Beamwidth Characterization and Hole Filling for Computational Ultrasound Reconstruction

Thomas Vaughan - CISC 499 Final Report
Mentors: Gabor Fichtinger, Andras Lasso, Thomas Chen
Submission on March 30th 2012

Laboratory for Percutaneous Surgery, Queen's University

Abstract

The work done for CISC 499 consisted of two projects: automatic ultrasound beamwidth characterization and extraction, and; development of a hole-filling algorithm for ultrasound-based volume reconstruction. This report will include an introduction where the significance of both projects is explained, and one section for each of the projects where the specific backgrounds, methods, results, and conclusions are explained in detail.

1. Introduction

Medical imaging plays a key role in many clinical and research settings. Various modalities of imaging are currently available, including X-ray, CT scan, magnetic resonance imaging, and Ultrasound. This report focuses mostly on Ultrasound, its accuracy, and its application in building high-quality three dimensional reconstructions.

Ultrasound is a technique where beams of high frequency sound are sent from a transducer. The transducer is able to detect reflections as they reflect off objects and return to the source of the ultrasound. Different surfaces and materials in the ultrasound field can be localized by the amount of time it takes for the sound to hit them and bounce back to the transducer. Ultrasound is used extensively in medical settings and research because it is cheap, and because it produces no ionizing radiation.

If the ultrasound transducer is tracked in real-time, accurate 3D volume reconstructions can be produced from localized signals. This should not be done, however, without being aware of sources of uncertainty within the ultrasound images. Uncertainty may arise from factors such as noise, speckle, artifacts, and the out-of-plane width of the ultrasound beam.

Ultrasound beamwidth is a source of localization errors. Ultrasound essentially assumes that sound is projected in a perfectly flat plane originating along the transducer. This is not a correct assumption because sound can in fact travel outside the plane of ultrasound. When this happens, and sound returns to the transducer, there is no way of telling a signal that originated within the plane from one that originated outside the plane. Some objects outside of the plane may, therefore, appear as though they are inside the plane (Figure 1.1). Ultrasound settings such as gain have been known to influence beamwidth [3].

Once the volume reconstruction has been produced, there are often holes due to a lack of data in-between collected images.

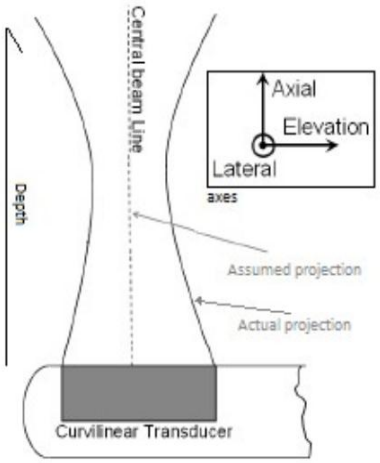

Figure 1.1: Ultrasound beamwidth visualized, adapted from Peikari et al [3].

The first project carried out as part of CISC 499 was automatic ultrasound beamwidth extraction and characterization using experimental data. This involved creating an algorithm to segment lines from ultrasound images. The lines could then be used to quantify the beamwidth.

The second project involved implementing a hole-filling algorithm into the Public software Library for Ultrasound (PLUS).

2. Automatic Ultrasound Beamwidth Extraction and Characterization

2.1. Background

Goldstein (1981) proposed that the beamwidth of ultrasound could be measured using images of a plane oriented 45 degrees to the transducer [1]. Because the height of the plane within the beamwidth will be approximately equal to the depth, we can quantify the beamwidth as the width of the visible beam inside the image (see Figure 2.1.1). This is currently the widely-accepted standard practice in industry for beamwidth characterization [5]. The Ultrasound beamwidth for a machine is therefore usually done in two phases: a Data Collection Phase and a Quantification Phase.

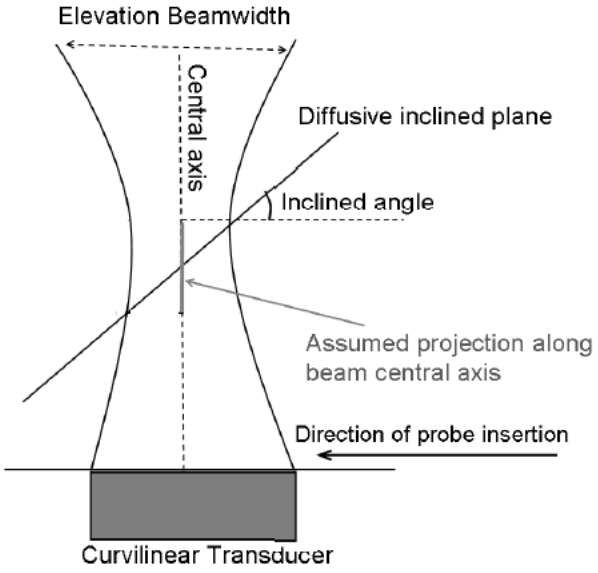

Figure 2: Illustration of inclined plane as it relates to beamwidth, adapted from Peikari et al [2].

1. The Data Collection Phase

The phantom consists of a flat rubber membrane oriented 45 degrees to the plane of the ultrasound. The ultrasound transducer, while collecting images, is slowly moved along a straight line between the center of the top of the plane and the center of the bottom of the plane (Figure 2.1.2) [2].

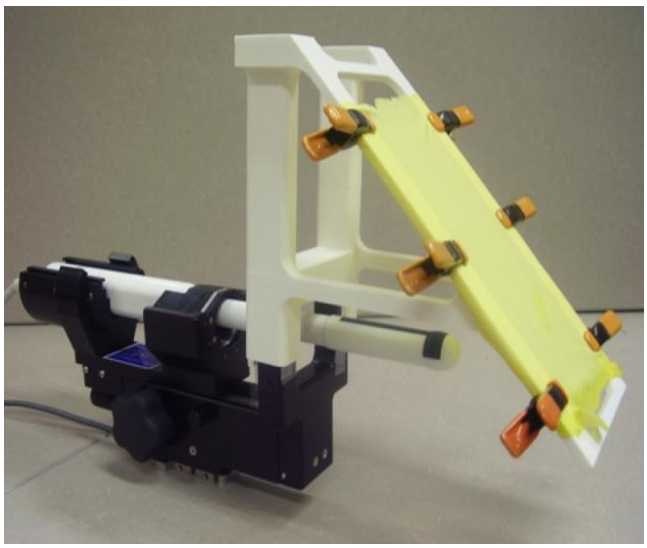

Figure 2.1.2: Phantom used during the Data Collection Phase

2. The Quantification Phase

The industrial standard is to look at the images obtained during the experiment, and to manually indicate the location of the band [5] (Figure 2.1.3). This is a very subjective task, however, and can result in bias or inconsistencies.

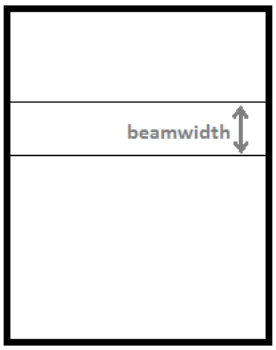

Figure 2.1.3: The beamwidth is quantified as the width of the high-intensity band in the ultrasound image

The **goal** of this project was to automate the quantification phase and to eliminate the bias and time investment that can result by requiring manual segmentation. Because the latter step only involves indicating where lines are in an image, the project is effectively an image processing problem.

2.2. Methods

The task is not trivial for several reasons. The first is that there is no gold standard to compare the final results of any algorithm against. The only way to validate any output is therefore by visual inspection. The second complication is that many known image segmentation procedures are sensitive to noise, which is quite common in ultrasound imaging. As a result, the images obtained during the experiment will likely require some form of pre-processing to reduce the amount of noise. Removing too much noise could be risky, however, since in preprocessing we are distorting the original image and losing potentially valuable information.

The data that were provided for the experiment had approximate line positions indicated. These were analyzed in order to try to gather potentially useful information about the beamwidth and how to extract it from an image. The mean intensities of each row of an ultrasound image were plotted together with lines that represent the segmented line elevations. It can be seen from such graphs (Figure 2.2.1) that the ultrasound beamwidth boundaries were usually located in local minima between areas of high intensity at two-thirds the maximum overall intensity and medium intensity at half the maximum overall intensity.

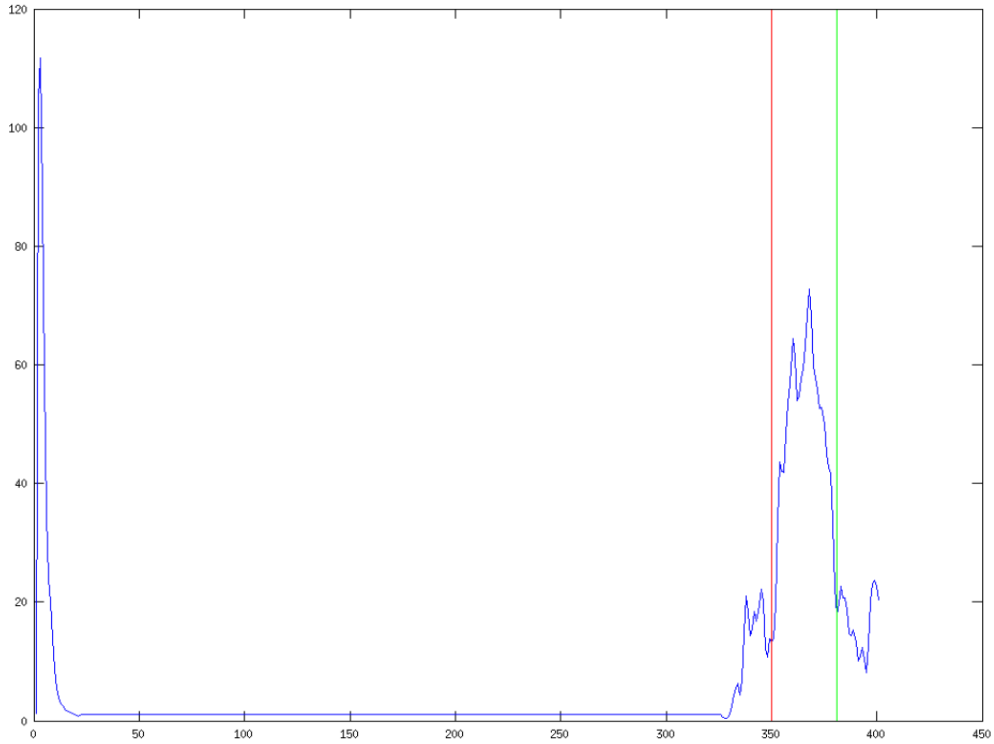

Figure 2.2.1: Representative graph showing the beam boundaries (in red and green) as they relate to the average intensity traversing the plane (blue). The y axis represents the pixel intensity, and the x axis represents the offset from the top of the image.

At this stage, some possible methods of segmenting the lines were proposed:

1. Traverse the graph of mean intensities, and find the minimum peak, in much the same manner as above

2. Region Splitting and Merging, as per Gonzalez et al (2004) [4], with three distinct regions corresponding to no band, weak band, and strong band

3. A combination of noise removal and edge detection using the Hough Transformation.

The final procedure that was decided upon was a combination of 1 and 3 above. The Hough Transformation is a standard edge-segmentation technique, and can be used to detect lines. The lines can then be filtered according to criteria similar to what was observed earlier that the basis of the first proposed process. Region Splitting and Merging would have been unsuitable because it does not detect lines, but rather it detects regions that may or may not form suitable or distinct lines. It possible that such a process could result in fuzzy boundaries where it is unclear what line would best divide the regions.

Refer to Figure 2.2.2 for the overall workflow.

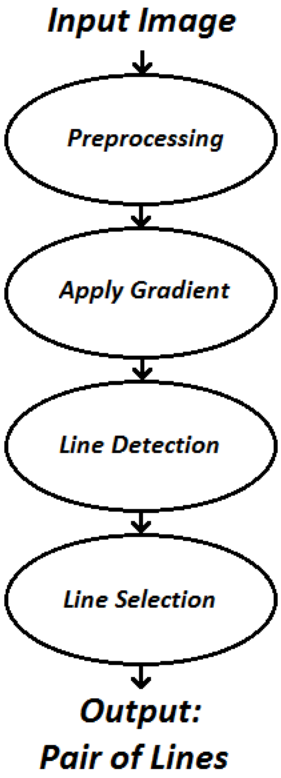

Figure 2.2.2: Workflow. The bubbles each represent a component of the overall algorithm and the arrows represent the data flow.

The Hough Transformation and Line Detection is sensitive to noise. The noise needed to be removed to at least some extent before the Line Detection was performed. Three filters were found for this purpose: a median filter, a blur filter, and a sticks filter. The sticks filter is an image processing technique for use in reducing noise in ultrasound images [6,7].

Once the image has been processed, an edge enhancement gradient is applied to enhance the results obtained by the algorithm as a whole. Sobel, Canny, and Prewitt vertical gradients were all applied to the data, but did not perform as well as a generalized edge detection matrix (Table 2.2.3) adapted from Hedrick et al [5].

0	-1	0
-1	4	-1
0	-1	0

Table 2.2.3: Edge detection matrix adapted from Hedrick et al [5].

The Hough Transformation and Line Detection algorithm transforms the image into two-dimensional space where the angles and positions of lines each correspond to an axis. The most likely lines are characterized by the brightest points in Hough space. The user can specify a maximum number of lines in addition to a lower threshold intensity beyond which bright spots are no longer considered lines. Unfortunately the first two lines do not always correspond to viable beamwidth boundaries. As such, several lines are collected, and there is a need for a final Line Selection step.

Line Selection is carried out by analyzing the area between each pair of lines, and applying what we know about the region corresponding to the ultrasound beamwidth in order to score that particular pair

of lines. Each criteria was scored on a scale of 0 (low correspondence) to 1 (high correspondence), and then the scores were all multiplied. The pair of lines with the highest overall score would then be considered to contain the beamwidth. The scores were as follows:

- Intensity score: The average intensity of pixels within the beamwidth is expected to be relatively high compared to the rest of the image. The average intensity was divided by half the highest intensity in the image, then bounded between 0 and 1, to compute this score.

- Parallel score: The two lines were expected to be parallel. This score was taken as the dot product of the normalized vectors that represent each of the lines. This was chosen because it exhibits the desired property of having a 0 result for lines that are absolutely not parallel (perpendicular) and a 1 result for lines that are absolutely parallel.

- Uniform score: If two lines include more than one band, there will be a very high standard deviation. The score should in this case be very low. When the two lines include a single region of reasonably consistent intensities, then the standard deviation should be reasonably low, and the score should be higher. This score is calculated as 1/stdev.

- Direction scores: Each pair of lines is scored as a product of single line direction scores. For the single line direction score: each line is scored according to whether it is in a region where the pixels above in the image are higher than the pixels below (negative score) or a region where the pixels below are higher in intensity than the pixels above (positive score). If the pixels are reasonably consistent on both sides of the line, the score for an individual line is approximately 0. The final direction score for the pair of lines is reported as -1 * the direction score for the first line * the direction score for the second line.

2.3. Results

As mentioned earlier, there is no gold standard or “Ground Truth” to compare the results of the algorithm against. The only reliability metric is the points that were manually segmented by Thomas K. Chen. As noted earlier, there may be slight bias in any manual segmentation.

Generally, in relatively noise-free data, the automatic algorithm detected lines that were off by only 2-4 pixels of the manually segmented lines. Occasionally lines outside this region were selected as well. The resulting band region in these cases, however, could not be dismissed as incorrect since we do not know the true band region. Whatever relatively small amount of noise was present can probably account for the variation in results. Figure 2.3.1 shows a typical segmentation result.

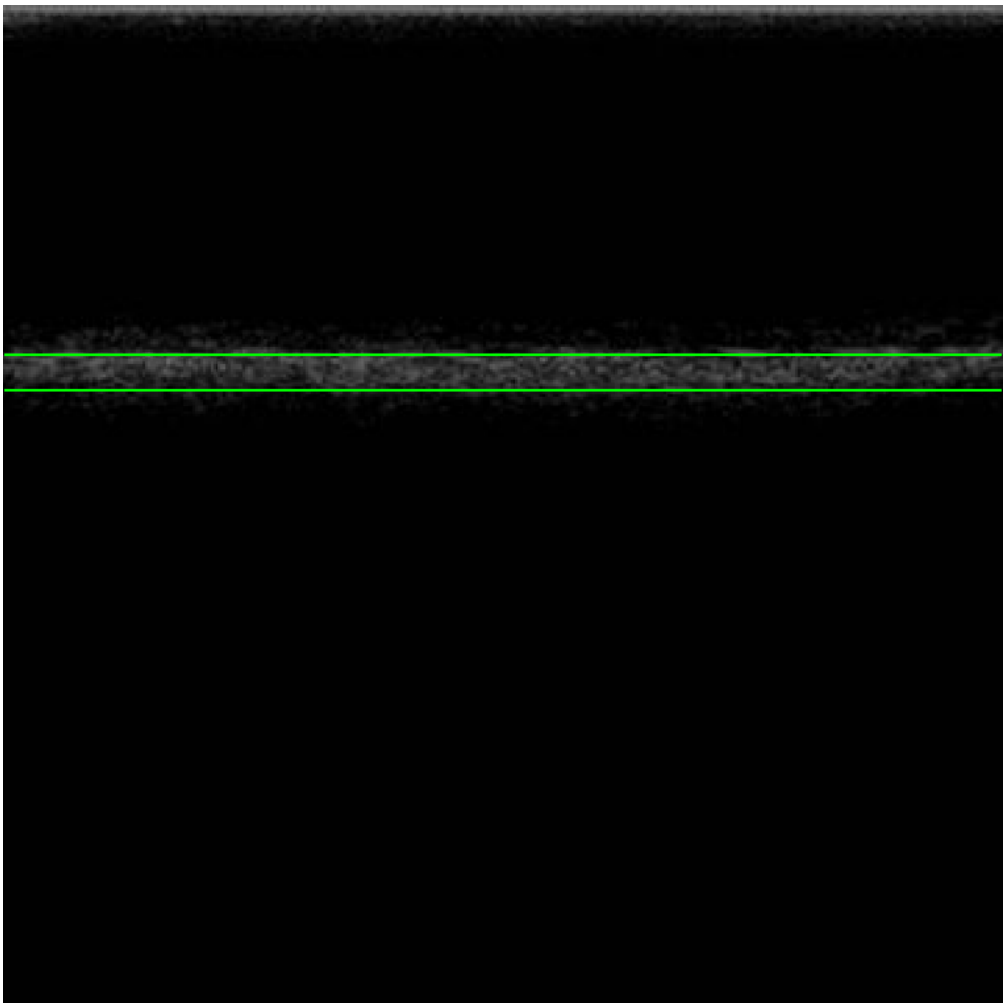

Figure 2.3.1: Typical Segmentation Result for relatively non-noisy data, the beamwidth boundaries are shown in green

When a dataset with much more noise was used, the results seemed much less consistent (Figure 2.3.2). Errors are introduced because some lines that should definitely not exist are detected (false positives), and some lines that should definitely exist are not detected at all (false negatives). False positive lines are particularly dangerous because if the region between them exhibits the criteria that are being searched for, even if it is all caused by noise, then an incorrect pair of lines may be selected instead of a reasonable pair of lines.

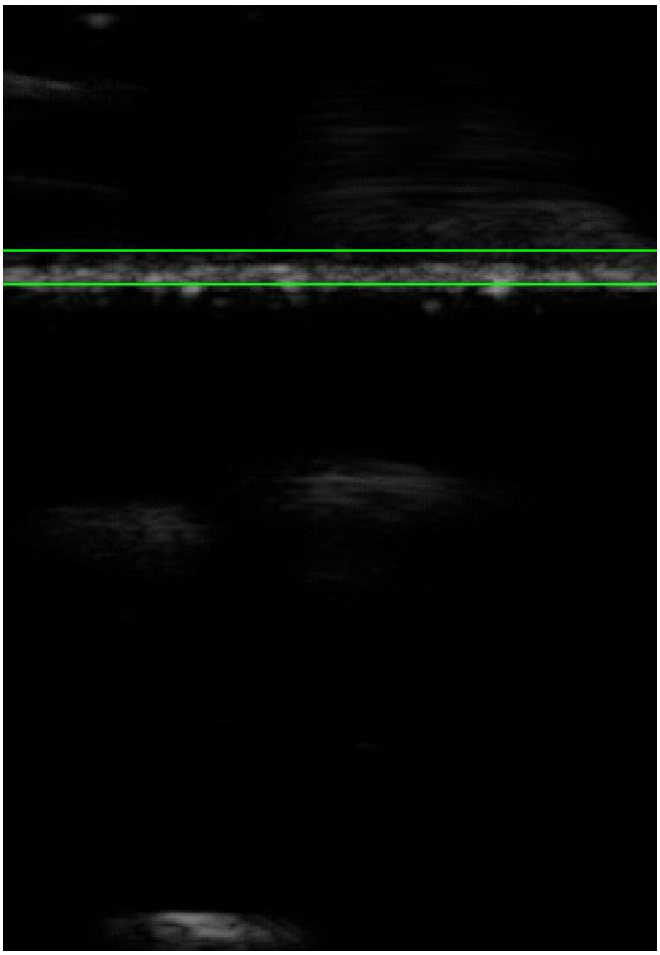

Figure 2.3.2: Typical Result on noisy data, the beamwidth boundaries are shown in green

2.4. Conclusion

After discussions with the project supervisors, it was concluded that the data collected for the purposes of generating an ultrasound beamwidth profile were too noisy to be reliable. The noise interfered with the line detection algorithm, and also made it difficult to verify the results that were obtained.

Furthermore, in order to generate a reliable profile, additional information about the position of the transducer would be needed. To my knowledge, this information does not exist for the datasets used in this project.

Finally, some of the images showed a band that was tilted slightly diagonally. This may be a result of the transducer slipping. In this case, the transducer was not constrained along a single axis, which means the beamwidth may not be uniform across the image.

To conclude this chapter of the report, we would like to propose the following improvements to the Data Collection phase in order to obtain results that are more reliable and more usable:

- The plane in the phantom must be of a material that is acoustically scattering, yet does not exhibit refractions within its own surface. The signals picked up by the transducer must be clear and distinct for there to be a reliable segmentation. Hydrofoam is the ideal material for this purpose [5].

- If complete removal of refraction is not possible, it would be beneficial to know in advance how the refraction affects the quality of ultrasound images. This may require for new experiments to be designed.

- The procedure should not ignore the thickness of the material. This should be accounted for in calculating the final beamwidth.

- The position and orientation of the ultrasound transducer must be tracked, that way the beamwidth can be mapped to the depth. This may help when it comes to visual validation of the segmented beamwidths because they can also be mapped to the original structure.

- If the transducer is not already constrained along a single axis of movement, then it should be. It was unclear from the procedure and experimental setup whether this was the case or not.

While all of these would improve the quality of the experiment, the whole process lacks any kind of Ground Truth so it is impossible to verify any results. Ultimately it will be impossible to be completely confident of the results of any algorithm that segments images acquired from this process.

For the purposes of having an estimate of beamwidth, we used the results published by Pekari et al. [2,3]. These were the result of manual segmentation of fiducials, so we can expect for some bias to be present. It is, however, indisputably better to use this as an error estimate than to assume no localization error at all.

3. Development of a Hole Filling Algorithm for Volume Reconstruction

3.1 Background

There are many applications of Volume Reconstruction, such as cross-modality registration, surface mesh generation, and the arbitrary slice visualization.

Cross modality registration (Figure 3.1.1) is a way where the information from different types of medical images (such as CT, MRI, or ultrasound) can be combined. CT scans provide images that are of high enough quality to produce detailed meshes of bones. Unfortunately CT scans expose the patient and medical staff to radiation, so using it in real-time during a medical intervention is generally unrealistic. Ultrasound images can provide real-time visualization of a patient's anatomy because it doesn't produce radiation, but it can't provide images that are usable for mesh construction because of noise and artifacts. Cross modality registration helps to get the best of both worlds - a mesh generated from a pre-operative CT scan can be fit to ultrasound data that is collected in real-time. A volume may provide a better registration than simple tracked ultrasound images.

Figure 3.1.1: Registration of a bone model to tracked ultrasound [8].

Surface mesh generation is the process by which a triangulated model of an anatomical structure is generated from an ultrasound volume. A threshold value is set, and an isosurface is calculated within the volume for that threshold value (Figure 3.1.2). A mesh is then generated for that isosurface. In cases where one needs a three-dimensional structure in virtual space, but where a CT or MRI scan is inappropriate, surface mesh generation from an ultrasound volume may be the only option.

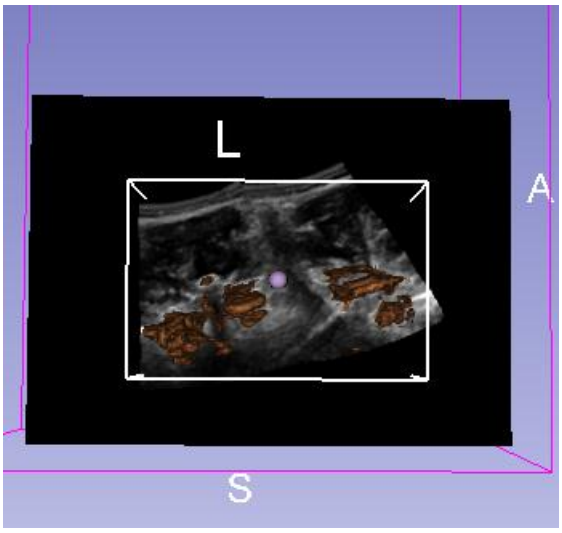

Figure 3.1.2: An isosurface (brown) is shown overtop an ultrasound image.

Finally, ultrasound volumes allow for arbitrary slice visualization (Figure 3.1.3) of the ultrasound image data. In traditional ultrasound imaging, a medical professional would only be able to see one image slice of data at a time, and only from the orientation in which it was collected. Volume reconstructions allow for a user to see, for example, an image slice that intersects all previously-collected image slices in a region of interest.

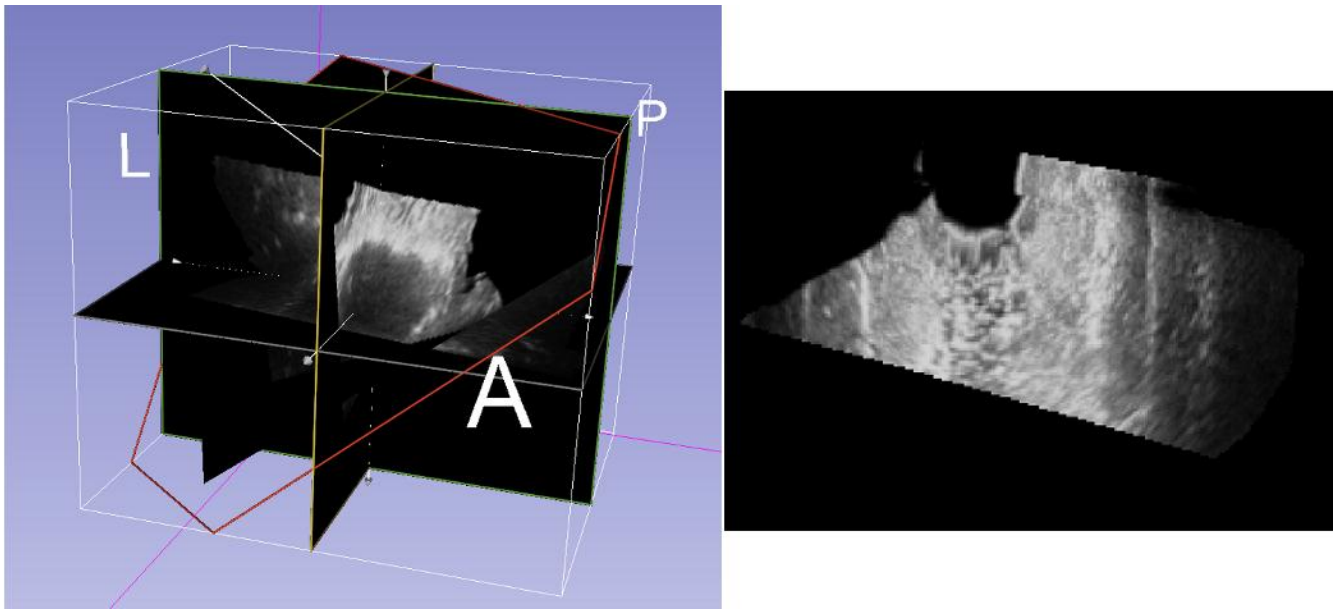

Figure 3.1.3: Arbitrary slice visualization. Left – the slice with a red outline is oriented arbitrarily with respect to the volume axes. Right – the arbitrary slice is shown as an image.

To our knowledge no free, working and open-source implementation of volume reconstructor software existed. The CISC 499 project therefore involved producing volume reconstruction software that would be made free and open-source. The implementation would require a minimal amount of knowledge on the part of the end-user and will be packaged as part of a toolkit that is used in several labs.

A literature survey revealed that there are many different ways to produce volume reconstructions, which in turn can have many variations. Solberg (2007) [9] proposed to classify existing algorithms

into three different general categories:

1. Pixel-Based Methods
2. Voxel-Based Methods
3. Function-Based Methods

Pixel-Based Methods involve first a Distribution Step where the pixels from a 2D ultrasound image are inserted into a 3D volume. There is often a second Hole-Filling Step, where holes in-between slices are filled with interpolated values.

Voxel-Based Methods involve iterating through all voxels of a volume and inserting a value based on the information from a subset of the original ultrasound images. These images might, for example, need to be within a certain distance of a voxel being calculated.

Function-Based Methods involve using the original ultrasound images to generate a mathematical function to describe the ultrasound data in the local neighborhood.

We decided that a Pixel-Based Method would make the most sense for us to implement since it allows for a great degree of modularity. The volume reconstruction algorithm will likely need to be decided based on the specific application. With a Pixel-Based Method, it would be easy to implement various Distribution Step algorithms or different Hole-Filling Step algorithms that can be selected by the user, and the sheer variety in options will help to ensure that even applications we did not consider might be possible.

Reverse tri-linear interpolation [10] (Figure 3.1.4) is the Distribution Step we used. There are two volumes: the Reconstructed Volume holds the voxel intensity values, and; the Accumulation Buffer holds the accumulated weight for that voxel. A pixel is found to be between eight voxels in three dimensional space. Its value will be contributed to each of those voxels with a weight depending on its relative distance to that voxel, and the cumulative weight for that voxel will be stored in the Accumulation Buffer. A voxel that has had more slices intersect it will be weighted more heavily in the future when other pixels are distributed into it. It has been claimed that this method is equivalent to a Function-Based Method called the Radial Basis Function, which is known to produce very high quality results at high computational expense. Reverse tri-linear interpolation is more computationally expensive than other Pixel-Based Methods, but it is still possible to perform real-time volume reconstruction using this method [10].

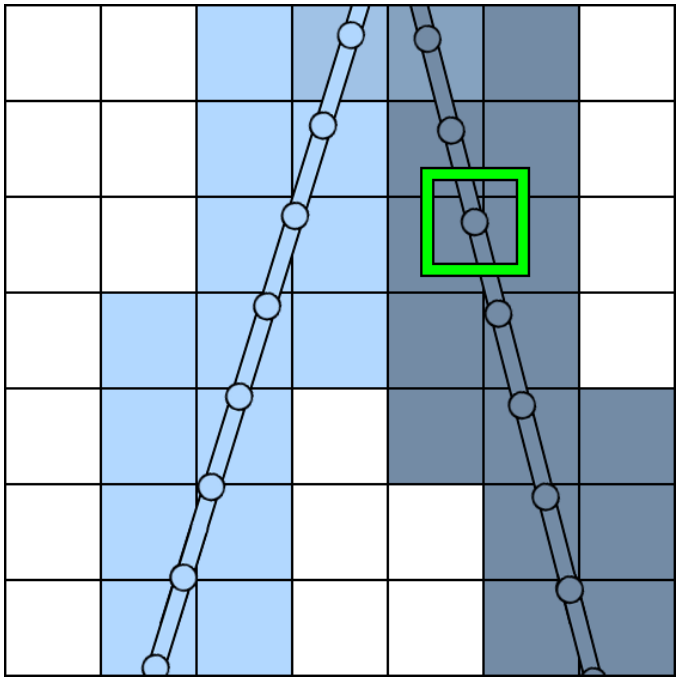

Figure 3.1.4: The distribution of a 2D pixel into a 3D volume is shown by a green box.

There are, once again, various Hole-Filling methods [9,11]. These can involve using kernels to distribute a single voxel over a larger area [12], using kernels to fill a hole voxel with a value interpolated from a surrounding neighborhood [13], or finding a number of nearest neighbors and using some median [14], mean, or maximum [15].

3.2. Methods

Weighted Average over a kernel region

We iterated through each of the hole voxels and applied a method similar to San José-Estépar (2003) where a distance weighted kernel was applied to a region around a hole voxel [13]. We essentially used an average over a box-shaped kernel region (Figure 3.2.1) that had two separate weights:

- We first applied a Gaussian Distance kernel. The reasoning for this is voxels that are farther away are less likely to share similar value to the hole.

- We used the Accumulation Buffer produced by reverse tri-linear interpolation [10] since voxels that are intersected by more image slices are more likely to be accurate.

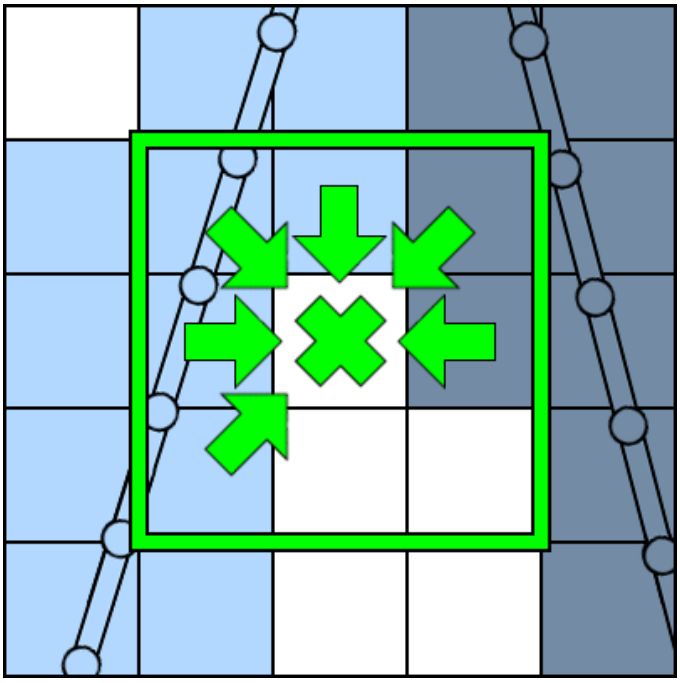

Figure 3.2.1: A hole is filled using a weighted average of the values around it

Parameters

There are several parameters that could be chosen by the user. These are:

- Size of the kernel region: The x, y, and z dimensions of the box-shaped kernel region in terms of voxels
- Standard deviation of the Gaussian distance weight: The standard deviation of the kernel for each of the x, y, and z axes, once again in terms of voxels
- Minimum ratio of known voxels to fill a hole: If the ratio of known voxels in the kernel region is less than this value, the hole shouldn't be filled because we don't have enough information.

Variable Kernel Size

If one voxel cannot be filled by a particular kernel due to lack of information, then we thought it should search a larger neighborhood (Figure 3.2.2). As such, multiple kernels can be defined. If the hole is still not filled after all kernels have been tried, it will be left as a hole with intensity value 0.

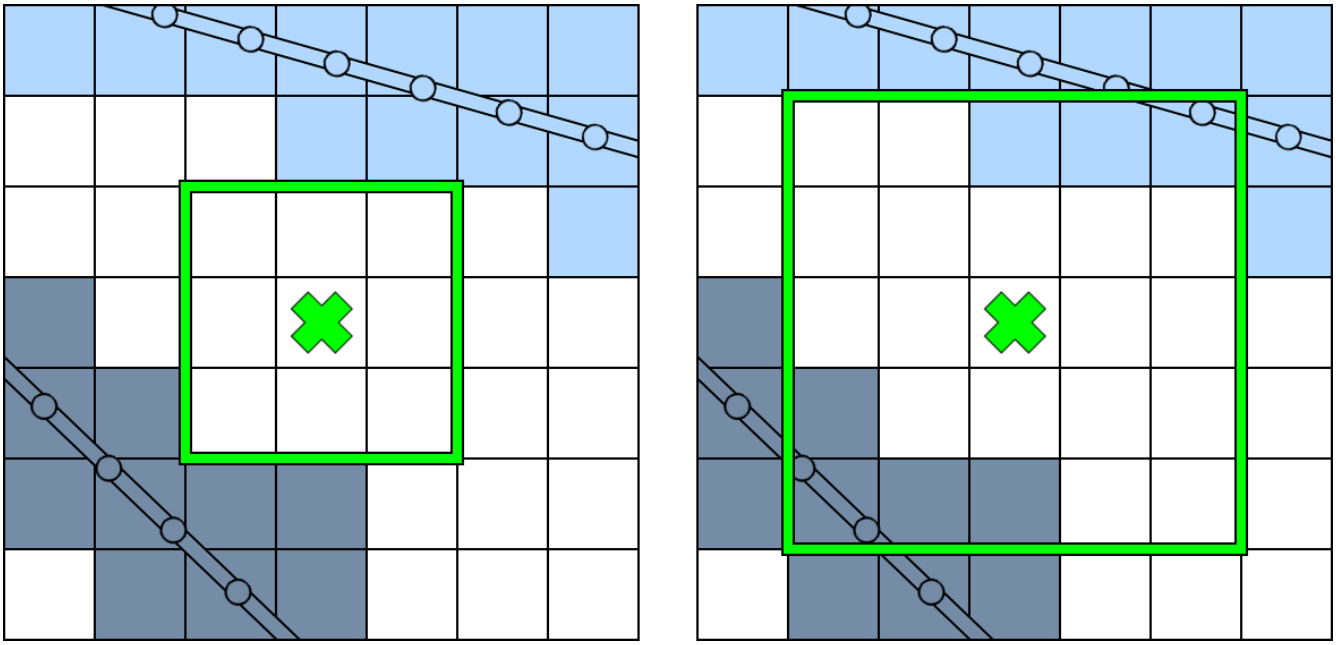

Figure 3.2.2: Left – there is inadequate information in a 3x3x3 kernel for interpolation. Right – there is sufficient information for interpolation after expanding the kernel region.

Quality Metrics

Error metrics are needed in order to determine how reliably the volume can be reconstructed. We first need a Ground Truth to which we can compare our volume reconstructions. We propose to collect a dense set of ultrasound images and to insert them into a 3D volume to be our Ground Truth. We will introduce holes by inserting only every 4^{th} slice into a volume. This simulates uniformly faster probe movement, and is therefore an accurate simulation of a real scenario.

We propose to use two methods to compare reconstructed volumes to the Ground Truth: Visual Assessment, and Mean Absolute Error (MAE) [14].

Visual Assessment is a practical, qualitative metric. From a practical point of view, one needs to be able to identify the features in the ultrasound volume, and that can currently only be done visually. From a qualitative perspective, visual assessment will be able to tell us if the holes have been filled, and if they have been filled with realistic values. There is, however, the potential for bias and as such another error metric is needed.

Absolute Error has been used in studies [11, 14] to quantitatively measure the accuracy of a volume reconstruction. We propose to use this metric solely on hole voxels since those are what we are trying to reconstruct. Any error that is present outside the holes is not representative of the hole-filling algorithm's performance. In addition, voxels that are filled outside the Ground Truth are considered to be outside the volume of interest, which in clinical cases would be well-defined, and can therefore be safely ignored for the purposes of our testing.

Data Collection

Dense ultrasound datasets were collected on simulated tissue phantoms at Queen's University for the purposes of testing the hole-filling algorithm (Figure 3.2.3). In addition, we tested the volume reconstructor on external data from the [various universities].

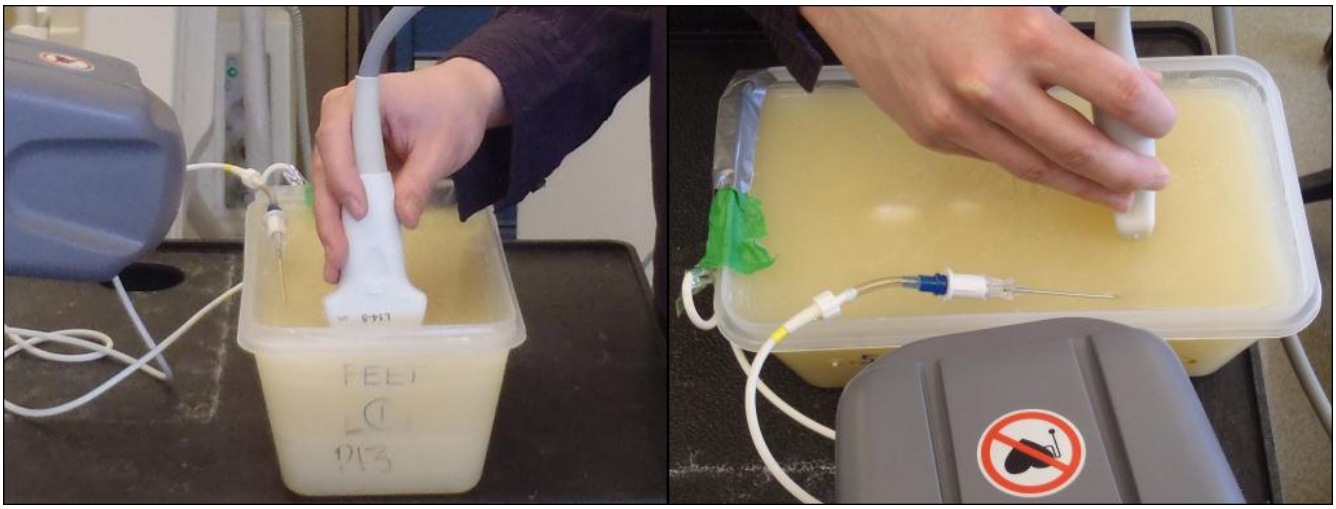

Figure 3.2.3: Ultrasound data is collected on a spine phantom at Queen's University.

3.3. Results

Several ultrasound datasets were collected. Volume reconstruction was carried out on many of these datasets to evaluate the hole-filling algorithm, and also to see how the volume reconstructor performed as a unit.

Qualitative Analysis

Visual assessment was carried out on reconstructed volumes. Generally, volumes that use hole filling are more usable than images that do not use hole filling. Figure 3.3.1 below shows typical examples of images that had holes filled using our algorithm. It can be seen that the holes are indeed filled, as intended. Smaller holes were filled with reasonable, continuous values, as we had hoped, but larger holes often presented seams.

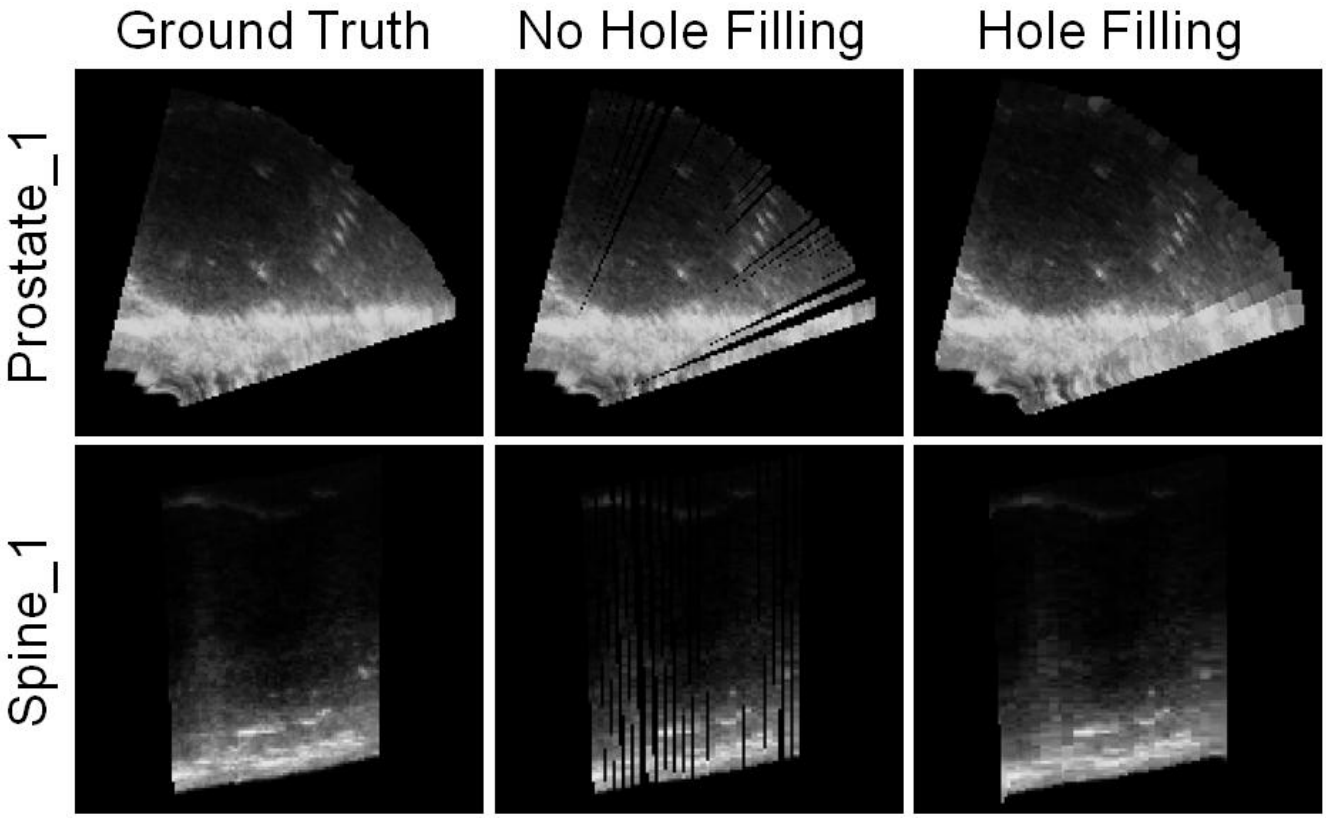

Figure 3.3.1: The Ground Truth is compared to the volume reconstructions with and without hole filling for different anatomical structures.

Some of the datasets, when reconstructed, resulted in very visible lines that were present regardless of whether hole filling was used. These datasets usually involved more than one sweep of data collection with the ultrasound probe. It can be reasoned that these voxels exist because two or more sweeps inserted conflicting values during the Distribution Step. The conflict may exist because the curvilinear transducer, for example, didn't make contact with the tissue surface. It may be that using a weighted average is inappropriate for cases when more than one sweep of a region is collected. We did not consider these datasets for evaluating the hole-filling step since there was a clear error in either the distribution step or in data collection.

In external datasets where the dataset consisted of very sparse images, there was simply not enough information to generate appropriate values to fill holes with. It should be noted, therefore, that the hole-filling algorithm is not a replacement for adequate data sampling, but only a supplement to help when the lack of data is unintentional.

Quantitative Analysis

We confirm that the values in hole voxels are now more appropriate after hole filling by comparing the Mean Absolute Error for hole-filled reconstructions against those for non-hole-filled reconstructions (Table 3.3.2). The hole-filled reconstructions consistently have less error than the non-hole-filled reconstructions. This can be further confirmed by examining histograms of hole voxel intensity error (Figure 3.3.3). This means that the hole-filled volume reconstructions are closer to the ground truth than the non-hole-filled volume reconstructions.

Dataset	No Hole Filling	Hole Filling
Spine_1	**17.10**	**2.37**
Prostate_1	**64.57**	**8.47**

Table 3.3.2: The MAE for volume reconstructions with and without hole filling.

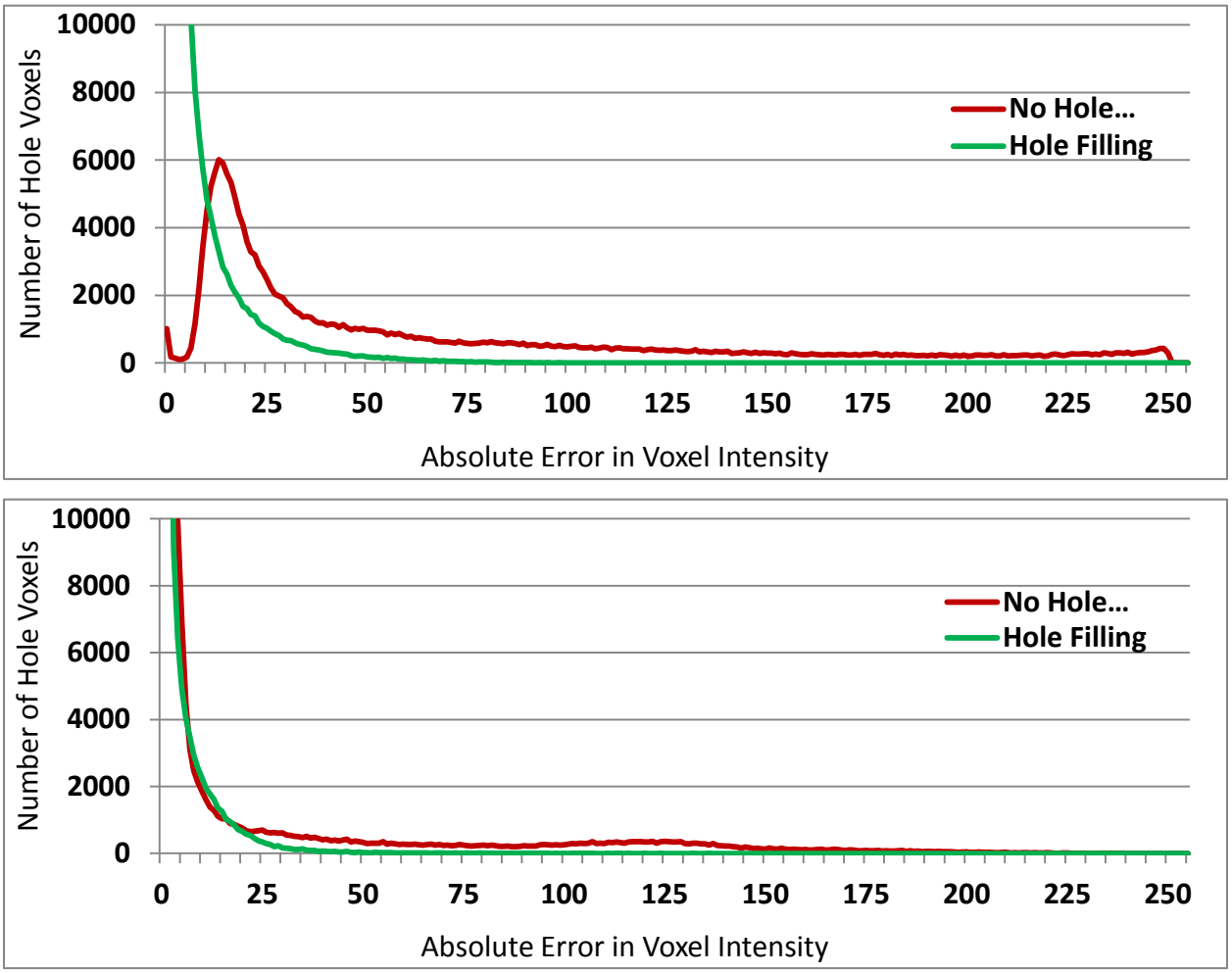

Figure 3.3.3: Histograms showing the absolute error distribution in voxel intensities for volume reconstructions with and without hole filling. *Above* – Prostate_1 reconstructions. *Below* – Spine_1 reconstruction.

3.4. Conclusions

We have implemented a free, open-source volume reconstructor. We have shown that the hole-filling algorithm enhances the accuracy of 3D volumes. The hole-filling algorithm should not, however, be regarded as a replacement for adequate data sampling, but should only be considered as being a failsafe for when a user unintentionally collects too little data.

The Distribution Step performs reliably on datasets where one sweep of ultrasound images is collected, but does not perform as well when multiple sweeps are collected. As such, other distribution steps need to be considered, and the user may need to be given the option to select the method based on how the data was collected.

We have not yet been able to compare the results of our hole filling implementation to the results of other hole filling implementations. In order to do this, the different methods should be compared using the same Distribution Step algorithm, the same datasets, and the same quality metrics.

Work on the volume reconstructor will continue during the month of April.

4. References

[1] Goldstein, A., and Madrazo, B.L. Slice Thickness Artifacts in Gray-Scale Ultrasound. Journal of Clinical Ultrasound. Vol. 9, pp 365-375, Sep (1981)

[2] Peikari, M., Chen, T.K., Burdette, C., and Fichtinger, G. Section-Thickness Profiling for Brachytherapy Ultrasound Guidance. SPIE Medical Imaging. (2011)

[3] Peikari, M., Chen, T.K., Lasso, A., Heffter, T., Fichtinger, and G., Burdette, C. Characterization of Ultrasound Elevation Bandwidth Artefects for Prostate Brachytherapy Needle Insertion. MSc thesis. Queen's University, Kingston, ON, Canada. (2011) Qspace. Web. 10 November 2011.

[4] Gonzalez R.C., Woods R.E., and Eddins S.L. Digital Image Processing Using MATLAB. Pearson-Prentice Hall. 2004.

[5] Hedrick W.R., Hykes D.L., and Starchman D.E. Ultrasound Physics and Instrumentation, fourth ed. Elsevier Mosby, Missouri. 2005.

[6] Czerwinski R.N., Jones D.L., and O'Brien W.D. Detection of Lines and Boundaries in Speckle Images. IEEE Transactions on Image Processing, vol. 7, no. 12. (1998)

[7] Czerwinski R.N., Jones D.L., and O'Brien W. D. Detection of Lines and Boundaries in Speckle Images - Application to Medical Ultrasound. IEEE Transactions on Medical Imaging, vol. 18, no. 2. (1999)

[8] Khallaghi S., Mousavi P., Gong R.H., Gill S., Boisvert J., Fichtinger G., Pichora D.R., Borschneck D.P., and Abolmaesumi P., Registration of a Statistical Shape Model of the Lumbar Spine to 3D Ultrasound Images. Medical Image Computing and Computer-Assisted Intervention (MICCAI), Beijing, China, Springer, pp. 68-75. (2010)

[9] Solberg O.V., Lindseth F., Torp H., Blake R.E., and Hernes T.A.N. Freehand 3D Ultrasound Reconstruction Algorithms – A Review. Ultrasound in Med. & Biol., Vol. 33, No. 7, pp. 991–1009. (2007)

[10] Gobbi D.G., and Peters T.M.. Interactive intra-operative 3D ultrasound reconstruction and visualization. Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI), Tokyo, Japan: Springer, pp. 156 –163. (2002)

[11] Rohling R., Gee A., and Berman L. A comparison of freehand threedimensional ultrasound reconstruction techniques. Medical Image Analysis, vol. 3, no. 4, pp. 339-359. (1999)

[12] Ohbuchi R., Chen D., and Fuchs H. Incremental volume reconstruction and rendering for 3-D ultrasound imaging. Proceedings of Visualization in Biomedical Computing, Chapel Hill, NC, USA: SPIE, pp. 312-323. (1992)

[13] San José-Estépar R.S., Martin-Ferandez M., Caballero-Martinez P.P., Alberola-Lopez C., and Ruiz-Alzola J. A theoretical framework to threedimensional ultrasound reconstruction from irregularly sampled data. Ultrasound Med Biol vol. 29 pp. 255-269. (2003)

[14] Dewi D.E.O., Wilkinson M.H.F., Mengko T.L.R., Purnama I.K.E., van Ooijen P.M.A., Veldhuizen A.G., Maurits N.M., and Verkerke G.J. 3D Ultrasound Reconstruction of Spinal Images using an Improved Olympic Hole-Filling Method. ICICI-BME, pp 1-5. (2009)

[15] Nelson T.R., and Pretorius D.H. Interactive acquisition, analysis and visualization of sonographic volume data. Int J Imag Systems Technol vol. 8 pp. 26-37. (1997)