3D Point Cloud Upsampling for Accurate Reconstruction of Dense 2.5D Thickness Maps

Bradley Skinner, Teresa Vidal-Calleja, Jaime Valls Miro, Freek De Bruijn, and Raphael Falque
Centre for Autonomous Systems,
Faculty of Engineering and Information Technology
University of Technology Sydney
Bradley.Skinner@uts.edu.au, teresa.vidalcalleja@uts.edu.au, Jaime.VallsMiro@uts.edu.au
Freek.DeBruijn@uts.edu.au, raphael.h.guenot-falque@student.uts.edu.au,

Abstract

This paper presents a novel robust processing methodology for computing 2.5D thickness maps from dense 3D collocated surfaces. The proposed pipeline is suitable to faithfully adjust data representation detailing as required, from preserving fine surface features to coarse interpretations. The foundations of the proposed technique exploit spatial point-based filtering, ray tracing techniques and the Robust Implicit Moving Least Squares (RIMLS) algorithm applied to dense 3D datasets, such as those acquired from laser scanners. The effectiveness of the proposed technique in overcoming traditional angular aliasing and corruption artifacts is validated with 3D ranging data acquired from internal and external surfaces of exhumed water pipes. It is shown that the resulting 2.5D maps can be more accurately and completely computed to higher resolutions, while significantly reducing the number of raytracing errors when compared with 2.5D thickness maps derived from our current approach.

1 Motivation

The remaining wall thickness of water pipes is a deciding parameter in the likelihood of sudden failure, particularly for the case of large trunk mains. Sensibly, the prediction of when, where and why these sudden and often catastrophic failures occur is of paramount importance to water utilities all over the world, and pipe thickness maps are indeed widely employed to assess the extent of the constituent wall material in large water mains. Thickness maps derived from the internal and external surfaces of exhumed water pipes are essentially 2.5D representations similar to the elevation maps used in robotics, cartography, geophysics, aerial photography etc. An accurate, precise and complete high resolution 2.5D thickness map derived from the acquired dense, irregular point cloud surface data is thus desirable for the implicit representation, or ground truth, of actual pipe sections. The new processing pipeline proposed in this work overcomes most of the current limitations faced when producing 2.5D thickness maps, which tend to contain angular aliasing effects, corruptions of the map with finer voxel resolution and errors resulting from unfiltered point clusters.

2 Introduction

There are currently several different condition assessment (CA) techniques available which aim at attaining remaining pipe wall thickness as the first step to evaluate the remaining life of a water pipe section. This information is subsequently used by water utilities to better target their critical main renewal programs and reduce negative impacts on customers [Miro *et al.*, 2014]. For metallic pipes the condition assessment techniques include those based on Magnetic Flux Leakage (MFL) [Wijerathna *et al.*, 2013], Broadband Electro-Magnetics (BEM) [Ulapane *et al.*, 2014], and the remote field Eddy current technique (also known as Remote Field Technique, or "RFT") [Atherton, 1995]. All of these CA sensor technologies are capable of providing a 2.5D map of varying resolution for the remaining pipe wall thickness. In addition, data driven approaches has been recently proposed to train probabilistic models in an attempt to interpret raw pipe wall measurements from these sensor technologies and produce a 2.5D thickness map. With this approach a Gaussian Process (GP) model [Rasmussen and Williams, 2005] is trained using both simulation and real CA sensor data [Wijerathna *et al.*, 2013] and [Ulapane *et al.*, 2014]. With new sensor data provided to the GP model an inferred pipe wall thickness can be computed as a 2.5D map for interpretation and Bayesian fusion [Vidal-Calleja *et al.*, 2014].

The effectiveness of both the inferred and measured methodologies used to produce 2.5D pipe wall thickness maps must be validated using an accurate ground truth representation of the same pipe section. Currently, ray tracing is performed using the point cloud acquired from the 3D laser scanning procedure. This point cloud is dense and can contain for example a large number (>20Million) points for a single pipe section of length=1500mm and outer diameter of ϕ=660mm. However, the non-uniformity of this point cloud greatly restricts the minimum voxel resolution, which is required in the ray tracing algorithm to determine remaining wall thickness. As a result, when the voxel resolution is decreased the resulting 2.5D remaining

wall thickness map contains angular aliasing effects and corruptions resulting from decreasing number of ray tracing intersections of the voxelised point cloud. In addition, the current approach does not employ any spatial filtering methods to remove outliers from the point cloud, resulting in errors in the computed thickness of the 2.5D remaining wall thickness map.

This paper describes a processing pipeline which takes as input the 3D laser scanner data acquired from the internal and external surfaces of exhumed water pipes. We perform spatial filtering on the 3D point cloud to remove outliers that are not part of the surface structure and voxelisation to create a uniformly sampled point cloud. To preserve the fine surface features of the internal and external pipe walls we employ the Robust Implicit Moving Least Squares (RIMLS) algorithm to generate a very dense and continuous isosurface [Oztireli *et al.*, 2009]. A computationally efficient ray tracing algorithm is then iterated along the central axis (x-axis), casting rays circumferentially around the voxelised representation of the isosurface to determine the distance between the inner and outer pipe walls. Using sub-millimeter voxel resolutions, allows for the computation of a dense and accurate 2.5D thickness map of the entire pipe section.

2 3D Point Cloud Acquisition

2.1 Laser Scanning of Exhumed Pipe Section

Selected sections of Cast Iron Cement-lined (CICL) water pipes are separated from the water main and then exhumed from the ground. To produce an accurate 3D point cloud as the ground truth representation and 2.5D thickness map, it is essential to perform scanning of the internal and external surfaces of the exhumed pipe section. The EXAscan® scanner (from Creaform) was used to produce the desired profile. This sensor suite contains a handheld 3D laser scanner with a resolution of 0.05±0.03mm and acquisition rate of 480,000 samples per second. It is also small enough to permit scanning of the inside surface of the exhumed pipe sections.

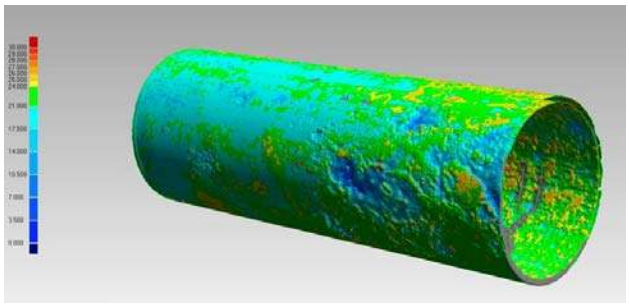

Figure 1: 3D Mesh of CICL pipe walls acquired from Exascan laser scanning suite.

The laser scanning process produces data in the form of a dense 3D mesh for the internal and external surfaces of the pipe, as illustrated in Figure 1. However, the vertices or points of the mesh are not uniformly distributed within the Euclidean coordinate frame. As such, the mesh contains areas of different point densities, which is directly related to the velocity of the laser scanner during the data acquisition phase. Before the laser scanning procedure is performed on the pipe walls, it is essential to remove all debris, the internal concrete lining and graphitization layer from the internal and external pipe walls. This is typically performed using high pressure grit-blasting tools and in the case of CICL pipes, results in the exposure of both the internal and external pipe wall surfaces consisting of cast iron only.

2.2 Voxel Grid Filtering

Typically, the raw point cloud data derived from the laser scanning procedure (§2.1) is of high dimensionality and has highly non-uniform point density. The non-uniformity in point density is due to variations with the velocity of the handheld laser scanner during the scanning process of a pipe section (the sampling rate of the laser scanning device is constant). As such, the raw point cloud will contain regions of highly non-uniform point densities. The regions of higher density are removed following application of RIMLS algorithm (§3.0) and we believe this is due to the RIMLS algorithm requiring point clouds with roughly uniform point densities [Oztireli *et al.*, 2009]. Regardless, this is an undesirable effect for our application as it results in a point cloud with large regions containing no point data. To circumvent this limitation a 3D voxel grid filter (VGF) is applied to the point cloud data, following the application of the bounding box filter. The voxel grid filter arranges a local 3D grid over the point cloud using a leaf size in the range of $0.8\text{mm} \leq leaf_{x,y,z} \leq 1.2\text{mm}$. The small leaf size reduces the amount of down sampling in the point cloud. In general, the VGF creates a 3D voxel grid (where a voxel grid is a set of small 3D cubes in space) over the input point cloud data. Then, in each voxel (i.e., 3D cube), all the points present will be approximated to a single centroid, computed on the average Euclidean distances within the voxel. The computation time of this approach is slower compared to approximating them with the center of the voxel, but it represents the underlying surface more accurately [Rusu and Cousins, 2011].

2.3 Spatial Point-based filtering

The resulting 3D point cloud of the internal and external pipe wall often contain noise induced artifacts, which are typically located around the ends of the pipe section. These noise induced artifacts are unwanted and feature in the point cloud as clusters of neighboring points, which are not actually part of the pipe surface. In order, to effectively remove these unwanted outliers, three different point-based filter techniques were developed using the Point Cloud Library (PCL) [Rusu and Cousins, 2011]. They include a bounding box filter, Radius Outlier Removal (ROR) filter and Statistical Outlier Removal (SOR) filter.

Since, the dimension in each of the 3D axis (x, y, z) is known for the pipe section the initial filter comprises of a simple rectangular prism known as a bounding box. The static dimensions for the bounding box are established to fully contain or 'bound' the cylindrical pipe section within a rectangular prism. Once the pipe section is effectively enclosed the filter can exclude (crops) all points outside of the rectangular prism. No translation or rotation of the point cloud is performed by the bounding box filter. However, due to the fundamental geometry of a rectangular prism bounding (enclosing) a cylindrical pipe section across its diameter and along its length, outliers

may remain in the point cloud after application of the bounding box filter. To remove any remaining outlier point clusters, two additional filtering steps are performed on the point cloud using a Radius Outlier Removal filter (ROR) and Statistical Outlier Removal (SOR) filter.

The Radius Outlier Removal filter is used to remove points from the point cloud based on the number of neighbors they have. The ROR filter iterates through the entire point cloud once, and for each point, retrieves the number of neighbors located within a sphere of predefined radius (r_{ROR}). A point is considered to be an outlier if it has too few neighbors enclosed within the sphere. The value of the radius (r_{ROR}) is computed using the K nearest neighbor search algorithm to determine the mean distance (point resolution) between a large number of randomly selected sample points and their associated nearest neighbor on their x-axis dimension only. Here, the value of K is an integer number of sample points selected uniformly at random from the point cloud. We empirically determined that using a subset of points consisting of 5.0% to 10.0% of the total number of points is representative and provides good computational performance.

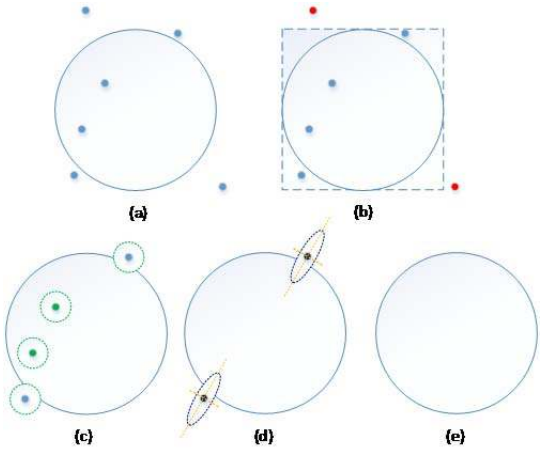

Figure 2: 2D illustration of Point Cloud Filtering: (a) Raw point cloud with outliers, (b) Bounding Box filter, (c) ROR Filter, (d) SOR Filter (e) Filtered point cloud (as convex hull).

The Statistical Outlier Removal filter is employed to remove unwanted outliers based on point neighborhood statistics. The SOR filter iterates through the entire input point cloud twice. For the first iteration it computes the mean distance that each point has to its nearest K neighbors. Here, the value of K is set according to the point resolution computed previously. Then, the mean and standard deviation of all these distances are computed to determine a distance threshold. During the next iteration of the SOR filter the points will be labelled as inlier or outlier if their average neighbor distance is below or above the distance threshold respectively. Figure 2 illustrates the effects of each filter from the raw input point cloud containing a number of outliers to the final point cloud with all outlier point clusters effectively removed.

3 RIMLS Upsampling

The RIMLS algorithm is a novel implementation of the non-robust Implicit Moving Least Squares (IMLS) algorithm. The IMLS portion of the algorithm is combined with non-linear local kernel regression (LKR) methods and techniques from robust statistics [Oztireli *et al.*, 2009]. The authors claim that RIMLS is capable of robustly handling outliers and high frequency features in point clouds. The algorithm is also designed to provide increased accuracy of the reconstruction in the case of globally and locally sharp features and to provide stability under sparse sampling. These major features of the RIMLS algorithm make it attractive for use in our application, which takes a uniformly sampled point cloud (§2.0) with the objective of generating an accurately upsampled point cloud that retains the fine details of the original pipe section. The projection of points onto the underlying RIMLS surface is done using a steepest gradient decent strategy [Oztireli *et al.*, 2009]. Evaluation of the scalar field $f(\mathbf{x_i})$ is performed using a simple weighted average over the nearest neighbors of $\mathbf{x}$ until some termination criteria is fulfilled. Both the RIMLS and Marching Cubes algorithm were implemented in Meshlab [Cignoni *et al.*, 2008].

3.1 Spatial weight function (MLS Filter Scale)

The RIMLS algorithm uses a C^3 continuous polynomial approximation of a Gaussian function as the spatial low pass filter:

$$\boldsymbol{\phi_i(x)} = \left(\mathbf{1} - \frac{\|x - x_i\|^2}{h_i^2}\right)^{\mathbf{4}} \quad (1)$$

where $x_i \in \mathbb{R}$ are sample points around point x, h_i is the weight radii the scale of which allows adaptation to local point density. It is established according to the relative local point spacing within the point cloud (§2.2). Typical values range from 1.4 to 4 times the local point spacing. We found that the default value of h_i = 2 provided good results.

3.2 Robust weight radii (MLS Sharpness)

The RIMLS algorithm uses two weight radii σ_r and σ_n. The σ_r term is used in another Gaussian weight function:

$$\boldsymbol{w(x)} = \boldsymbol{e}^{-\left(\frac{x}{\sigma_r h_i}\right)^2} \quad (2)$$

which is used to scale a residual term, representing a difference of projection distance to the computed surface. It is set locally as a fraction of the spatial weight radius h_i. The value of σ_r is scale independent and in this paper is we used σ_r = 0.5. Selecting a value for the σ_n term is more subjective and depends on the desired degree of sharpness. It defines the width of the filter used by the normal refitting weight function. This weight function is a Gaussian on the distance between two unit vectors, the current gradients and the input normal. The authors of RIMLS assume the norm of the gradient is very close to one, resulting in a typical range of $0.5 \leq \sigma_n \leq 2.0$. As such, lower values provide sharper results and higher values provide smoothing. In this paper, we selected a global value for σ_n = 1.0 for all upsampled point clouds.

3.3 Termination criteria (Projection Accuracy and Maximum Iterations)

During the projection step the RIMLS algorithm iterates until convergence is detected or the number of iterations

has been meet. We fix a constant number of iterations while keeping the convergence test as an early termination optimisation. In RIMLS, convergence of the projection accuracy is measured according to the relative difference of the refitting weights. Namely, $w(r_i^k)$ which is a measure of the weighted least squares minimisation during projection and $w_n(\Delta n_i^k)$ which is a measure of the difference between the predicted gradient and a sample normal of the IMLS surface. We used the default threshold of $t = 10^{-4}$ to terminate projections. The value of t is scaled by the mean point spacing (density) to get the actual threshold value. In addition, our application does not have any real time constraints and the scalar field is extremely large (~10^6 points), so we increased the maximum number of projection iterations from 15 to 30 and the maximum number of MLS fitting iterations from 3 to 5.

3.4 Sampling Grid Resolution

Before the RIMLS surface projection procedure can be performed, extracting a polygonal mesh of an isosurface from the 3D scalar field $f(\mathbf{x}_i)$ and associated vertex normals (n_i is the normal at sample point $\mathbf{x}_i$) is performed using the Marching Cubes algorithm [Lorensen and Cline, 1987]. The marching cubes algorithm uses a divide-and-conquer technique to locate a polygonal surface in a logical *cube* created from eight points; four each from two adjacent slices of the logical cube. It then computes the number of polygons required to represent the part of the isosurface that intersects this cube, then moves (or *marches*) to the next logical cube. The individual polygons are then fused into the desired surface. There exists 2^8 possible polygon configurations bounded by a cube, because there are 8 vertices in each cube and two states, inside and outside of the cube. As such, there are only $2^8 = 256$ ways a polygonal surface can interest the cube. The 256 cases are enumerated and form an index of surface-edge intersections. The final steps to compute the surface involves placing each vertex of the generated polygons on to the appropriate position along the edge of the imaginary cube by linearly interpolating the two scalar values that are connected by that edge. Additionally, the unit normal at each cube vertex is calculated using the method of central differences. Since, the gradient of the scalar field at each grid point is also the normal vector of a hypothetical isosurface passing from that point, linear interpolation is used to cast this normal vector to each the generated triangle vertices.

When the Marching Cubes algorithm performs polygonising of the scalar field, the resolution of the sampling grid determines the level (course or fine) of approximation for the computed isosurface. Marching Cubes uses a structured or uniform grid for the implicit high resolution surface reconstruction algorithm. Since our application does not have any real time constraints and the large computer memory subsystems available (>16GBytes), we employed grid sizes in the range of $1800 \leq R \leq 2400$. Using grid sizes within this range permitted the IMLS fitting function and projection step in the RIMLS algorithm to effectively upsample the pipe section point clouds by a factor of ×2 (R = 1800) and ×3 (R=2400).

4 Spatial Partitioning and Octree Search

Here we describe the spatial partitioning of the point cloud into a voxels (voxelisation) using the octree data structure and associated search mechanisms for determining the intersection of voxels when performing vector ray tracing. The Octree library in PCL provides efficient methods for creating a hierarchical octree data structure from point cloud data, which enables spatial partitioning and search operations [Rusu and Cousins, 2011].

4.1 Spatial Partitioning

Each node in the octree contains either eight or zero children. The root node describes a cubic bounding box encapsulating all the points in the entire input point cloud. At every higher level in the tree, this space becomes subdivided by a factor of 2 which results in an increasing voxel resolution. The pcl_octree implementation can automatically adjust its dimensionality to the size of the point data set and it also provides efficient nearest neighbor search capabilities. In addition, the leaf nodes of the octree data structure provide a mechanism to map voxel occupancy and point density per voxel methods.

The filtered (§2.2 and §2.3) and reconstructed/upsampled (§3.0) point cloud of a pipe section can now be fully mapped into the octree data structure with a specified voxel resolution. In PCL, the voxel resolution specifies the resolution at the lowest octree level. In other words, it defines the length of the sides of the smallest voxel cuboid which is represented as a leaf node in the octree. The 3D results of the computed octree structure for a single CICL water pipe section (l=1.50m, ϕ_{outer}=550mm) are illustrated in Figure 3 for 250,000 voxel cuboids and in Figure 4 using the centroids for 5.1million voxels.

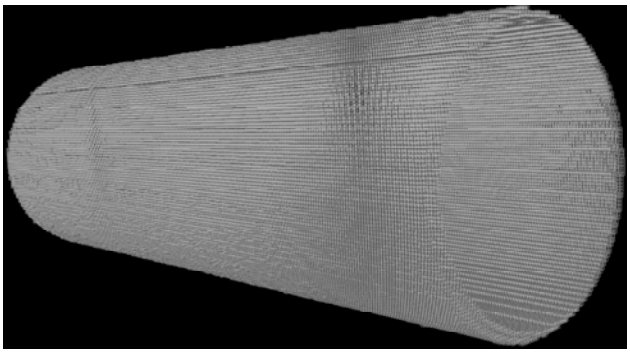

Figure 3: 250,000 voxel cuboids computed using PCL for a 1.50m length and 550mm diameter CICL water pipe section.

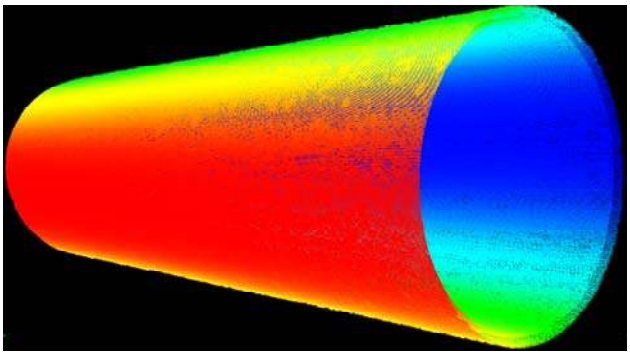

Figure 4: 5.1Million voxel centroids computed using PCL for a 1.50m length and 550mm diameter CICL water pipe section.

4.2 Searching the Octree for Raycasting Intersections

Once the octree has been constructed, we can simply cast a ray defined only by its origin and direction to get a vector of the centres of all the voxels that are intersected by the casting of the ray. We compute the number of occupied voxels in the octree to determine an upper bound on the maximum number of ray casting intersections that could occur when comparable voxel resolution and sampling rate is used. The octree leaf count and branch count are also computed to determine the size of the computed octree. The ray tracing scheme traverses the entire central x-axis (axial length) of the 3D pipe section in discrete steps (Δx) from a specified starting position (x_{start}) to a termination position (x_{end}), ensuring the complete length of a pipe section is covered ($x_{start} \leq \Delta x \leq x_{end}$). At every discrete Δx position, a ray is casted from the origin (Δx, 0, 0) in the outward radial direction to the limit of the cubic bounding box encapsulating all the voxels in the cloud. The entire circumference is discretely sampled from 0 to 2π, in steps of $\Delta\theta$ steps, as illustrated in Figure 5.

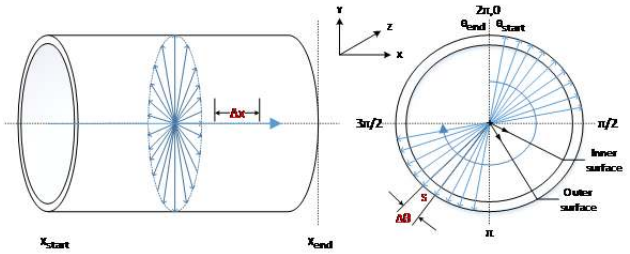

Figure 5: The ray tracing scheme traverses the entire axial length (x-axis) of a pipe section in discrete steps of Δx (mm) and casts rays in the outward radial direction in discrete angular displacements of $\Delta\theta^{o}$.

The size of Δx and the outer surface arc length ($s_{outer} = r_{outer} \cdot \Delta\theta$) are both fixed at half the voxel resolution of the voxelised point cloud, ensuring Nyquist criterion is minimally achieved. For each ray cast operation a vector of the centroids of all the voxels that are intersected by the casting operation is efficiently computed using a search method to recursively traverse the octree data structure [Revelles *et al.*, 2000].

5 Results

The initial voxel grid filtering (§2.2), spatial filtering (§2.3) and the coupled use of the marching cube surface reconstruction algorithm with the RIMLS algorithm (§3.0), were sequentially applied to the point cloud data produced directly from the laser scanning process. This did overcome the existing limitations arising from the non-uniformly dense point cloud (§1.0), which produces angular aliasing effects and corruptions for a decreasing (finer) voxel resolution in the resulting 2.5D thickness map of the pipe section.

A single CICL water pipe section of l=1.50m and ϕ_{outer}=550mm was used to produce the results in this paper. Figure 6 shows the total number of occupied voxels (octree leaf nodes) computed from the input point clouds when the *Raw Laser Data* and *MC+RIMLS Upsampled* data are mapped into the octree data structure with decreasing voxel resolutions.

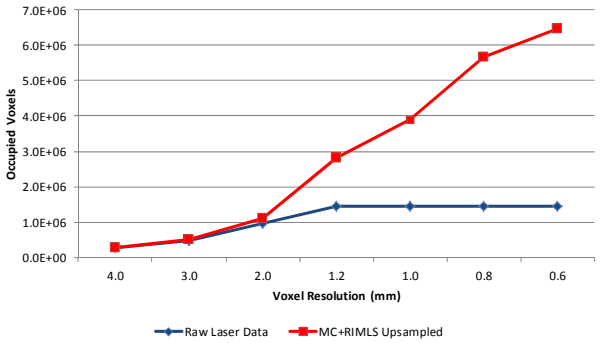

Figure 6: Total occupied voxel count (~octree leaf nodes) for decreasing voxel resolution.

Here, the number of occupied voxels in the octree continues to increase when the voxel resolution decreases for the MC+RIMLS upsampled point cloud, due to the increased uniformity and density of the point cloud. Whereas, the number of occupied voxels computed using the raw laser scanning data does not significantly increase beyond a 2mm voxel resolution.

It is the voxel occupancy that determines the success of voxel intersections from the ray casting operation. As such, Figure 7 shows the number of voxel intersections containing just two voxels (inner and outer pipe wall voxels as illustrated in Figure **5**) that are returned from the ray casting operation. Due to the increased uniformity and density of the MC+RIMLS upsampled point cloud the number of Voxel[2] intersections returned from the ray casting operation continuously increases with decreasing voxel resolution. Whereas, the limited number of occupied voxels in the point cloud from the raw laser scanning data severely restricts the Voxel[2] intersection count.

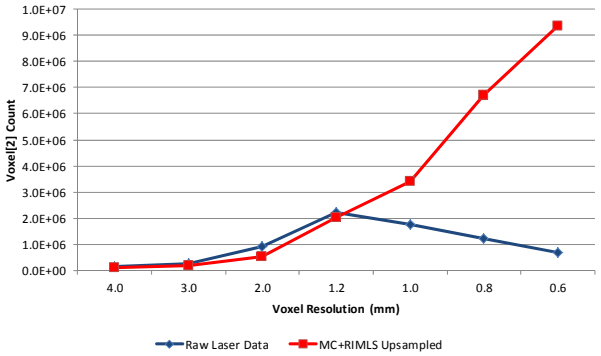

Figure 7: Voxel intersections containing only two voxels (Voxel[2]) in the return ray cast vector for decreasing voxel resolution.

Ray casting operations resulting in zero voxel intersections occur for two reasons. Firstly, if the original point cloud data does not contain information at a certain location, then voxels will not be mapped into the octree at this location. This is related to the initial data acquisition technique and not the post processing. Secondly, as the voxel resolution decreases the Δx and $\Delta\theta$ discrete ray casting parameters also decrease (§4.2), producing very fine grained raytracing. If the voxel occupancy does not increase the number of ray cast operations resulting in zero intersections will increase as shown in Figure 8.

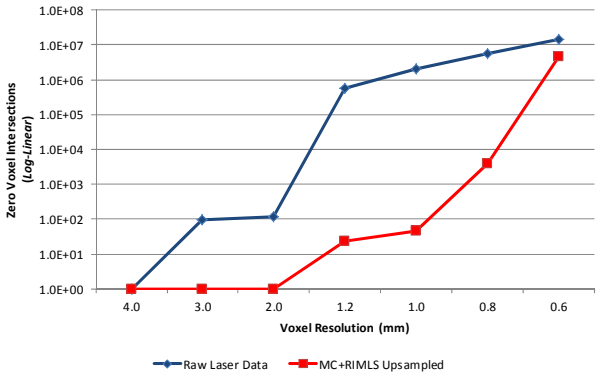

Figure 8: Number of ray casting operations resulting in zero voxel intersections for decreasing voxel resolution (Log-Linear).

Tracking voxel intersections returned from the raytracing operation can determine the effectiveness of the input point cloud as shown in the results of Figure 6 to Figure 8. However, transformation of the computed raytracing data into the required 2.5D thickness map clearly illustrates the angular aliasing and pixel (point) corruptions resulting from voxelisation of a non-uniformly dense point cloud. As such, Figure 10 (over page) shows the 2.5D thickness maps for MC+RIMLS upsampled data (A1-F1:top row) and Raw Scanner Data (A2-F2:bottom row) for the same CICL water pipe section (l=550mm - 1100m, ϕ_{outer}=550mm).

The 2.5D maps in A2-F2 are derived from the raw scanner data and contain an increasing number of holes or black pixels (z_p=0) resulting from the increasing number of zero voxel intersections as the voxel resolution decreases. For voxel resolutions less than 2mm the 2.5D maps are unusable as a representation for ground truth. Alternatively, it is only at a voxel resolution of 0.6mm that the zero voxel intersections become problematic for the MC+RIMLS upsampled data as shown in image F1. Additionally, the 2.5D maps in A2 and B2 contain a considerable amount of angular aliasing, which is evident from the presence of the horizontal banding artifacts. Since the MC+RIMLS upsampled data permits use of a much smaller voxel resolution, the angular aliasing artifacts are significantly reduced when compared with the 2.5D maps of C1, D1 and E1.

5.1 Filling Holes in 2.5D Thickness Maps

If the original point cloud data does not contain information at a particular location, then occupied voxels will not be mapped into the octree at this location and the resulting 2.5D thickness map will contain a zero thickness value(z_p=0) or *hole* at this location. In addition, zero voxel intersections from the raytracing operation also produce *holes* in the thickness map as illustrated in Figure 9 and discussed in (§5.0). For holes that have a diameter less than 10mm, we employ a morphological filter based on a specific 3x3 Gaussian filter kernel and iteratively apply convolution *only* in regions containing holes.

This method effectively fills existing holes using the information from the nine nearest neighbours. As such, the hole filling algorithm uses convolution to iteratively processes the entire image until all holes less than a specified diameter are filled using the 3x3 kernel k = [1 2 1; 2 0 2; 1 2 1]. Specifically, a temporary map of size ($m \times n$) is created through the application of convolution to pixels with zero value using the pixel weights defined in kernel k. A normalisation map of size ($m \times n$) is then generated to store the final weight of each updated pixel. The initial map ($m \times n$) is then updated as a normalised version of the temporary map and the process is repeated until all holes less than a specified diameter have been filled. This is the final step in the entire pipeline and produces a 2.5D thickness map that is accurate, precise and complete.

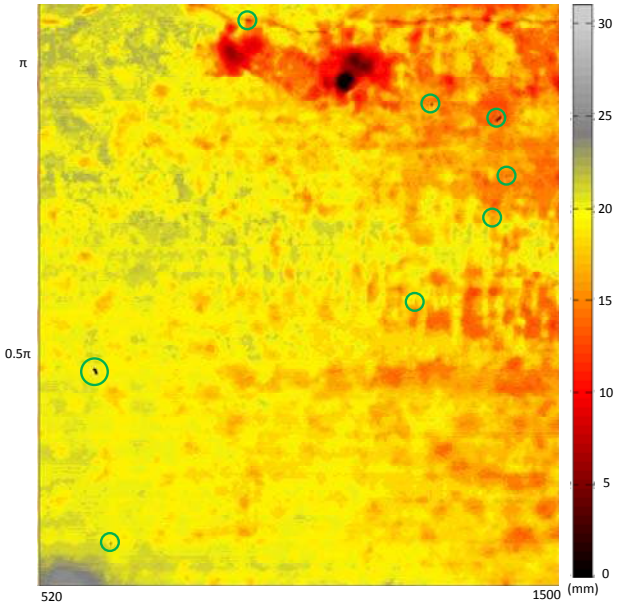

Figure 9: Green circles highlight the holes present in the 2.5D thickness map, which are caused by zero voxel intersections during the raytracing operation.

6 Conclusions

This paper described a robust, improved point cloud processing pipeline to compute 2.5D maps of arbitrary resolution. The proposed technique is tested to calculate the remaining wall thickness of exhumed cast iron cement lined water pipes from 3D laser scanner data acquired from the internal and external surfaces of the pipe. The computed 2.5D thickness map is being used as ground truth for comparison with inference-based learning methods. As such, the accuracy and precision of the ground truth is critical. However, prevailing 2.5D mapping methods suffer from significant angular aliasing and corruption artifacts. We overcome these limitations with a novel approach which introduces grid, radius and statistical spatial point-based filtering methods, the computation of an accurate isosurface using the Marching Cubes algorithm coupled with Robust Implicit Moving Least Squares (RIMLS) algorithm to produce a uniformly dense point cloud. A computationally efficient ray tracing algorithm can then be used to cast rays circumferentially around the voxelised representation of the isosurface to accurately determine the distance between the inner and outer pipe walls. As a result, we significantly increased the total number of occupied voxels, the total number of voxel intersections and significantly reduced the total number of

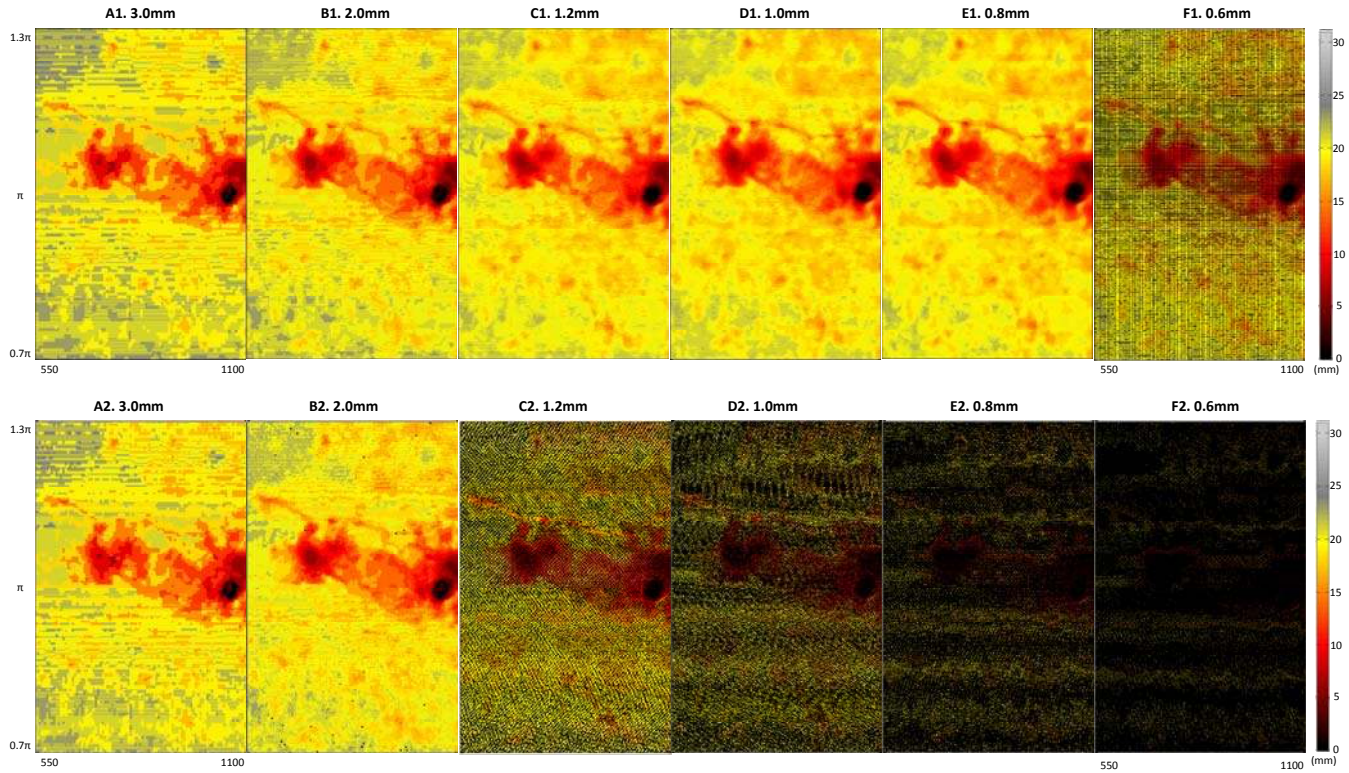

Figure 10: 2.5D Thickness Maps for MC+RIMLS upsampled data (A1-F1 top row) and Raw Scanner Data (A2-F2 bottom row), shown against decreasing voxel resolution (3.0mm-0.6mm). The sample pipe section is located axially between 550mm to 1100mm and radially between 0.7π and 1.3π.

zero voxel intersections (errors). In addition, we incorporated a 3x3 Gaussian filter kernel to iteratively apply convolution to the 2.5D thickness map to perform hole filling to finally produce an accurate, precise and complete sub-millimeter 2.5D thickness map.

Acknowledgements

This publication is an outcome from the Critical Pipes Project (www.criticalpipes.com) funded by Sydney Water Corporation, Water Research Foundation of USA, Melbourne Water, Water Corporation (Western Australia), UK Water Industry Research Ltd, South Australia Water Corporation, South East Water, Hunter Water Corporation, City West Water, Monash University, University of Technology Sydney and University of Newcastle. The research partners are Monash University (lead), University of Technology Sydney and University of Newcastle.

References

[Atherton, D., 1995]. *Remote field eddy current inspection*, IEEE Transactions on Magnetics, pp. 4142-4147.

[Cignoni, P., Corsini, M., et al., 2008]. *MeshLab: an Open-Source 3D Mesh Processing System.* ERCIM News 2008.

[Lorensen, W.E., Cline, H.E., 1987]. *Marching cubes: A high resolution 3D surface construction algorithm.* SIGGRAPH Comput. Graph. 21, 163-169.

[Miro, J.V., Rajalingam, J., et al., 2014]. *A live test-bed for the advancement of condition assessment and failure prediction research on critical pipes.* Water Asset Management International 10, 3-8.

[Oztireli, C., Guennebaud, G., et al., 2009]. *Feature Preserving Point Set Surfaces based on Non-Linear Kernel Regression.* Computer Graphics Forum 28, 493--501.

[Rasmussen, C.E., Williams, C.K.I., 2005]. *Gaussian Processes for Machine Learning*. MIT Press, London.

[Revelles, J., Urena, C., et al., 2000]. *An Efficient Parametric Algorithm for Octree Traversal*, Proceedings of 8th International Conference on Computer Graphics and Visualization, Czech Republic, pp. 212-219.

[Rusu, R.B., Cousins, S., 2011]. *3D is here: Point Cloud Library (PCL)*, IEEE International Conference on Robotics and Automation (ICRA). IEEE, China, pp. 1-4.

[Ulapane, N., Alempijevic, A., et al., 2014]. *Gaussian Process for Interpreting Pulsed Eddy Current Signals for Ferromagnetic Pipe Profiling*, IEEE Conference on Industrial Electronics and Applications (ICIEA), China, pp. 1762-1767.

[Vidal-Calleja, T., Su, D., et al., 2014]. *Learning Spatial Correlations for Bayesian fusion in Pipe Thickness Mapping*, IEEE International Conference on Robotics and Automation. IEEE, Hong Kong.

[Wijerathna, B., Vidal-Calleja, T., et al., 2013]. *Multiple defect interpretation based on Gaussian processes for MFL technology*, in: Yu, T.Y. (Ed.), Proc. SPIE 8694, Nondestructive Characterization for Composite Materials, Aerospace Engineering, Civil Infrastructure, and Homeland Security 2013, San Diego, California, USA, pp. 1-12.