COSC470 Research Project Report

Contour Splitting for Branching Structures in CT Image Reconstructions

Cameron Stevenson

Supervisor: Ramakrishnan Mukundan

October 7, 2021

Abstract

abstract text

Contents

1 Overview

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

2 Introduction

introduction text

3 Background

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

3.1 Overview

Images from scans (also referred to as sections or slices) 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.

Any medical imaging reconstruction/render is typically concerned with a particular area of the body, dependent on the application. Most applications tend to use either volumetric rendering or surface rendering.

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 generated through contour correspondence followed by mesh triangulation (with point correspondence as an optional middle step).

3.2 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.

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

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.

3.3 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 considered.

3.3.1 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 [6] 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 [6] 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 [7] 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.

3.3.2 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. [8] 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.

3.3.3 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. [9] 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. [10] 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.

3.4 Typical Methods

generic methods text

3.4.1 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. [11] use elliptic Gaussian kernels as the basis of the shape of each splat.

Texture-based volume rendering intersects many planes with the volume [12]. 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 [6]. 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 [6].

Fishman et al. [13] 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 [14].

3.4.2 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 [15] 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. [16] 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. [17].

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.

Models Braude et al. [18] 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. [19] 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. [20] 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. [21] 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. [22] 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.

3.5 Correspondence Methods

subsection preamble text

3.5.1 Contour Correspondence

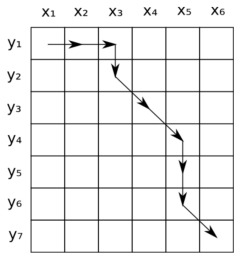
contour correspondence text

Example list:

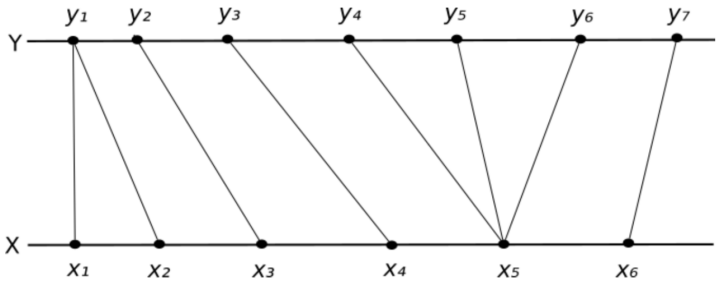
- item1
- item2
- item3

3.5.2 Point Correspondence and Triangulation

pc and t text



(a) DTW path through cost matrix



(b) DTW point correspondence

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

text after figure declaration

3.5.3 Branching Problem

branching problem text

4 Method

method text

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).

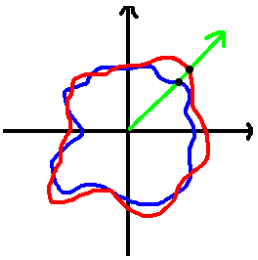


Figure 2: Points matched by angle from shared centroid

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.

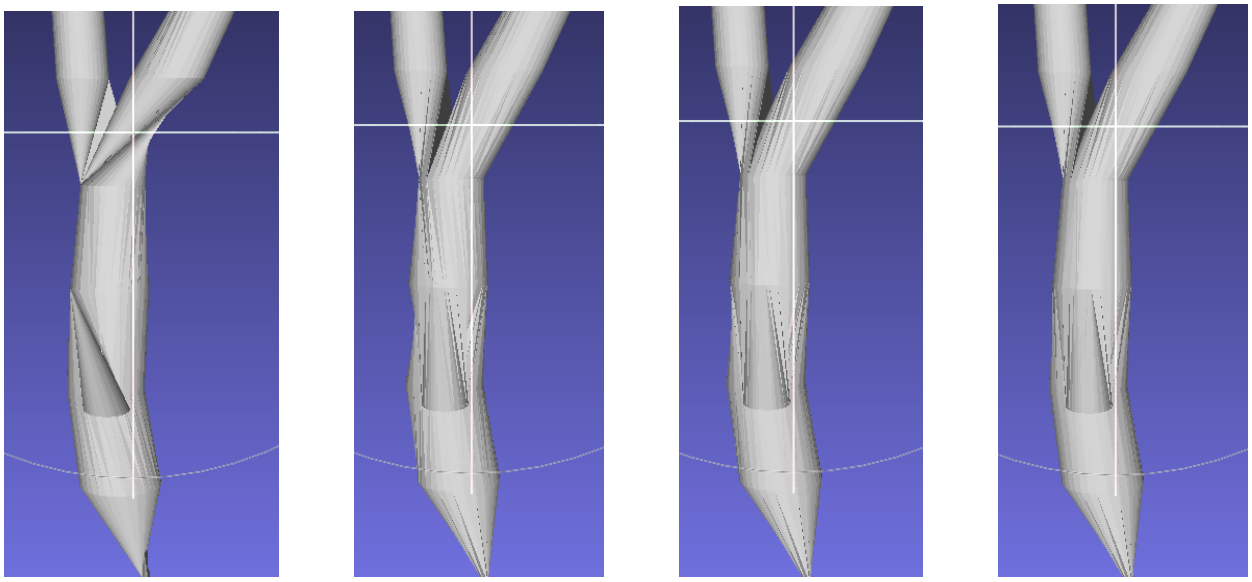
- The ratio of contour areas on the 2 contour side can be reflected in the split contour by adjusting which points the split line connects to. This preserves the internal cross section of each branch half as they join.
- To achieve a smooth point correspondence along the inside of the branch (where the 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

5.2 Visual Results

visual results text

5.3 Measurements

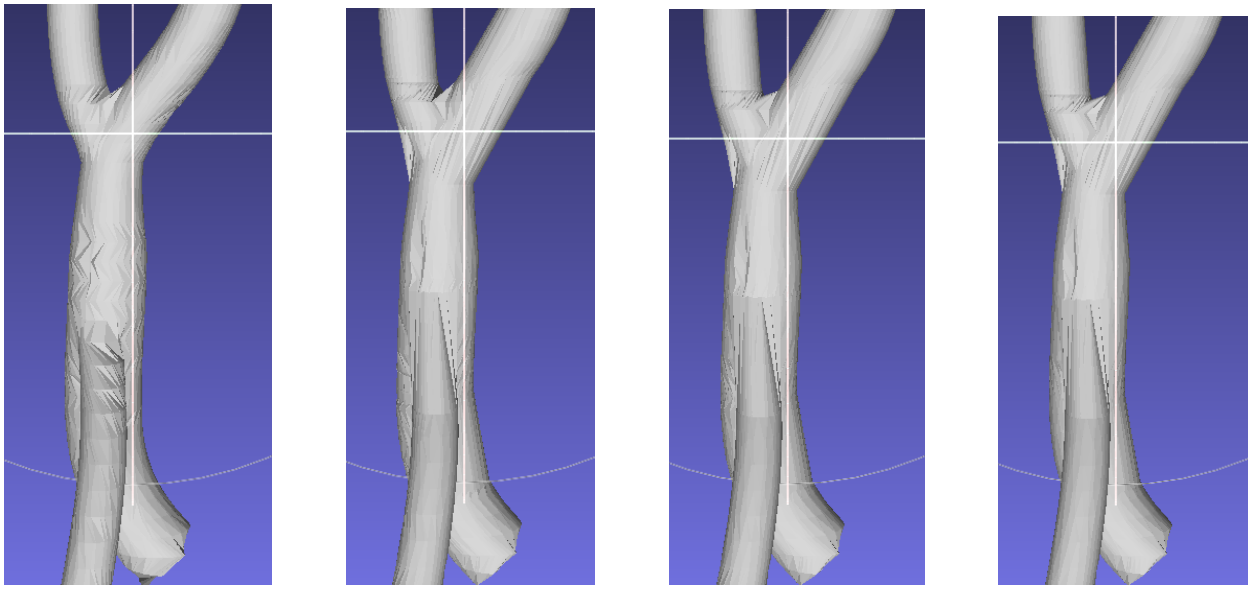
measurements text

5.4 Summary

summary text

6 Conclusion

conclusion 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

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.

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

[7] J. Carr, "Surface reconstruction in 3d medical imaging," 1996.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

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

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

[18] 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.

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

[20] 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.

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

[22] 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.

7 Appendix

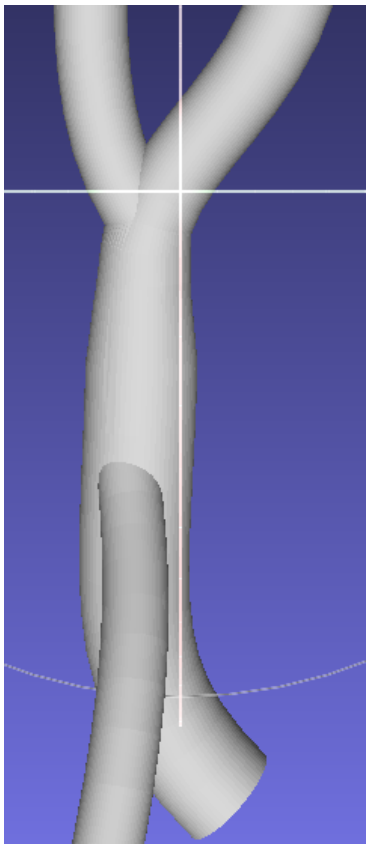


Figure 5: Original multi branch model