University of Victoria
Engineering & Computer Science/Math Co-op
Spring 2017

Airway Segmentation & Quantitative Image Analysis of MicroCT Scanned Lung Tissue

Heart and Lung Innovation
Hogg Lab
Vancouver, BC, Canada

Sukhdip Sandhu
Software Engineering
sukhdips@uvic.ca

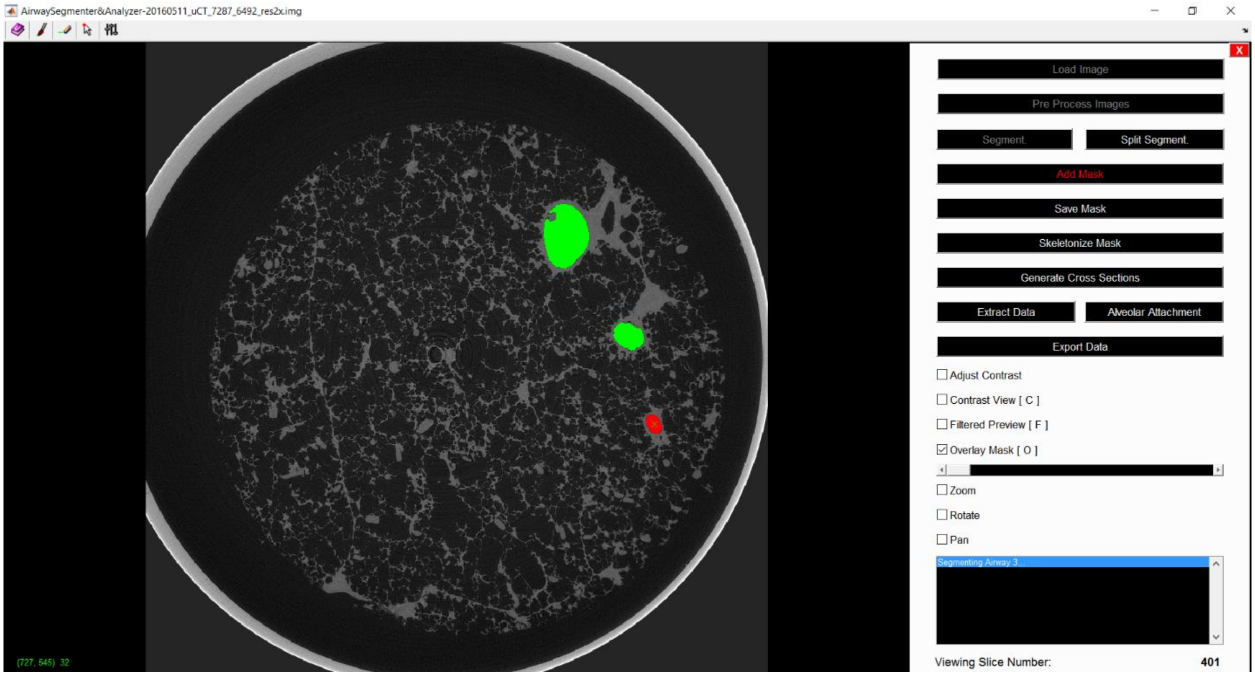
April 28th, 2017

Table of Contents

List of Figures and Tables

Summary

In continual efforts to understand chronic obstructive pulmonary disease (COPD), researchers spend a great deal of time working with microCT scans of lung cores. Currently, there are no fully developed programs available that enable researchers to gain quantitative data of lung airways given a microCT scan. A single program is required for researchers to get the data they need and for this reason, Airway Segmenter & Analyzer (ASA) was developed. ASA, shown in Fig. 1, is a standalone application created in MATLAB that can segment airways, and provide a wide range of measureable data in an unbiased manner using SURS technique. Behind the scenes of its current implementation, however, ASA is heavily reliant on the help of ImageJ – a open source java-based image processing/analysis program. As such, it is recommended that if wanting to publish ASA and make it a truly standalone professional program, it must be written entirely in one language.



□ Adjust Contrast
□ Contrast View [C]
□ Filtered Preview [F]
☑ Overlay Mask [O]

Segmenting Airway 3...

(727, 545) 32

Viewing Slice Number: 401

Figure 1: Graphical User Interface for Airway Segmenter & Analyzer

Glossary

Word	*Definition.*
Alveoli	*Tiny air sacs clustered in the lungs which allow for rapid exchange of oxygen and carbon dioxide.*
Clinic CT Scans	*A computerized tomography scan which combines a series of X-ray images taken from various angles and processes them to create cross sectional images, or slices.*
COPD	*Chronic Obstructive Pulmonary Disease. A disease that obstructs or blocks lung airways making it difficult to breathe.*
In Vivo	*Preformed or occurred inside a living organism.*
Lumen	*The inside space of a tubular structure, such as lung airways or arteries.*
Slice	*A single 2D image in a group of 3D images (stack).*
Stack	*A group of 2D images.*
SURS	*Systematic Uniform Random Sampling. An unbiased sampling technique in which an element in a list in chosen at random and then every k^{th} element in the frame is selected where k is the sampling interval.*
Voxel	*A unit of graphic information that defines a point in 3D space. A pixel having an x y and z coordinate.*

Introduction

The Centre for Heart Lung Innovation (HLI) is a UBC research lab housed in St. Paul's Hospital in the heart of downtown Vancouver. It was established as the Pulmonary Research Laboratory in 1977 by Dr. James Hogg and Peter Pare. Since then, it has grown exponentially in size and welcomed top researchers around the world. As HLI's vision is to be a world leader in understanding and eliminating heart and lung disease, research groups here tackle the deadliest diseases – including coronary artery disease (CAD), and chronic obstructive pulmonary disease (COPD).

Chronic Obstructive Pulmonary Disease

The Hogg Lab, run by Dr. Hogg himself is in the process of understanding emphysema, and other COPDs. In emphysema, the alveolar air sacs in the lungs become damaged and increase in size. This results in breathing difficulty as over-inflated alveoli do not exchange gases due to little or no movement of the alveoli. Emphysema also alters the anatomy of the lung airways, by reducing their elasticity which results in collapsed lungs and further worsening of gas exchange. With the advancements in microscopy – specifically with the introduction of the microCT scanner, researchers are better equipped to spot the signs of emphysemas and COPDs right away.

MicroCT Imaging of Lung Cores

Micro computed tomography is x-ray imaging with the same principles as CT (or CAT) scans but on a much smaller scale and with a huge increase in resolution. Due to its high magnification and smaller depth of view, microCT scans require a core cut out from a donor lung, as the human lungs cannot be scanned in vivo. The lung cores are cylindrical in shape with a diameter of 1.6cm and a depth of 2cm, as reconstructed in Fig. 2. The microCT creates a stack of 1000 slices where each slice is a mere 20μm thick.

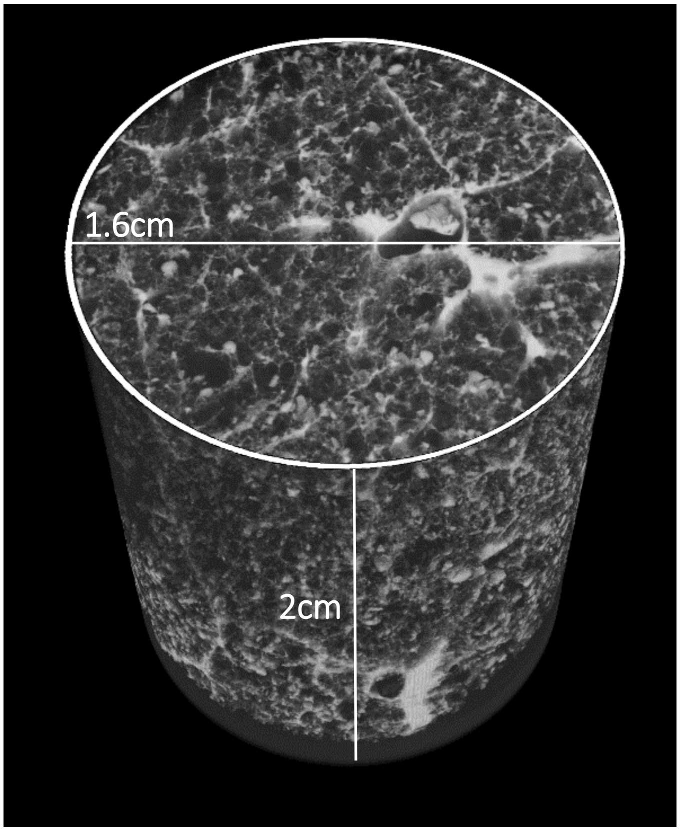


Figure 2 MicroCT scan reconstructed lung core

Airway Segmenter & Analyzer

ASA is the first developed program that is able to segment microCT scans and provide quantitative analysis of airways. A very successful program developed by VIDA Diagnostics Inc. at the University of Iowa is designed to do quantitative imaging analysis, but in its current stage is unable to work with the small, low contrast airways trees of the microCT scans. Located in the appendix as a supplementary documentary is the full user manual for ASA which gives a more detailed overview on how to use the program.

Airway Segmenting Algorithm

The airway segmentation algorithm can be broken up into two steps. The first is preprocessing the image stack, and second is searching the image stack for the selected airway. The supplementary user manual in Appendix 2 – 6 covers segmentation.

Preprocessing the Image Stack

Preprocessing the image stack consists of applying 5 filtering and image processing techniques. First, the slices (Fig. 3A) in the image stack get sharpened (Fig. 3B) to increase the border contrast. Second, the sharpened stack gets binarized (Fig. 3C), meaning every value above a certain threshold becomes a 1, and every value below the threshold becomes 0. This threshold is adjustable by the user using a slider tool and can be adjusted afterwards to find the best binarization based off the target airway the user wishes to segment. Afterwards, noise removal (Fig. 3D) is applied to get rid of everything under 8 pixels in size that is not connected to a larger structure. Next, the image simply gets inverted so 1s becomes 0s and vice versa (Fig. 3E). Lastly, the image gets eroded to thicken the cell walls and improves connections between shapes (Fig. 3F). Evidently, a lot of information disappears during the erosion stage, but it is a crucial step as it helps distinguish airways from their surroundings.

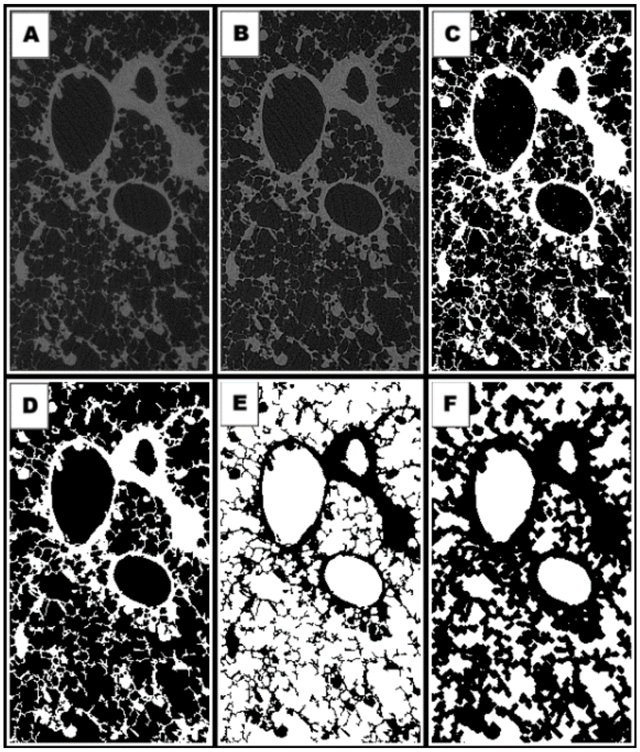


Figure 3 Filtering and thresholding algorithm

To calculate the average amount of time it took to preprocess, varying file sizes were preprocessed and execution time was calculated using MATLABs built in timer. The results, as shown in Table 1, are based off a machine with an Intel® Xeon® Dual processor E5520 @ 2.27GHz. The average MB/second to preprocess is an image is: 9.95MB/s.

Table 1: Pre-processing execution time based off file size

File Size [MB]	47.818	127.513	255.025	352.538	510.688	632.803	790.016
Time [s]	9.816904	12.66019	26.61178	39.85082	53.21301	61.76651	76.08085

Located the Image Stack

The next stage in the segmentation algorithm is finding and segmenting the desired airways. The algorithm works by assigning a different label to each unique shape. If there is a matrix with at least 8 connected components around it in the slice, it is given a unique label which is used for identification, as shown in Fig. 4. From here, to select a distinct shape, all that is needed is its identification number. *Note here the two airways we will explore are labeled 55 and 81.*

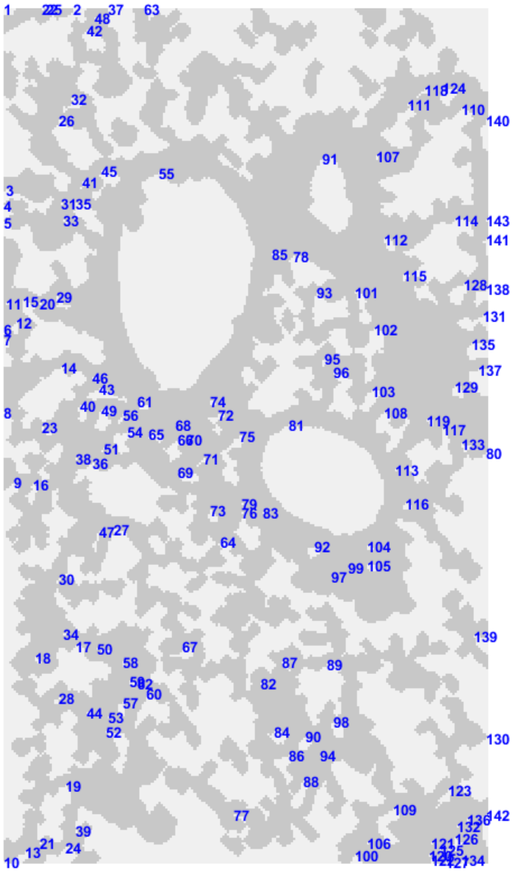


Figure 4 Labeled Slice

To get the identification number, the user must hover their cursor over the desired airway, and when pressing the ‘x’ key, the mouse’s X and Y coordinates are collected and mapped to the image to find the label of the desired airway. As some datasets can be up to 1000 images, the X and Y data gets linearly interpolated between slices so the user does not have to click on every single slice in the stack.

After locating the specific airway, it gets dilated with the exact structuring element as it was eroded in hopes of preserving the true segmentation shape as best as possible. Since erosion and dilation are identical operations in different directions, this approach works well. Fig. 5A shows the original airway, where Fig. 5B is the pre-dilated airway mask and Fig. 5C is it overlaid onto the original. Likewise, Fig. 5D is the dilated airway mask and Fig. 5E it is overlaid onto the original. Fig. 5E is very close to what one would expect given the original cell shape.

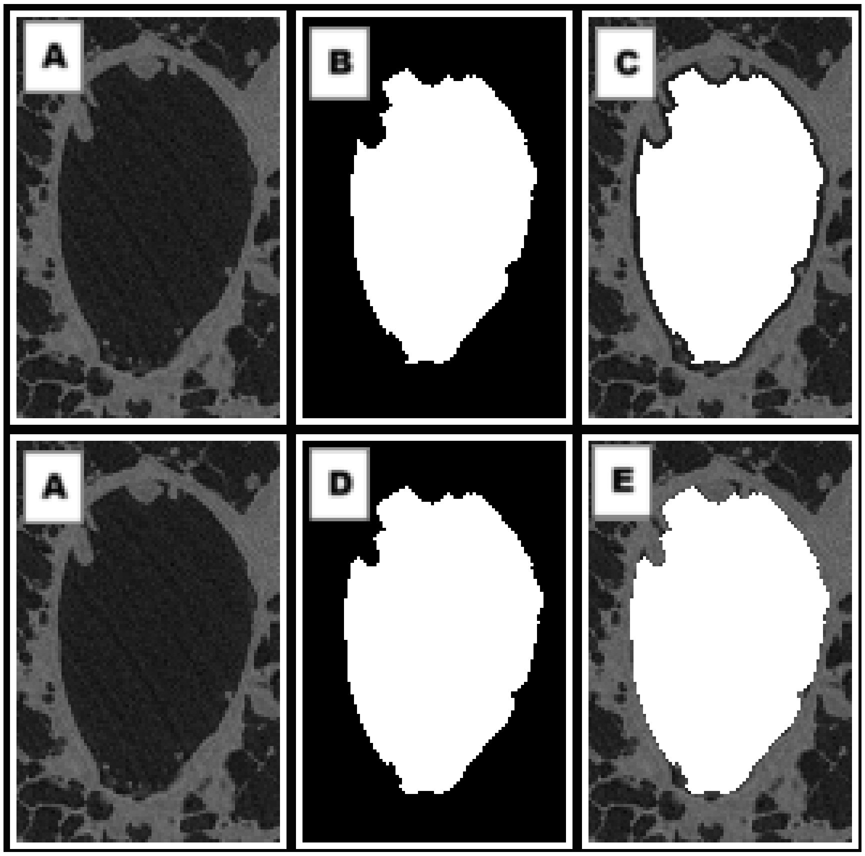


Figure 5 Dilating Airway after Segmenting

Airway Segmenting Tools

One problem that occurs from this algorithm are leakages, or explosions, which occur when the targeting airways are not perfectly closed off and are connected to an undesired area, as shown in Fig 6A and Fig. 6B. If a user were to select such an airway, a mask as shown is Fig. 6C would be produced. To deal with this, there is a tool that can be used to 'split' the desired region from the undesired one. Any lines drawn (Fig. 6D), get *burned* into the image and turn the 1's in that region to 0's, which can be used to trick the computer into thinking there is a boundary (Fig. 5E). When executing the segmentation now, a desirable mask, Fig. 6F, gets produced.

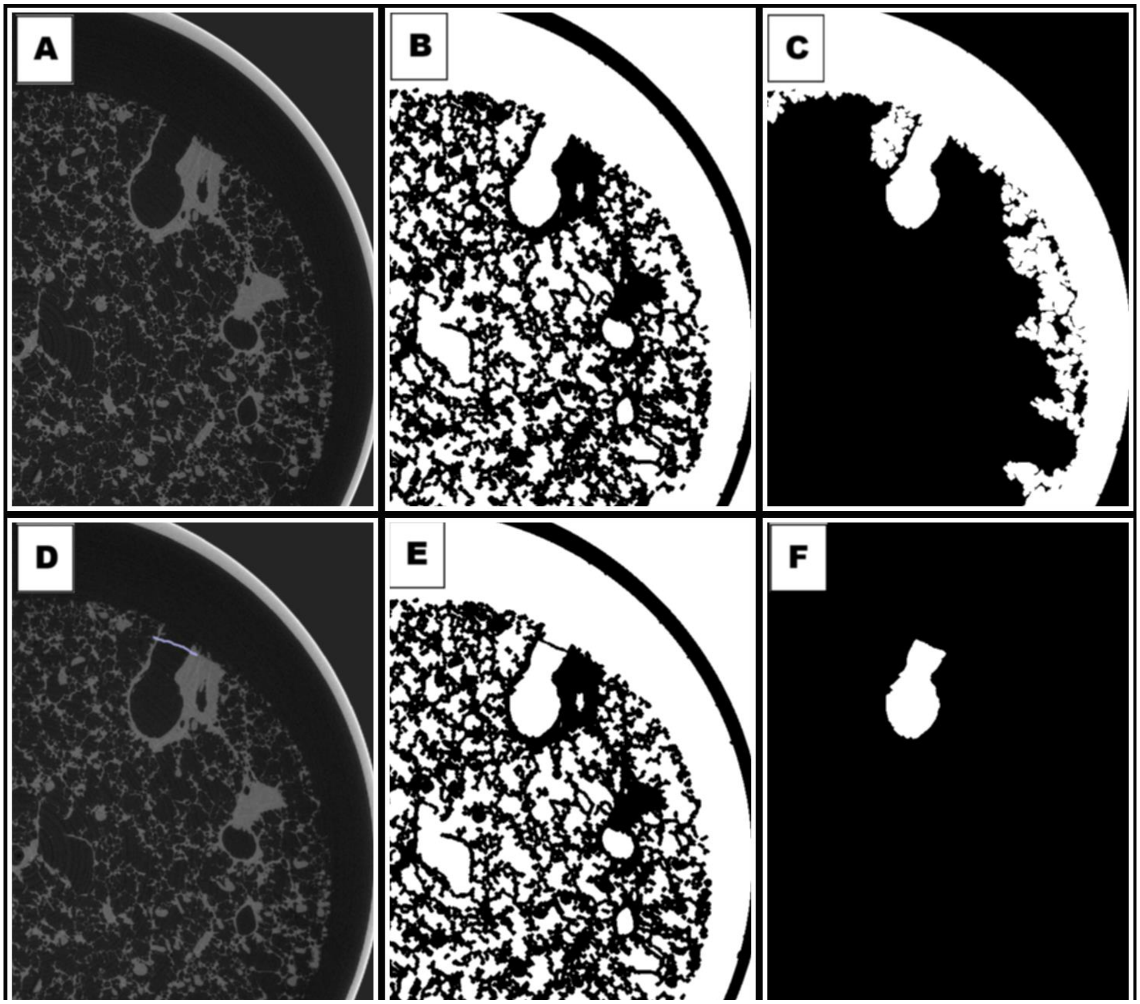


Figure 6 Explosion Handling

Moreover, there is a paintbrush tool that enables the user to draw onto the image (Fig. 7), a region adding tool which allows the user to create distinct shaped objects onto the image (Fig. 8), and a connecting tool that places a 3px by 3px rectangle onto the image that is used when the airway branch is obstructed (Fig. 9). Fig. 9A, 9B, and 9C display an airway becoming obstructed overtime, and Fig. 9D shows the connected region tool being placed over the obstructed airway to simulate a connection. This is a crucial tool for skeletonizing, during the airway analysis step.

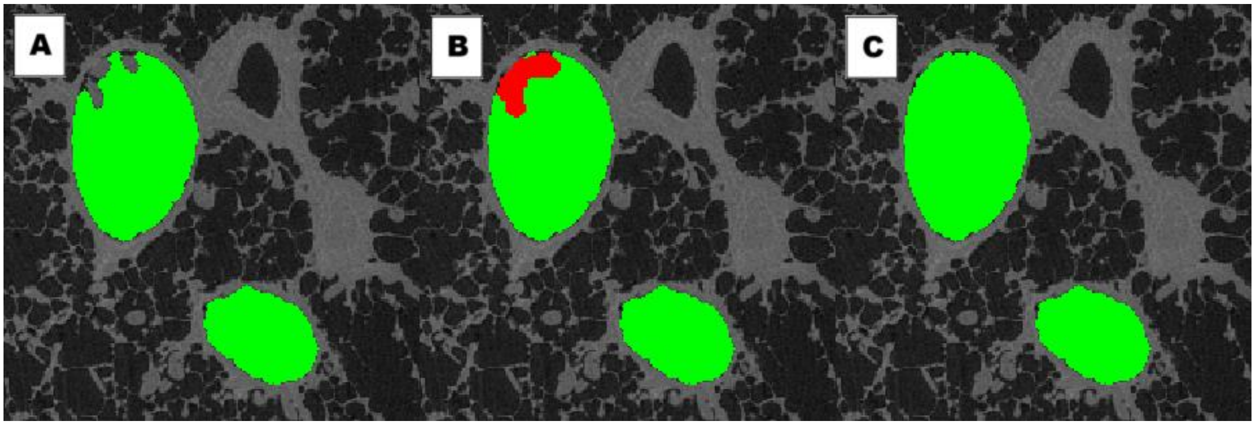


Figure 7 Paintbrush Tool for Adding Manual Segmented Airways

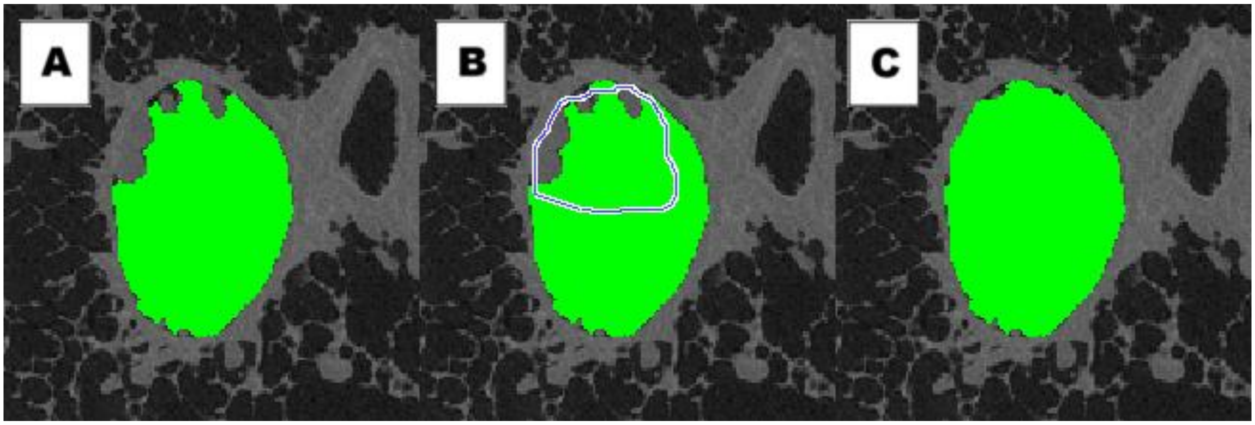


Figure 8 Region Adding Tool for Adding Segmented Airway

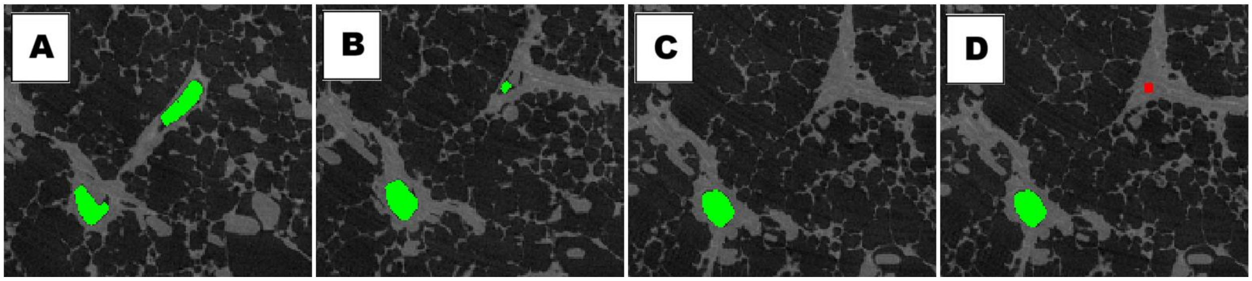


Figure 9 Connecting Tool for Obstructed Airways

Airway Analyzing

Once the desired airways are selected, the quantitative analysis can begin. For this, the airways will need to get skeletonized and have their branch points locations collected. Then the stack needs to be manipulated to get cross sectional cuts based off the centerline of the generated skeleton. The supplementary user manual in Appendix A8 – A11 covers airway analysis in finer detail.

Up until now, everything program wise is done purely in MATLAB. As more complex algorithms need to be performed, a MATLAB-to-ImageJ Bridge, MIJ is installed to allow bi-directional communication and data exchange between the two respective programs.

Skeletonize3D

The image masks get sent over to ImageJ where they first undergo a 3D Gaussian Blur filter to smoothen out the mask (Fig 10A) in the X Y and Z directions, and then undergo a 3D skeleton operation [1]. The Skeleton operation symmetrically erodes the surfaces of the mask until there is only 1 pixel that remains (Fig. 10B) which is guaranteed to be the medial position of the slice, and the centerline of the stack. The centerline of the stack is of importance as it will be used to determine the planes/cross sectional cuts of the image stack.

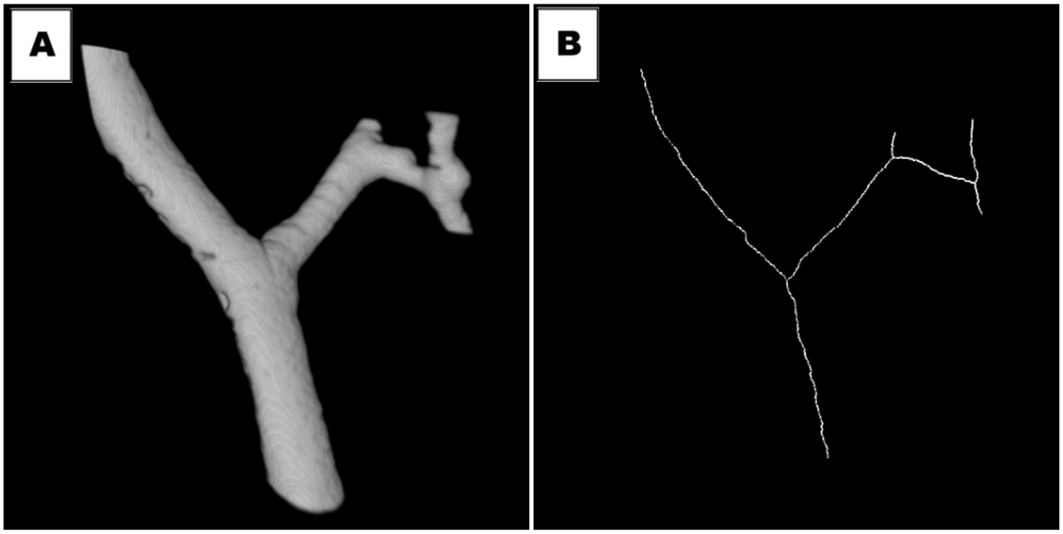


Figure 10 Reconstruced airway and its respective skeleton

Branch Coordinates

The branch coordinates are detected via an ImageJ plugin called Analyze Skeleton (2D/3D) [2] that Dr. Naoya Tanabe, a researcher at HLI modified. This plugin analyzes the voxels in the skeletonized stack and classifies them into three categories. In Fig. 12, end-point voxels (blue) are determined as having less than 2 neighboring pixels. Junction voxels (purple) are classified as having more than neighboring 2 pixels, and slab voxels (yellow) are determined as having exactly 2 neighboring pixels. In 2D, neighboring pixels are defined as any of the 8 pixels that are around a central pixel. In 3D however, neighboring pixels are the 26 pixels located around the central pixel. End-point voxels are of interest so the X Y and Z coordinates of those get determined and overlaid onto the skeleton, as shown in Fig. 11. These points can then be clicked on to select the start and end of the branch the user wants to analyze.

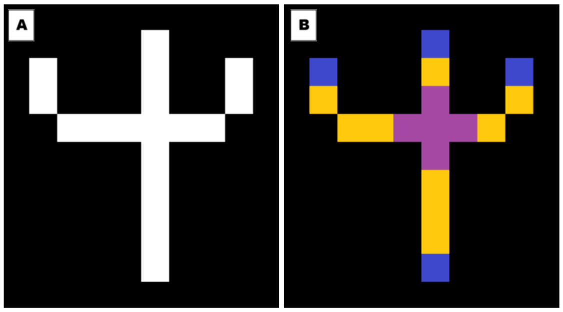


Figure 12 Voxel classifications

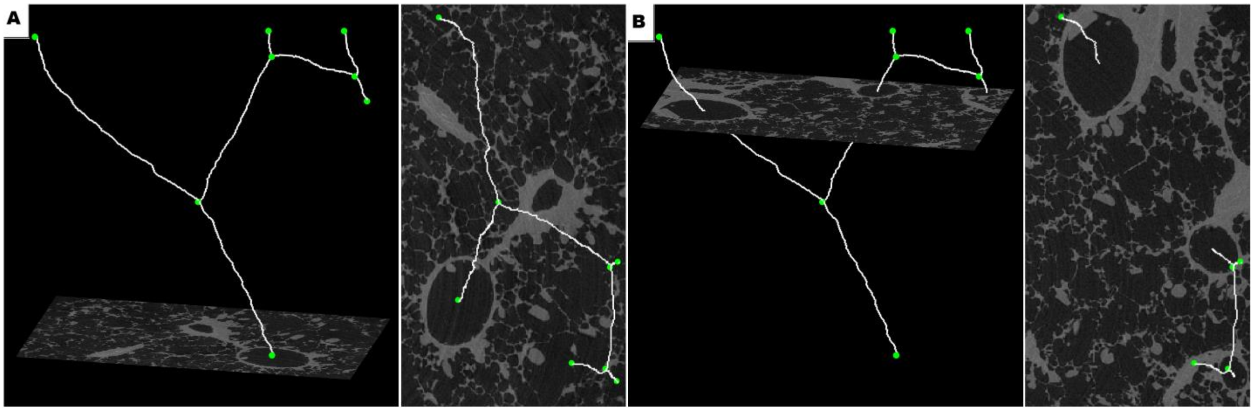


Figure 11 Branch endpoints overlaid the skeleton from various angles and slice positions

Cross Sectional Cuts

With the starting and ending branch points selected, and the centerline determined, linear algebra comes into use to rotate the stack in 3D to determine the cross sections in the correct plane. First, based off the skeleton, perpendicular vectors are created throughout the branch. Next, a random point is selected in the branch, and two orthogonal vectors are determined which represent the plane for the cross section. Lastly, the pixel density values get interpolated from the original stack to the new one, effectively creating a cross sectional view of the branch.

In Fig. 13A, the original stack is viewed from the original scan in the XY direction with (x,y,z) rotation at (0,0,0). Figure 13B is rotated 45 degrees in the X direction with a rotation of (45,0,0), and lastly, 13C is rotated 90 degrees in the XZ direction, with a rotation of (90,0,0). It is aligned in the same direction as the perpendicular vector based off the skeleton.

For each branch selected, 10 cross sectional images get generated randomly to avoid bias.

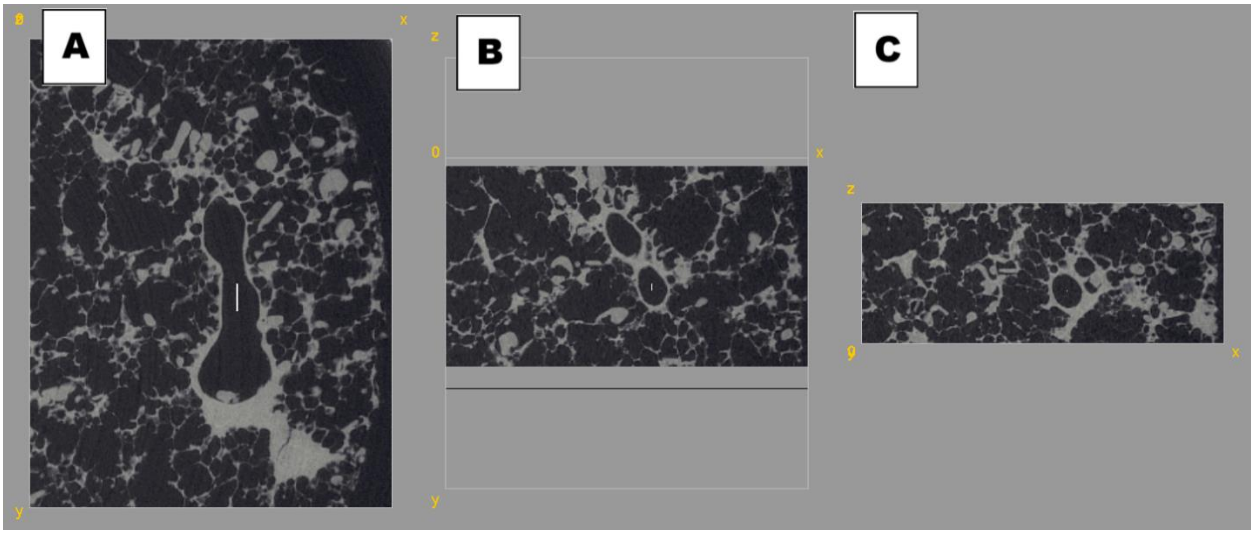


Figure 13 Cross sectional reconstruction based off skeleton centerline

Analysis of Cross Section Slices

To analyze the cross sectional areas, as demonstrated in Fig. 14A, 128 rays get projected 360 degrees around the airway from the center of lumen. Along the ray, the pixel density values get calculated and using the full width at half-maximum principle, the inner Fig. 14B, and outer Fig. 14C, edges of the shape can be determined. 128 editable points are created around the airway and based off these points, the area and perimeters can be determined for inner and outer walls.

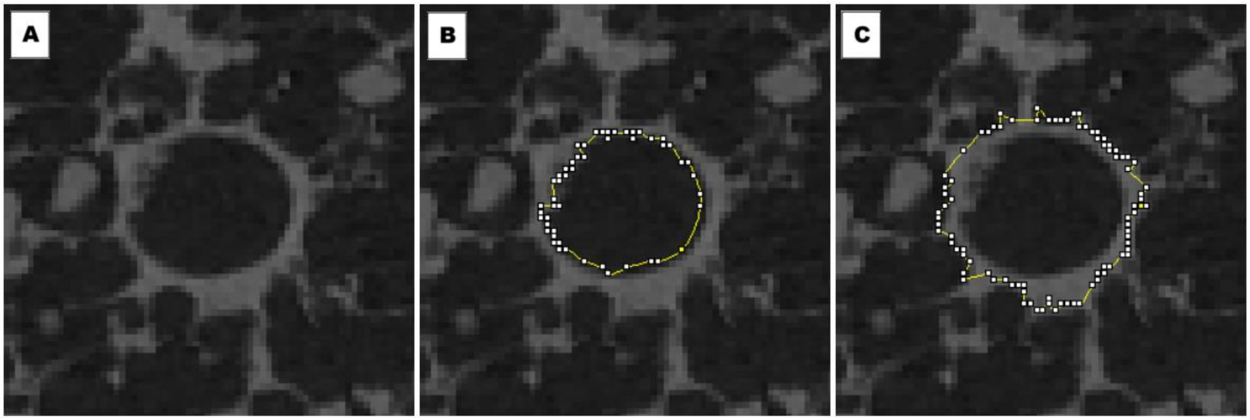


Figure 14 Sample cross section with inner and outer wall analysis

Based off inner and outer areas and perimeters, as well as basic geometry formulas, several critical formulas can be derived.

Airway luminal area:	**Ai** = given
Total area:	**Ao** = given
Luminal perimeter:	**Pi** = given
Outer perimeter:	**Po** = given

Inner diameter:	**ID** = $2x\sqrt{}(Ai/\pi)$
Wall thickness:	**WT** = $\sqrt{}(Ao/\pi) - (ID/2)$
Wall area:	**WA** = $Ao - Ai$
Wall area percent:	**WA%** = $100 \times (WA/Ao)$
Circulatory	**Circularity** = $4\pi \times (Ai/Pi^2)$

Exported Data

Upon completion, the data will get formatted and entered into an excel spreadsheet (Table 2) with all the details about the branch being analyzed, including its identifications, core # and sub branch (both user inputted) as well as alveolar attachment counting which is explained in detail in appendix A10, as well as the length of the branch.

Table 2 Sample exported data

Core #	Sub Branch	Slice #	Ai	Ao	Pi	Po	ID	WT	WA	WA%	Circularity	AA#	Length
6492	Parent	1	1.38889	2.01747	4.52166	5.77364	1.32981	0.13646	0.62858	31.15699	0.85365	9.00000	
6492	Parent	2	1.33707	2.03103	4.65374	5.87888	1.30476	0.15167	0.69396	34.16786	0.77582	1.00000	
6492	Parent	3	1.29252	1.94870	4.49953	6.04398	1.28284	0.14616	0.65619	33.67296	0.80225	4.00000	
6492	Parent	4	1.32399	1.97921	4.38863	5.68927	1.29837	0.14454	0.65522	33.10497	0.86385	5.00000	
6492	Parent	5	1.34143	2.01650	4.57737	5.87071	1.30689	0.14772	0.67507	33.47743	0.80453	9.00000	
6492	Parent	6	1.38889	2.13854	4.81937	5.91501	1.32981	0.16015	0.74965	35.05435	0.75144	9.00000	
6492	Parent	7	1.46976	2.14289	4.55040	5.85869	1.36797	0.14191	0.67314	31.41243	0.89198	4.00000	
6492	Parent	8	1.51092	2.14677	4.72556	6.12989	1.38700	0.13314	0.63585	29.61877	0.85025	8.00000	
6492	Parent	9	1.60196	2.28624	4.86234	6.11771	1.42818	0.13898	0.68427	29.93010	0.85148	5.00000	
6492	Parent	10	1.61940	2.57971	4.91404	7.20749	1.43593	0.18821	0.96031	37.22546	0.84273	6.00000	4.5253
6492	Daughter1	1	0.83537	1.56225	3.60024	5.33713	1.03132	0.18952	0.72689	46.52821	0.80988	7.00000	
6492	Daughter1	2	0.79808	1.24699	3.45666	4.70321	1.00804	0.12600	0.44892	36.00000	0.83935	6.00000	
6492	Daughter1	3	0.67991	1.06927	3.24776	4.57555	0.93043	0.11819	0.38935	36.41304	0.81002	9.00000	
6492	Daughter1	4	0.56175	0.90219	3.05760	3.83958	0.84572	0.11303	0.34044	37.73483	0.75508	10.00000	
6492	Daughter1	5	0.55013	0.92205	2.92976	4.18332	0.83693	0.12329	0.37192	40.33614	0.80540	11.00000	
6492	Daughter1	6	0.52253	0.91672	3.02710	4.24933	0.81566	0.13236	0.39420	43.00053	0.71658	9.00000	
6492	Daughter1	7	0.47991	0.94287	2.79598	4.98411	0.78169	0.15699	0.46296	49.10118	0.77144	8.00000	
6492	Daughter1	8	0.51042	1.02181	7.19656	6.05556	0.80616	0.16723	0.51139	50.04739	0.12385	8.00000	
6492	Daughter1	9	0.50025	1.19372	3.07275	5.56434	0.79808	0.21738	0.69347	58.09330	0.66580	10.00000	
6492	Daughter1	10	0.52689	1.23973	6.09063	6.66877	0.81906	0.21866	0.71285	57.50000	0.17848	8.00000	4.3850
6492	Daughter2	1	1.29397	1.96710	4.29329	5.99753	1.28356	0.14951	0.67313	34.21959	0.88217	13.00000	
6492	Daughter2	2	1.26637	2.16856	4.39467	6.46974	1.26980	0.19593	0.90219	41.60339	0.82398	10.00000	
6492	Daughter2	3	1.27557	1.78938	4.38290	5.53212	1.27440	0.11750	0.51381	28.71449	0.83443	11.00000	
6492	Daughter2	4	1.37000	2.09834	4.54175	6.05912	1.32073	0.15690	0.72834	34.71036	0.83461	14.00000	
6492	Daughter2	5	1.42133	2.19277	4.49709	6.09035	1.34525	0.16283	0.77144	35.18109	0.88317	13.00000	
6492	Daughter2	6	1.39518	2.28914	4.70297	6.81824	1.33282	0.18721	0.89396	39.05225	0.79268	15.00000	
6492	Daughter2	7	1.46298	2.20391	4.59455	6.42253	1.36482	0.15516	0.74093	33.61899	0.87089	8.00000	
6492	Daughter2	8	1.44070	2.46687	4.64152	6.83052	1.35439	0.20894	1.02617	41.59796	0.84036	9.00000	
6492	Daughter2	9	1.35305	2.62668	4.62844	7.57135	1.31254	0.25811	1.27363	48.48820	0.79369	7.00000	
6492	Daughter2	10	1.16709	1.80729	4.67377	6.77467	1.21901	0.14897	0.64020	35.42337	0.67140	9.00000	6.7164

Limitations

A major limitation in the airway analysis occurs due to memory issues. MATLAB by default can only allocate 1/4th of the computers RAM to created objects. MIJ, the bridge between MATLAB and ImageJ is an object itself, therefore, on a 12GB machine, only a maximum of 3GB can be handled.

As the full image stacks themselves can be up to 1GB in size, using the full stack is not feasible. To compensate, the user will be required to crop the data sets to display only the required information. In its current implementation, another way around this would be to move to a computer with greater RAM as memory is the only thing holding back the program.

Conclusion

So far, ASA has proven to be a successful, functional program. It is able to segment selected airways and provide a multitude of information about said airways in a short period of time. It has a clean user interface that takes advantage of many object properties. It is straight forward to use with a very linear workflow, but also offers many tools to aid along in the processes. This program in its current stage can be used by researchers around the globe wanting to have a unified program both segment microCT airways, and then analyze them in a quick and simple way.

Recommendations

It is recommended that further development is made on ASA if it is to be a program that is comparable to that of the VIDA Diagnostics Inc. team. The MIJ Bridge between MATLAB and ImageJ works fairly well as it utilizes the best of MATLABS image processing toolbox and ImageJ's professionally written skeleton analysis code, however, there is only a limited supply of memory that can be exchanged. A recommendation for future developers working on ASA would be to allow sending only a cropped data to ImageJ and have it be mapped back onto the original full stack.

On a more ambitious scale, I would propose having ASA be developed entirely in either MATLAB, entirely in Java to be used as an ImageJ plugin, or entirely in languages like C++/C# which is what the programmers at VIDA use.

References

[1] Ignacio Arganda-Carreras. Skeletonize3D. Available at http://imagej.net/Skeletonize3D

[2] Ignacio Arganda-Carreras. AnalyzeSkeleton. Available at http://imagej.net/AnalyzeSkeleton