison via compactness ratios, and surroundness (how deep is the contour nested in larger contours).

Herbert et al. [29] include semantic information given by the user on the expected components in a reconstruction, and their spatial relation to each other. A starting contour is found for each component before the growing process starts.

Mukundan [2] begins point correspondence without an explicit contour correspondence already found. If two point correspondences are found as neighbours belonging to the same pair of contours, the search for further point correspondences is narrowed to these two contours. This effectively gives a temporary contour correspondence.

2.7.2 Point Correspondence

Point correspondence is an optional step in surface reconstructions, where points on matched contours are matched to each other as a precursor to triangulation.

Mukundan [2] uses distance in the XZ plane as a metric for matching points. Other optional constraints include matching the next point close to the previous point.

Mackay [1] proposes Dynamic Time Warping (DTW) as a method of point correspondence. DTW is intended to match features on the same structure across different times. In point correspondence, it matches points on contours which are from the same structure but in different slices, so slightly warped. At any time during DTW a pair of points from both contours is matched, then the next points in both contours are considered as matches for the current points. DTW has some basic constraints when finding a warp path, but Mackay suggests adding further constraints as a potential improvement. It tends to be that as planes are more separated, the contours undergoing point correspondence are more dissimilar, and DTW performs worse in reconstructing these.

2.7.3 Mesh Triangulation

A surface can be represented as a set of primitive elements, with triangles being the simplest. They are defined by three points in 3D space. The final step of mesh reconstruction is taking the relationships inferred by contour and point correspondence, and generating triangles to connect points whilst taking these relationships into account.

Mackay [1] begins with two ordered sets of points X and Y, from the two contours matched, with some edges provided by point correspondence. As a result of the constraints on DTW, there are three cases for each point x_m on the first contour:

- x_m has an edge with y_n and x_{m+1} has an edge with y_{n+1}. These points are direct neighbours on their respective contours, and form a quad which is trivial to triangulate.
- x_m has edges with a sequence of points $\{y_n, y_{n+1}, ..., y_{n+i}\}$. x_m has a one-to-many point correspondence with these points. This can be triangulated with a triangle fan centering about x_m.
- Each point in the sequence $\{x_m, x_{m+1}, ..., x_{m+i}\}$ has edges with a point y_n. This is the opposite of the previous case and can likewise be triangulated with a triangle fan centering about y_n.

Li et al. [30] observe that when m and n differ greatly, one-to-many point correspondences are common, and the triangulation becomes rough.

2.7.4 Branching Problem

A simple structure is easy to infer from its images. On each slice the structure is present in, we see one contour, and these contours are similarly positioned and shaped in adjacent slices. More complex structures such as a branch show up differently, and it becomes harder to infer the original structure. Contour correspondence must be adjusted to deduce these unusual structures when they appear. Even when the general layout of the original structure is inferred through contour correspondence, point correspondence and mesh triangulation can be difficult when the shapes of contours change suddenly.

Mackay [1] looks for lung branches as one contour approximately splitting in half into two contours, for contour correspondence.

Mackay [1] uses contour merging to help DTW point correspondence in the branching case. In a contour correspondence where there is a slice with a single contour and a slice with two contours, the two contours on the same slice are merged at their nearest point to give one larger contour. Point correspondence between the single contour and the merged contour can then proceed as normal. In some branching cases however, the triangulated mesh is twisted, as though DTW has not matched the correct points together. This problem is left for further research.

2.8 Background Summary

Scans have multiple planes separated by some distance. For the same spatial resolution, smaller objects will have fewer planes scanning through them. Alternatively, for a given sized object, lower plane counts are better for lowering scan time. Reconstruction methods which can handle lower plane counts should give more accurate reconstructions of smaller objects, or allow for shorter scan times.

General methods related to volumetric rendering and mesh reconstructions have been applied to scans. These all have some drawback. For volumetric rendering it is the render time, for marching cubes it is the jagged meshes, and for point cloud methods it is the approximation of structure.

Counter-based correspondence methods aim to solve these issues. Contour correspondence and point correspondence are achieved through variations of Euclidean distance and other metrics. Mackay was able to achieve similar or better accuracy to prior methods with substantially lower triangle count. Mackay found

that branches and other structures cause problems in both contour and point correspondence. His method using DTW, like others, becomes less accurate as plane count decreases.

A new point correspondence method can improve upon prior approaches by reconstructing branches and other structures more accurately, and handling lower plane counts better.

3 Method

In this section I detail how a new point correspondence method has been constructed, with the intention of handling branching structures and lower plane counts better.

3.1 Proposal

The proposed system consists of:

- Contour Splitting, a new approach to enabling point correspondence on branches and other structures
- Point Angle, an alternative algorithm for point correspondence.

3.1.1 Contour Splitting

For brevity, contour correspondences of 1-to-2 will be considered. Point correspondence algorithms act on 1-to-1 contour matchings, so 1-to-2 cases must be reduced to these.

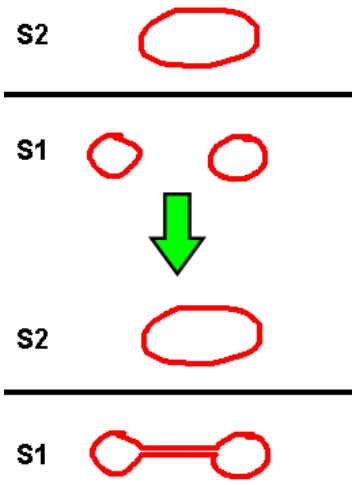


Figure 1: Contour merging for DTW

Mackay's approach was contour merging, where the 2 contour side of the correspondence is merged. The closest pair of points across the contours is found, to join them into a single contour (See Figure 1). This gives a single 1-to-1 case for point correspondence to act on. A disadvantage of this method is that the merged contour has an unusual shape, which can cause point correspondence algorithms to behave poorly.

The proposed technique instead splits the 1 contour side of the correspondence. The best fit line to divide the 2 contour side is found, giving the angle of the line to split the 1 contour (See Figure 2). Each half of the split 1 contour is paired with its corresponding contour on the 2 contour side. This gives two 1-to-1 cases for point correspondence to act on. The contours produced are well shaped and suitable for point correspondence algorithms designed for simpler cases.

Adjustments can be made to the position of the split line to improve accuracy.

- The ratio of contour areas on the 2 contour side can be reflected in the split contour by adjusting which points the split line connects to (See Figure 3). This preserves the internal cross section of each branch half as they join.
- To achieve a smooth point correspondence along the inside of the branch (where the branches join each other), the split line must have points added along it (See Figure 4). This is in proportion to the number of points on the original contour.

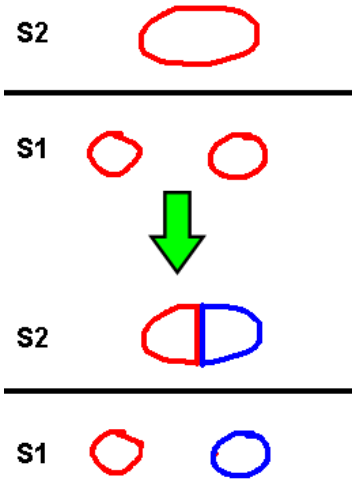


Figure 2: Contour splitting

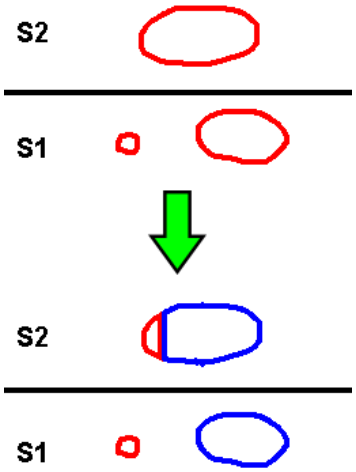


Figure 3: Ratio of areas under splitting

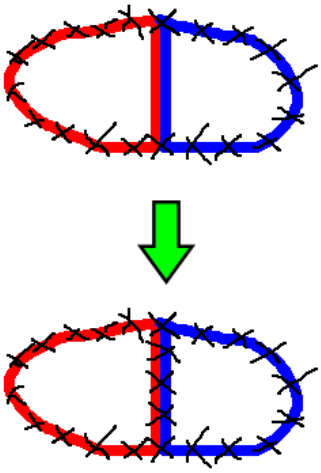


Figure 4: Additional points added to split line

- The split line may also be adjusted in height, to reflect the likelihood the branch split is somewhere between the two planes of contours. With no further calculation, the height is assumed to be halfway. A semi-circular curve (changing in height above the split line) creates a split line joint which mimics the intersection of two cylinders, which is approximately what is expected from two branches coming together.

3.1.2 Point Angle

Prior methods of point correspondence consider the Euclidean distance between points on corresponded contours. The proposed algorithm instead considers similar angular distance relative to the contour's centroid (See Figure 5). For contours where every border point can be seen from the centroid (similar to star-shaped polygons), the angular distance metric is monotonically increasing as we progress along the contour's specified order (if we start from the smallest angle point). In point correspondence, the contours' point angle metrics are leapfrogged between to join points (See Figure 6). This leapfrogging assumes the monotonically increasing property. For contours which "double back" (are not star-shaped), the monotonically increasing property can be enforced when filling in the point angle metric, by recording the previous angle if the current angle is smaller.

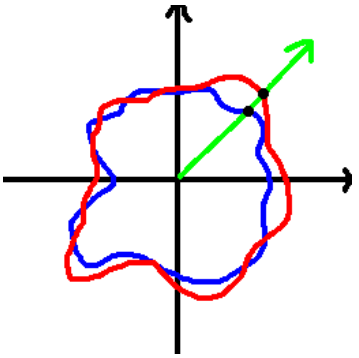


Figure 5: Correspondence by point angle

This can lead to sections of points with the same point angle metric, and leapfrogging which produces unideal many-to-one mappings. To counter this, a second metric is added, which is simply progression along the number of points in the contour (normalized), starting from the same point as the angle metric. The angle and progression metrics are weighted and summed before the leapfrogging correspondence. This progression metric is actually well suited to contour data with similarly distanced points, which is what we get from uncompressed image contours. In these cases it may be suitable to only use progression.

3.2 Implementation

Mackay's report provides a complete implementation of contour and point correspondence[2]. Contour correspondence is implemented by checking the distance between centroids relative to the radius of the contours, which indicates if there is any overlap. The complete implementation was modified to add the options of contour splitting and point angle for point correspondence. The modified source code and tools are available here [3].

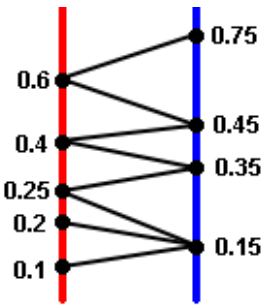


Figure 6: Leapfrogging of point angle metric

This paper contributes a new tool for testing. A python script has been developed to automate reconstruction and analysis of multiple models and plane counts. The script calls the reconstruction binary and feeds it every combination of model and plane count to obtain reconstructions. PyMeshLab [31] is then used to call the Hausdorff distance filter with reconstructions and their original counterparts, giving measurements which are then stored in a JSON file. The script includes methods to filter this file, to quickly look up measurements across different reconstruction methods for a given model and plane count.

[2] https://drive.google.com/open?id=1EHIx3dOOwcP1KTxWxhv8NWyROHAyVZ-P
[3] https://github.com/cstevenson3/cosc470code

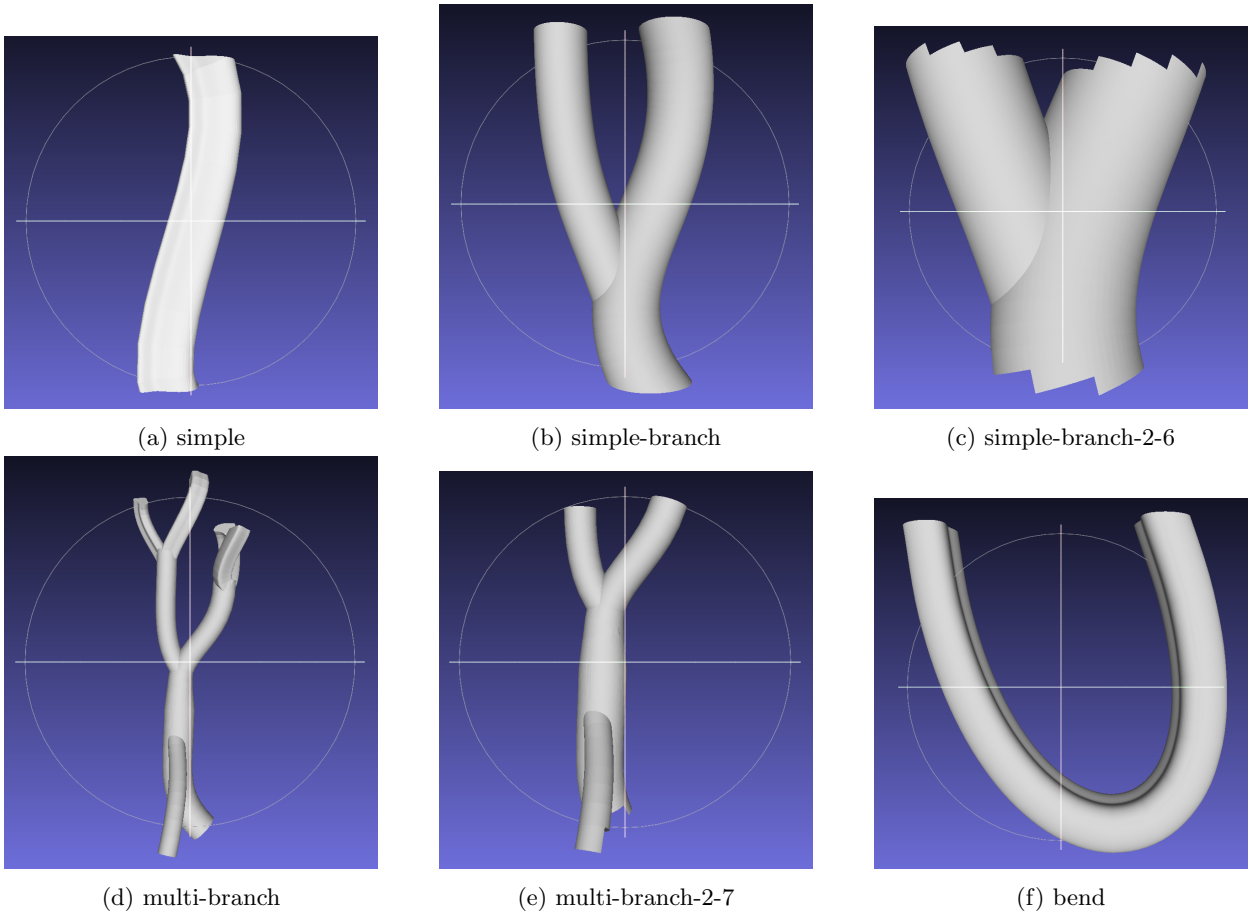
4 Analysis

To analyse the effectiveness of the proposed method in improving reconstruction accuracy, synthetic models are intersected with planes to give contours of points, which are then given as input to reconstruction. These reconstructions from different methods are compared to the original and against each other. The observations are then confirmed with objective measurements such as Hausdorff distance. Finally a reconstruction from real lung scans will be explored.

4.1 Ground Truth

For ground truth I used the same synthetic models Mackay generated for analysis (See Figure 7). These can be downloaded standalone[4], but are also included alongside my implementation[5]. Planes are intersected with the models to give the contours of points expected from scanning and segmenting a real object of the same shape. Plane counts of 10, 20, 30, 40 and 50 are used to emulate different scan slice thicknesses.

Additionally, Meshlab was used to copy specific sections of these models into new files, to focus reconstructions on the difficult cases the proposed method is intended to improve on. These are named after the original model and which slice range (based on the 10 slice levels version) is being copied.



(a) simple (b) simple-branch (c) simple-branch-2-6

(d) multi-branch (e) multi-branch-2-7 (f) bend

Figure 7: Original Models

[4] https://drive.google.com/open?id=1QYsQa-qsz9mwDRz_nk4ohtD4pITV9AD3
[5] https://github.com/cstevenson3/cosc470code

4.2 Measurements

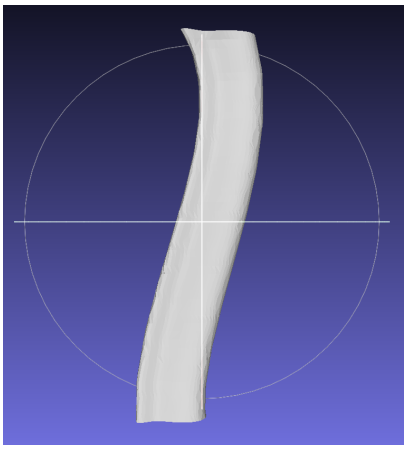
For reconstruction accuracy, Mackay used the Hausdorff distance filter built into Meshlab, which takes as input a sampled mesh (the mesh to sample points from), and a target mesh (to find the closest distance from each sample point). Many points are sampled and mean distance is reported. Lower mean distance is better. I may refer to the forward direction as when the original synthetic mesh is the sampled mesh, and the reconstructed mesh is the targest mesh, with reverse being the opposite.

The filter also has parameters related to sampling, such as whether vertices, edges, or faces are sampled. The choices of sample/target meshes and these parameters are important, as the nature of how the models and reconstructions are generated affect different methods differently. For example, all vertices in Mackay's reconstructions are the original points given by plane sampling, and so lie on a face in the original mesh. Hence the reverse Hausdorff distance filter with only vertices sampled gives a measurement of zero. Whereas the proposed method adds additional points, and so is unlikely to achieve this zero measurement. For this reason only faces will be sampled. When it comes to direction, extraneous faces on a target mesh may reduce Hausdorff distance, but extraneous faces on a sampled mesh increase Hausdorff distance (usually the preferred interpretation). For this reason, in same examined cases a direction may be omitted, for reasons which will be explained.

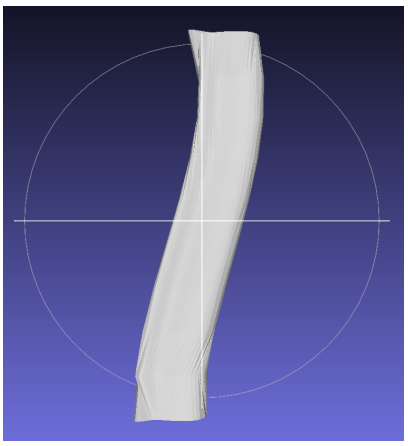
For contour merging and DTW, Mackay's implementation is used. For the proposed method, the same implementation is used except where contour splitting and point angle replace the point correspondence stage. To start with, 50% angle weight (and 50% progression) is used in the point angle metric.

4.2.1 Simple Model Reconstruction

A simple tube is reconstructed.

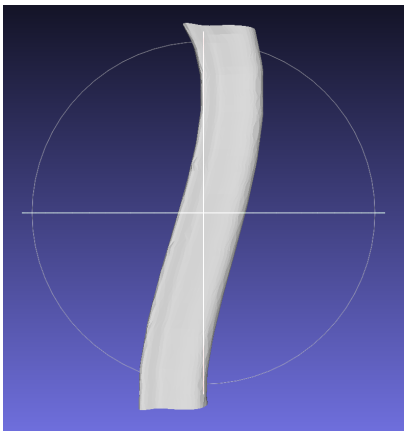


(b) DTW 50 planes

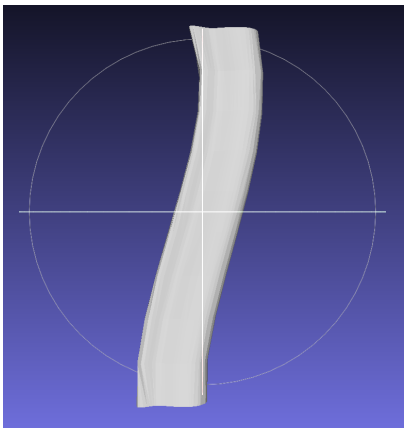


(c) DTW 10 planes

(a) Original



(d) CSPA 50 planes



(e) CSPA 10 planes

Figure 8: Reconstructions of simple model

Both reconstruction methods handle this simple case well, with both 50 and 10 plane samples. In the 10 plane reconstruction, they both have visible jagged parts at the bottom, although CSPA seems to be better in this area. We now look at Hausdorff distance to confirm this.

Method	No. of Slices Sampled				
	10	20	30	40	50
Contour Merging + DTW (MDTW)	0.0368	0.0274	0.0174	0.0195	0.0145
Contour Splitting + Point Angle (CSPA)	0.0297	0.0240	0.0165	0.0190	0.0140

Table 1: Simple model, mean Hausdorff distance from original to reconstruction

In Table 1, the anamoly at 40 samples (being higher than expected) is caused by the 40 plane samples not spanning the full original model. This affects both methods equivalently so has been left in for relative comparison.

Method	No. of Slices Sampled				
	10	20	30	40	50
Contour Merging + DTW (MDTW)	0.0186	0.00836	0.00479	0.00336	0.00275
Contour Splitting + Point Angle (CSPA)	0.0102	0.00401	0.00281	0.00236	0.00215

Table 2: Simple model, mean Hausdorff distance from reconstruction to original

As we can see from Table 2 and Figure 9, our data agrees with the trend in Mackay's measurements, in

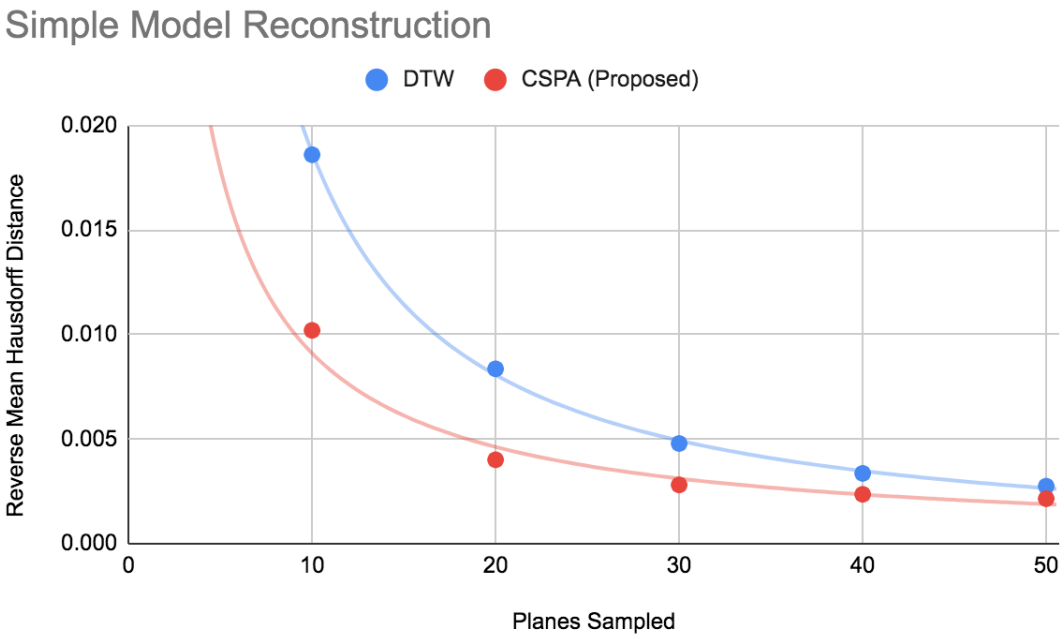
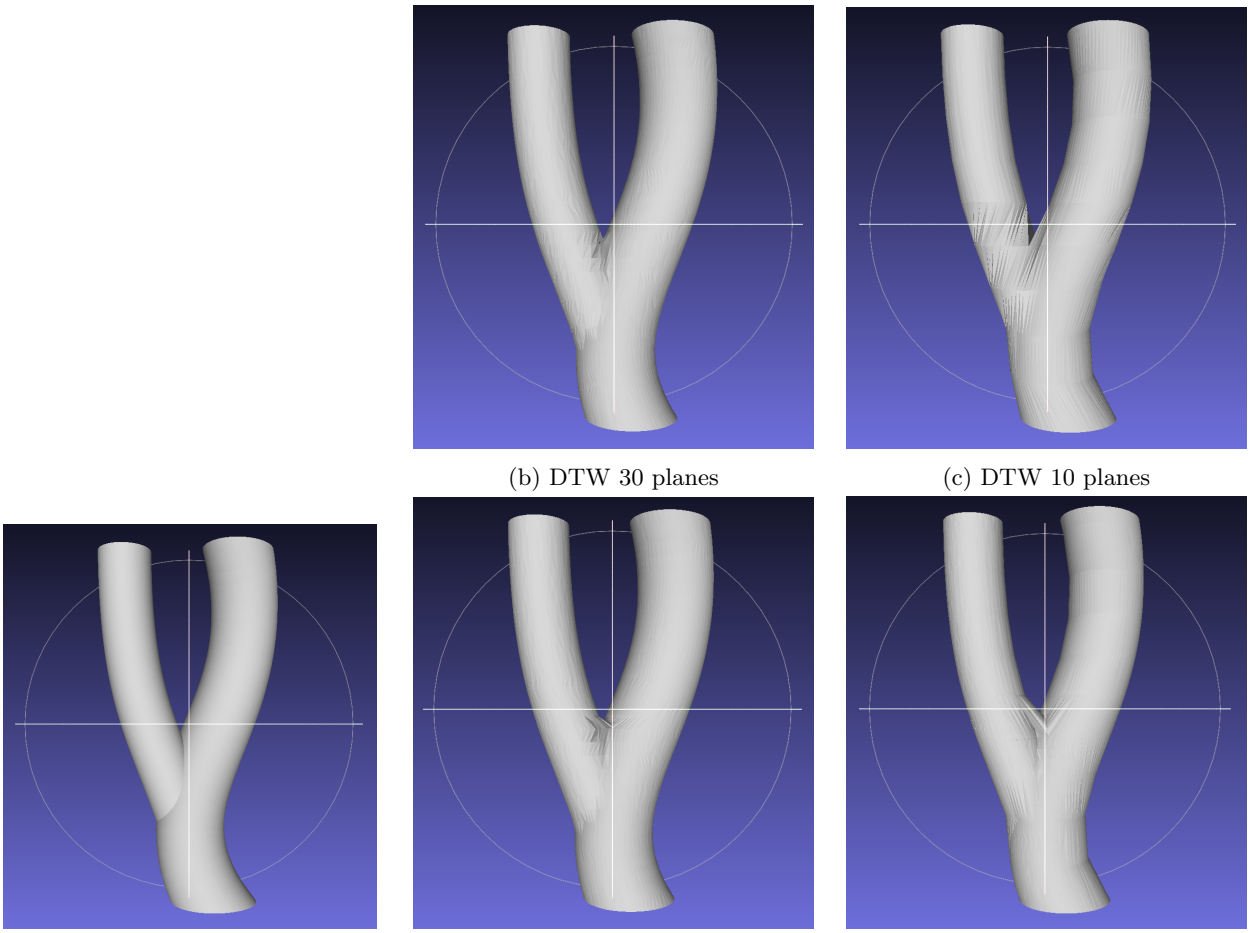


Figure 9: Simple model, mean Hausdorff distance from reconstruction to original

that more slices sampled results in more accurate reconstructions. At high sample rates, the two methods have similar accuracies. However, as the sample rate lowers, CSPA has better measured accuracy than MDTW. This is a trend which will continue for the remaining models.

4.2.2 Simple Branch Reconstruction

A simple model containing a branch is reconstructed. Here there are cases where contour correspondence is invalid (for 20 and 50 plane samples), and 40 plane samples has an issue which will be explained later, so only 10 and 30 plane samples are shown first.



(a) Original (b) DTW 30 planes (c) DTW 10 planes

(d) CSPA 30 planes (e) CSPA 10 planes

Figure 10: Reconstructions of simple branch model

At 30 plane samples we see decent reconstructions, with CSPA having a slightly less clean split. At 10 plane samples, DTW reconstructs one side of the branch well, but the other side is twisted, and the split is missing triangles. CSPA on the other hand has a clean reconstruction in terms of being closed and smooth on the branches, but there is an obvious protrusion where the split line has been slightly misplaced.

These 10-plane figures highlight the behaviours of the two approaches. With merging + DTW, the constraints on DTW force the point correspondence to move along on both sides. When the correspondence arrives at one of the merging points on the two-contour, it can only connect to a few points on the single contour before being forced to jump across the merge. This results in the split on the single contour being placed not exactly halfway between the branches. In Figure 10, this is visible as the front split being more left than it should. With CSPA, the split line is placed based on an estimation of the ratio of areas of the two contours, and the ratio of areas in the split contour. This gets the split line close to where it should be, but not exactly. In Figure 10, we see it has estimated the split line to be further right than it should be. Perhaps a metric accounting for contour shape to find where the single contour starts to pinch would be more accurate, but also more sensitive to irregular contour shapes.

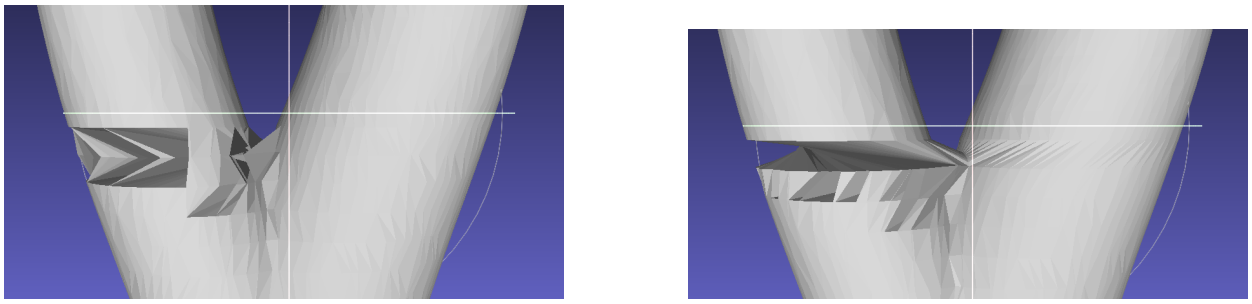
Method	No. of Slices Sampled				
	10	20	30	40	50
Contour Merging + DTW (MDTW)	0.0727	-	0.0641	0.0605	-
Contour Splitting + Point Angle (CSPA)	0.0685	-	0.0640	0.0607	-

Table 3: Simple branch model, mean Hausdorff distance from original to reconstruction

Method	No. of Slices Sampled				
	10	20	30	40	50
Contour Merging + DTW (MDTW)	0.0195	-	0.0107	0.0538	-
Contour Splitting + Point Angle (CSPA)	0.0147	-	0.0115	0.0768	-

Table 4: Simple branch model, mean Hausdorff distance from reconstruction to original

For higher plane counts the two methods have almost equal accuracy, with CSPA actually being worse at 40 plane samples. At a plane count of 10, CSPA is more accurate. The 40 plane sample version has poor accuracy for both reconstructions. Inspecting the models (See Figure 11), the issue appears to be with triangulation of the point correspondences, rather than a bad contour correspondence. This may be worth investigating, to see if this is an input case needing improvement or an implementation error.



(a) DTW

(b) CSPA

Figure 11: Simple branch reconstruction error

I follow this up with a model containing only the branch section of the simple branch model. I compared the complete simple model reconstruction to this focussed original model, to specifically see Hausdorff distance at the branch section. Only the forward Hausdorff distance is valid here.

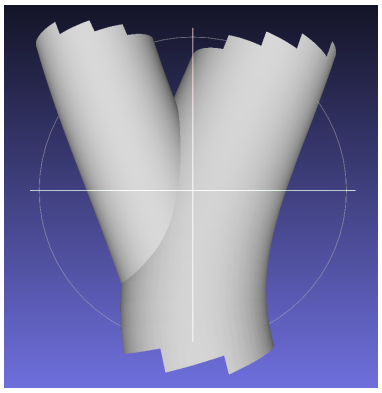


Figure 12: Simple branch model focussed on the branch

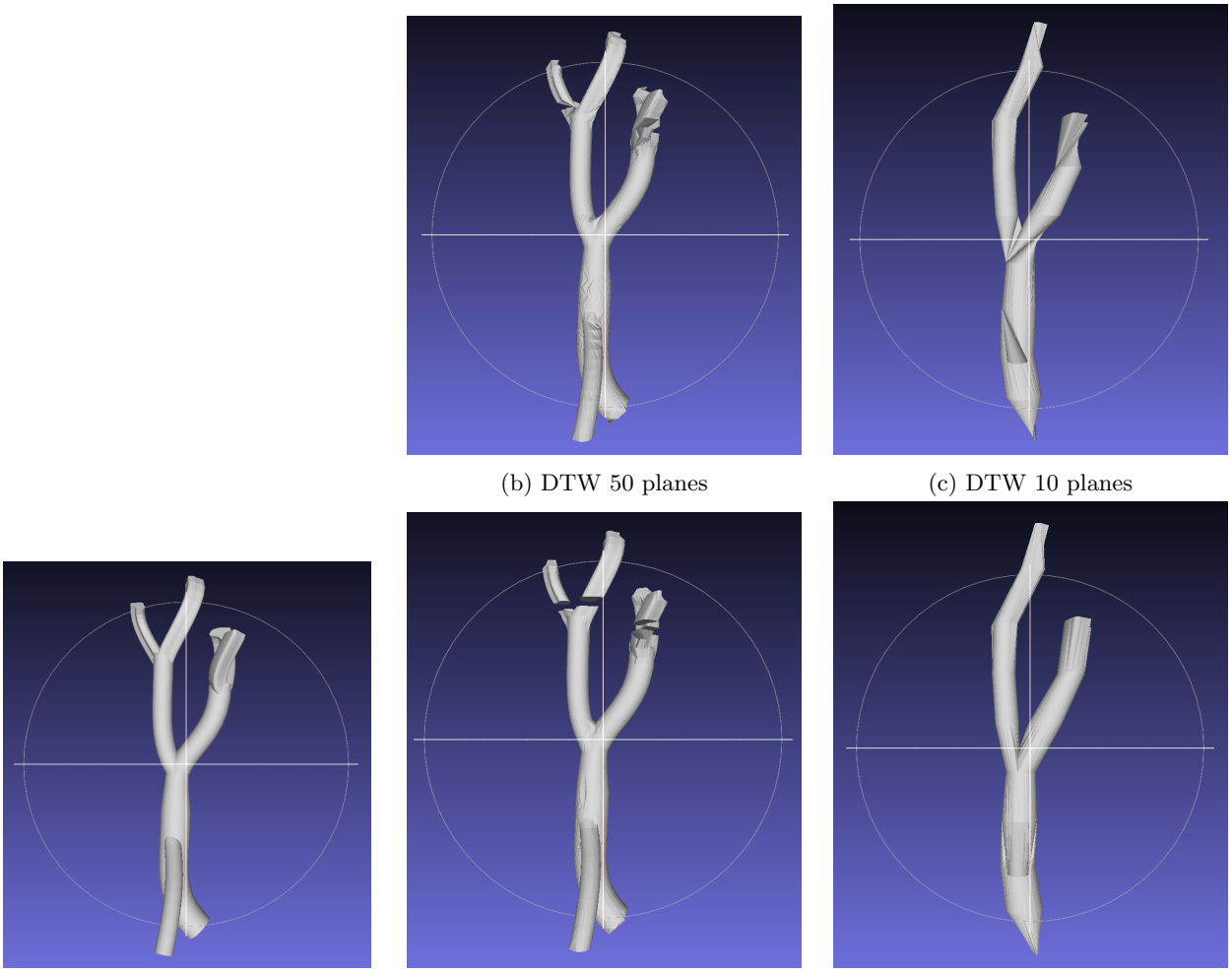
Method	No. of Slices Sampled				
	10	20	30	40	50
Contour Merging + DTW (MDTW)	0.0315	-	0.00453	0.00440	-
Contour Splitting + Point Angle (CSPA)	0.0207	-	0.00469	0.00482	-

Table 5: Simple branch focussed model, mean Hausdorff distance from original to reconstruction

This effectively shows the same results as the complete version of the model.

4.2.3 Multi Branch Reconstruction

A model with multiple branches at different orientations is reconstructed.



(a) Original (b) DTW 50 planes (c) DTW 10 planes

(d) CSPA 50 planes (e) CSPA 10 planes

Figure 13: Reconstructions of multi branch model

In the branches with correct contour correspondence (the middle and lower branches), we can compare the point correspondence. In the 50-plane reconstructions, we see similar quality in the middle branch from both methods. However in the lower branch, CSPA produces a smoother point correspondence. In the 10-plane reconstructions, DTW has bad artefacts in both branches, such as twisting, and too many points being matched to a single point. CSPA solves these problems, giving better branch reconstructions.

The 50-plane reconstructions show invalid contour correspondence in the top branches. In the 10-plane versions, entire branch halves are missing from the contour correspondence. These issues affect most of the multi-branch reconstructions. For the latter reason we will consider only the reverse Hausdorff distance, from reconstruction to original. Note that this favours CSPA more as its implementation omits some sections with invalid contour correspondence, and so that section is not sampled from. For this reason we will revisit a version of this model focussing only on the lower two branches.

Method	No. of Slices Sampled				
	10	20	30	40	50
Contour Merging + DTW (MDTW)	0.0964	0.0469	0.0320	0.0400	0.0236
Contour Splitting + Point Angle (CSPA)	0.0773	0.0371	0.0157	0.0350	0.0135

Table 6: Multi branch model, mean Hausdorff distance from reconstruction to original

The reverse mean Hausdorff distance indicates CSPA gives more accurate reconstructions of the multi

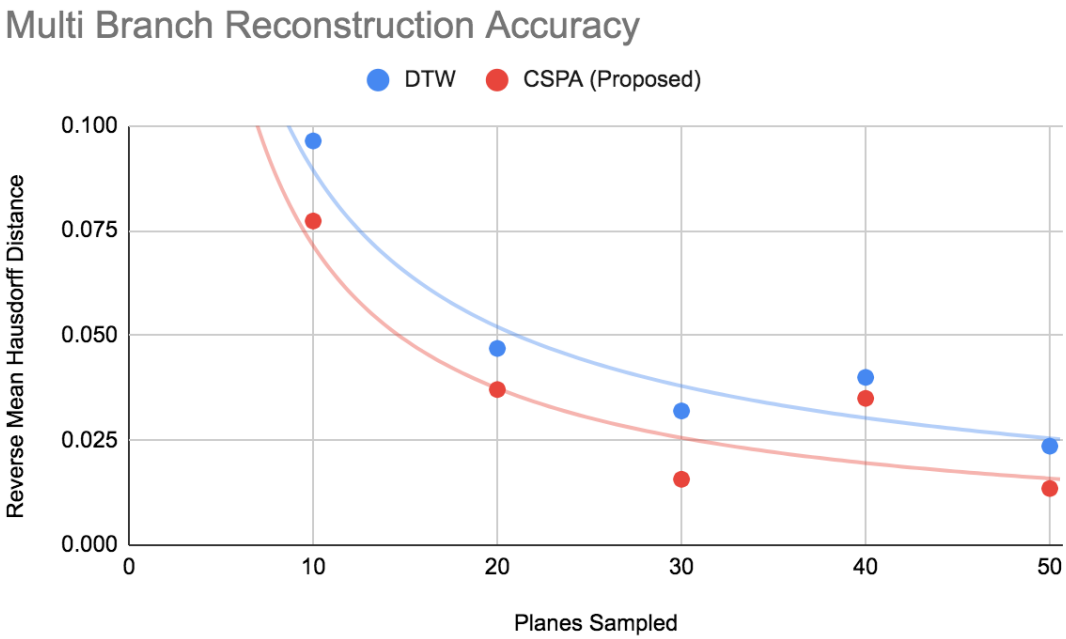


Figure 14: Multi branch reconstruction, mean Hausdorff distance from reconstruction to original

branch model than DTW at all plane sample counts. To confirm this is not down to the omissions caused by the CSPA implementation, we use the focussed multi-branch model. As with the simple branch focussed, only forward Hausdorff distance is valid here.

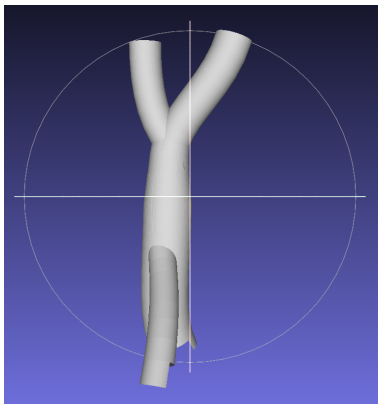


Figure 15: Multi branch model focussed on lower two branches

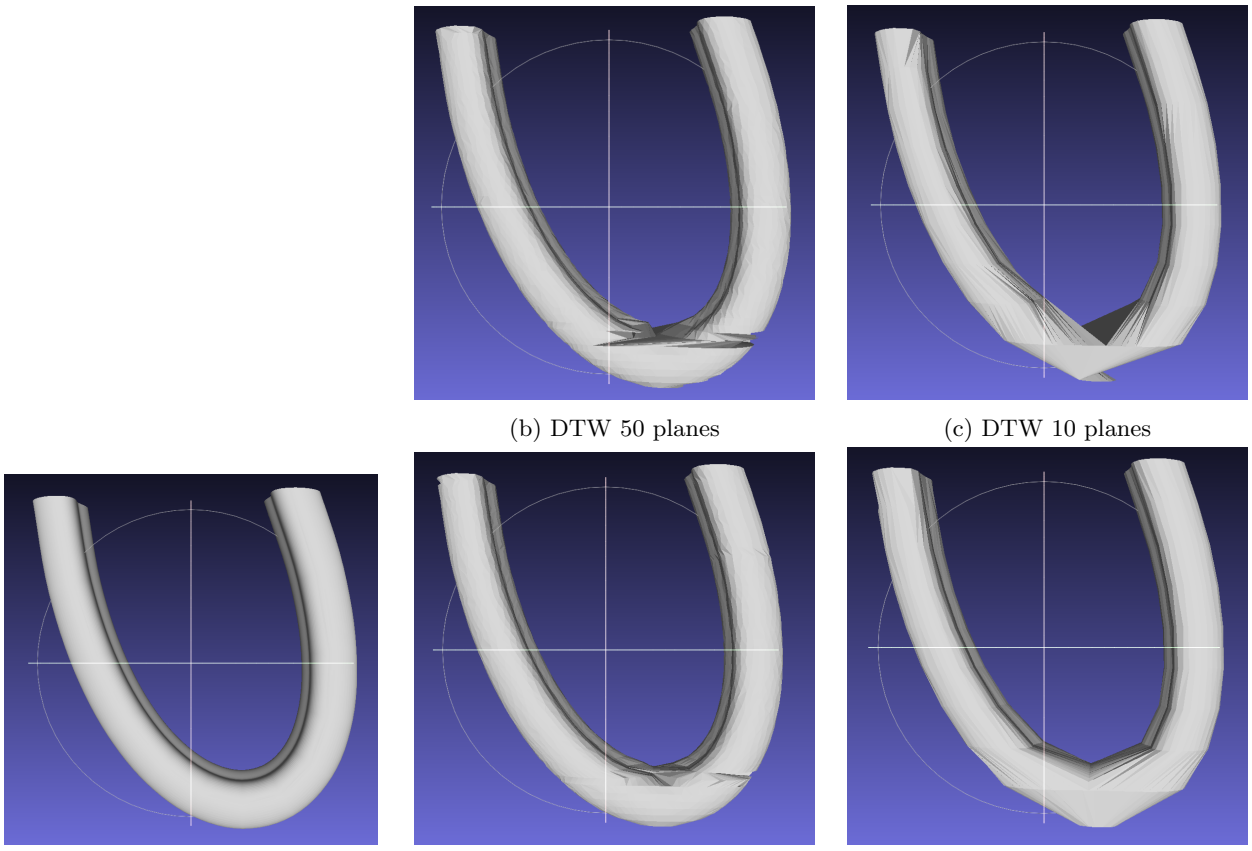
Method	No. of Slices Sampled				
	10	20	30	40	50
Contour Merging + DTW (MDTW)	0.149	0.0238	0.0134	0.0109	0.00674
Contour Splitting + Point Angle (CSPA)	0.143	0.0175	0.00934	0.0108	0.00497

Table 7: Multi branch focussed model, mean Hausdorff distance from original to reconstruction

These Hausdorff distances show that CSPA either equates with or improves in accuracy compared to DTW in the lower branching sections.

4.2.4 Bend Reconstruction

In this section a bended tube is reconstructed. This tube is oriented so that when scanned, it appears as two contours coming down to one, similar to a branch.



(a) Original (b) DTW 50 planes (c) DTW 10 planes (d) CSPA 50 planes (e) CSPA 10 planes

Figure 16: Reconstructions of bend model

For any plane count on the bend model, CSPA appears to have a cleaner reconstruction than DTW.

Method	No. of Slices Sampled				
	10	20	30	40	50
Contour Merging + DTW (MDTW)	0.0764	0.0363	0.0321	0.0314	0.0266
Contour Splitting + Point Angle (CSPA)	0.0647	0.0325	0.0281	0.0296	0.0256

Table 8: Bend model, mean Hausdorff distance from original to reconstruction

Method	No. of Slices Sampled				
	10	20	30	40	50
Contour Merging + DTW (MDTW)	0.0618	0.0313	0.0158	0.101	0.0752
Contour Splitting + Point Angle (CSPA)	0.0535	0.0173	0.0127	0.0650	0.0491

Table 9: Bend model, mean Hausdorff distance from reconstruction to original

For every plane count, CSPA reconstructions have a lower mean Hausdorrf distance than DTW, both in forward and reverse directions. Interestingly, the reverse Hausdorff distance is best for both at 30 plane samples, with worse accuracy as plane samples increase. I suspect this is because as the plane count increases, sampled contours get closer and closer to the part of the bend which is tangent to the planes. These contours are shaped more strangely, and both methods have worse accuracy reconstructing these. In the case of bends, lower plane count may be an advantage, provided the bend is not missed by any planes.

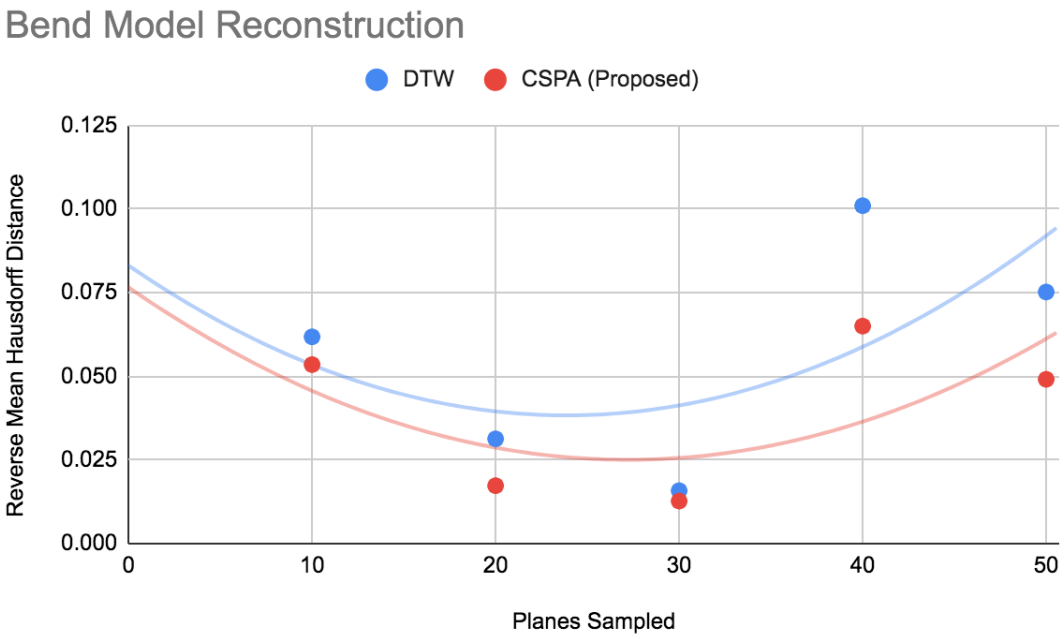


Figure 17: Bend model, mean Hausdorff distance from reconstruction to original

4.3 Angle Weight Comparison

The point angle metric is a weighted sum of angle and progression for each point. The weighting given to both may influence how accurate CSPA reconstructions are. We will test this with the simple model and the multi-branch focussed model.

CSPA Angle Weight	No. of Slices Sampled				
	10	20	30	40	50
0%	0.0107	0.00457	0.00329	0.00296	0.00277
25%	0.0102	0.00431	0.00304	0.00253	0.00234
50%	0.0102	0.00401	0.00281	0.00236	0.00215
75%	0.0103	0.00387	0.00268	0.00216	0.00206
100%	0.00998	0.00380	0.00264	0.00216	0.00204

Table 10: Angle weight comparison on simple model, mean Hausdorff distance from reconstruction to original

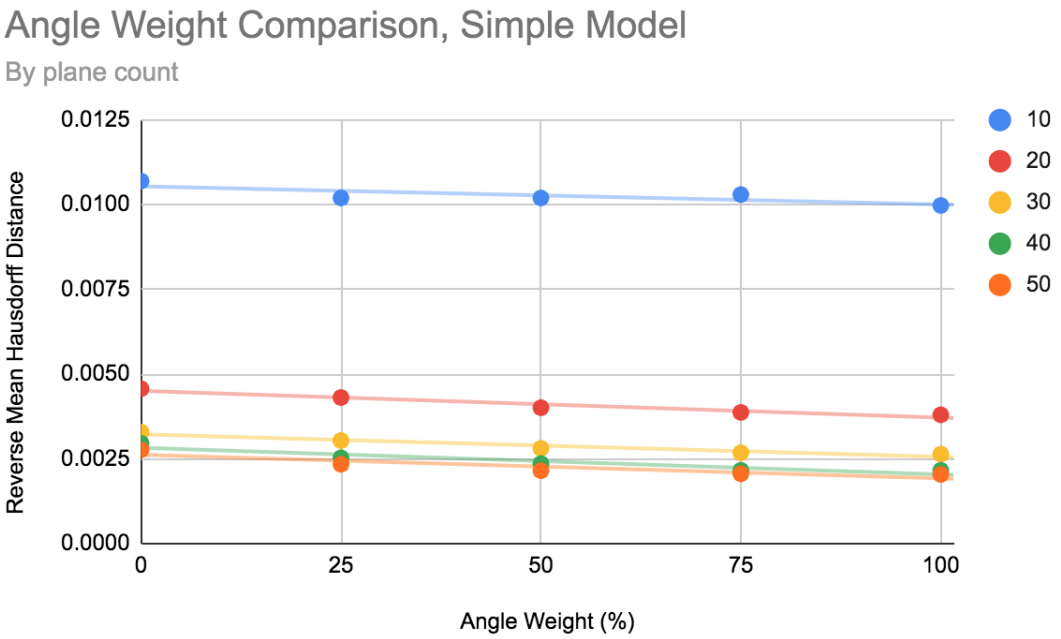


Figure 18: Comparison of accuracy for various angle weights

For the simple model (See Table 10) we can see that for every plane count, higher angle weight produces more accurate reconstructions. There seems to be diminishing returns on increasing angle weight, so using an angle weight of about 75% could be a good way to keep some progression weighting, in order to handle more oddly shaped contours. At low plane counts such as 10, the effect of changing angle weight is lessened. These trends are just for a simple model synthetically produced, and so may not apply to real data. Robust behaviour across varying structures is probably more important than optimising accuracy for simpler cases, and the difference seen here is minimal. We now investigate the multi-branch focussed model.

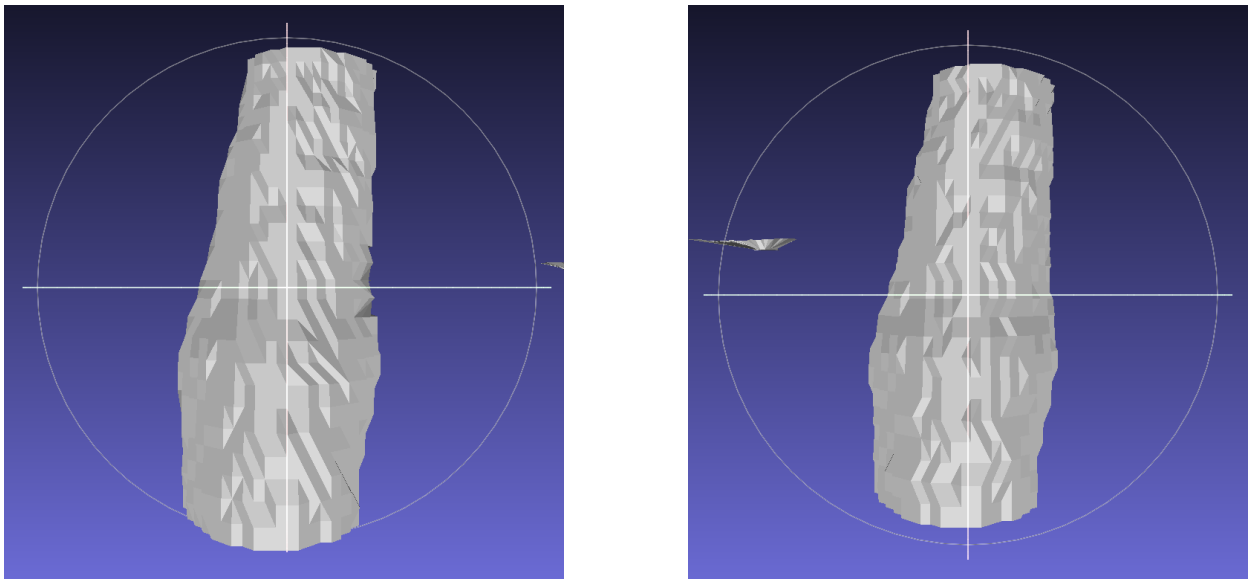
	No. of Slices Sampled				
CSPA Angle Weight	10	20	30	40	50
0%	0.146	0.0203	0.0124	0.0144	0.00689
25%	0.145	0.0186	0.0107	0.0127	0.00574
50%	0.143	0.0175	0.00934	0.0109	0.00497
75%	0.143	0.0169	0.00849	0.0100	0.00448
100%	0.142	0.0166	0.00803	0.00973	0.00437

Table 11: Angle weight comparison on multi branch focussed model, mean Hausdorff distance from original to reconstruction

For the multi-branch model, we see similar trends to the simple model. It appears the angle metric can handle the synthetic branching cases well enough to be better than the progression metric.

4.4 Real Lung Reconstruction

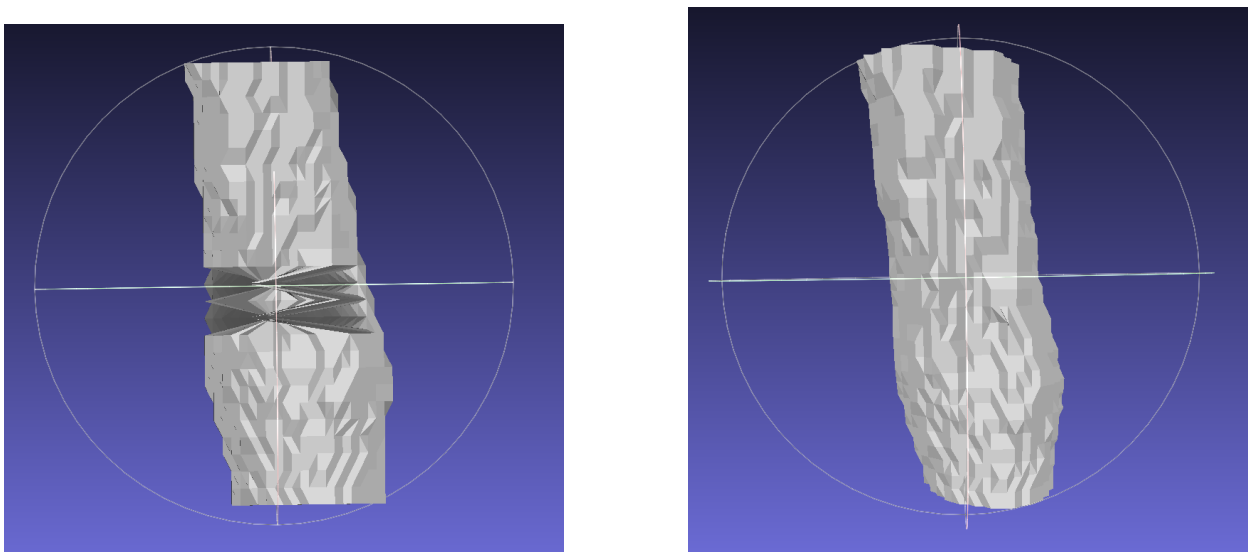
Due to technical difficulties with the implementations, only some parts of the lung were reconstructed, which unfortunately did not include any branching structures. However we can verify that CSPA gets the simpler parts correct, by taking a look at the trachea.



(a) DTW (b) CSPA (Proposed)

Figure 19: Real trachea reconstruction, front side

In the scan data, the image resolution is similar to the plane thickness, and so we can see the individual triangles. Both methods have sufficient accuracy on the front side (Figure 19) of the trachea.



(a) DTW (b) CSPA (Proposed)

Figure 20: Real trachea reconstruction, rear side

On the rear side (Figure 20) we see DTW fail at a section with less circular contours (there is an indent which CSPA accurately reconstructs). This is a good indication that point angle may be superior to DTW in some instances. Further testing on more real data would be needed to establish whether this is consistent.

4.5 Performance

We take a brief look at performance, to validate that even a naive implementation of contour splitting and point angle correspondence is not unreasonably inefficient.

PC specs:

- OS: macOS 10.13.4
- CPU: 2.4 GHz dual core I5-4258U
- RAM: 8GB DDR3
- Disk: SSD
- C++ compiled with debug flag

Method	No. of Slices Sampled				
	10	20	30	40	50
Contour Merging + DTW (MDTW)	35	54	92	106	147
Contour Splitting + Point Angle (CSPA)	36	45	70	79	98

Table 12: Time for one reconstruction of the simple model (ms)

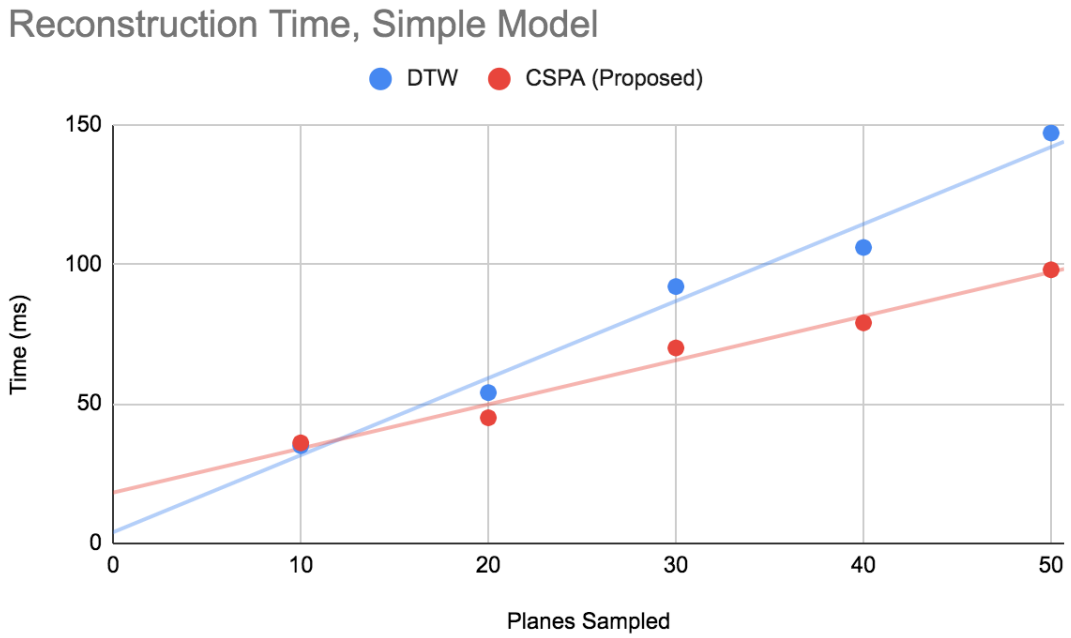


Figure 21: Performance of reconstruction methods on simple model

CSPA performs at least as well as DTW, and even performs better as plane count goes up.

Next we turn to reconstruction performance on the multi branch model. 10 plane samples gives the highest proportion of contour correspondences which are branch cases.

Method	No. of Slices Sampled				
	10	20	30	40	50
Contour Merging + DTW (MDTW)	44	65	94	115	140
Contour Splitting + Point Angle (CSPA)	29	44	58	90	104

Table 13: Time for one reconstruction of the multi branch model (ms)

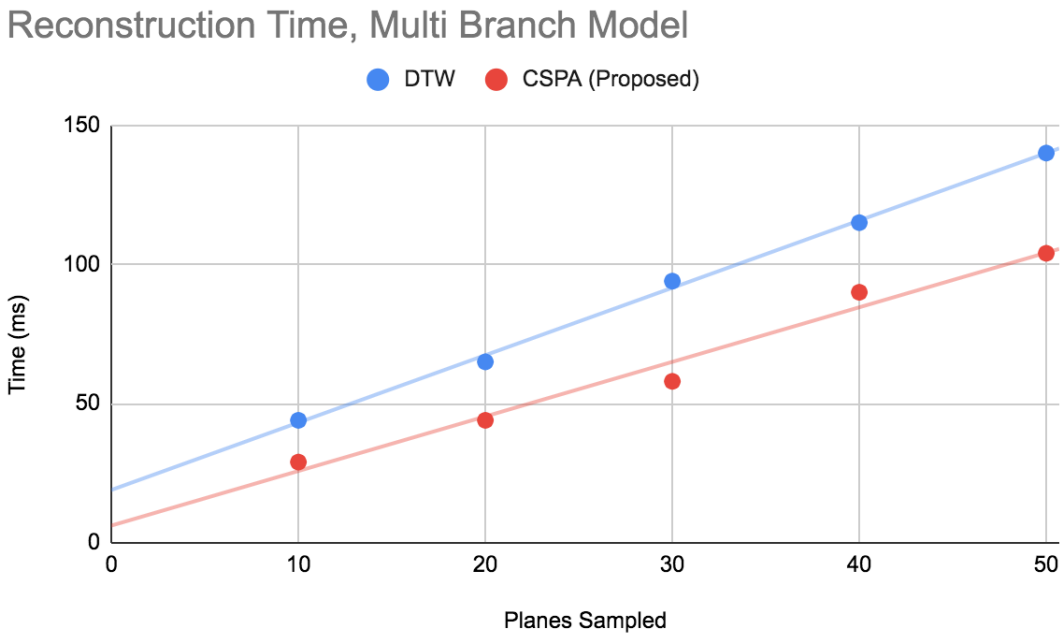


Figure 22: Performance of reconstruction methods on multi branch model

In the branching cases, CSPA splits contours into separate point correspondences, and each correspondence has linear complexity. Mackay's implementation with merging creates larger contours for DTW to act on. Filling in the cost matrix gives DTW quadratic complexity, and so the slowdown for DTW relative to CSPA can probably be attributed to this.

On the real dataset, DTW took an average of 88 seconds for a reconstruction, whilst CSPA took 83 seconds. Both implementations have some naive elements, but this is a good indicator that CSPA does not have any major flaws with regards to performance. It handles larger contours (on the order of hundreds of points) about as well as DTW.

4.6 Analysis Summary

CSPA gave more accurate reconstructions than DTW in branches and similar structures. As plane count lowered for a given object, CSPA's accuracy worsened at a rate slower than DTW's. This can be useful for reconstructing small objects. Also, there may be image stack data which has incomplete contour data (missing contours). In cases where contour correspondences traversing multiple slices are allowed, CSPA may handle the larger gap better.

5 Conclusion

This paper proposes a new method for handling branching cases in point correspondence, and a new point correspondence algorithm. Contours are split in branching cases, and angle/progression metrics are used for point correspondence. The intent was to more deliberately choose key points in the point correspondence process (where splits occur, where correspondence starts from etc.). Synthetic models had contours sampled from them, to be reconstructed by the proposed method and the most comparable prior method by Mackay. The reconstructions were compared to the originals, with Hausdorff distance providing an accuracy metric. It was found that the proposed method on average handles branching cases better, especially at lower plane sample counts. Averaged across all models and plane sample counts, CSPA improved on DTW by an average of 15.2%. Averaged across all models with 10 sample planes, there was an improvement of 17.5%. The proposed method may provide an improvement for real applications with many branching cases and small objects, such as lung scans. The improvement in visual quality should help with understanding scanned structures, and the improvement in mesh quality and accuracy may help with further processing steps.

5.1 Future Research

Contour splitting is currently implemented to handle 1-to-2 cases. However the idea is simple enough to be generalised to 1-to-many. In addition, the ratio of areas can be explicitly calculated instead of approximated by number of points. This will cope with real contour shapes better. The split line curve can be modelled differently depending on factors such as how close the two contours are. For point correspondence, metrics other than point angle and progression can be considered.

Absent from the analysis was a reconstruction of a real branching structure. With a fixed contour correspondence, a comparison between DTW and CSPA (and other non-contour reconstruction methods) on real lung scans would be useful. An alternative would be to implement the synthetic lung model generation by Pluta et al. [25]. Sample planes could then be taken from this model and reconstructed, to allow for analysis on a more realistic structure than the models used in this paper.

References

[1] D. Mackay, "Robust contour based surface reconstruction algorithms for applications in medical imaging," *University of Canterbury*, 2019.

[2] R. Mukundan, "Reconstruction of high resolution 3d meshes of lung geometry from hrct contours," in *2016 IEEE International Symposium on Multimedia (ISM)*. IEEE, 2016, pp. 247–252.

[3] Z. Pan, S. Tian, M. Guo, J. Zhang, N. Yu, and Y. Xin, "Comparison of medical image 3d reconstruction rendering methods for robot-assisted surgery," in *2017 2nd International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE, 2017, pp. 94–99.

[4] W. Xuyi, P. Jianping, Z. Junfeng, S. Chao, C. Yimin, and C. Xiaodong, "Application of three-dimensional computerised tomography reconstruction and image processing technology in individual operation design of developmental dysplasia of the hip patients," *International orthopaedics*, vol. 40, no. 2, pp. 255–265, 2016.

[5] K. H. A. Lim, Z. Y. Loo, S. J. Goldie, J. W. Adams, and P. G. McMenamin, "Use of 3d printed models in medical education: a randomized control trial comparing 3d prints versus cadaveric materials for learning external cardiac anatomy," *Anatomical sciences education*, vol. 9, no. 3, pp. 213–221, 2016.

[6] F. Fischbach, F. Knollmann, V. Griesshaber, T. Freund, E. Akkol, and R. Felix, "Detection of pulmonary nodules by multislice computed tomography: improved detection rate with reduced slice thickness," *European radiology*, vol. 13, no. 10, pp. 2378–2383, 2003.

[7] M. P. Laakso, K. Juottonen, K. Partanen, P. Vainio, and H. Soininen, "Mri volumetry of the hippocampus: the effect of slice thickness on volume formation," *Magnetic resonance imaging*, vol. 15, no. 2, pp. 263–265, 1997.

[8] W. Birkfellner, *Applied medical image processing: a basic course*. CRC Press, 2016.

[9] J. Carr, "Surface reconstruction in 3d medical imaging," *University of Canterbury*, 1996.

[10] J. Pu, J. Roos, A. Y. Chin, S. Napel, G. D. Rubin, and D. S. Paik, "Adaptive border marching algorithm: automatic lung segmentation on chest ct images," *Computerized Medical Imaging and Graphics*, vol. 32, no. 6, pp. 452–462, 2008.

[11] W. Barrett, E. Mortensen, and D. Taylor, "An image space algorithm for morphological contour interpolation," in *Graphics Interface*. CANADIAN INFORMATION PROCESSING SOCIETY, 1994, pp. 16–16.

[12] J. Chai, T. Miyoshi, and E. Nakamae, "Contour interpolation and surface reconstruction of smooth terrain models," in *Proceedings Visualization'98 (Cat. No. 98CB36276)*. IEEE, 1998, pp. 27–33.

[13] M. Zwicker, H. Pfister, J. Van Baar, and M. Gross, "Ewa splatting," *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 3, pp. 223–238, 2002.

[14] "Texture-based volume rendering," accessed from https://web.cse.ohio-state.edu/ crawfis.3/cis694L/Slides/TextureSlicing.pdf.

[15] E. K. Fishman, D. R. Ney, D. G. Heath, F. M. Corl, K. M. Horton, and P. T. Johnson, "Volume rendering versus maximum intensity projection in ct angiography: what works best, when, and why," *Radiographics*, vol. 26, no. 3, pp. 905–922, 2006.

[16] P. Lacroute and M. Levoy, "Fast volume rendering using a shear-warp factorization of the viewing transformation," in *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, 1994, pp. 451–458.

[17] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *ACM siggraph computer graphics*, vol. 21, no. 4, pp. 163–169, 1987.

[18] T. S. Newman and H. Yi, "A survey of the marching cubes algorithm," *Computers & Graphics*, vol. 30, no. 5, pp. 854–879, 2006.

[19] G. M. Nielson, "On marching cubes," *IEEE Transactions on visualization and computer graphics*, vol. 9, no. 3, pp. 283–297, 2003.

[20] I. Braude, J. Marker, K. Museth, J. Nissanov, and D. Breen, "Contour-based surface reconstruction using mpu implicit models," *Graphical models*, vol. 69, no. 2, pp. 139–157, 2007.

[21] G. Guennebaud and M. Gross, "Algebraic point set surfaces," in *ACM SIGGRAPH 2007 papers*, 2007, pp. 23–es.

[22] A. C. Öztireli, G. Guennebaud, and M. Gross, "Feature preserving point set surfaces based on non-linear kernel regression," in *Computer Graphics Forum*, vol. 28, no. 2. Wiley Online Library, 2009, pp. 493–501.

[23] G. Taubin, "Smooth signed distance surface reconstruction and applications," in *Iberoamerican Congress on Pattern Recognition*. Springer, 2012, pp. 38–45.

[24] N. J. Mitra and A. Nguyen, "Estimating surface normals in noisy point cloud data," in *Proceedings of the nineteenth annual symposium on Computational geometry*, 2003, pp. 322–328.

[25] K. Pluta, M. Janaszewski, and M. Postolski, "New algorithm for modeling of bronchial trees," *Image Processing & Communications*, vol. 17, no. 4, pp. 179–190, 2012.

[26] H.-Y. Shum, M. Hebert, and K. Ikeuchi, "On 3d shape similarity," in *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 1996, pp. 526–531.

[27] S. Saxena, N. Sharma, and S. Sharma, "Image processing tasks using parallel computing in multi core architecture and its applications in medical imaging," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 4, pp. 1896–1900, 2013.

[28] M. A. Alsmirat, Y. Jararweh, M. Al-Ayyoub, M. A. Shehab, and B. B. Gupta, "Accelerating compute intensive medical imaging segmentation algorithms using hybrid cpu-gpu implementations," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 3537–3555, 2017.

[29] M. J. Herbert and C. B. Jones, "Contour correspondence for serial section reconstruction: complex scenarios in palaeontology," *Computers & geosciences*, vol. 27, no. 4, pp. 427–440, 2001.

[30] H. Li and R. Wang, "Method of real-time wellbore surface reconstruction based on spiral contour," *Energies*, vol. 14, no. 2, p. 291, 2021.

[31] A. Muntoni and P. Cignoni, "PyMeshLab," *Zenodo*, Jan. 2021.

6 Appendix