COSC470 Research Project Report

Contour Splitting for Branching Structures in CT Image Reconstructions

Cameron Stevenson

Supervisor: Ramakrishnan Mukundan

October 6, 2021

Abstract

abstract text

Contents

1 Overview

overview text Example citation [1, 2, 3]. Example URL [1].

2 Introduction

introduction text

3 Background

background text

3.1 Generic Methods

generic methods text

3.2 Correspondence Methods

subsection preamble text

3.2.1 Contour Correspondence

contour correspondence text

Example list:

- item1
- item2
- item3

3.2.2 Point Correspondence and Triangulation

pc and t text

text after figure declaration

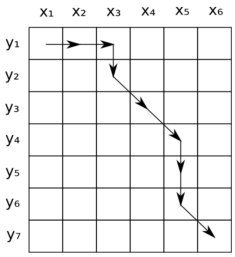
3.2.3 Branching Problem

branching problem text

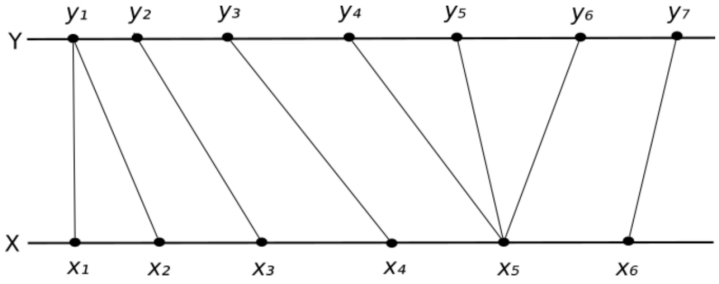
4 Method

method text

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



(a) DTW path through cost matrix



(b) DTW point correspondence

Figure 1: Two examples of DTW paths on contours X and Y [1]

4.1 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.

Example figure ref (See Figure 2).

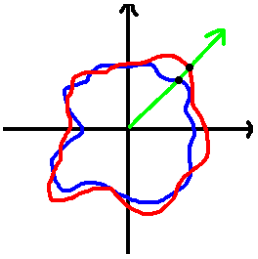


Figure 2: Points matched by angle from shared centroid

4.1.1 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.

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 TODO). 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 TODO). 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.

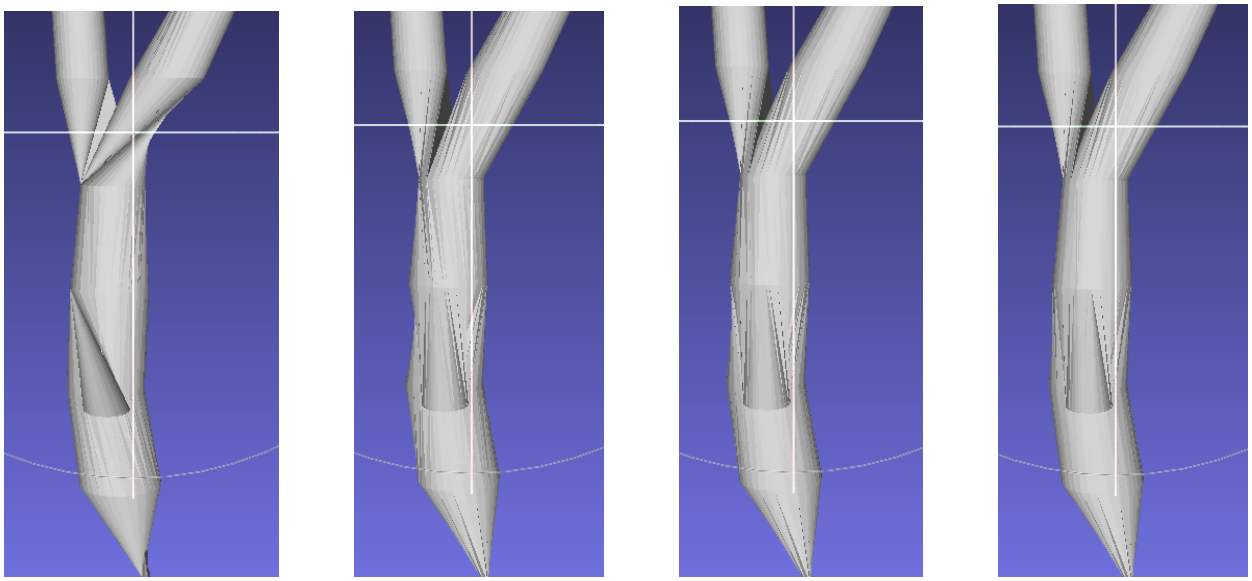
- The ratio of contour areas on the 2 contour side can be reflected in the split contour by adjusting which points the split line connects to. This preserves the internal cross section of each branch half as they join.
- To achieve a smooth point correspondence along the inside of the branch (where the branches join each other), the split line must have points added along it. This is in proportion to the number of points on the original contour.
- 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 creates a split line joint which mimics the intersection of two cylinders, which is approximately what is expected from two branches coming together.

4.1.2 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. For contours where every border point can be seen from the centroid (similar to star-shaped polygons), the angular distance metric is monotonically increasing. In point correspondence, the contours' point angle metrics are leapfrogged between to join points (See Figure TODO). 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. 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, starting from the same point as the angle metric. These two metrics are weighted and summed before the leapfrogging correspondence.

4.2 Implementation

Mackay's report provides a complete implementation of contour and point correspondence (TODO cite software). This implementation was modified to add the options of contour splitting and point angle for point correspondence. The modified source code is available here (TODO git link).



(a) DTW

(b) Point angle, 0% angle weight

(c) Point angle, 50% angle weight

(d) Point angle, 100% angle weight

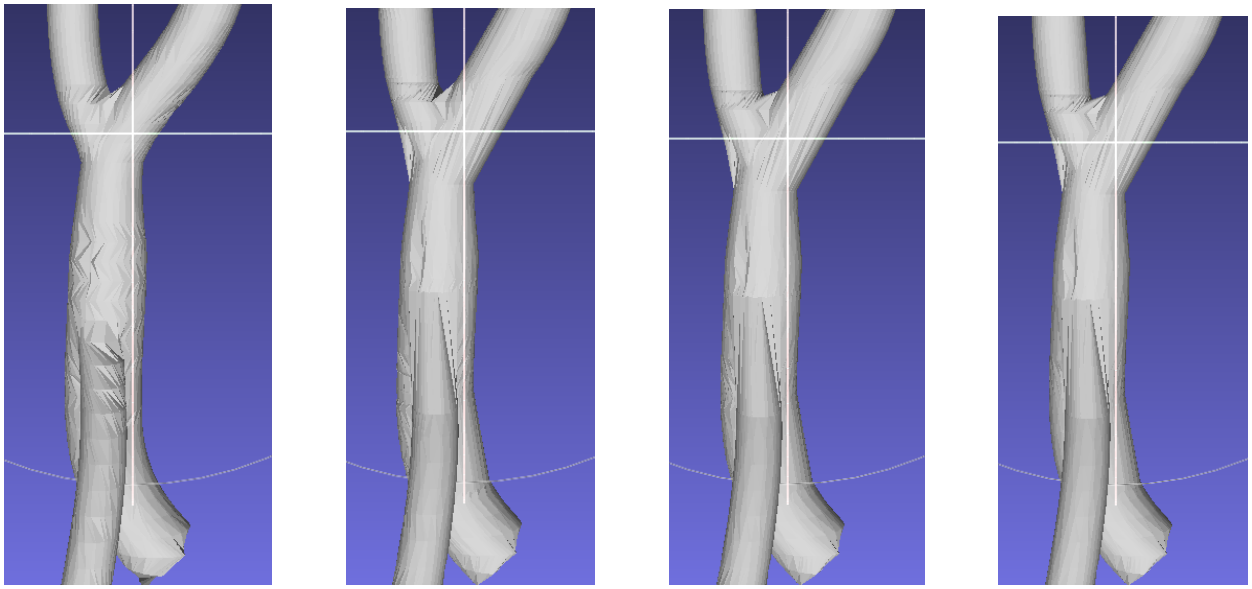
Figure 3: Reconstructions with 10 plane samples

5 Analysis

analysis text

5.1 Ground Truth

ground truth text



(a) DTW (b) Point angle, 0% angle weight (c) Point angle, 50% angle weight (d) Point angle, 90% angle weight

Figure 4: Reconstructions with 50 plane samples

5.2 Visual Results

visual results text

5.3 Measurements

measurements text

5.4 Summary

summary text

6 Conclusion

conclusion text

References

[1] D. Mackay, "Robust contour based surface reconstruction algorithms for applications in medical imaging," 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.

7 Appendix

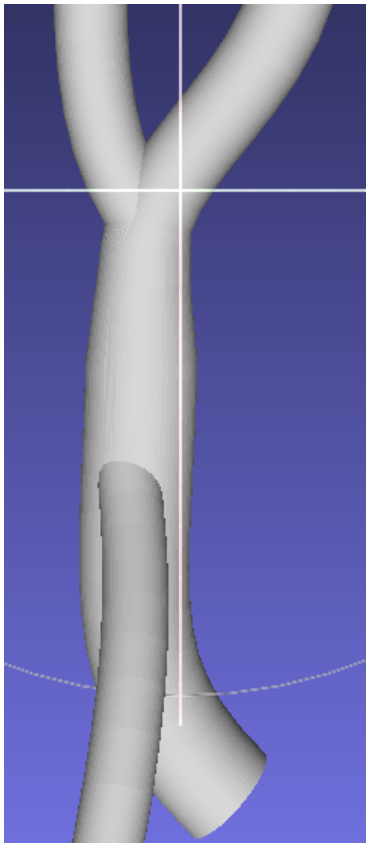


Figure 5: Original multi branch model