

Creating AustinMan: An Electromagnetic Voxel Model of the Visible Human

Jackson W. Massey

Undergraduate Thesis

Department of Electrical and Computer Engineering

Cockrell School of Engineering

University of Texas at Austin

May 2011

Faculty supervisor: Dr. Ali E. Yilmaz

Abstract

The objective of this thesis was to develop an electromagnetic voxel model of the human body. The male dataset from the Visible Human Project was used to create the so called AustinMan Electromagnetic Voxels model. A graphical user interface, the Boundary and Region Software Kit (BRISKit), was created in MATLAB for users to label image slices with their respective tissues, identify tissue boundaries, edit the labeled tissues, and review the tissues labeled in each slice. A semi-automated, highly-interactive method was developed to ensure accurate results within a reasonable timeframe. The methods and algorithms used to develop the AustinMan model are described in this thesis. Additionally, all assumptions regarding tissue labeling and problematic regions have been carefully documented and are made available on the web. The AustinMan model is currently available for download for research and teaching purposes.

Table of Contents

1	Introduction	1
2	Initial Attempts	2
2.1	MRI Dataset	2
2.2	CAT Dataset	5
2.3	Cross-Sectional Photograph Dataset	5
3	Adjustments to Cross-Sectional Images	7
4	Mask Slices	9
4.1	Cropping	10
4.2	Color-Space Filtering	10
4.3	Manual Adjustments	14
5	Image Segmentation and Tissue Labeling	15
5.1	Independent Processing	15
5.1.1	Algorithm 1: Region Growing	15
5.1.2	Algorithm 2: Manual Region Stitching	18
5.1.3	Algorithm 3: Boundary Competition for a Fixed Number of Iterations	19
5.1.4	Manual Adjustments	19
5.1.5	Tissue Labeling	20
5.2	Prediction-Correction	21
5.2.1	Algorithm 4: Boundary Competition with User-Selected Mean Colors	22
5.2.2	Algorithm 5: Boundary Competition among User-Selected Regions	23
5.2.3	Algorithm 6: Adjust Size	23
5.2.4	Final Algorithm	23
5.3	Independent Processing vs. Prediction-Correction	23
6	Boundary and Region Identification Software Kit (BRISKit)	24
6.1	Home Mode	24
6.2	Connect Mode	25
6.3	ID Materials Mode	26
6.4	Review Mode	26
7	Coarsening and Extrusion	28
7.1	2-D Coarsening: AustinMan v1.0	28

7.2	3-D Coarsening: AustinMan v1.1 and Later	29
8	Results	31
9	Future Work.....	38
10	Conclusion	38
11	Acknowledgments.....	39
	Works Cited	40
	Appendix A: Tissue Assumptions and Problematic Regions	42

1 Introduction

The proliferation of wireless communication devices that radiate electromagnetic power has led to an increased interest in the study of electromagnetic fields around the human body. Electromagnetic simulations complement experimental measurements on human phantom models (Figure 1) as they can account for the effects of tissue inhomogeneities and body-specific variations. A major roadblock for these simulations is the scarcity of accurate models of the human body that limits their predictive power. The objective of the research detailed in this thesis is to create a publicly available highly accurate anthropomorphic human model (henceforth referred to as the AustinMan model) that will initially be used to study the effects of cell phone radiation [1].



Figure 1: The IEEE standard SAM phantom that is currently used to certify compliance of cell phones with relevant safety standards [2].

There are several commercially available human models for electromagnetic simulations, such as the HUGO and Virtual Family models [3-5]. Unfortunately, information on how these models were created (such as how tissue boundaries were identified) is often not available, perhaps for intellectual property reasons. As a result, these commercial models are not reproducible and their shortcomings are not transparent: The models cannot be verified for their accuracy or improved by independent groups. In contrast, the AustinMan model that has been developed in this thesis research is not only highly accurate but is also meticulously documented: All the assumptions made during model development and the problematic regions in the model are detailed in public documents; moreover, all the raw data needed for reproducing the model is also published on the internet; thus, AustinMan is a reproducible and ‘open’ human model.

The AustinMan model was constructed in four steps as detailed in the following sections. First, an appropriate image dataset was selected as a basis for the model; this selection process is described in Section 2; some of the discrepancies in the images in the selected dataset were corrected with minor adjustments as detailed in Section 3. Second, the irrelevant background information was removed in the images by creating “mask slices” as explained in Section 4.

Third, the images were segmented and different tissue types were identified, i.e. the pixels in each image were labeled with an appropriate tissue ID as discussed in Section 5. This segmentation process, which can be extremely time consuming, was semi-automated by developing and using the Boundary and Region Identification Software Kit (BRISKit), which provides image processing tools to speed up the process. BRISKit is described in Section 6. Fourth and last, the slices were coarsened and extruded to create a three-dimensional model and converted to a format that can be used by electromagnetic simulations as detailed in Section 7. Images from the latest version of the model, AustinMan v1.1, are shown in Section 8; future work for improving the model and developing a complete-body version are detailed in Section 9; and the conclusions of the thesis are presented in Section 10.

2 Initial Attempts

The AustinMan model was created by using the high-resolution images obtained by the National Library of Medicine's Visible Human Project (VHP) [6,7]. The VHP dataset contains images for a male and a female body. Each body was imaged using multiple modalities: MRI, CAT scan, and cross-sectional photographs of the bodies are available. All of the data are stored in their original formats (*.t1, *.t2, *.pd, *.fre, *.raw) and also converted to PNG format by the VHP. After converting a portion of the original images to a PNG format to verify the images, the PNG files that were already converted by the VHP were used for convenience.

2.1 MRI Dataset

Initially, the MRI data was planned to be used as the basis for the AustinMan model. The converted PNG files in this dataset are 16-bit grayscale images, with the MRI data stored in the lower 12 bits. To view the images, the bits were shifted six bits to the left, which provided the best contrast. The images were then converted to 8-bit grayscale images and saved (Figure 2).

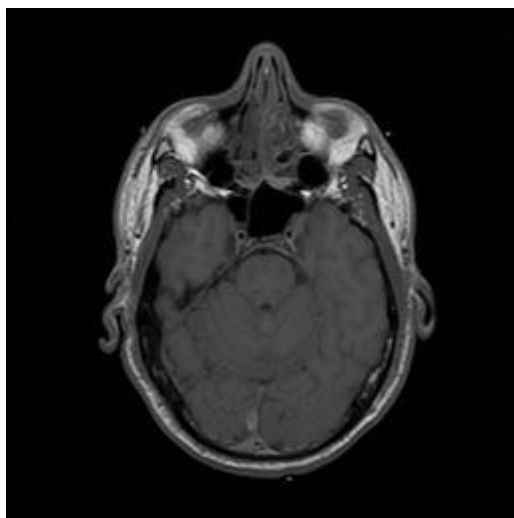


Figure 2: An MRI image that is bit-shifted and reduced to 8-bit grayscale for easy viewing (mvml120t.png).

Upon closer examination of the MRI head dataset, additional artifacts that did not appear to be part of the human body were found (Figure 3). It was determined that these artifacts were from two Tygon tubes (3 mm outer diameter) filled with contrast agents that continue along the outside of the body and crisscross at the head to help with image alignment [7]. These artifacts were manually edited out by creating “mask slices” (see Section 4) that are white for the area of interest (the body) and black for the background including the Tygon tubes.

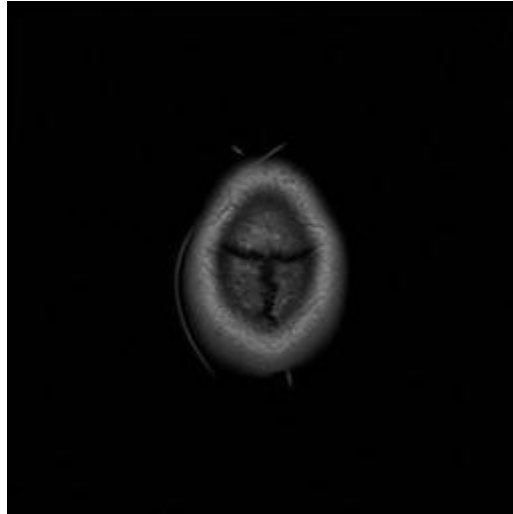


Figure 3: Tygon tube artifacts appear in mvm1020t.png at the top of the MRI dataset.

These masks were then extruded (see Section 7) to create a homogeneous model (Figure 4).

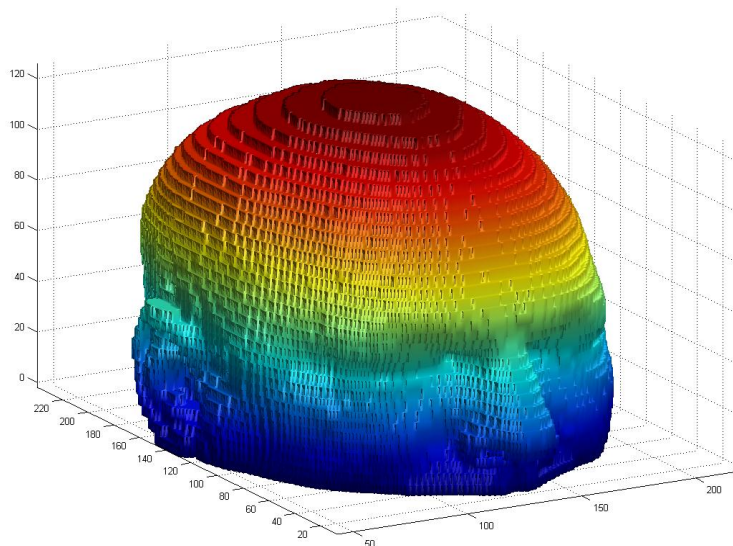


Figure 4: Homogeneous model created using mask slices for the MRI head dataset.

The resulting model resembled a human head; however, the head portion of the MRI dataset did not include the mouth or neck and stopped around the upper lip. Several difficulties arose when the same method was applied to the thorax subset of the MRI data. First, the head MRI images are in the transverse plane while the rest of the MRI images (including the thorax subset) are in

the coronal plane of the body (Figure 5). Second, the pixel and slice resolutions are different for each region (Table 1).

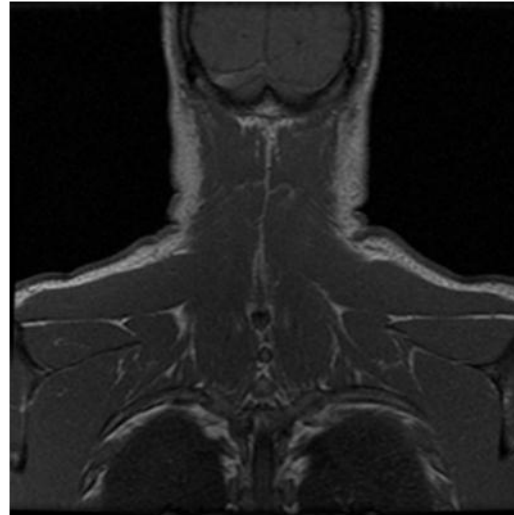
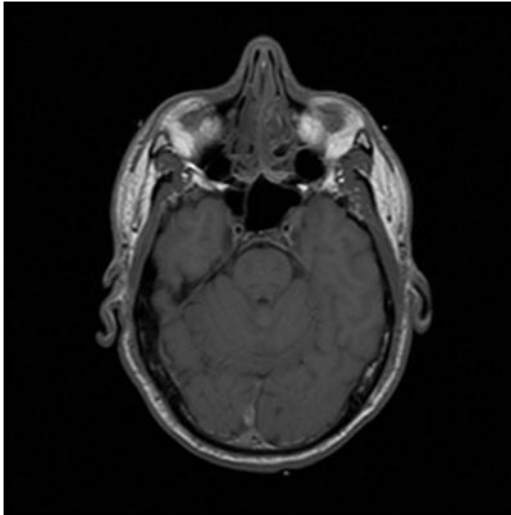
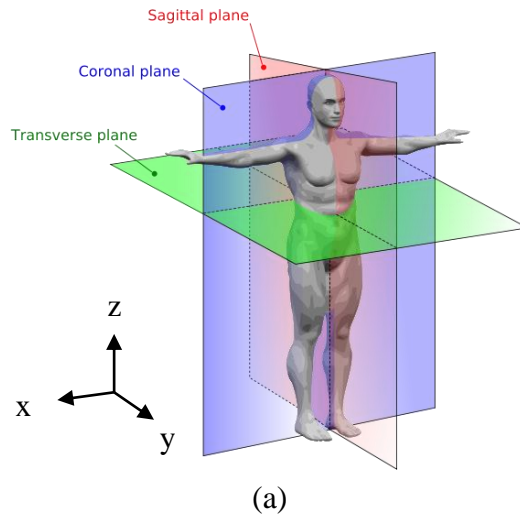


Figure 5: (a) Definition of anatomical planes [8]. (b) A transverse slice from the MRI head dataset (mvm1120t.png). (c) A coronal slice from the MRI thorax dataset (mvm3464t.png).

Table 1: Region Resolutions of MRI data set

Region	Head	Thorax	Remaining
Slice Plane	Transverse (x-y plane)	Coronal (x-z plane)	Coronal (x-z plane)
Slice Resolution	260 mm / 256 pixels	300 mm / 256 pixels	480 mm / 256 pixels
Distance between	4 mm (z direction)	4 mm (y direction)	4 mm (y direction)

Slices			
--------	--	--	--

Because of the mismatch in resolutions and orientations of the different parts of the body, the MRI dataset was found to be inappropriate for creating the AustinMan model.

2.2 CAT Dataset

All images in the CAT dataset are in the transverse plane; thus, the mismatch problem is not encountered and a consistent model can be developed. Again, the already converted 16-bit grayscale PNG images were shifted and reduced to 8-bit grayscale images for viewing. Unfortunately, portions of the body, most notably the arms are cropped in the images (Figure 6); thus, this dataset was also found to be inappropriate for creating the AustinMan model.

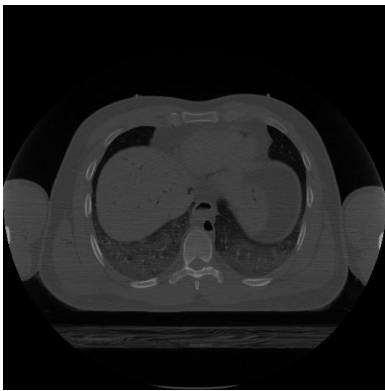


Figure 6: A CAT image with the cropped arms (cvm1481f.png).

2.3 Cross-Sectional Photograph Dataset

The cross-sectional photographs in the VHP dataset (Figure 7) are all in the transverse plane and the full body is visible in each image, which avoids the problems encountered for the MRI and CAT datasets. Therefore, this dataset was used to create the AustinMan model. Note that this dataset has its own difficulties: The images are of much higher resolution and are in color, which means that data sizes are larger and more complicated image processing operations are needed compared to the other datasets. Moreover, several adjustments were needed to the dataset as detailed next.

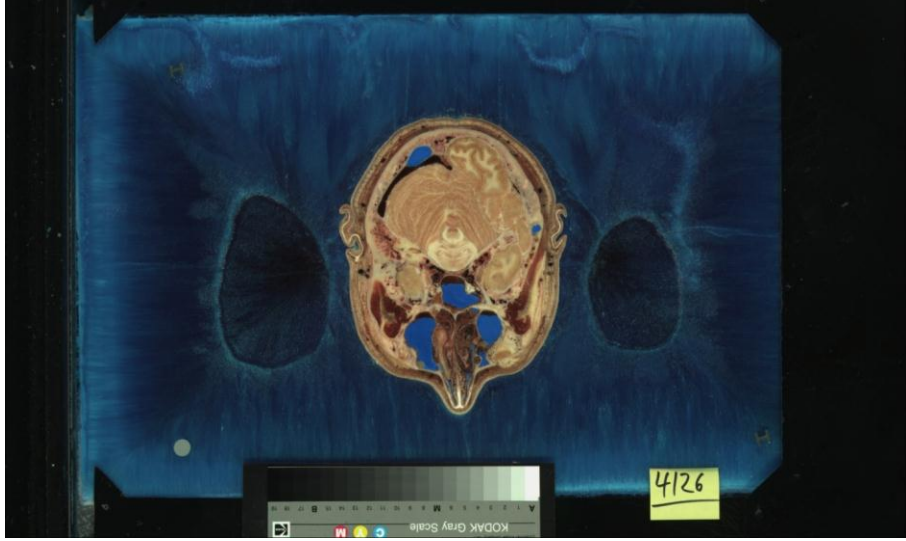


Figure 7: A cross-sectional image of the head (a_vm1126.png).

3 Adjustments to Cross-Sectional Images

The cross-sectional photographs in the VHP dataset have two types of problems that require adjusting the images before using them for constructing the AustinMan model. First, the images are not perfectly aligned (Figure 8). Two methods were used to determine the amount of the shift in pixels of each slice, relative to the slice above it. One method looked for a unique part of the slice, e.g., the vertebrae in Figure 8, and counted the number of pixels visually. The other method created temporary “mask slices” (see Section 4), and shifted the second mask one pixel at a time. It then performed an “exclusive-or” operation of each pixel in the first mask with the corresponding pixel in the second mask and counted the number of 1s; thus, the number of non-overlapping pixels was counted. This was performed for each pixel shift and the shift that minimized the count was identified. This method uses the assumption that the body grows symmetrically compared to the slice above. If this assumption was not true, the minimum would still center the masks on top of each other, but the shift in the body would not be corrected. This could be prevented by also using the first method to verify the shifts. Table 2 presents these shifts, which must be countered to align the images. Note that each shift is relative to the one above it; thus, every image below the shift must be shifted the same amount for alignment.

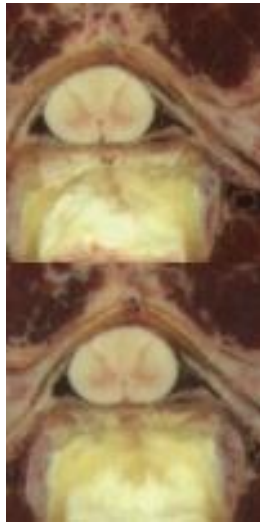


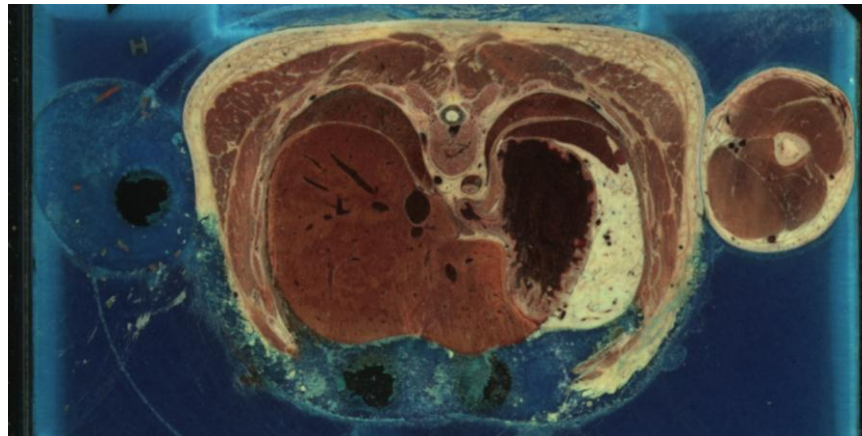
Figure 8: The same cropped portion of two image slices separated by 1 mm: a_vm1230.png (top) and a_vm1231.png (bottom). The images indicate a horizontal shift of 9 pixels, i.e., the bottom image should be shifted 9 pixels to the left for alignment.

Second, some images in the cross-sectional photograph dataset are corrupted because the body was cut into fourths prior to photographing so that it was easier to handle and would fit the available tools for processing. Cutting the body into fourths caused saw kerfs, which appear as extra blue noise on the images (a_vm1505 – a_vm1507; a_vm2017 – a_vm2023; a_vm2424 – a_vm2432) [7] (Figure 9). This noise must be removed to retain continuity among the image slices and to retain the resolution. In this thesis, this part of the body was not reached so this problem was not addressed. The noise can potentially be overcome by careful visual inspection; when this is not possible (e.g., the left side of Figure 9a), the vertical-resolution of the model

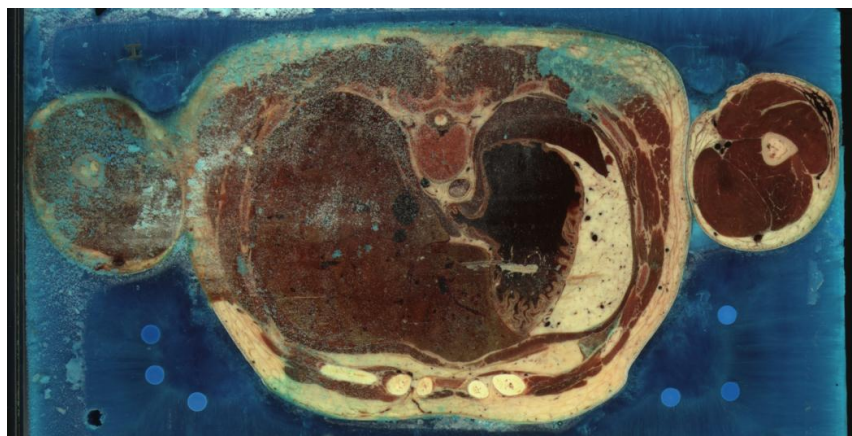
may be reduced for this part of the model by copying the closest known image slice to overwrite (parts of) the problem slice.

Table 2: Horizontal Shifts in the Cross-Sectional Images

Relative Pixel Shift	Cumulative Pixel Shifts
Slice 1231: 9	Slices 1231-1252: 9
Slice 1253: -11	Slices 1253-1360: -2
Slice 1361: 13	Slices 1361-1390: 11
Slice 1391: -13	Slices 1391-1416: -2
Slice 1417: 13	Slices 1417-1729: 11
Slice 1730: 6	Slices 1730-1790: 17
Slice 1791: 1	Slices 1791-1880: 18
Slice 1881: -2	Slices 1881-2546: 16
Slice 2547: 2	Slices 2547-2702: 18
Slice 2703: 2	Slices 2703-2878: 20



(a)



(b)

Figure 9: Saw kerf seen in slices (a) a_vm1505.png and (b) a_vm1506.png as a spread of the blue gelatin.

4 Mask Slices

After the adjustments described in Section 3 were completed, “mask slices” were created; these are black-and-white images with pixels set to one for the desired regions in the actual image and zero otherwise. The desired regions of an image slice can be isolated by computing an “and” operation with its mask slice; as a result, the pixels in the image that correspond to a zero in the mask are blacked out while the remaining pixels are unmodified and appear on a black background. Mask slices are a simple method of keeping track of which parts of the images are actually showing the body; they can be used to create smaller images, which reduce image processing times. They also present a simple test for image-processing of color images, as there is a significant color difference between the background and the body. To create mask slices, images were cropped, converted to and filtered in color space, and manually adjusted as detailed next¹. These mask slices can be downloaded from [1].

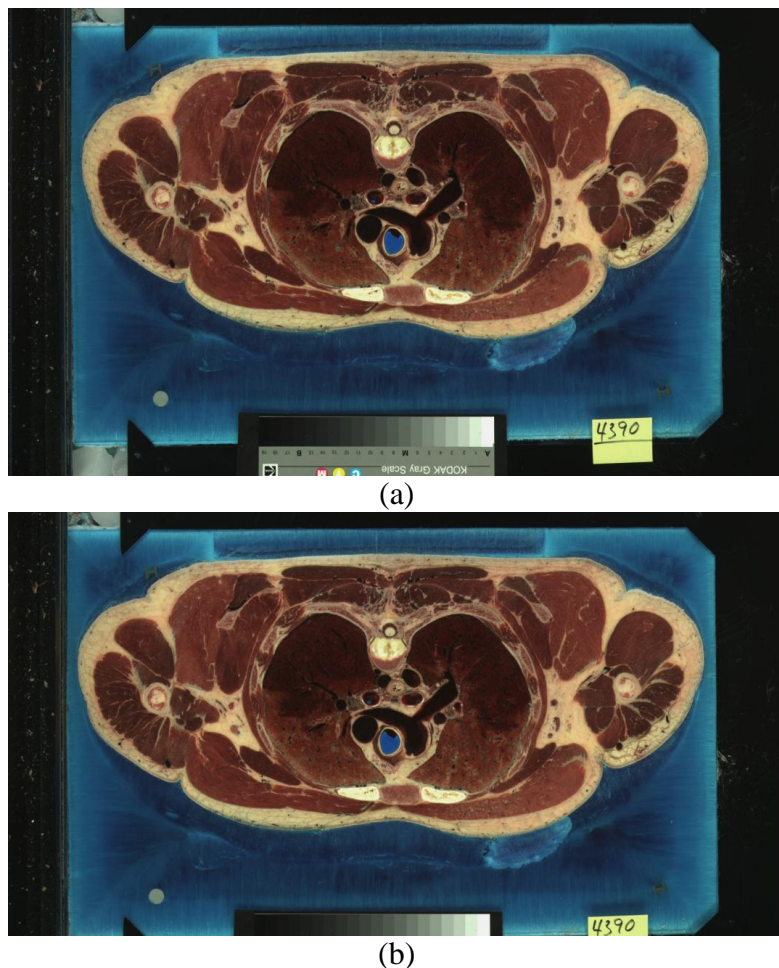


Figure 10: (a) Original image a_vm1390.png. (b) Cropped image a_vm1390.png.

¹ Mask slices can also be used to create homogeneous body models; however, such models are of limited use as they are larger than the actual body because the mask slices do not contain any cavities and are by design (slightly) larger than the actual body due to complications listed in Section 4.2.

4.1 Cropping

The images were cropped to reduce the file sizes and extraneous information, such as the black border, post-it note, and color bar. The cropped images were chosen by visual inspection to ensure that the complete specimen fit in each cropped image. The cropped images have an upper-left corner at pixel (0, 0) in the original image and are of size 2048×1120 pixels (see Figure 10a vs. Figure 10b) instead of 2048×1216 .

4.2 Color-Space Filtering

The images were then converted to the CIE 1976 $L^*a^*b^*$ color space. This color space was developed “to get a perceptually uniform color space that could be correlated with the visual appearance of colors” [9]. As a result, Euclidian distances in the $L^*a^*b^*$ color space are approximately proportional to perceptual changes [9,10]. To create mask slices, first two reference regions of pixels were selected (in the blue background and skin) and reference L^* , a^* , and b^* values were determined by finding the mean of the L^* , a^* , and b^* values of all pixels in these region. The resulting L^* , a^* , and b^* values are shown in Table 3. Then, the Euclidian distance from each pixel in the image to that of reference values for the skin and blue background regions were found in the $L^*a^*b^*$ color space; if the former distance was smaller, the pixel was assigned a white color in the mask, otherwise it was assigned black.

Table 3: The Color Space Values Used for Creating Masks

Region	L^*	a^*	b^*
Blue Background	107.0000	132.2143	128.5714
Skin	134.5263	152.2632	138.5789

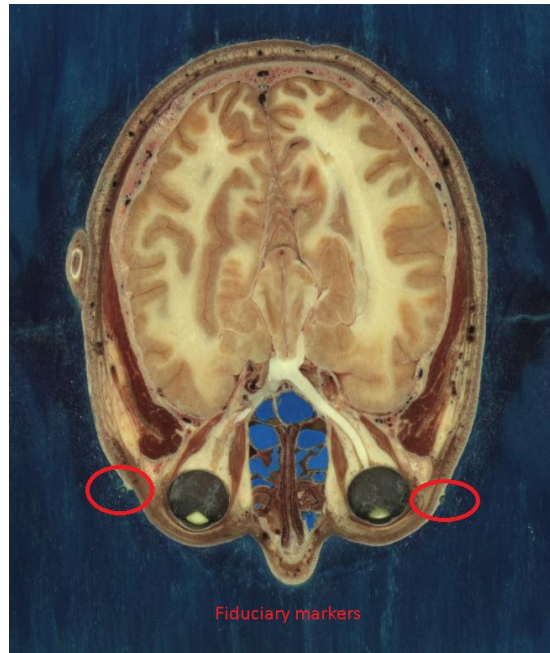


Figure 11: Highlighted fiduciary markers in slice a_vm1108.png.

Unfortunately, this operation cannot be performed independently for each pixel: Because the image slices often contain body cavities, the mask would incorrectly ‘black out’ pixels corresponding to these holes. To avoid this problem, holes were pre-identified and filled in using the MATLAB “imfill” function, which fills in holes using an algorithm based on morphological reconstruction [11,12]. This automated approach resulted in inaccurate masks because of several reasons related to the nature of the images and the method used to obtain them:

1. Two Tygon tubes (3-mm outer diameter) filled with contrast agents were attached to the body to serve as fiduciary markers between the different data sets [7]. The tubes, which run along the skin surface of the anterior aspect from the head down to each foot, are a different color than the blue background (Figures 3 and 11).
2. In some image slices, the body parts underneath the slice are partially visible through the dyed gelatin solution. This causes the gelatin solution’s color to be similar to skin color in places where the body expands in the slices underneath current slice, e.g., the shoulders are broader than the neck and are visible in the slices above them (Figure 12).

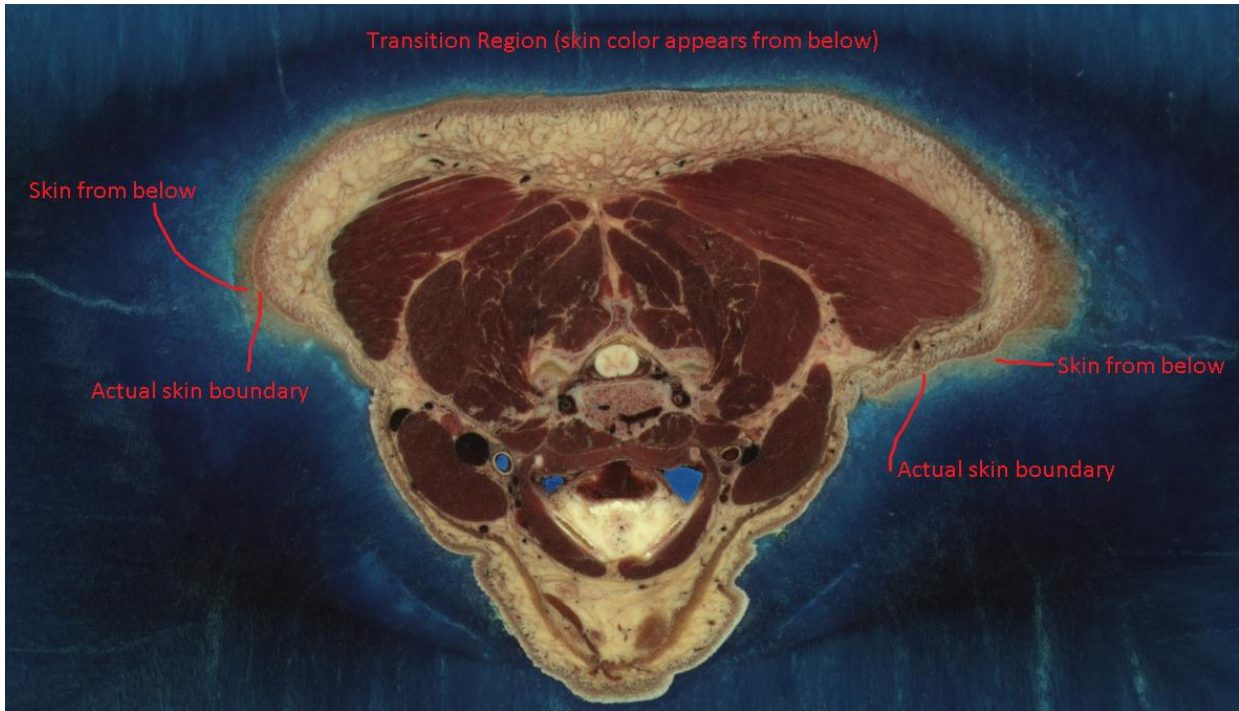


Figure 12: Partially visible skin from slices below in slice a_vm1242.png.

3. Some of the body cavities were filled with the dyed gelatin solution (such as the ear canal) while other cavities were filled with latex to prevent debris from collecting in the cavities during the preparation of the body. Both the dyed gelatin solution and the latex are blue in the image slices [7]. This requires a careful labeling of the boundaries of these cavities to avoid filtering out important body features (see Section 5). This labeling step can be delayed by filling in the cavities in the mask slices, e.g., see the ear canals in Figure 13.

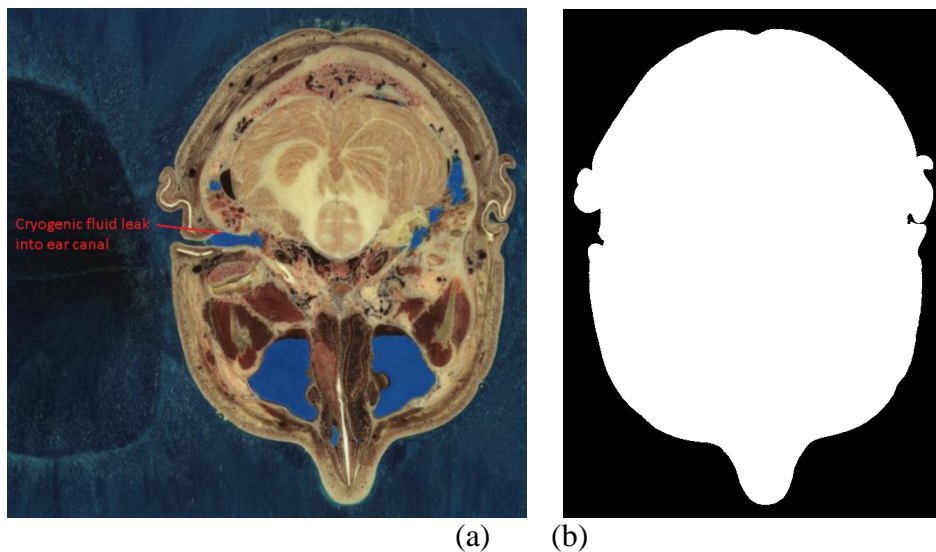


Figure 13: (a) Blue fluid leaked into the ear canal in slice a_vm135.png. (b) The mask for slice a_vm135.png with the ear canals closed in.

4. Other odd protrusions from the skin also appeared in some of the images where the background was lighter in color due to the intentionally conservative choice of the color space values when creating the masks (Table 3). If the odd protrusions were not fixed in the masks, they should be labeled as air while labeling the tissues (Section 5). These protrusions can be seen in the uncorrected mask in Figure 14 and are removed (using 1. below) in the final mask in Figure 14, where both images are zoomed in to show the detail.

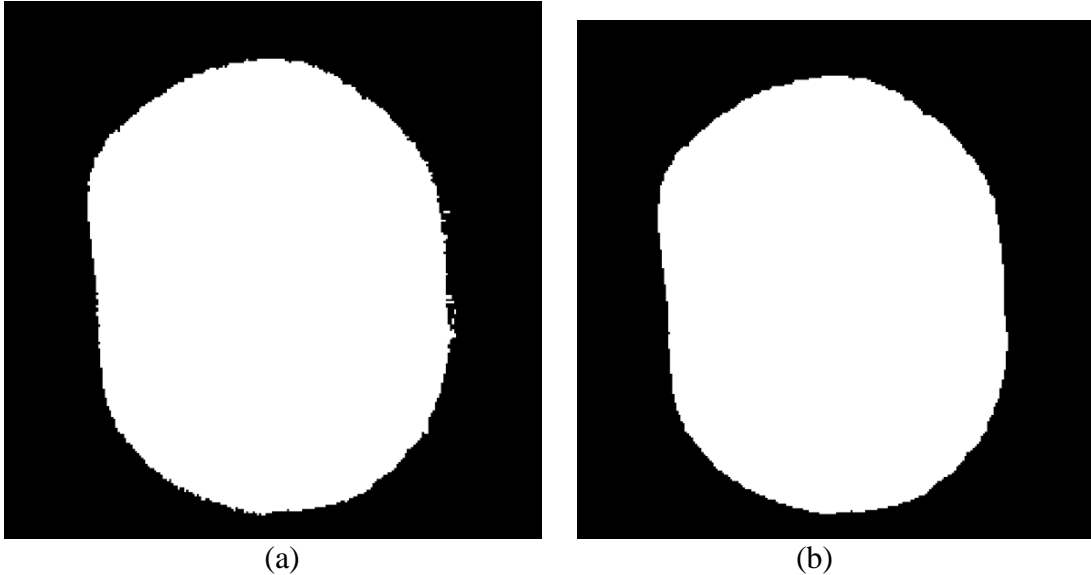


Figure 14: (a) The uncorrected mask for slice a_vm1009.png. (b) The corrected mask after dilation and erosion. Note the numerous protrusions in (a) compared to the smoothed mask of slice a_vm1009.png.

5. Some of the image slices are severely degraded or missing because of the saw kerf that occurred during the preparation of the body for the cross-sectional slices [7]; see Section 3.

These complications were addressed by editing the mask slices as follows:

1. The bumps in the images from the fiduciary markers and the odd protrusions from the skin were removed by dilating and then eroding (using a disk-shaped structuring element with a radius of 5 pixels) the white regions [13].
2. The resulting black-and-white masks were individually reviewed for any discrepancies. These discrepancies were corrected by displaying the outline of the initial mask on top of the original image in the background, manually selecting problem pixels, and assigning them to either the foreground or background.
3. The degraded or missing image slices were replaced with the closest valid slice. In effect, the closest valid slices were extruded further and the z -resolution of the model decreased at slices 1505-1507, 2016-2022, and 2424-2435.

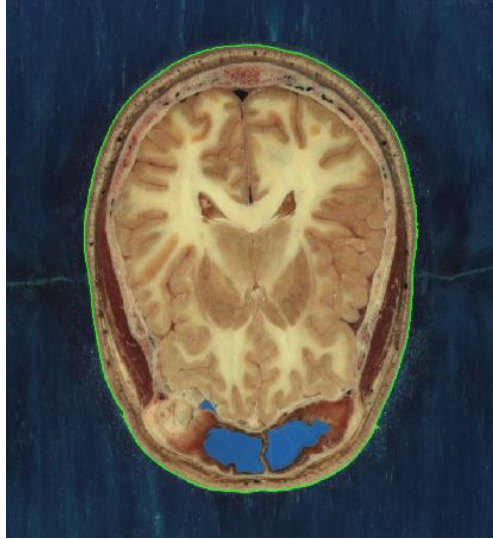


Figure 15: Outline of the mask (green) on top of slice a_vm1090.png.

4.3 Manual Adjustments

Occasionally, masks would need to be manually edited; e.g., for the auditory canal that often filled with the blue gelatin and went inside the head (Figure 13). For each manual edit, an outline of the mask was shown on top of the cropped image and several adjustment options were considered. Regions were added or omitted from the original mask as necessary (similar to using the lasso tool in most image editing programs) on the outlined image (Figure 15) and any holes in the final mask would be filled; e.g., filling in an auditory canal only required sealing off the outermost part, which leaves a hole on the inside that would later be filled. Some images at the top and bottom of the model (e.g., slice 1002) did not contain any desired regions, in which case a completely black mask was created.

5 Image Segmentation and Tissue Labeling

Image segmentation and the subsequent tissue labeling are significantly more challenging than creating a mask. Initially, the color-space filtering technique in Section 4 (sometimes called a ‘thresholding’ technique [14]) was applied to the segmentation problem, but did not yield satisfactory results due to the similarity in color of some of the tissues and the inherent variation of color within tissues. Over the course of the project, it became apparent that a careful balance between automated algorithms and manual user input was required to obtain an accurate model within a reasonable amount of time. As a result, a variety of image processing methods were utilized; these methods were chosen to balance automation with real-time and interactive manual processing to utilize the feedback from a human in the loop. These methods were chosen based on two different strategies for processing the images: (i) independent processing strategy and (ii) prediction-correction strategy. These strategies are detailed and contrasted next.

Note that the VHP images were prepared for the below processing steps in three stages. First, each image slice was masked (using ‘and’ operation with its mask slice, see Section 4) to remove the blue background; then, the color value of each pixel of each image was converted to the $L^*a^*b^*$ color space and sent through a 2-D Wiener filter to remove noise from each channel. The 2-D Wiener filter is an adaptive noise-removal filter that preserves the edges of objects in the image better than a linear filter [15]. The noise comes from each tissue being represented by a range of colors and from noise or debris that may have been present on the slice when picture was taken. By smoothing out the noise, the algorithms presented in this section yielded better results because the color was more uniform within each region. Finally, each image was masked again to remove any blurring on the edges of the original image slice from the filter.

5.1 Independent Processing

In this strategy, each slice of the model was constructed independently and went through the same image segmentation and tissue labeling process. First, the images were pre-processed without any user interaction. The pre-processing script, which took around two hours per slice, performed three automated steps: (i) Region growing, (ii) Region stitching, and (iii) Boundary competition. Next, the results from the pre-processing script were loaded into BRISKit (Section 6) where a user assigned each resulting region a tissue and manually adjusted the boundaries of the regions as needed. Slices 1002,1018,...,1354 were processed using this approach, As more slices were processed with the strategy; however, it became clear that the user spent significant amount of time correcting boundaries. Thus, the pre-processing steps were modified as follows: (i) Region growing, (ii) Manual region stitching, (iii) Boundary competition for a fixed number of iterations. These steps are detailed next.

5.1.1 Algorithm 1: Region Growing

The region growing algorithm uses ‘seed pixels’ as a starting point to identify the regions in an image [14]; here, the seed pixels are chosen by uniformly sampling the image with a spacing of

50 pixels in the x - and y -directions. Each seed pixel is assigned a unique ‘region ID’ $r \in 1, \dots, R$ and the pixels that are not seed pixels are considered “unclaimed” and assigned to the ‘null region’ $r=0$; for the VHP dataset, this results in $1 < R < 375$ regions for each image. Each region $r=1, \dots, R$ contains only 1 pixel initially and the ‘mean color’ of the region m_{color}^r is set to the color of that pixel, i.e., $m_{color}^r = \{L^*, a^*, b^*\}$ of the seed pixel. The algorithm then loops over each region $r=1, \dots, R$ and examines all unclaimed pixels that share one side with region r . If the color of a pixel is similar enough to the mean color of region r —determined by comparing their Euclidian distance in the $L^*a^*b^*$ color space to a threshold value—then the pixel is assigned the ID of the region; otherwise, the pixel is left unclaimed and its region ID remains zero. After the algorithm has iterated through all of the regions, it updates the mean colors by finding the average value of the colors of all the pixels in each region. Then, the process is repeated until all of the regions have converged, i.e., the regions do not grow and there are no changes from one iteration to next [14]. Since only unclaimed pixels are compared, the regions are not allowed to grow into each other or reclaim pixels already belonging to another region (Figure 16).

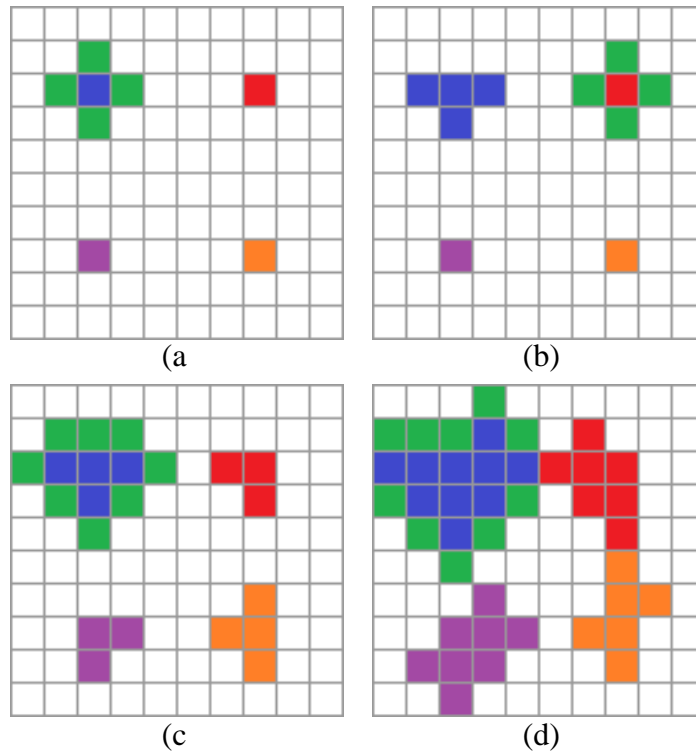


Figure 16: The region growing algorithm with 4 seeds (blue, red, magenta, orange). (a) Compare unclaimed (green) pixels adjacent to the boundary of region 1 ($m_{color}^1 = \text{blue}$). Add pixels whose color is similar to m_{color}^1 . (b) Perform the same operation for regions 2 to 4. (c) Second iteration: Update mean colors, evaluate the pixels adjacent to the new boundaries (green pixels), and add pixels whose color is similar to the new mean color of each region. (d) Regions are not allowed to grow into each other (notice the lack of green pixels at the boundary of regions 1 and 2).

There are several shortcomings of the above region growing algorithm: (i) There can be multiple seed pixels in the same tissue, as shown for the white matter in Figure 17, which introduces unnecessary boundaries. (ii) It results in a first-come, first-served boundary approximation with a bias toward the region whose seed is closest to the actual boundary. (iii) Some pixels remain unclaimed by their surrounding regions, e.g., as shown by red circles in Figure 17, because their color is significantly different from the regions around them and because there was not a seed pixel of similar color in the region. (iv) The boundaries are sensitive to the choice of the threshold value, which was experimentally chosen after reviewing the results obtained from different threshold values; typical threshold values were in the range 750 to 1500. The first three shortcomings can be ameliorated as detailed next; the last was an inherent problem with the region growing algorithm, which could potentially be overcome using a more adaptive/localized algorithm to compare pixels as opposed to comparing them to the global mean color.

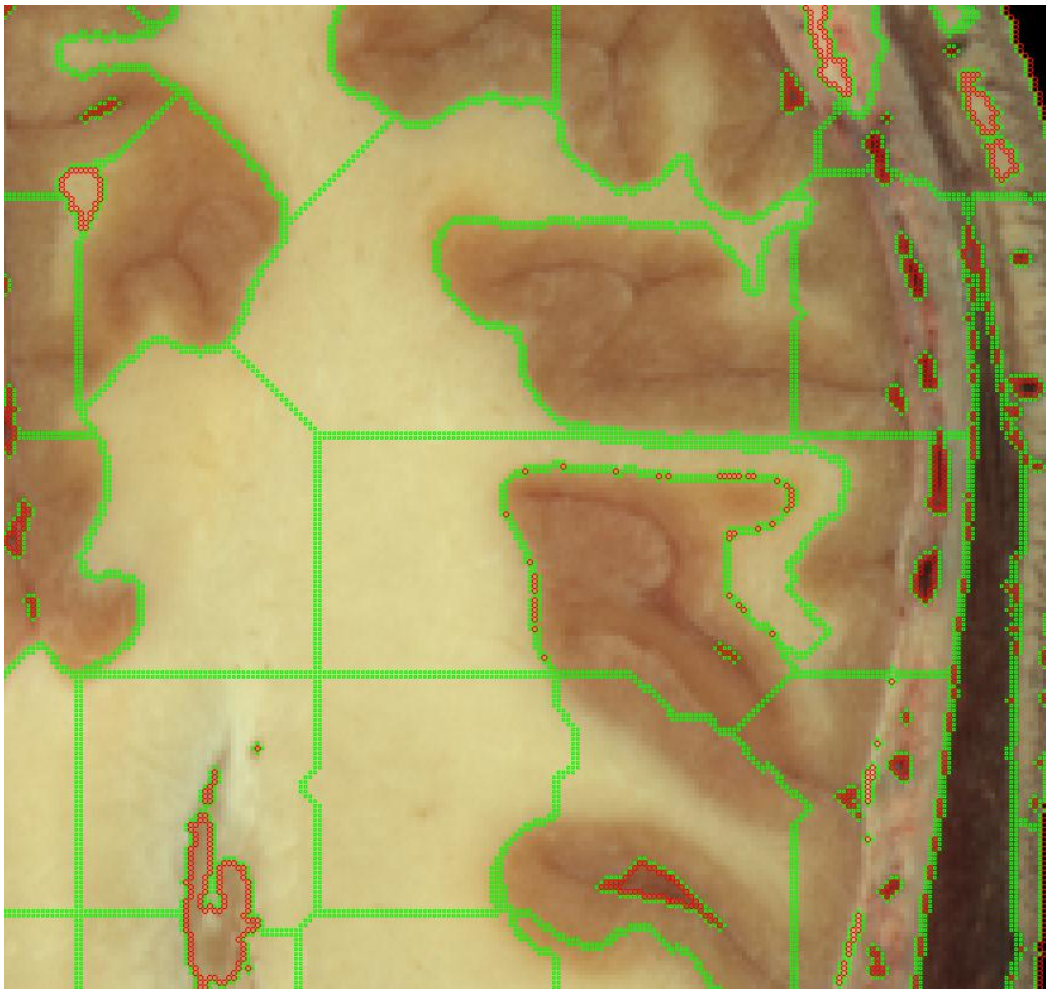
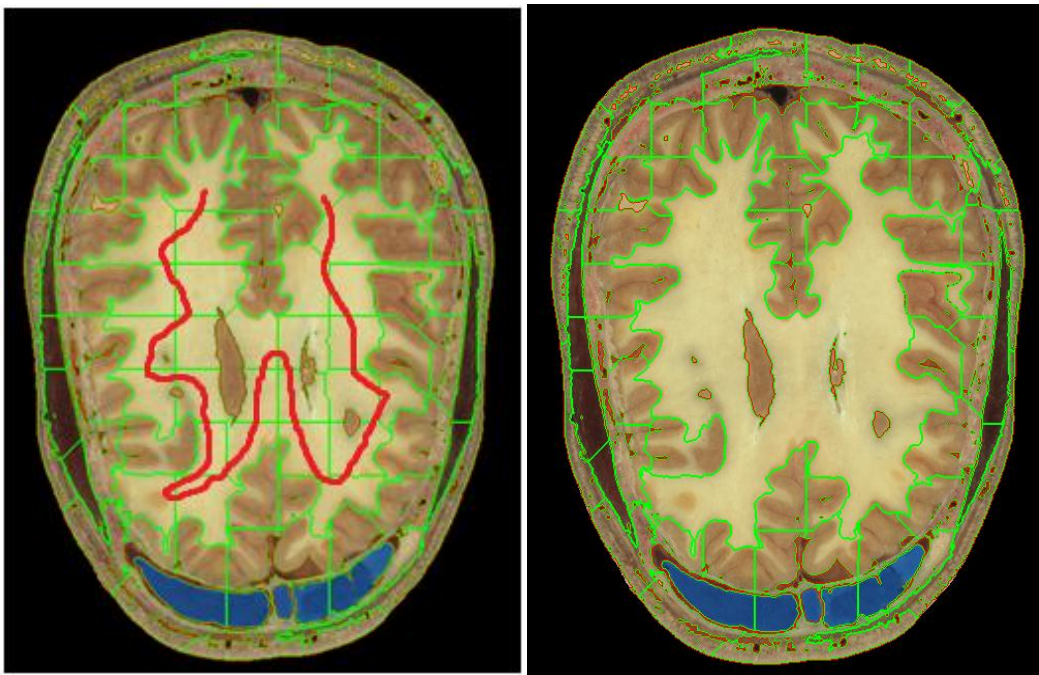


Figure 17: Converged result of the region growing algorithm for image slice a_vm1070.png. Green pixels are at the boundaries of each region; red pixels are at the boundaries of the null region.

5.1.2 Algorithm 2: Manual Region Stitching

As seen for the white matter in Figure 17, some of the regions can be combined to form a larger tissue. The region stitching algorithm evaluates regions that share a common boundary and determines if they are part of the same tissue by comparing their mean colors. If the Euclidian distance between the two regions' mean colors in the $L^*a^*b^*$ color space is below a threshold, then the regions are stitched by assigning them the same region ID. The algorithm is similar to the region growing algorithm except that it compares and adds regions to regions as opposed to pixels to regions. Like the region growing algorithm, it also continues iterating, updating the mean colors of regions in between iterations, until there are no changes. This stitching algorithm is also sensitive to the threshold value; moreover, it can stitch two regions that have similar mean colors but are in fact different tissues; this is a more severe problem than for the region growing case because after several iterations the regions can grow very large and their mean colors can be very different from the colors at their boundaries. To avoid this problem, a manual stitching option was implemented in BRISKit: To manually connect the regions, the user selects the *Connect* mode in BRISKit (Section 6) and then draws a line through the regions to be stitched as shown in Figure 18. This allows the user to employ their anatomical knowledge to create a more accurate estimate of the actual boundaries before the next steps.



(a) (b)
Figure 18: Manually stitching the regions corresponding to white matter in image slice a_vm1070.png. (a) The user drawn line. (b) The stitched regions.

5.1.3 Algorithm 3: Boundary Competition for a Fixed Number of Iterations

The boundary competition algorithm was implemented to counter the first-come, first-served boundary bias of the region growing algorithm and to improve the accuracy of boundaries. While the region growing algorithm does not allow regions to grow into one another, the boundary-competition algorithm does. The algorithm starts after the region growing and stitching algorithms are completed and each pixel is assigned to a region. In the first iteration, the algorithm examines pixels that are adjacent to the boundary of each region and do not belong to the null region: Each neighboring pixel's color is compared to the mean color of the region it belongs to and that of the current region (Figure 19); the pixel is assigned to the region that is closer to it in mean color. Note that pixels that belong to the null region are not modified in this process. The algorithm performs the same evaluation for every region to complete the first iteration. After the first iteration is completed, the mean color assigned to each region is updated and the process is repeated until there are no changes. Toy examples can easily be constructed to demonstrate that convergence is not guaranteed for this algorithm and the pixels can flip-flop ad infinitum between different regions as the mean colors change in between the iterations. Indeed, when used for the slices in the VHP dataset, the boundary competition algorithm gave mixed results: It did correct the first-come, first-served bias; however, it grew spurious, thin, and elongated regions that kept expanding a few pixels over many iterations; as a result, the competition took many iterations for convergence and several hours of processing time. To avoid these problems, the maximum number of iterations in the algorithm was limited to only 5 in BRISKit; i.e., the algorithm was terminated before convergence.

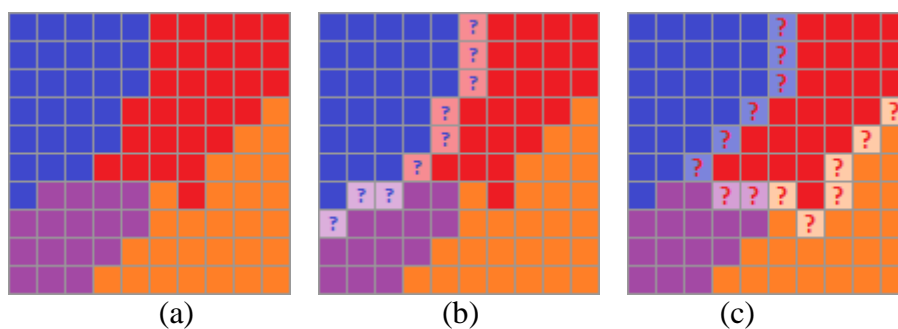


Figure 19: Boundary competition algorithm. (a) The input based on the result of region growing and stitching algorithms. (b) First step of first iteration: Evaluate pixels at the boundary of region 1 (mean color m_{color}^1 : blue). The pixels that are examined are shown with a lighter color and a blue question mark. The colors of these pixels are compared to m_{color}^1 and the mean color of their current region. (c) Second step of first iteration: Evaluate pixels at the boundary of region 2 (mean color: red). The mean colors are updated only at the end of the first iteration.

5.1.4 Manual Adjustments

For some regions, none of the methods in this section produced accurate results, in which case the accuracy of the boundaries can be improved manually. This was implemented in BRISKit and appears in many of the user modes, such as *ID Materials* and *Review*, because it is the most

elementary tool that provides the greatest flexibility for the user to correct discrepancies. It is important to note that hand-drawing boundaries one pixel at a time can be extremely time consuming and error prone; it should be used as a last resort and only with a good initial approximation to the actual boundaries. Manual adjustments also are used to improve the null region boundaries, which are not modified by the stitching and boundary competition algorithms: In the BRISKit *Review* mode, the user can choose to only adjust the null region without changing any labeled regions (see Section 6).

5.1.5 Tissue Labeling

Finally, the regions found from the above algorithms are manually labeled with tissue names. This was performed in BRISKit as follows: The user loads the region information then enters the *ID Materials* mode of BRISKit where the program outlines unlabeled regions individually and allows the user to select the tissue that corresponds to the region (Figure 20). The region ID is then replaced with the corresponding tissue ID, indicating that the region is part of that tissue.

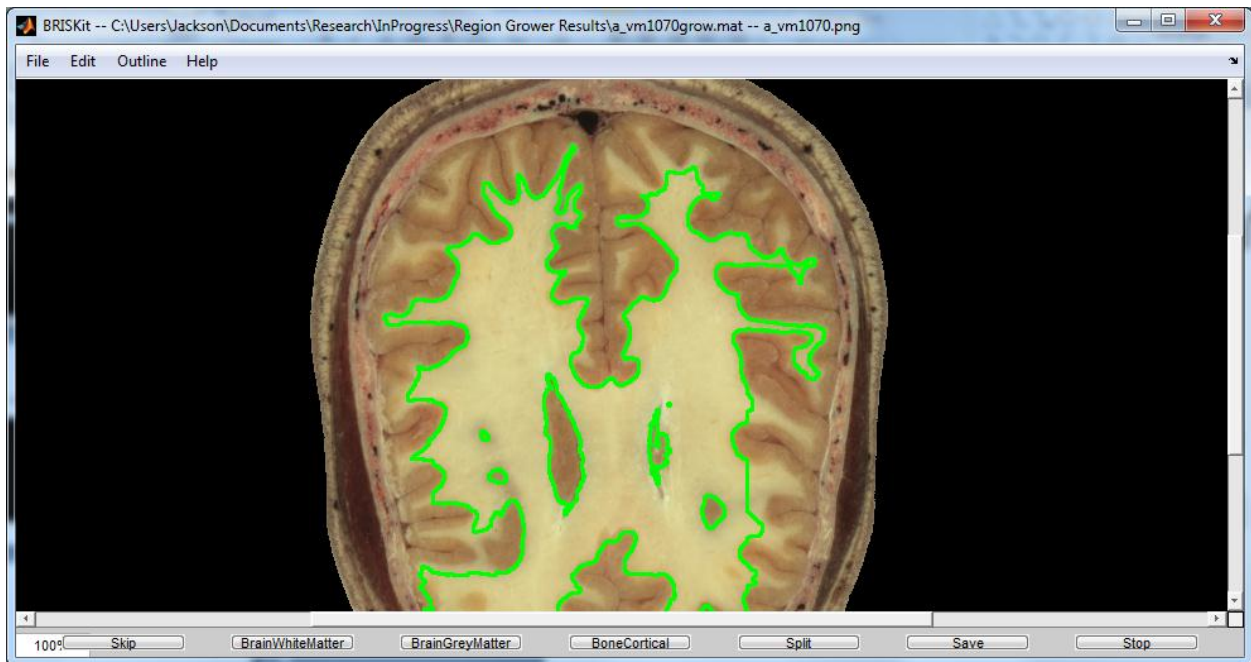
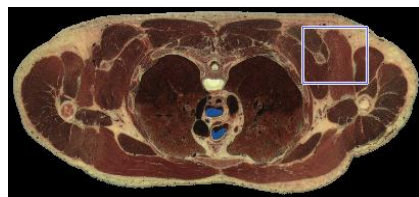


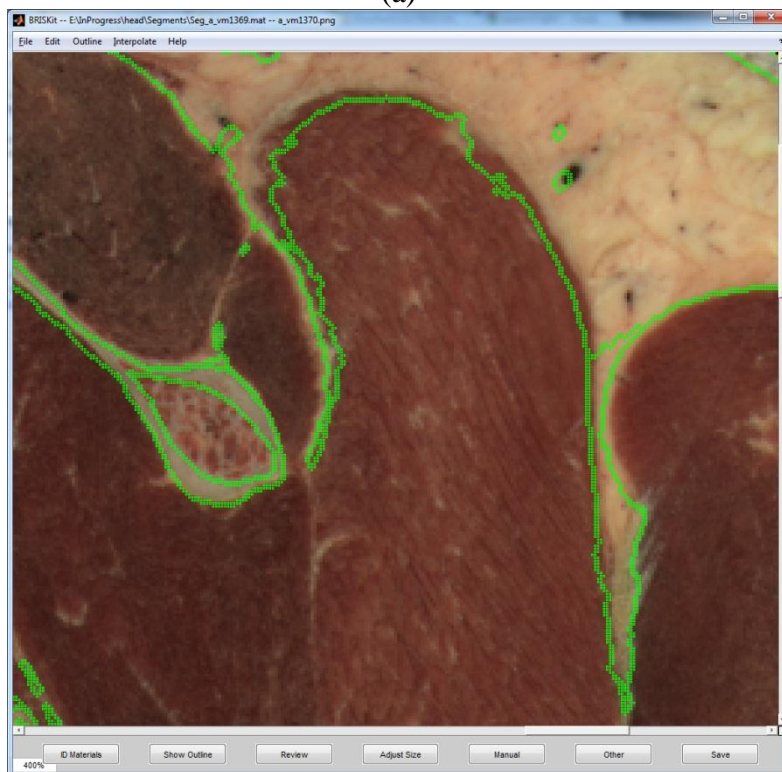
Figure 20: The *ID Materials* mode of BRISKit for slice a_vm1070.png. Here, the outlined region is currently unidentified. The user can assign the region a tissue label by pressing the correct tissue button at the bottom; here, BrainWhiteMatter, BrianGreyMatter, or BoneCortical. The user can also skip labeling the region, split the region into smaller regions, or exit the *ID Materials* mode.

5.2 Prediction-Correction

In this strategy, once all the image processing steps for a slice are completed and tissues are labeled (each pixel is labeled), this information is used to estimate the tissues in the next slice, i.e., the pixel information is copied on to the next slice. Of course, some adjustments are needed to account for the changes between the slices (Figure 21). This strategy attempts to utilize the information on a nearby slice and the effort that went into segmentation and labeling of the pixels in that slice to speed up the image processing. To be effective, this prediction-correction strategy requires that the regions in a given slice are similar enough to the ones in the next one.



(a)



(b)

Figure 21: The initial prediction (for a_vm1370.png without any adjustments) for the regions using the data from the previous slice (a_vm1369.png). (a) The entire image and (b) an enlarged portion. Note the minute changes in the location of the muscle-fat boundary and the shifts in the blood vessel locations at the top right.

When implementing this strategy, the region growing algorithm, the stitching algorithm, and the tissue labeling step are not required because the regions are already labeled in the previously completed slice and that information is passed along to the new slice. The main effort is then “to

correct” the images by adjusting the boundaries (typically done in the *Review* mode in BRISKit (Section 6)). To this end, two algorithms are implemented: (i) Boundary competition with user-selected mean colors and (ii) Adjust image size.

5.2.1 Algorithm 4: Boundary Competition with User-Selected Mean Colors

As noted before, there can be large variations in color across a tissue and the mean color can be a poor indicator of the color at the boundaries of a region; this causes the mean-color based automated algorithms to label the boundaries of the regions poorly. A new version of the competition algorithm that uses the mean colors of sub-regions selected by the user (rather than the entire region) was developed to ameliorate this problem. This boundary competition algorithm with user-selected mean colors was designed to give the user more control over the competition algorithm and to increase the boundary accuracy by choosing more appropriate colors.

This algorithm was implemented as follows in BRISKit: The user specifies/limits the area in which the competition can take place, i.e., the competition is localized to a portion of the image. The user also specifies the regions that will compete and the colors representing those regions by manually drawing a sub-region that is representative of the desired region color (Figure 22). Similar to algorithm 3, the pixels at the boundaries of the regions are evaluated and re-assigned iteratively; this competition continues among the selected regions until there are no changes, i.e., until convergence. It should be noted that the mean color is not updated from iteration to iteration and the algorithm only uses the mean color of the sub-regions specified by the user. Note that the competition does not affect regions that do not have a representative sub-region selected inside them (these regions are ‘frozen’). This algorithm proved to be an effective method of obtaining accurate boundaries although it requires significant user intervention.



Figure 22: Sample boundary competition with user-selected mean colors for a white and gray matter boundary (a_vm1070.png). (a) The required user inputs: The outermost blue loop defines/limits the area in which competition will take place. The smaller loops indicate the sub-regions selected to represent the colors for the two tissues. (b) The result of the competition. Notice the improved boundaries in the competition area and the unchanged boundaries outside of it.

5.2.2 Algorithm 5: Boundary Competition among User-Selected Regions

In the prediction-correction strategy, essentially all of the boundaries must be corrected. Doing this correction using boundary competition with user-selected mean colors (algorithm 4) would be too tedious as there are many regions in each image (30-300 regions depending on the complexity and location of the image). It was observed that boundary competition with automated (global) mean colors (algorithm 3) worked well for regions that had very different colors (such as muscle compared to fat and blood vessels compared to dense connective tissue). Thus, algorithm 3 was modified so that the user selected a subset of the regions (typically in pairs, e.g., muscle-fat, blood vessel-connective tissue, etc.); the competition was performed only among these regions; and only their boundaries were modified (rather than all boundaries of all regions) using the mean color of the entire regions.

5.2.3 Algorithm 6: Adjust Size

The adjust size algorithm was developed to account for the area of the body changing between slices. The area of the body in each image slice is found by using its mask slices. When the area expands, there are new unclaimed pixels that appear as a result. To account for these new pixels, the tissue at the existing boundary is simply extended outward until the new boundary is reached. This typically results in the new pixels being labeled as skin or air. When the area contracts, there are pixels with a tissue that are supposed to be free space and belong in the background. To remove these pixels, the existing regions are combined with the new mask using an 'and' operation; this blacks out any labeled pixels that are supposed to be in the background of the new mask and labels them as free space.

5.2.4 Final Algorithm

These algorithms for the prediction-correction strategy were implemented as follows: A preprocessing script was developed to allow the user to easily improve the prediction. The script first runs the adjust size algorithm (algorithm 6) and then competes user-specified pairs of regions using the modified boundary competition algorithm (algorithm 5); these steps reduce the amount of adjustments that require human input. Then, the user applies algorithm 4 as well as any manual adjustments to improve the boundaries.

5.3 Independent Processing vs. Prediction-Correction

On the one hand, the independent processing strategy is an inherently parallel approach as each slice of the model is labeled independently of the others. This allows for multiple users to work on the model at once with relative ease (each user must work on a separate slice). This strategy was also beneficial for creating initial, coarser models where the slices change a lot because slices were skipped for 2-D coarsening (Section 7). As a result, the independent processing strategy was used to create the AustinMan v1.0 $16 \times 16 \times 16 \text{mm}^3$, $8 \times 8 \times 8 \text{mm}^3$, and $4 \times 4 \times 4 \text{mm}^3$ models.

On the other hand, the prediction-correction strategy is a sequential process because a previous slice must be completed to use it as a prediction for the next slice. Coordination is required for multiple users to work on the model simultaneously. Two approaches have been tried: Users can work on different parts of the body (e.g., one user works on the chest working down while another user starts at the knees and works up); or a group of users can work in the same region, splitting up the tissues to label in each slice, which requires a great deal of coordination to ensure that the users do not overwrite each other's work by editing the same slice at the same time. Nevertheless, there are several advantages of prediction-correction over independent processing: First, a significant amount of time is saved by no longer having to label tissues. The tissues labeled are also more consistently labeled and continuous across slices because the tissues labeled in the completed slice appear in the new one for the user to correct. Thus, all of the blood vessels that were labeled in the completed slice will continue to propagate downward as the slices are completed and used as the prediction for the next slice. Also, users acquire a more specialized knowledge of anatomy pertaining to the slices that they are currently working on which yields more accurate results. Lastly, there is no overhead required for the pre-processing script, which is only a minor advantage.

The prediction-correction strategy was used for completing the slices between the slices from the 4-mm dataset to create the $1 \times 1 \times 1 \text{mm}^3$ AustinMan v1.1 model. Due to the significant time savings by already having labeled tissues and the increased accuracy that results from this strategy, it is currently used as the primary method for completing the model.

6 Boundary and Region Identification Software Kit (BRISKit)

BRISKit is a MATLAB based graphical user interface for overlaying labeled regions on top of the original image, for modifying boundaries of these regions, and for labeling them with tissue names. It allows users to label tissues, review boundaries, and redraw them as necessary. The interface has four modes: *Home*, *Connect*, *ID Materials*, and *Review*.

6.1 Home Mode

When users first open BRISKit, they enter the *Home* mode where the buttons on the bottom are designed to let users perform basic tasks (such as Adjust Size and Manual) or easily enter another (more specialized) mode (Figure 23). The *Other* button at the bottom contains several additional methods that do not belong in the other modes. These include: (i) Fill in holes which looks for single pixels that are completely surrounded (4 sides and 4 corners) by another (single) region/tissue and changes the single pixel to be part of the region surrounding it. (ii) Assign tissue X to Y which allows users to reassign all of a labeled tissue to another type of tissue, which is beneficial for correcting wrong assumptions. (iii) Change display which allows users to change the background to display the intensity of the individual color channels (L^* , a^* , b^* , red, green, or blue) or the grayscale image.

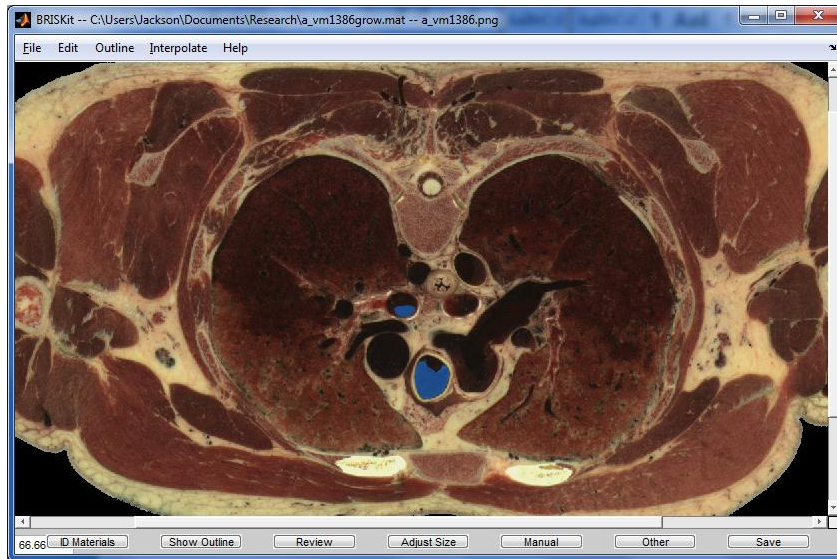


Figure 23: The BRISKit *Home* mode (a_vm1386.png).

6.2 Connect Mode

The *Connect* mode (Figure 24) displays the results from the Region Growing algorithm (algorithm 1, Section 5) for users to stitch the regions (algorithm 2) and perform both boundary competition with global mean colors (algorithm 3) and user-selected colors (algorithm 4). This mode was designed to improve the independent processing strategy by including more input from users.

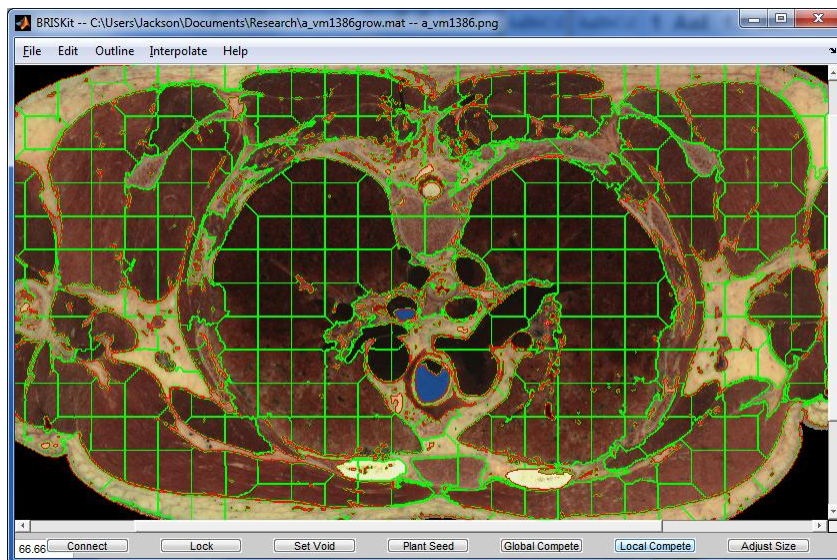


Figure 24: The BRISKit *Connect* mode (a_vm1386.png).

6.3 ID Materials Mode

The *ID Materials* mode asks users for three tissues to label (Figure 25). It displays these tissues as buttons at the bottom of the window and then iterates through the unidentified regions by overlaying them one at a time. Users can then select to label the overlaid region with a tissue by pressing one of the three tissue buttons, or skip to the next unidentified region. This mode also gives users the option of splitting a region into two different regions by pressing the *Split* button and manually encircling part of the region to split from the other part. The split region will later appear in the *ID Materials* mode as a separate, unidentified region.

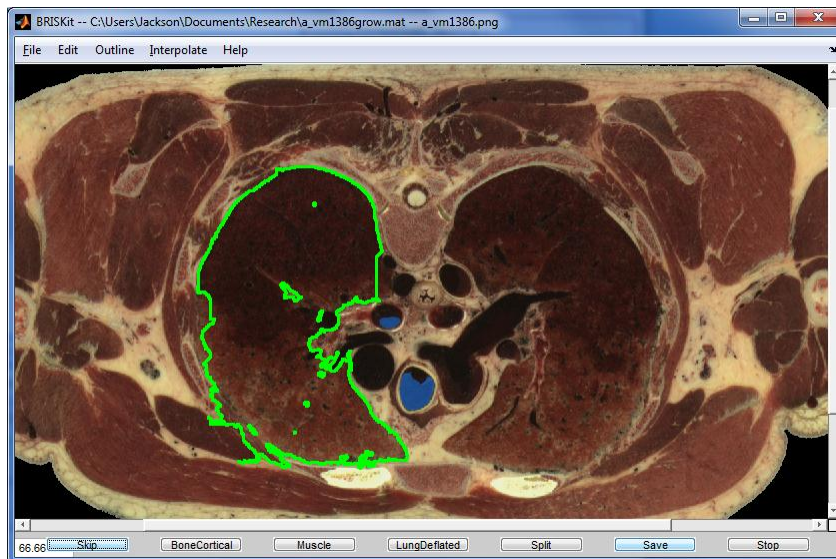


Figure 25: The BRISKit *ID Materials* mode (a_vm1386.png). The unidentified region will be labeled as LungDeflated using the fourth button at the bottom of the screen.

6.4 Review Mode

The *Review* mode is used to iterate through the tissues labeled in the slice, showing an overlay of each labeled region on top of the original picture (Figure 26). Users can then adjust the region's boundaries using either the boundary competition method (algorithm 5, Section 5) or by manually redrawing the boundaries. Also, when BRISKit is in the *Review* mode, users have the option of setting the exclusive mode for manually adjusting regions. If the exclusive mode is enabled, then BRISKit will take different actions depending on which tissue is selected for the manually drawn regions. If the selected tissue matches the tissue currently outlined by the *Review* mode, then BRISKit will make the drawn regions the selected tissue, allowing the user to manually add to the regions. If the selected tissue does not match the tissue currently outlined, then BRISKit will take the intersection of the drawn regions with the outlined tissue's regions and only replace intersected regions with the new, selected tissue. This allows users to be less precise when correcting a region because they cannot make changes outside of the currently outlined region. If the exclusive mode is not enabled, BRISKit will assign the drawn regions the selected tissue, regardless of which tissue is selected or where the drawn regions are.

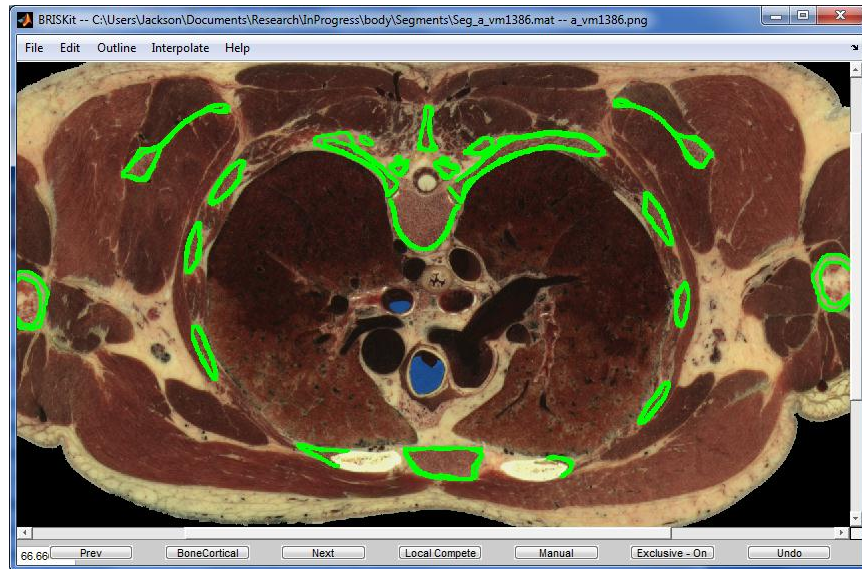


Figure 26: The BRISKit *Review* mode (a_vm1386.png). The labeled BoneCortical regions are outlined in green and overlaid on top of the original image.

Once the user is finished labeling tissues in an image, the results are then saved to a matrix file. A color image is also stored that uses each region's mean color as the basis for its color in the stored image for easy visualization between the original image and the segmented image (Figure 27). The results can then be coarsened with the coarsening algorithm and created into an input file (Section 7).

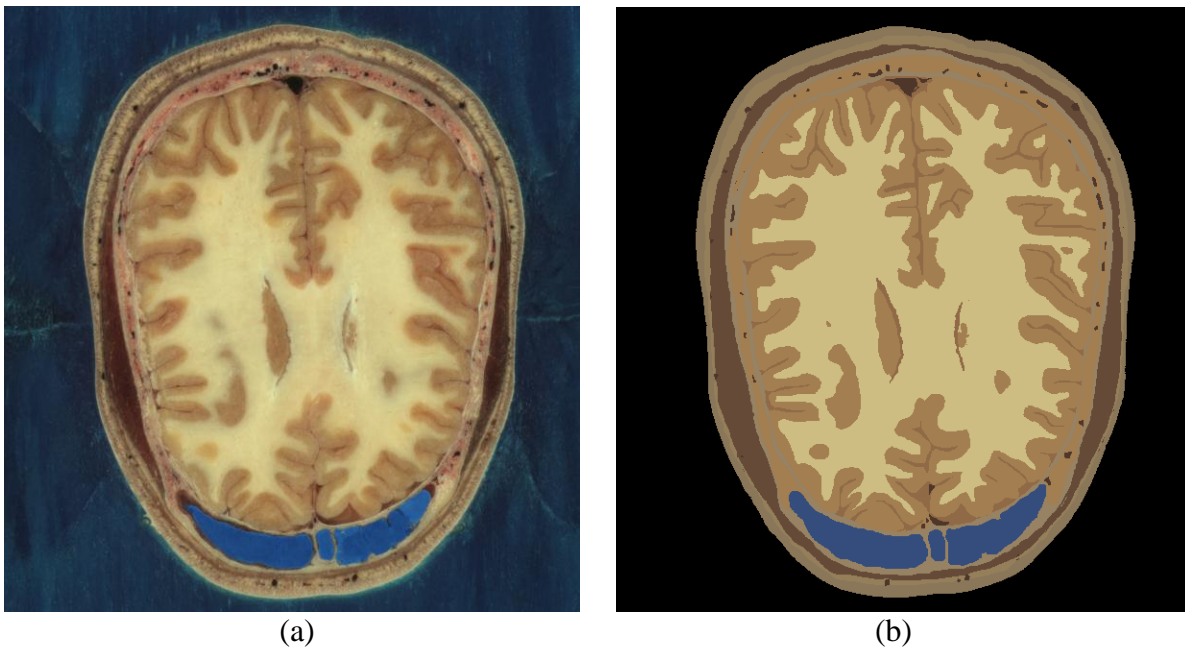


Figure 27: (a) The original image for slice a_vm1070.png. (b) The segmented and labeled color image created by using BRISKit.

7 Coarsening and Extrusion

After segmenting the images and labeling the tissues with electromagnetic tissues, a voxel model can be obtained by extruding each pixel in the $+z$ and $-z$ directions. This results in a very large dataset that can be used only by a few electromagnetic simulators at this time. To enable quicker (but less accurate) analysis, this data set was ‘coarsened’ and smaller models were developed. Various different approaches can be used for obtaining coarser models; as detailed below, two different methods were used for the AustinMan models: A 2-D coarsening method was used for the AustinMan v1.0 model and a 3-D one was used for the AustinMan v1.1. The 2-D coarsening method discards some of the slices and does not rely on/take advantage of the information in those slices, whereas the 3-D coarsening method uses information from all the slices. The 2-D coarsening method can be used to create models even when not all the slices are segmented whereas the 3-D results in more accurate models. After coarsening and extrusion, the model is stored in the UCD file format [16] that can be easily read by visualization software and computational simulators.

7.1 2-D Coarsening: AustinMan v1.0

In the 2-D coarsening method, first the z -direction is coarsened by discarding image slices, e.g., to create the $4 \times 4 \times 4 \text{ mm}^3$ AustinMan v1.0 model, slices 1002, 1006, 1010, etc. were kept and slices 1003-1005, 1007-1009, etc. were discarded. Then, the pixels in each of the remaining slices are coarsened independently by combining multiple pixels to create larger pixels (starting the combination from the $(x,y) = (1,1)$ pixel (top left corner of the cropped image) of each slice and shifting): Each new pixel is assigned a tissue by finding the mode of the tissues (the most frequent tissue) in an area of pixels. For example, for the $4 \times 4 \times 4 \text{ mm}^3$ resolution model, all pixels in an image slice are grouped into 12×12 sets of $(1/3 \times 1/3 \text{ mm}^2)$ pixels and each 12×12 set of pixels is replaced with a single $4 \times 4 \text{ mm}^2$ pixel. If the 12×12 set of pixels consisted of 35 pixels of Vacuum, 42 pixels of SkinDry, 56 pixels of Fat, and 11 pixels of Muscle, then the new pixel would be considered Fat. In the rare instances where two or more tissues had the same number of pixels in an area, the tissue with the lowest tissue ID number in the electromagnetic tissues table was chosen (essentially a random choice among the tissues). Once the larger pixels are created they are extruded $\Delta z/2$ in the $+z$ and $-z$ directions to create cubic voxels. Note that, to keep the top and bottom of the different resolution models at the same location, the first and last slices (slices 1002 and 1354) were extruded $\Delta z/2$ only in the $-z$ and $+z$ directions, respectively. As a result, the voxels at the top and bottom of the model are half as high as the remaining voxels (Figure 28).

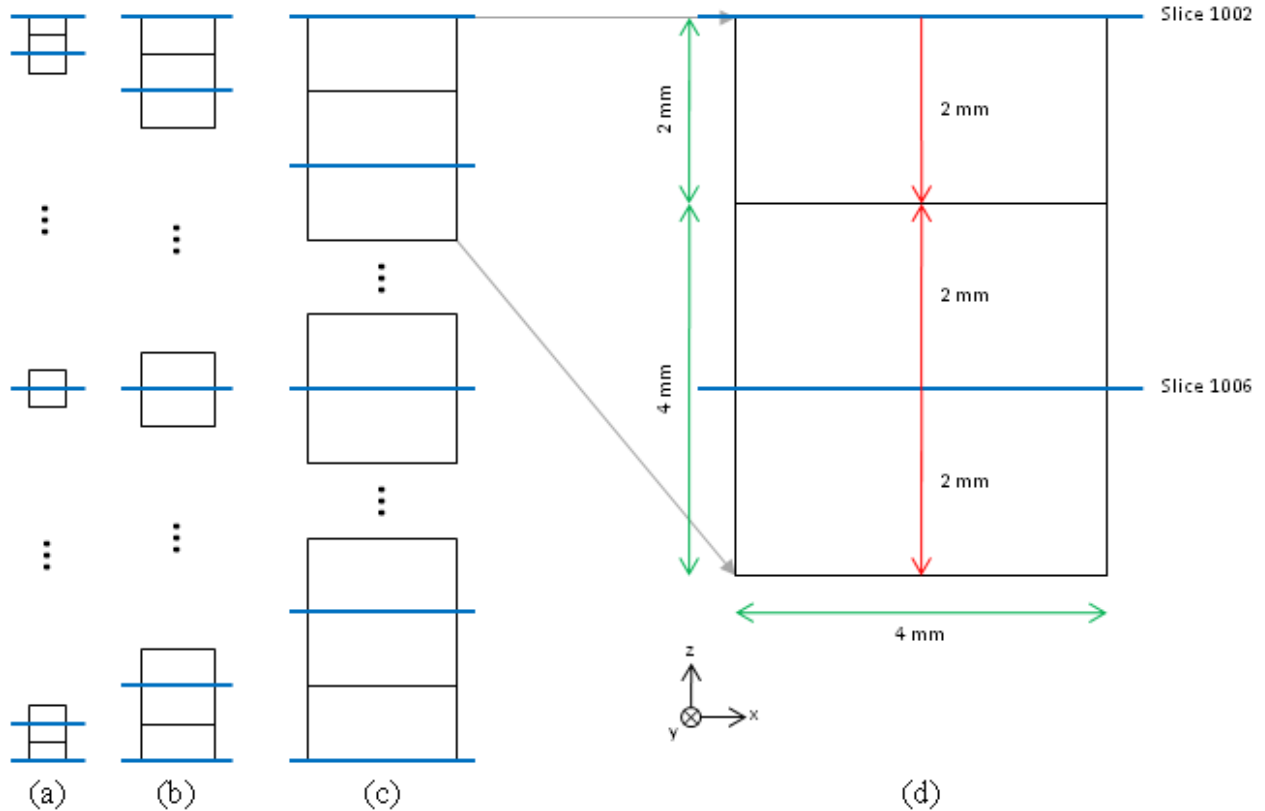
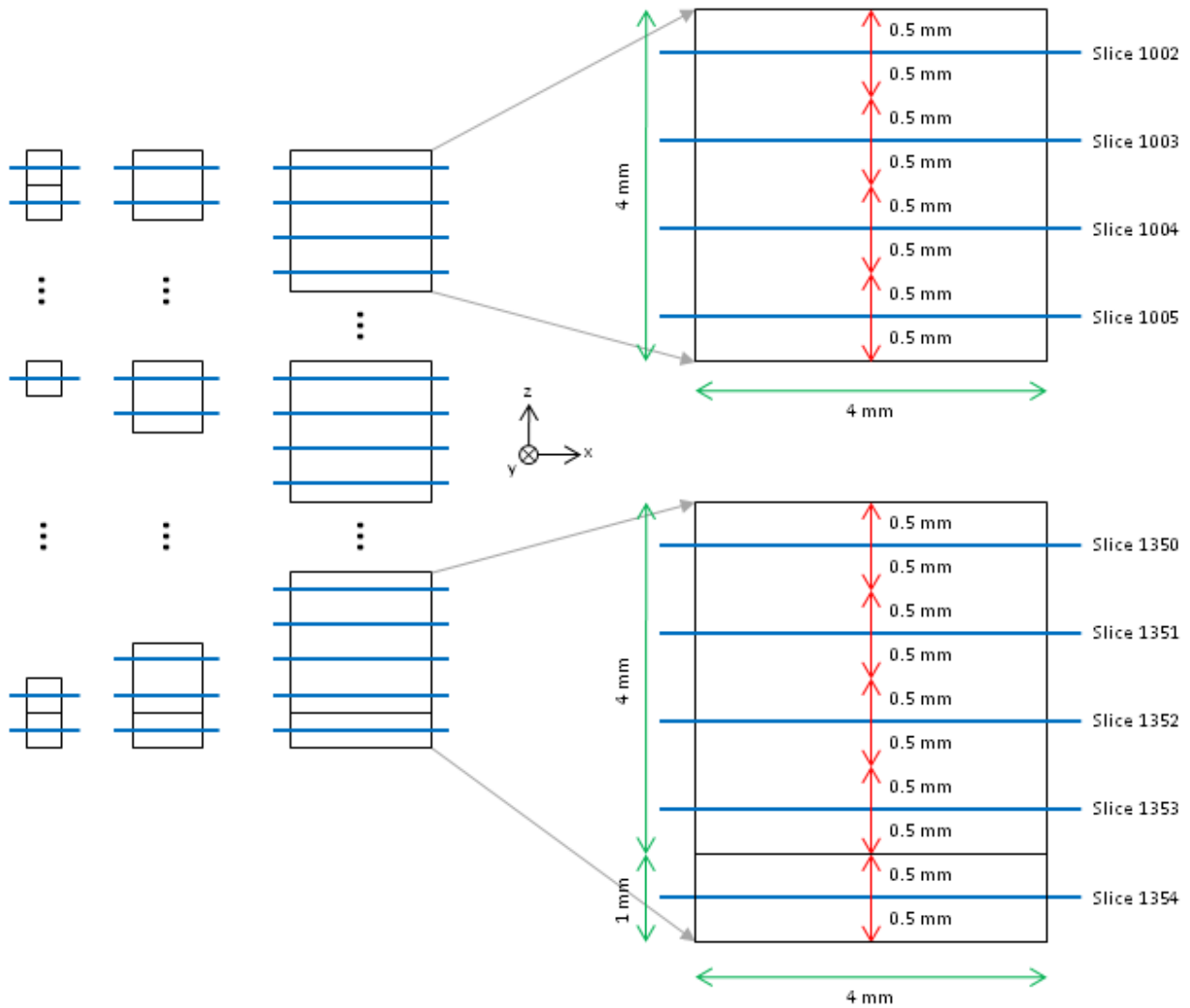


Figure 28: 2-D coarsening. The (a) $1 \times 1 \times 1 \text{ mm}^3$, (b) $2 \times 2 \times 2 \text{ mm}^3$, and (c) $4 \times 4 \times 4 \text{ mm}^3$ voxel resolutions in the AustinMan v1.0 model. The blue lines represent the image slices while the black lines represent the boundaries of the voxels. (d) A zoomed-in version of the top of the $4 \times 4 \times 4 \text{ mm}^3$ resolution model that shows how the top slice is only extruded downward 2 mm ($\Delta z/2$). The red arrows indicate the direction and distance that each slice was extruded and the green arrows indicate voxel dimensions.

7.2 3-D Coarsening: AustinMan v1.1 and Later

In the 3-D coarsening method, first pixels are extruded in the z direction to create $1/3 \times 1/3 \times 1 \text{ mm}^3$ voxels. Then, the voxels are coarsened by combining multiple voxels (starting from the top left corner of the top slice, i.e. $(x,y,z) = (1,1,1)$ voxel of the model): Each new voxel is assigned a tissue by finding the mode of the tissues (the most frequent tissue) in a volume of voxels. For example, for the $4 \times 4 \times 4 \text{ mm}^3$ resolution model, the voxels are grouped into $12 \times 12 \times 4$ sets of $(1/3 \times 1/3 \times 1 \text{ mm}^3)$ voxels and each $12 \times 12 \times 4$ set of voxels is replaced with a single $4 \times 4 \times 4 \text{ mm}^3$ voxel (Figure 29). Note that unlike the 2-D coarsening the pixels in the first and last image slices are extruded both up and down 0.5 mm for the initial (finest resolution) model. This method ensures that each of the initial voxels is given equal weight during coarsening and that the top and bottom of the model has the same locations regardless of the resolution. Note that when the number of image slices is not a multiple of the coarsening factor, the height of the voxels at the bottom are set to the height of the remaining slices and the voxels at the bottom of the coarsened model have a different height than the rest. For example, if there are 353 image slices in the finest-resolution model, then the voxels at the bottom of the $1 \times 1 \times 1 \text{ mm}^3$, $2 \times 2 \times$

2 mm^3 , and $4 \times 4 \times 4 \text{ mm}^3$ resolution models would have sizes of $1 \times 1 \times 1 \text{ mm}^3$, $2 \times 2 \times 1 \text{ mm}^3$, and $4 \times 4 \times 1 \text{ mm}^3$, respectively.

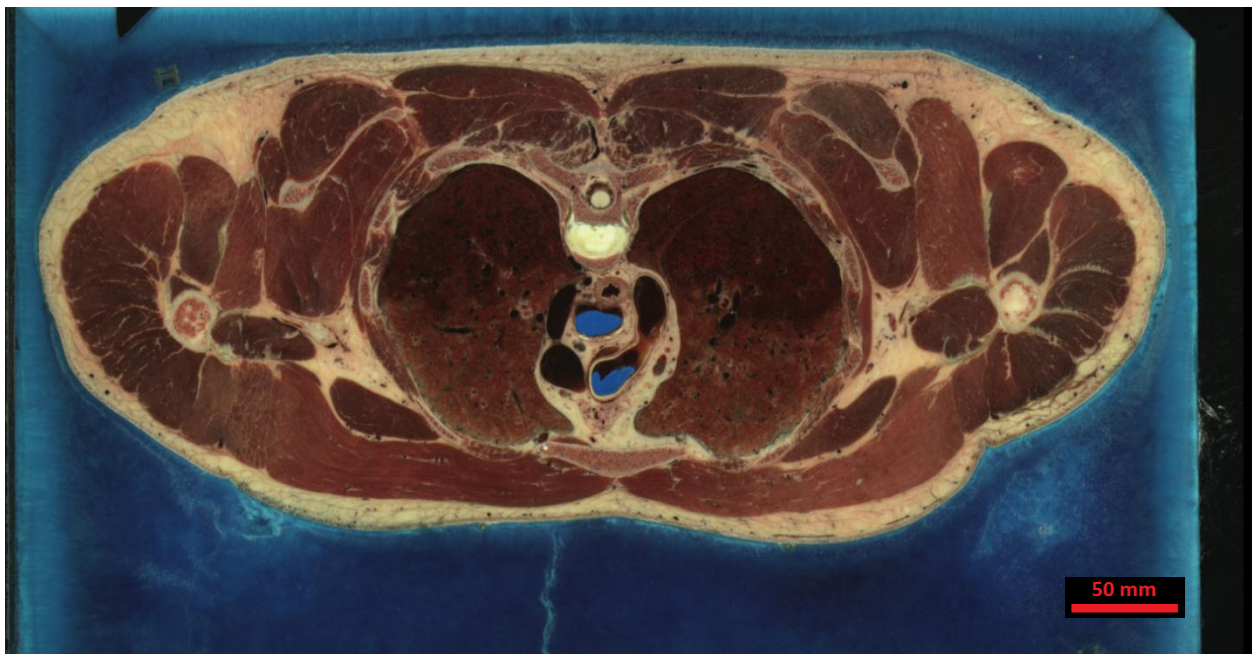


(a) (b) (c) (d)
 Figure 29: 3-D coarsening. The (a) $1 \times 1 \times 1 \text{ mm}^3$, (b) $2 \times 2 \times 2 \text{ mm}^3$, and (c) $4 \times 4 \times 4 \text{ mm}^3$ voxel resolutions in the AustinMan v1.1 model. The blue lines represent the image slices while the black lines represent the boundaries of the voxels. (d) A zoomed-in version of the top and bottom of the $4 \times 4 \times 4 \text{ mm}^3$ voxel resolution model. Notice that the voxels at the bottom can have a different height than others. The red arrows indicate the direction and distance that each slice was extruded and the green arrows indicate voxel dimensions.

8 Results

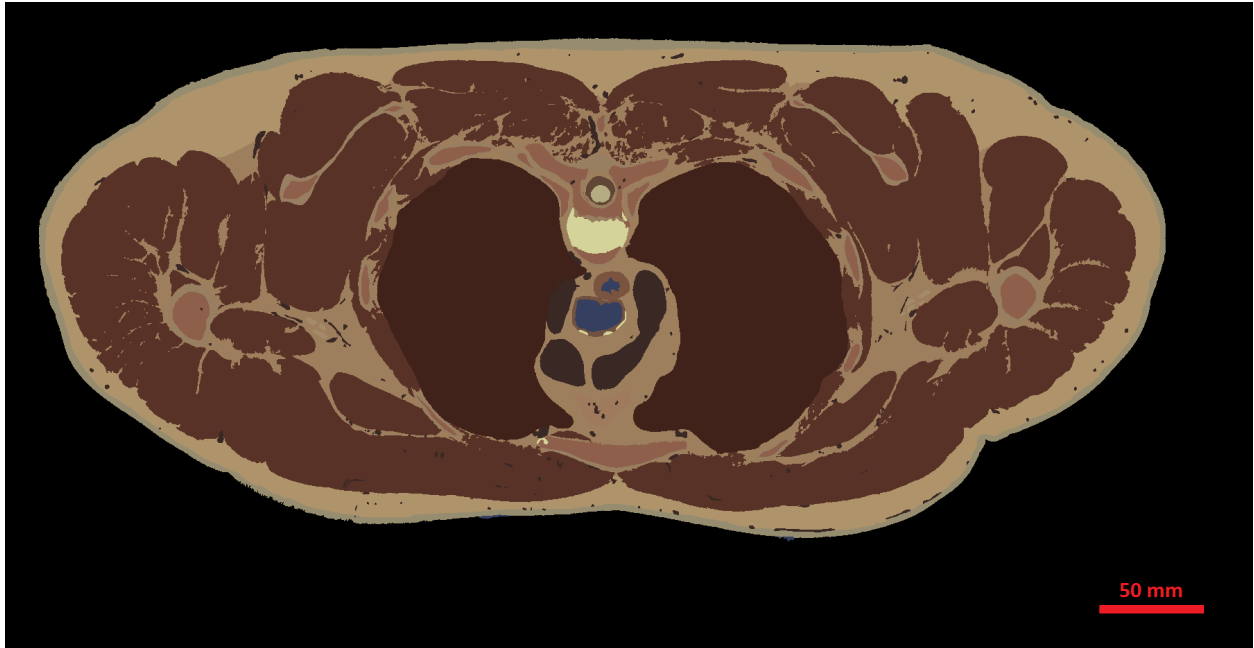
At the time of this writing, the processing of slices 1002-1397 has been completed. These slices contain approximately 160 million pixels that belong to the body and 31 different tissues have been labeled in the model thus far. The accuracy of the boundaries for each slice is estimated to be ± 3 pixels (± 1 mm).

The results from this project are largely confirmed through visual inspection. There are several different ways to view the model. Figure 30 shows the comparison between the original image and segmented color image. The segmented color image shows a fairly good agreement with the original image, including some of the areas of with high detail, such as the tendons in the muscles by the shoulders and the blood vessels in the fat along the bottom. It should again be emphasized that this is an iterative process that improves the accuracy of the model with adjustments made each time the slice is reviewed. Figure 30 also highlights why user-intervention is required to obtain an accurate model. One of the large blood vessels in the original picture is partially blue due to the gelatin added during the slice preparation process [7]. This shows the deficiency of labeling tissues strictly based on color, regardless of how accurate the algorithm is.



(a)

Figure 30: (a) The original image for slice a_vm1369.png. (b) The segmented color image created by BRISKit.



(b)

Figure 30: Continued.

The models can also be visualized in volume-rendering environments such as ParaView. To do this, the model files must be converted to a more standard UCD format [16]. Visualizing the larger datasets require significant resources, e.g., TACC visualization clusters. Figure 31 shows the 3-D visualization of the AustinMan model down to slice 1354. The colors for the tissues are based on their mean color throughout the dataset to give a more realistic looking model for comparison. ParaView also has several features that aid in visually inspecting the model for continuity, such as cutting or clipping with an arbitrary plane or using a threshold operation to only view a certain tissue. In Figure 31, two planar cuts are used to display the internal tissues of the model. A discontinuity in the shoulder bone marrow becomes readily apparent when visualizing the model in 3-D.

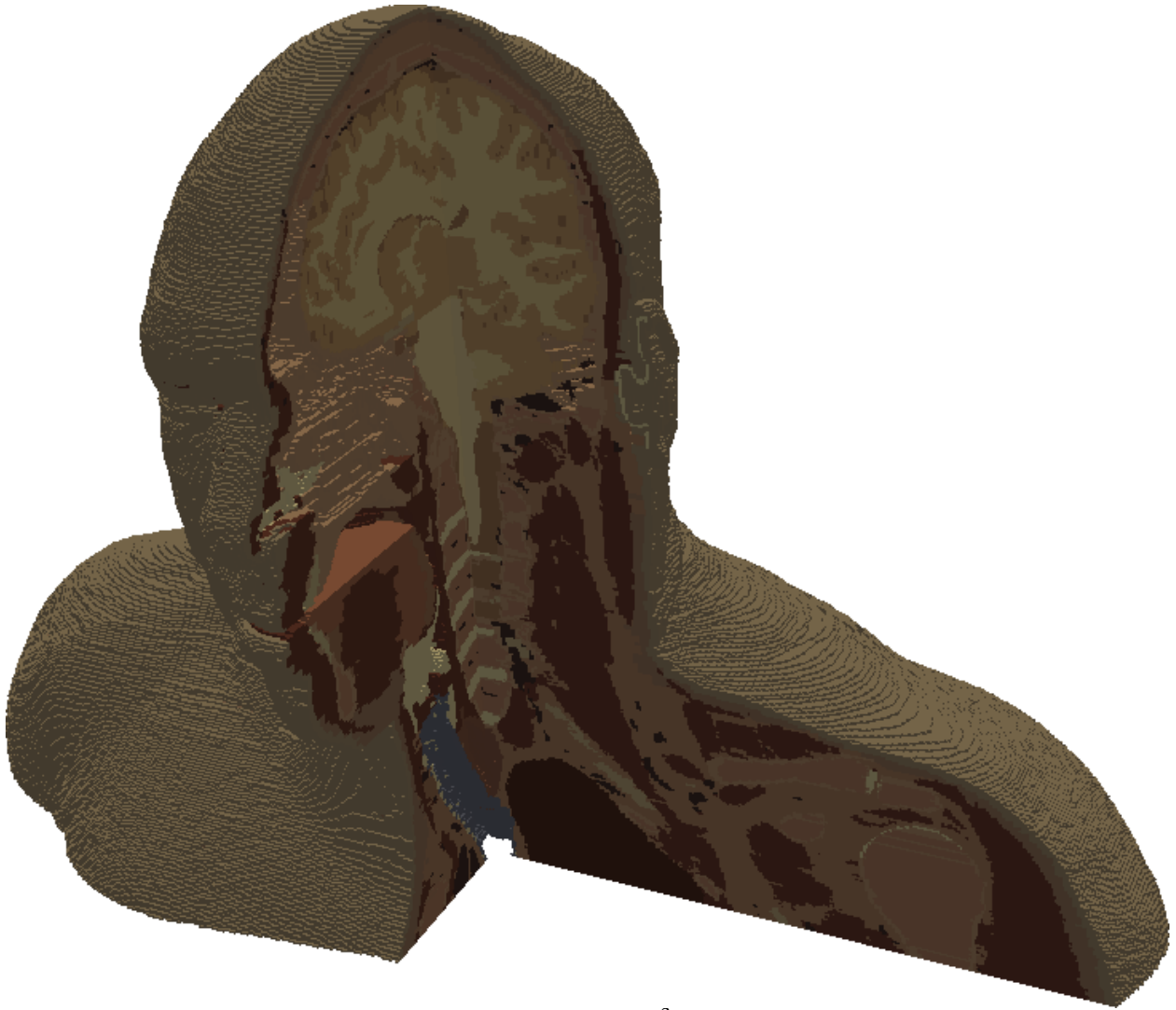


Figure 31: Visualization of the $1 \times 1 \times 1 \text{ mm}^3$ resolution model in ParaView.

Figure 32 also uses ParaView for visualizing the AustinMan model, this time by assigning thresholds on the tissue ID to only display certain tissues. This feature is very beneficial for checking the accuracy of each tissue. The visualization of the blood vessels shows the level of detail that was obtained. The tendon top in the muscle in Figure 32 is due to a tissue assumption at the the top of the head that is outlined at the website (<http://web2.corral.tacc.utexas.edu/AustinManEMVoxels>) and Appendix A.

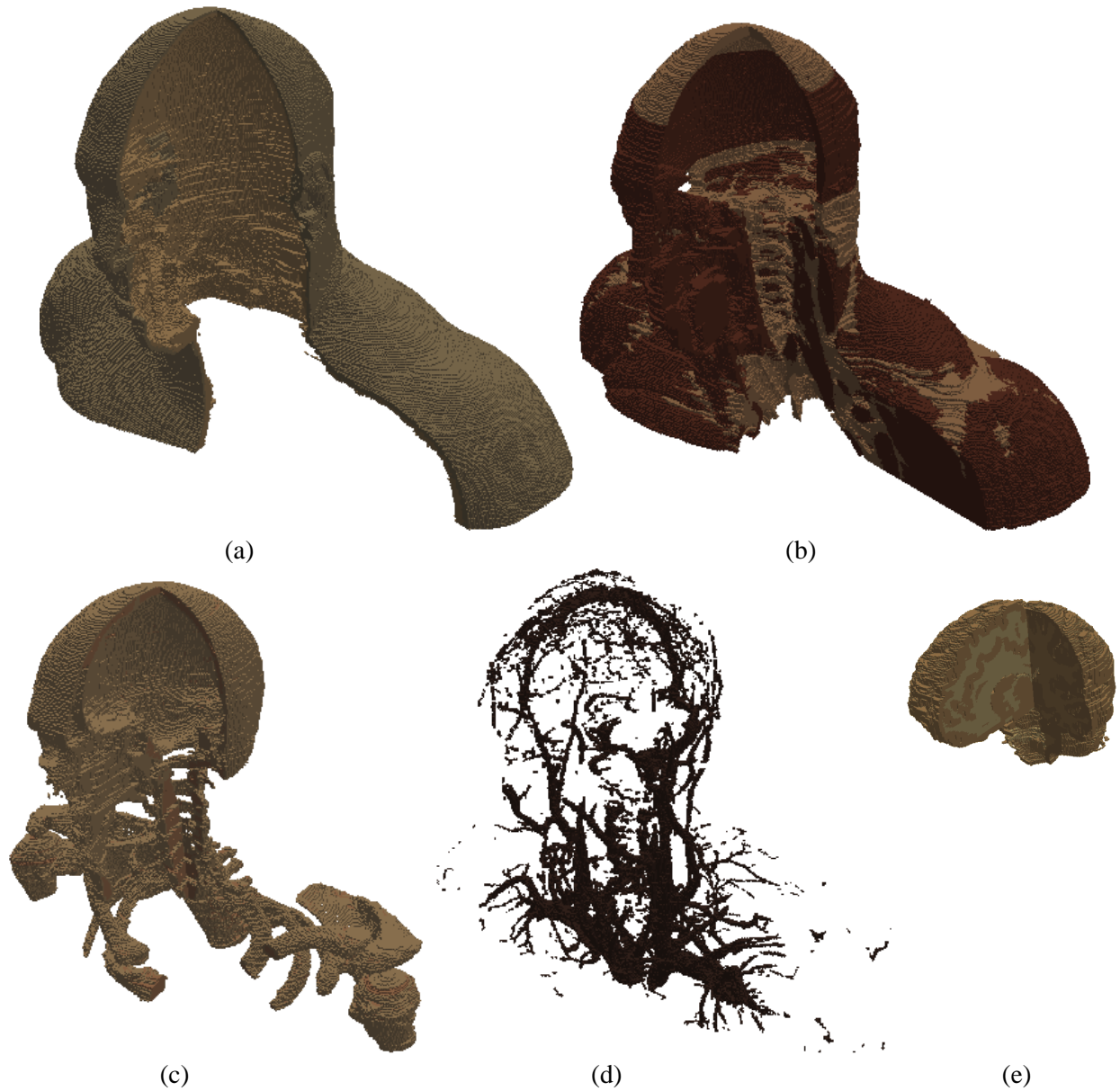


Figure 32: Visualization of the different tissues in the $1 \times 1 \times 1 \text{ mm}^3$ resolution model. (a) Skin and fat. (b) Muscle and tendon. (c) Bone cortical and bone marrow. (e) Gray and white matter in the brain.

Figures 31 and 32 displayed the visualization results for the $1 \times 1 \times 1 \text{ mm}^3$ resolution model. Coarser models were also produced as discussed in Section 7. Figure 33 shows two views of various resolution models to compare the effects of coarsening.

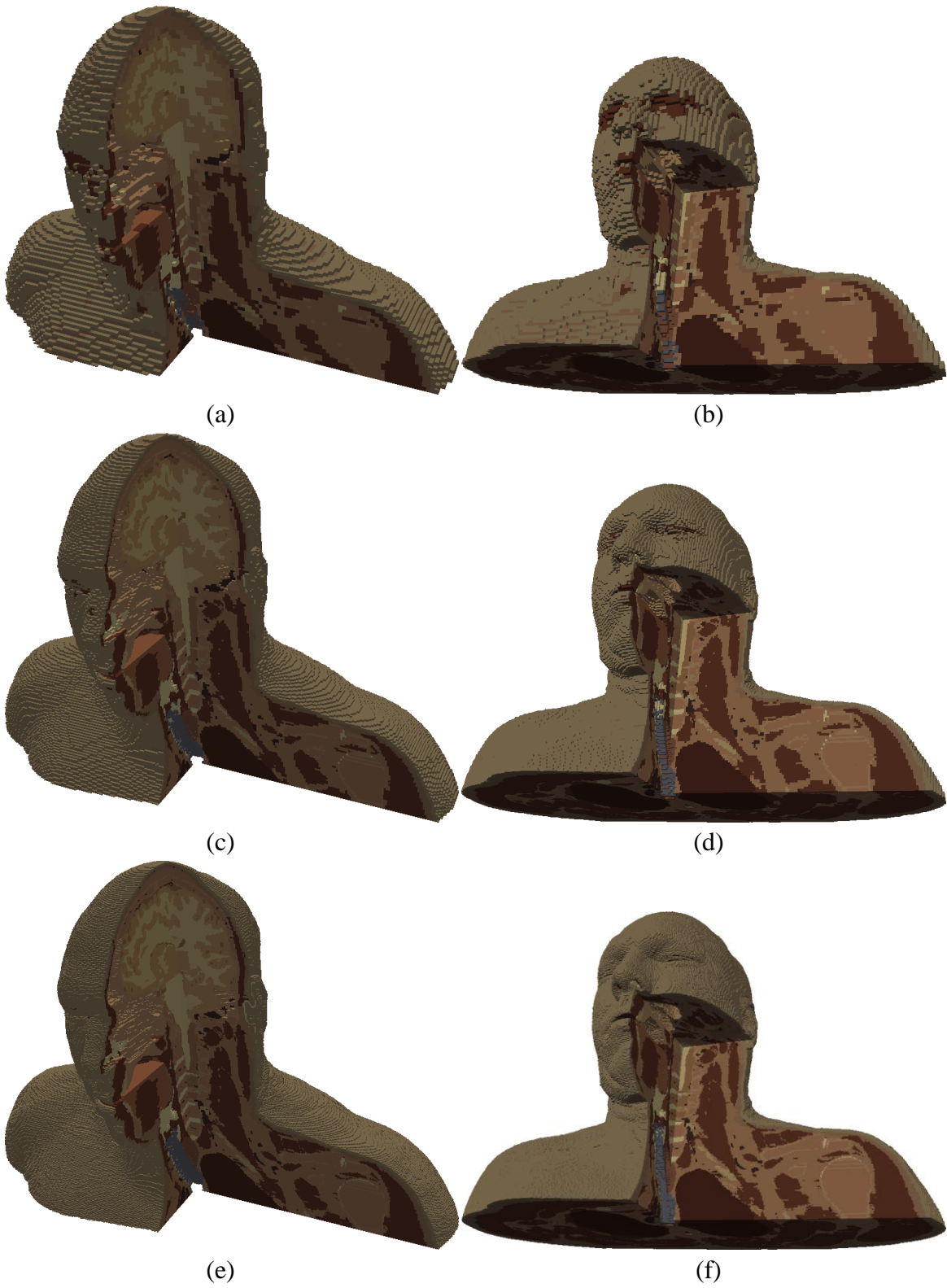


Figure 33: The effects of coarsening. The (a, b) $4 \times 4 \times 4 \text{ mm}^3$, (c, d) $2 \times 2 \times 2 \text{ mm}^3$, and (e, f) $1 \times 1 \times 1 \text{ mm}^3$ resolution models.

It is apparent from Figure 33 that the $4 \times 4 \times 4 \text{ mm}^3$ resolution model is not fine enough to resolve the skin region, resulting in red (muscle) voxels that appear on the surface of the model instead of skin. The $2 \times 2 \times 2 \text{ mm}^3$ and $1 \times 1 \times 1 \text{ mm}^3$ resolution models do not appear to suffer from this problem and both look fairly smooth. Figures 34 through 37 show the effect of coarsening on the mask slices (which affects both the homogeneous and heterogeneous models).

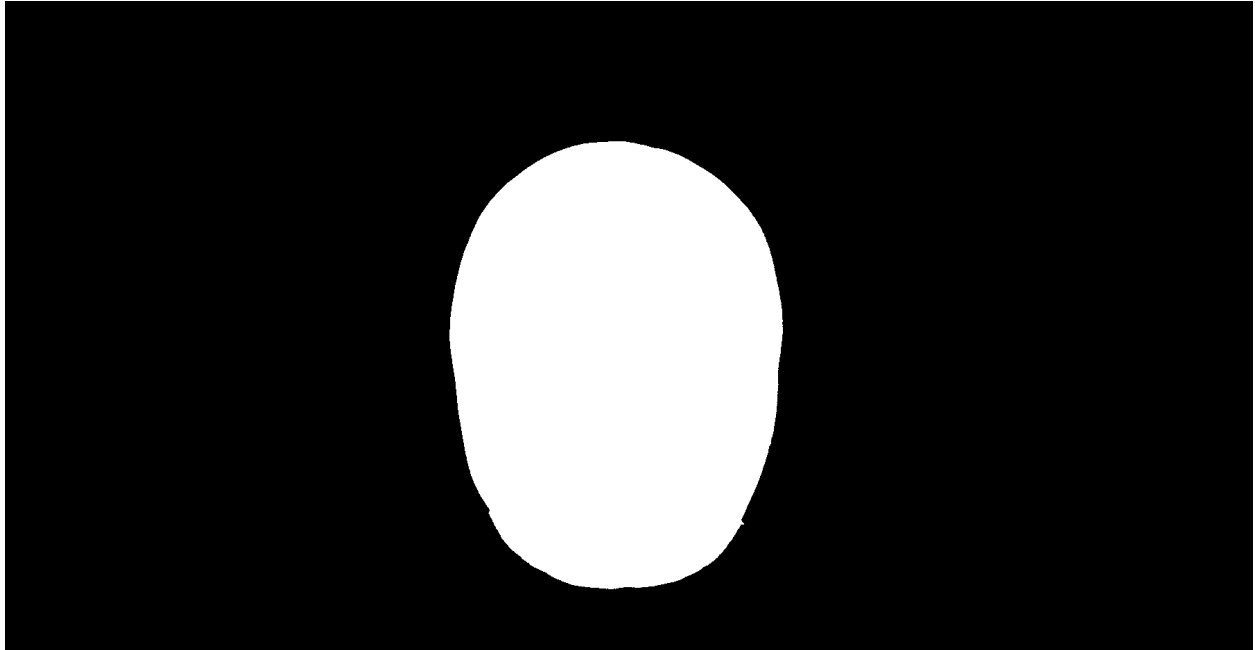


Figure 34: Mask_a_vm1090.png at the original resolution (1 mm / 3 pixels). (Resized to 35% of original to fit page).

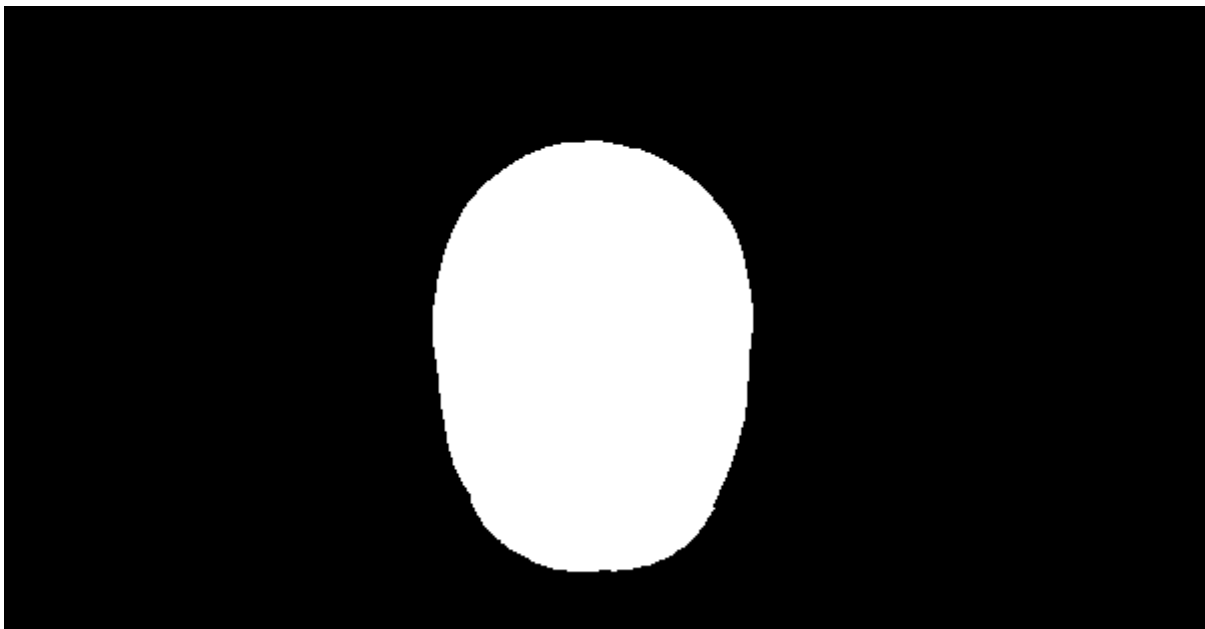


Figure 35: Mask_a_vm1090.png reduced by a factor of 3 (1 mm / pixel).

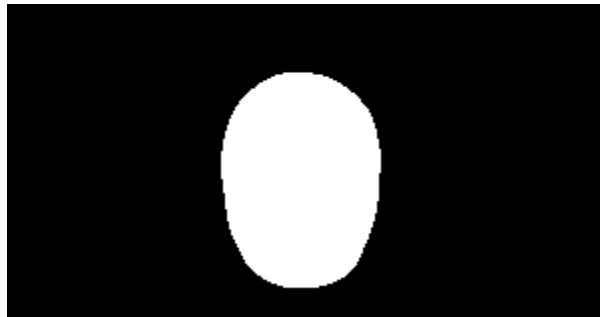


Figure 36: Mask_a_vm1090.png reduced by a factor of 6 (2 mm / pixel).



Figure 37: Mask_a_vm1090.png reduced by a factor of 12 (4 mm / pixel).

Figure 38 shows the result from MATLAB displaying the homogeneous model using the 3-D isosurface visualization tool. This method was not as fast and interactive as ParaView. As a result, ParaView became the primary visualization tool.

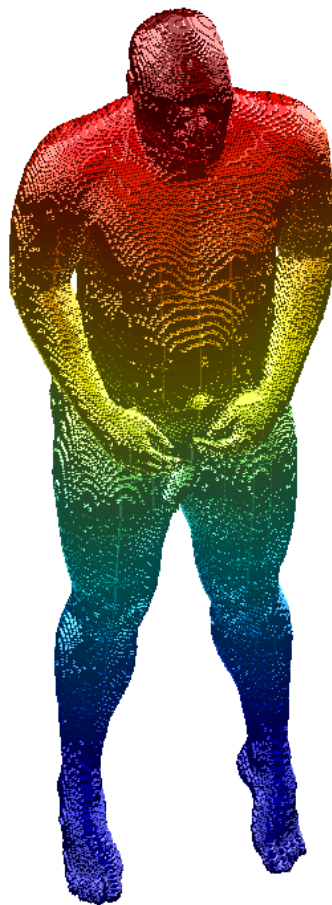


Figure 38: The homogeneous model from the masks.

9 Future Work

At the time this thesis was submitted, slices 1400-2050 and slices 2251-2878 still needed to be analyzed to complete the male dataset. BRISKit and the algorithms described in this thesis are being applied to these slices and fine-tuned, e.g., by changing thresholds for competition algorithms. It is expected that the entire AustinMan model will be completed by the end of Summer 2011. The methods presented here should be directly applicable to the female dataset in the VHP project and should facilitate a relatively faster development of an AustinWoman model.

10 Conclusion

A suite of algorithms and a graphical user interface were developed for the purpose of creating a virtual human model by performing image processing on the anatomical, cross-sectional slices of the VHP. Image segmentation of medical images is still an active research area and there is no single automated algorithm that can segment an image into different regions corresponding to physical tissues. As the authors continued working on the project, it became quite clear that there needed to be a large amount of interaction with the user to obtain accurate results in a timely fashion. As a result, a significant effort was spent on user interface and real-time processing algorithms instead of improving the accuracy of automated algorithms. This led to the development of BRISKit – a software tool that can speed up segmentation of images with judicious user inputs.

The AustinMan human model has been posted online along with extensive documentation about its development. Within its first month online, there have already been seven requests for the data sets from research groups across the world. Additionally, the model has already been used by the Computational Electromagnetics Group at the University of Texas at Austin for simulating the effects of cell phone radiation through simulations [17].

11 Acknowledgments

The following students have dedicated time to process the image slices:

- Cemil S. Geyik
- Robin Q. Nguyen
- Trevor Latson
- Madison Ball

Dr. Ali E. Yilmaz from the Department of Electrical and Computer Engineering at the University of Texas at Austin served as the faculty advisor for this thesis. Without his mentoring and expertise, this project would not have been possible.

Dr. Paul Findell from the Department of Biological Sciences at the University of Texas at Austin provided his insight and expertise to improve the anatomical accuracy of the model. Dr. Findell assisted both with determining unknown tissues in the pictures and with matching the many different tissue types to a corresponding tissue with known electromagnetic properties.

This material is based partially upon work supported by the National Science Foundation (NSF) under Grant No. 0904907. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of the NSF. The authors acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing visualization resources that have contributed to the development of this model. The authors acknowledge the National Library of Medicine (NLM) and the Visible Human Project as the source of the data used to create the AustinMan model. The Visible Human Data Set was developed under a contract from NLM by the Departments of Cellular and Structural Biology, and Radiology at the University of Colorado School of Medicine.

Works Cited

- [1] J.W. Massey, C.S. Geyik, M. Ball, P. Findell, and A.E. Yilmaz, "AustinMan Electromagnetic Voxels," 2011, <http://web2.corral.tacc.utexas.edu/AustinManEMVoxels/> (current 5 May 2011).
- [2] D. Seabury, "An Update on SAR Standards and the Basic Requirements for SAR Assessment," 2005, http://www.conformity.com/artman/publish/printer_74.shtml (current 5 May 2011).
- [3] CST Studio Suite, "HUGO Human Body Model," 2005, <https://www.cst.com/Content/Applications/Article/HUGO+Human+Body+Model> (current 5 May 2011).
- [4] IT'IS Foundation, "High-Resolution Whole-Body Human Models of the Virtual Population," 2010, <http://www.itis.ethz.ch/services/human-and-animal-models/human-models/> (current 5 May 2011).
- [5] A. Christ, W. Kainz, E.G. Hahn, K. Honegger, M. Zefferer, E. Neufeld, W. Rascher, R. Janka, W. Bautz, J. Chen, B. Kiefer, P. Schmitt, H.-P. Hollenbach, J. Shen, M. Oberle, D. Szczerba, A. Kam, J.W. Guag, and N. Kuster, "The Virtual Family--development of surface-based anatomical models of two adults and two children for dosimetric simulations.," *Physics in medicine and biology*, vol. 55, Jan. 2010, pp. N23-38.
- [6] U.S. National Library of Medicine, "The National Library of Medicines Visible Human Project," 2003, http://www.nlm.nih.gov/research/visible/visible_human.html (current 5 May 2011).
- [7] V. Spitzer, M.J. Ackerman, a L. Scherzinger, and D. Whitlock, "The visible human male: a technical report.," *Journal of the American Medical Informatics Association : JAMIA*, vol. 3, 1996, pp. 118-30.
- [8] YassineMrabet, "Planes of Human Anatomy," *Wikipedia*, 2008, http://en.wikipedia.org/wiki/File:Human_anatomy_planes.svg (current 5 May 2011).
- [9] International Organization for Standardization, "ISO 11664-4:2008 (CIE S 014-4/E:2007): Colorimetry -- Part 4: CIE 1976 L*a*b* Colour space," 2007.
- [10] Mathworks, "Color-Based Segmentation Using the L*a*b* Color Space," <http://www.mathworks.com/products/image/demos.html?file=/products/demos/shipping/images/ipexfabric.html> (current 5 May 2011).
- [11] Mathworks, "imfill - MATLAB Image Processing Toolbox Documentation," <http://www.mathworks.com/help/toolbox/images/ref/imfill.html> (current 5 May 2011).

- [12] P. Soille, *Morphological Image Analysis: Principles and Applications*, Secaucus, NJ, USA: Springer-Verlag Telos, 1999.
- [13] L.G. Shapiro and G.C. Stockman, *Computer Vision*, Upper Saddle River, NJ: Prentice Hall, 2001.
- [14] R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, Jun. 1994, pp. 641-647.
- [15] Mathworks, "Removing Noise from Images," <http://www.mathworks.com/help/toolbox/images/f11-12251.html> (current 5 May 2011).
- [16] "UCD File Format," http://help.avs.com/Express/doc/help/reference/dvmac/UCD_Form.htm (current 5 May 2011).
- [17] T. Malas, F. Wei, J.W. Massey, C.S. Geyik, and A.E. Yilmaz, "Application of AIM to high-resolution bioelectromagnetics simulations," *Proc. Appl. Comp. Electromagnetics Symp.*, 2011, p. 558-563.

Appendix A: Tissue Assumptions and Problematic Regions

The most recent tissue assumptions and problematic regions are listed on the AustinMan Electromagnetic Voxels website: <http://web2.corral.tacc.utexas.edu/AustinManEMVoxels/>



AustinMan Electromagnetic Voxels

[Introduction](#)

[Methodology](#)

[Crop](#)

[Mask](#)

[Segment & Identify](#)

[Coarsen & Extrude](#)

[Homogeneous Model](#)

[Assumptions](#)

[Image Shifts](#)

[Tissue Assumptions](#)

[Problem Regions](#)

[EM Material Properties](#)

[Known Errors](#)

[Citing the Model](#)

[Acknowledgments](#)

[Conditions of Use](#)

[Contact](#)

[FAQ](#)

[References](#)

[Download Data](#)

[AustinMan v1.1](#)

[Homogeneous v1.1](#)

[AustinMan v1.0](#)

[Homogeneous v1.0](#)

ASSUMPTIONS

A variety of assumptions were made when creating the AustinMan electromagnetic voxel models. The Image Shifts section lists the shifts that were made to align some of the image slices. The Material Assumptions section lists the assumptions made for tissues that do not have a corresponding entry in the electromagnetic materials table. The Problem Regions section outlines some of the regions where boundaries were identified less precisely than typical.

IMAGE SHIFTS

The cross-sectional image slices in the Visible Human Dataset do not align perfectly as observed in [9]. At some of the image slices, the picture is significantly shifted compared to the one above it and thus all the slices below that are misaligned by that shift (Fig. 3). To counter the misalignment, the slices where these shifts occur are identified, the amount of the shift is determined, and the slices are shifted back. The shifts were quantized in terms of pixels and their values were determined by maximizing the overlap between the mask of the slices under operation and the slices above them (e.g., mask of slice 1231 was compared to that of slice 1230). The shifts were then confirmed by visual inspection. Table 1 shows the pixel shifts made to the slices while developing the AustinMan model.

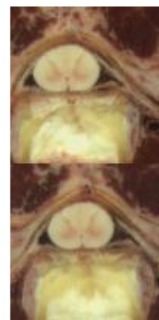


Figure 3: A cropped portion (at the same position of each slice) of the spinal cord from slices a_vm1230.png (top) and a_vm1231.png (bottom) indicating the relative shift of 9 pixels to the right.

Table 1: Pixel Shifts Used to Align the Dataset

TISSUE MATERIAL ASSUMPTIONS

The following assumptions were made when assigning known electromagnetic material ids to tissues whose electromagnetic properties are not known.

Figure 39: Assumptions page of the website from May 1, 2011.

About the Author



Jackson Massey is an undergraduate electrical engineering student at the University of Texas at Austin. He began working with the University of Texas Computational Electromagnetics Group in January 2010 developing the AustinMan Electromagnetic Voxels model. He will receive his B.S. in electrical engineering in August 2011 with high honors. In Fall 2011, Jackson will begin his graduate studies in pursuit of a Ph.D. at the University of Texas at Austin.