26th International Congress on Modelling and Simulation, Adelaide, SA, Australia, 30 Nov–4 Dec 2025
mssanz.org.au/modsim2025

Using Workspace to Explore Large Point Cloud Datasets

Lachlan Hetherton, Matt Bolger, Paul W. Cleary, David Thomas, Damien Watkins

CSIRO Data61, Clayton, Victoria
Email: Lachlan.hetherton@data61.csiro.au

Abstract: Workspace is a cross-platform scientific application development framework and Scientific Workflow System (SWS) that enables researchers to create impact from their research capabilities by encapsulating them within scientific workflows and domain-specific applications. Simulating and visualising physical processes, such as granular and fluid flows, and working with 3D object / environmental scans, are core focus areas of the Workspace framework and as such, large point cloud datasets feature prominently in its catalogue of visualisation and data processing operations. In this paper, we provide an example of a workflow for visualising large point clouds using Workspace, carefully considering its geometric structure and the visualisation requirements for the end-user application. The objective of this workflow is to enable interactive visualisation of a large, sensor-generated point cloud created through point cloud 3D reconstruction. We seek to demonstrate a method of visualising large point cloud datasets in a highly configurable manner using Workspace to create an executable, interactive scientific workflow, which can be used as the basis for a scientific application involving interactive 3D visualisation of point clouds.

Keywords: *Point Clouds, Workflows, Workspace, Visualisation*

1. INTRODUCTION

Workspace is a cross-platform scientific application development framework and Scientific Workflow System (SWS) that enables researchers to create impact from their research capabilities by encapsulating them within scientific workflows and domain-specific applications. Designed around the core themes of Analyse, Collaborate, Commercialise, Everywhere, Workspace aims to streamline the pathway from research to impact (Watkins et al. 2017). Originally conceived in 2005 (Cleary 2004) to aid researchers in the computational modelling space in the development and maintenance of commercial research applications, several case studies of Workspace applications have been published (Cohen et al. 2017) (Hetherton et al. 2019). Workspace was released for free public usage in 2014 (‘Workspace’ 2014).

Workspace’s origins in computational modelling research have resulted in it accumulating a substantial library of visualisation capabilities that provide insight into simulation outputs. These capabilities range from simple 2D plots and charts to complex interactive 3D visualisations, augmented reality and volume rendering. Each capability can be incorporated into standalone commercial desktop applications through configurable widgets.

2. BACKGROUND

Simulating and visualising physical processes, such as granular and fluid flows (Cleary 2004 and Cleary et al. 2021) and working with 3D object / environmental scans, are core focus areas of the Workspace framework. As such, point cloud processing and visualisation operation feature prominently in its catalogue. Over time, researchers have sought to leverage these capabilities in a variety of projects and applications, exposing Workspace to a wide range of different point cloud types, formats, topologies, and usage scenarios. This presents a design challenge to the Workspace development team, as it is not desirable to develop and maintain multiple independent, specialist point-cloud visualisation solutions. To provide a flavour of the types of datasets that Workspace caters for, we describe several of the most frequently encountered dataset types below:

- Point clouds originating from **sensor data**, such as light detection and ranging (LiDAR) and Time-of-Flight (ToF) scanners, and those generated using stereo photogrammetry techniques such as Structured Light, CSIRO Stereo Depth Fusion (SDF) (‘Stereo-depth Fusion’ 2025), or Structure from Motion (SFM). Points in these datasets represent recorded detections of a scanned real-world surface. Researchers in the computer vision, manufacturing and robotics fields typically encounter these kinds of point clouds. Frequently the range-based sources (such as LiDAR) are highly accurate in the depth (typically Z) dimension, while photogrammetry-based sources are more accurate in the transverse (X and Y) dimensions. Sensor fusion techniques attempt to combine the best of both worlds, producing dense, highly accurate point clouds by combining sensors of different types and different effective ranges. Point clouds originating from sensor data typically manifest as groups of thin bands of points along the surface of the scanned object and so their data can often be quite sparse in some regions. Such datasets can also be quite noisy, depending on the effective range of the source sensors relative to the object / environment being scanned, and the quality of the sensors' calibration.

- Point clouds originating from **geographic sources**, such as topographical / bathymetrical datasets captured using satellite imagery. Researchers in the earth observation, environmental science or natural hazards domains typically encounter these kinds of datasets. Points in such datasets represent specific attributes at a particular location on the earth's surface. Such datasets are typically grid-like in nature in at least 2 out of 3 dimensions, as they closely represent the images from which they originate.
- Point clouds generated from **particle simulations** that model fluid or granular flows. These simulations are typically generated using techniques such as the Discrete Element Method (DEM) or Smoothed Particle Hydrodynamics (SPH) or a combination of multiple such methods. Researchers in the computational modelling or fluid dynamics domains typically encounter these kinds of datasets. In these datasets, points represent particles, each associated with multiple simulation states, e.g. position, size, shape, velocity or pressure. Point clouds such as these have the potential to be the most varied: depending upon the simulation setup and parameters, the point clouds could be tightly clustered in patches; sparse in regions; tremendously varied in size and structure; or semi-homogenous.

Depending on the structure of the point cloud, its physical scale, the number of points, and its intended processing method, the underlying structure of the point cloud in memory (or on disk) may differ. Workspace supports different in-memory storage techniques for point clouds, enabling libraries such as Point Cloud Library (PCL) (Rusu and Cousins 9 May 2011) to be leveraged as needed. The main options to consider when selecting a particular point cloud in-memory storage method are the following:

- **Organised vs Unorganised Point Clouds**
  Workspace supports storing point clouds in either an "unorganised" form, as $N \times S$ arrays of points, where N is the number of points in the dataset and S is the number of states (including spatial dimensions X,Y and Z), or in an "organised" form, as $M \times N \times D$ arrays, where M and N are the X and Y dimensions of the grid on which the points lie. If the point cloud's source is a 2D grid, then using an organised point cloud has numerous advantages, as assumptions can be made about the resolution and density of the dataset when applying image-like filters to the datasets which can dramatically speed up the process. For example, being able to quickly access all neighbours of a point enables spatial filtering / down-sampling to be performed far more quickly, similarly normal and gradients can be computed far more efficiently. Using an unorganised point cloud is advantageous for situations where no fixed grid exists, where the data is very noisy, where flexibility is required in data acquisition, or where memory efficiency is paramount. Workspace supports conversion between the two storage methods if necessary.
- **Patterns of data access**
  Both in-memory and on-disk storage are key considerations in point cloud data representation. It is important with large point clouds to try and index the data so that repeated read operations do not result in frequent cache misses - the sequence in which data blocks are retrieved could be very different from the sequence in which they were written. With very large datasets (e.g. multiple millions to hundreds of millions of points) it is not always possible to achieve this, so the data should be stored in a manner that closely considers the ultimate sequence of the data accesses. For example, in the case where points are likely to be processed at the same time as their approximate neighbours, a spatial "split" operation could be useful to partition points into groups of small, manageable subsets (or "tiles"), prior to processing. Or, if points are likely to be filtered by their size, it may be appropriate to re-index the points in the cloud in order of size prior to processing.

Off-the-shelf solutions exist for many of these problems, such as those provided by ParaView (Ahrens et al. 2005) or MeshLab (Cigioni et al. 2008), however, the Workspace team's requirements for specialist capabilities and the need to embed the capabilities in domain-specific applications require a custom solution. The inherent variability of point cloud datasets and their usage lend themselves to being processed and visualized using flexible, configurable pipelines in tools like Workspace that can be easily altered to meet the needs of any use case.

3. VISUALISING POINT CLOUDS IN WORKSPACE

Here we provide an example of a workflow for visualising large point clouds using Workspace, which carefully considers the point cloud's geometric structure and the visualisation requirements for the end-user application. The objective of this workflow is to enable interactive visualisation of a large, sensor-generated point cloud created through 3D reconstruction in CSIRO's Mixed Reality Laboratory (MRL) (Watkins et al. 9 July 2023).

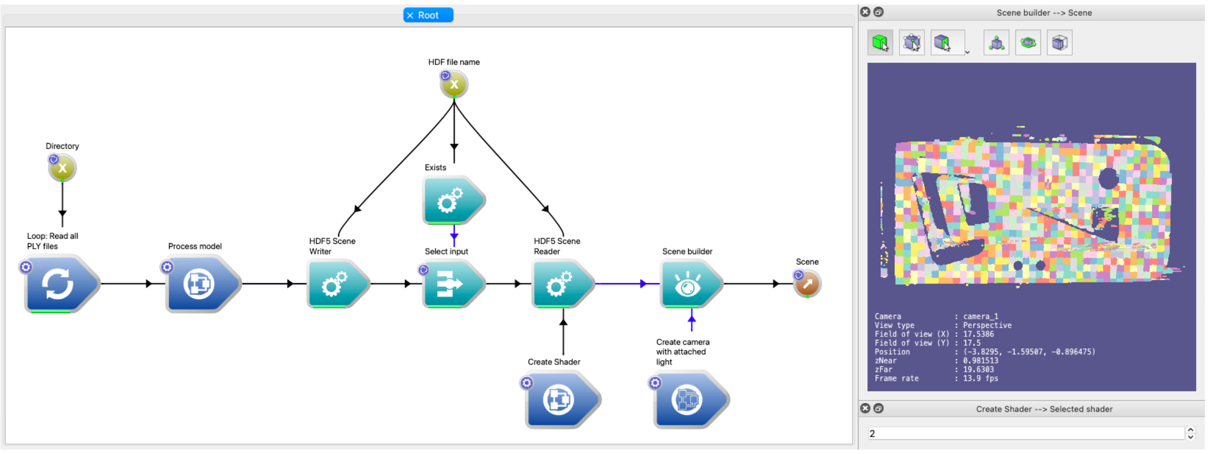

Figure 1 - Workspace-gui showing a workflow for importing, processing a visualising a large point cloud dataset

The point cloud being imported by this workflow represents the results of a data capture pass by sensors in the MRL. This scan is of a large steel panel (approx. 2 m × 1 m), which has several features designed to test surface reconstruction algorithms. Measurement is performed using a both vertically and horizontally aligned stereo camera pairs at a distance of approx. 5 m. The view volumes of the stereo pairs overlap, resulting in point clouds that are dense in some regions and sparse in others. The dataset consists of 4 PLY files, each a 3D stereo reconstruction using a pair of Ximea CB500CG-CM or CB500MG-CM cameras, totalling 131,068,883M unorganised points (~4 GB on disk when saved as PLY files with single-precision floating point values). In the following sections we discuss each step of the processing workflow, including the specifics of the operations used to aid visualisation.

3.1. Importing the point cloud

The first stage of the workflow involves importing point cloud data that has been saved to disk by the sensor data capture workflow in the MRL. The point clouds are stored in 4 separate PLY files on disk, each approximately 1 GB in size, which we use Workspace's FileLoop operation (see Figure 2) is used to iterate over all PLY files in the directory and merge them into a single model comprised of all the points. The FileLoop is a nested workflow, which can be modified to achieve any goal that requires iterating over files in one or more directories (including recursively). As shown in Figure 2, we have modified the workflow to read the PLY file associated with the loop's current file name and merge it with those that have already been read. The "First iteration" input is used to clear the merged model in the case that the workflow is reset.

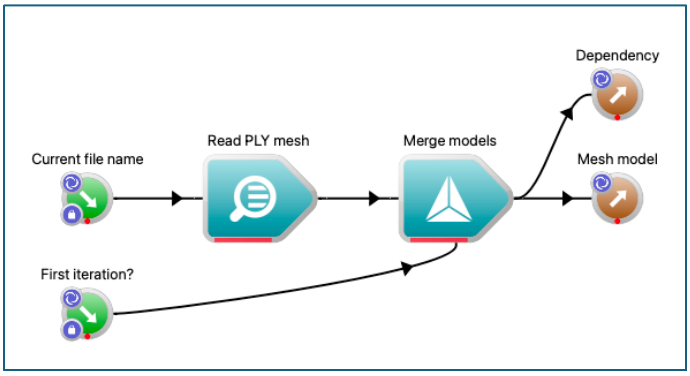

Figure 2 - Loop workflow for importing PLY data. (Left) Nested loop operation on the top-level workflow, (Right) Contents of this nested loop workflow

With smaller models, this step alone would be sufficient to visualise the model, however, in this case we have over 100M points to visualise, which will be difficult without further processing, as the performance of most visualisations for a model of this scale will drop well below interactive framerates (less than 1 frames per second), depending on the available compute / graphics processing potential of the machine.

3.2. Abstracting the point cloud

Once the point cloud has been imported into a single model, it can be processed it so that it is manageable for the visualisation engine. This functionality is encapsulated within a nested workflow, "Process model" (Figure 3), in which two key Workspace operations are used:

- **Create RGBA state**
  This operation adds an RGBA state to the model by combining several source states. In this case, each point already has a state RED, GREEN, BLUE, so we combine them into a single RGBA state, leaving the alpha blank so that each point will have an alpha value of 255 (or fully opaque).
- **Split and LOD Point cloud**
  This operation is the core part of this workflow and where most of the work happens. This operation splits a point cloud into multiple tiles, and for each tile builds a level-of-detail (LOD) pyramid of models, where each level in the pyramid contains progressively more points than the last. The intention is to allow models that are further away from the camera to be rendered with fewer points, while those closer to the camera can be rendered in higher detail. Figure 4 outlines how this operation functions.

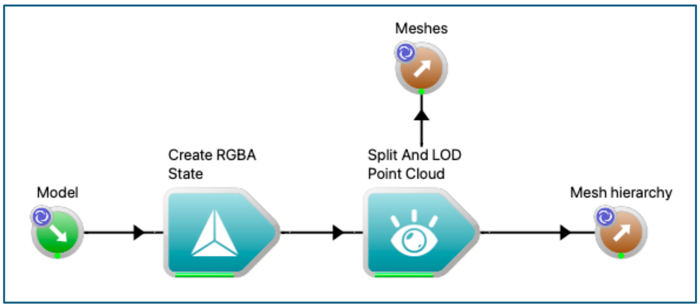

Figure 3 - Nested workflow for processing the merged point cloud. (Left) The nested workflow operation as shown on the top-level workflow, (Right) The contents of this nested workflow.

There are multiple methods that can be used to do this, such as by using discrete levels of detail (where each level of detail is statically generated and stored, retrieved as needed), or using continuous levels of detail (where each level of detail is generated on the fly based on the view distance, (Van Oosterom et al. 2022)); the advantages, disadvantages and performance characteristics of which are beyond the scope of this paper. A brief description follows of this particular algorithm, which uses a similar approach to that outlined by Cura et al (Cura et al. 2016):

1. The model is divided up spatially into a set of individual tiles. Workspace provides options to divide into a specific number of tiles, or it can automatically estimate the appropriate number of tiles based on how many points are desired per patch.
2. Each tile is subdivided further by using an octree (or quadtree for the case where the model is a long horizontal scan). Octrees are stored in memory using a sparse linear octree for processing. For each level in the octree, a LOD is created comprised of the points of the previous LOD, plus, for each cell at that level, the point closest to each cell's centroid.
3. Workspace provides a user-configurable threshold for subdivision, below which no further subdivision occurs. In this example, the threshold is 3000 points. Figure 5 shows the full set of configurable parameters for the SplitAndLodPointCloud operation.
4. Once processing is complete, Workspace stores points using a hierarchical data model, where the points are stored in the parent model, and lower-detail models reference a subset of the points in the parent.
5. The set of tiles, as well as the library of LOD models are output by the operation for display in a Scene. When rendering the scene for visualisation, Workspace uses the camera frustum to determine the appropriate tiles to load, and the distance from the camera (scaled by the field of view) to determine the LOD to load for a given tile. Figure 4 shows 2D example of this.

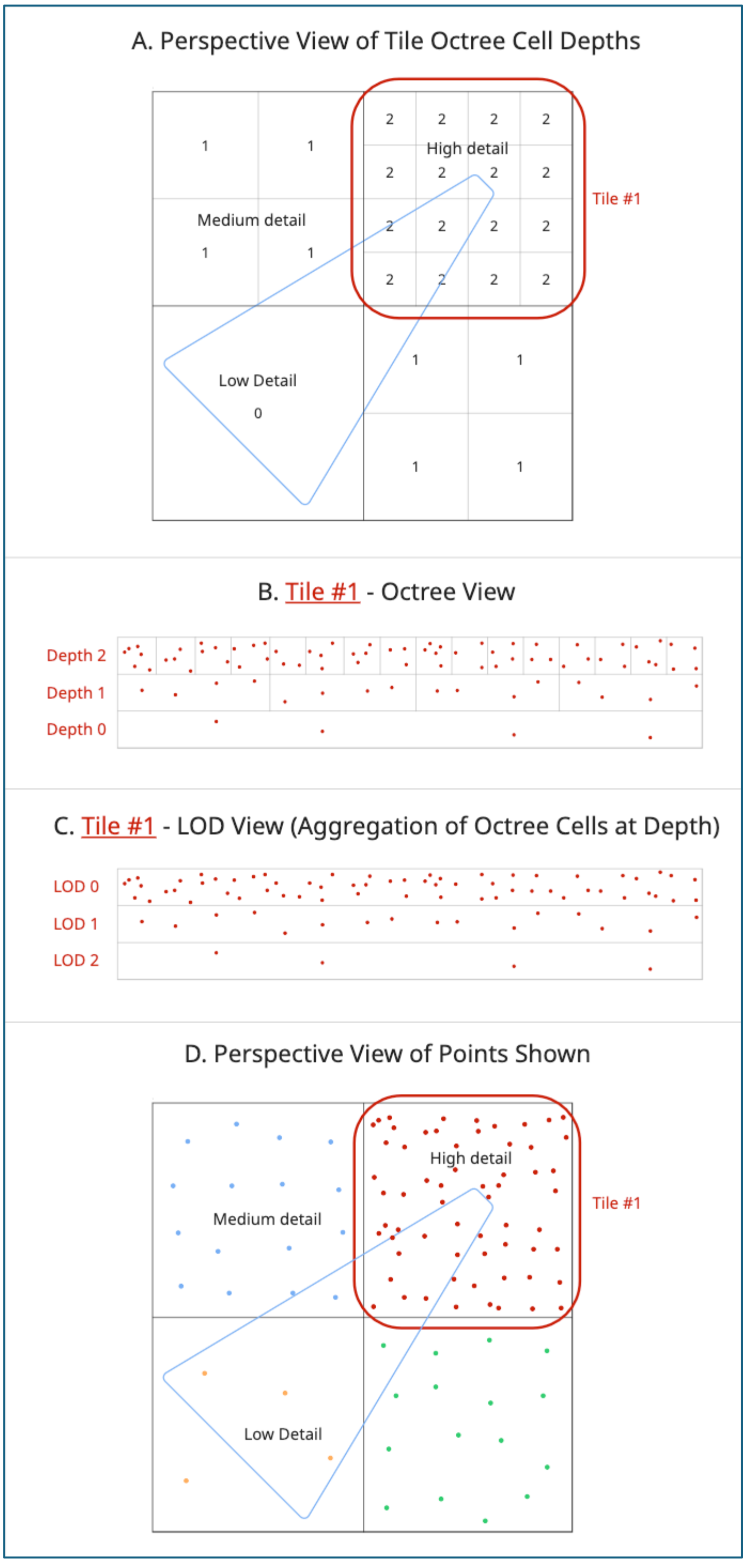

Figure 4 (Above) - Different views of a LOD visualisation in 2D (A) Perspective view, with blue line showing the area visible to the camera, (B) Points separated by octree tile, (C) Points separated by LOD level, (D) Perspective view of points at each LOD level, with the blue line showing the camera view area.

3.3. Exporting / importing processed data

Once the point cloud is appropriately split up and decomposed into smaller, manageable pieces, it is serialized to disk in the workflow through the use of the HDF5SceneWriter, which uses a custom HDF5-based (The HDF Group n.d.) format, where each tile forms a group and its LOD pyramid is stored directly underneath it, along with sufficient metadata to retrieve it when required. When rendering, this is combined with the hierarchical model architecture through use of the HDF5SceneReader, abstracting the data storage from the rendering completely. The model contains “virtual” points in a dataset, which can be retrieved on demand from the underlying HDF5 dataset on an as-needed basis using a query. Presently, only basic queries are possible, which generally involve retrieving the points in the LOD associated with a given draw distance and camera view vector). This structure also includes a built-in least recently used (LRU) cache to avoid loading data that is already available and in use in memory. The HDF5 format was chosen for efficiency and reusability reasons.

This allows the renderer to load each tile’s LOD from disk and into memory as required, rather than having to load the entire dataset into memory prior to rendering. As can be seen in Figure 5, we have some conditional logic built into our workflow to check whether a HDF5 file of the appropriate name already exists, and to use it rather than generating a new HDF5 dataset every time we load and execute the workflow.

3.4. Creating the shader

In this workflow, we have provided three separate shading options which can be selected via the UI through a drop-down-box connected to the index input of the SelectInput operation (see Figure 5). The selected shader is forwarded to the HDF5SceneReader operation and attached to each tile. The different shader types are described briefly below:

- DepthShader (Figure 6, Left): colors points according to how far away from the camera they are in the Z dimension.

- InstanceColorShader (Figure 6, Middle): randomly assigns a color to each tile. This results in a patchwork style visualisation, showing each unique tile in the point cloud.
- ColorStateShader (Figure 6, Right): colors each point using embedded RGBA state information.

3.5. Creating the scene

Finally, the scene is created in which the point cloud will be visible. This is done simply by passing the outputs of the HDF5SceneReader into the SceneBuilder operation, along with a Camera. We add the camera from a separate nested workflow called "Create camera with attached light". This creates a camera which has a light attached to it pointing in the same direction, along with an ambient light source to raise the overall illumination.

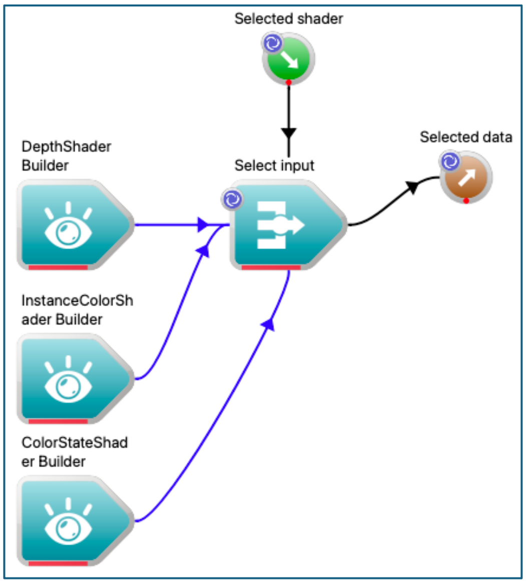

Figure 5 - Contents of the nested workflow for shading.

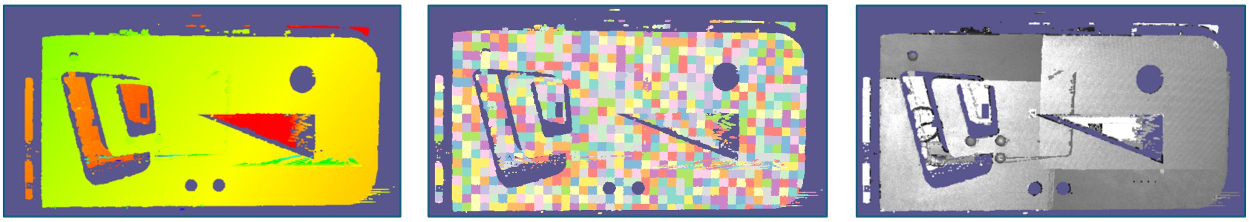

Figure 6 - Visualisations of the point cloud using different shaders. (Left) The DepthShader, (Middle) The InstanceColorShader, (Right) The ColorStateShader

4. DISCUSSION AND CONCLUSION

In this paper, we have demonstrated a way of visualising large point cloud datasets in a highly configurable manner using Workspace. We have demonstrated an executable, interactive scientific workflow, which can be used as the basis for a scientific application involving interactive 3D visualisation of point clouds. The configurability of the approach enables the substitution of different techniques at all levels of the workflow – from techniques to split up the point cloud into more manageable sub-clouds, through to different IO and shading techniques – in order to cater for different kinds of point clouds and application domains.

Table 1 compares interactive performance of the visualisation using different configurations of SplitAndLodPointCloud. Comparisons are made from multiple distances; one where the entire scan is in view (view distance of 25 scene units), one at a close distance (view distance 5), and one fully zoomed in (view distance 0.5). Experiments are performed on an Apple MacBook Pro with 32 GB RAM and an Apple M2 Pro processor (CPU and GPU). Rendering is output at 1080p resolution.

Table 1 - Performance measurements of visualisation in different scenarios

	Frames per second		
Scenario	View distance 25	View distance 5	View distance 0.5
1 – Full model, No LOD	0.1	4	20
2 – Quadtree, 5×5 tiles	11.5	35	65
3 – Quadtree, 10×10 tiles	15	6.5	81
4 – Quadtree, 25×25 tiles	36	15	85
5 – Quadtree, 50×50 tiles	55	25	80

From this table, it can be seen that our workflow has significantly improved the interactivity of the scene, with higher tile counts appearing to offer greater performance improvements. However, there is clearly room for improvement – it is desirable to ensure performance is consistent across view distances and tile counts. The team plans to build on this solution to demonstrate the flexibility of the tools when applied to other types of point clouds, in particular those output by computational simulations.

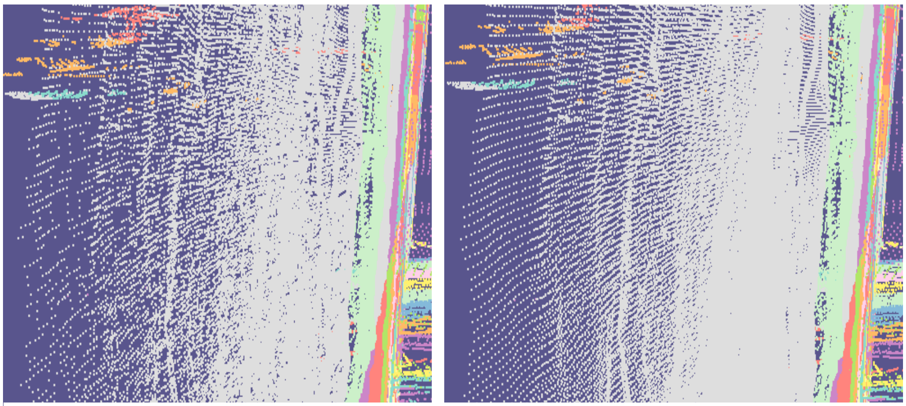

Figure 7 - Surface of the rendered point cloud visualised in the WSGL widget. (Left) LOD level 1 of the nearest tile, (Right), LOD level 0 - highest detail, equivalent to a brute force complete render.

5. REFERENCES

Ahrens J., Geveci B., Law C. (2005) *ParaView: An End-User Tool for Large Data Visualization*, Elsevier.
Cigioni P., Callieri M., Corsini M., Dellepiane M., Ganovelli F. and Ranzuglia G. (2008) 'MeshLab: an Open-Source Mesh Processing Tool', *Sixth Eurographics Italian Chapter Conference*.

Cleary, P. W., Large scale industrial DEM modelling, Engineering Computations, 21,169-204, (2004).

Cleary, P. W., Harrison, S. M., Sinnott, M. D., Pereira, G. G., Prakash, M., Cohen, R. C. Z., Rudman, M., and Stokes, N., Application of SPH to single and multiphase geophysical, biophysical and industrial fluid flows, Int. J. Comp. Fluid Dynamics, 35, 22-78, (2021).

Cleary, P. W., Thomas, D., Hetherton, L., Bolger, M., Hilton, J. E., and Watkins, D., Workspace: a workflow platform for supporting development and deployment of modelling and simulation, Mathematics and Computers in Simulation, 175, 25-61, (2020).

Cohen R.C.Z., Harrison S.M., Cleary P.W., Bolger M. (3 December 2017) 'Dive Mechanic: Bringing 3D virtual experimentation to elite level diving using the Workspace workflow engine', *22nd International Congress on Modelling and Simulation*, Hobart, Tasmania, Australia, https://www.mssanz.org.au/modsim2017/C3/cohen.pdf.

Cura R., Perret J., Paparoditis N. (2016) *Implicit LOD for Processing, Visualisation and Classification in Point Cloud Servers*, doi:10.13140/RG.2.1.1457.6400.

Hetherton L., Bolger M., Zhang Z., Oakes N., Watkins D. and Cleary P.W. (1 December 2019) 'Workspace for Industry 4.0', *23rd International Congress on Modelling and Simulation*, Canberra, ACT, Australia, doi:https://doi.org/10.36334/modsim.2019.D2.hetherton.

Rusu R.B. and Cousins S. (9 May 2011) '3D is here: Point Cloud Library (PCL)', *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China.

CSIRO, 'Stereo-depth Fusion' (2025), https://research.csiro.au/msci/our-research/imaging/sdf/.

The HDF Group (n.d.) 'Hierarchical Data Format, version 5', https://www.hdfgroup.org/HDF5/.

Van Oosterom P., Van Oosterom C., Liu H., Thompson R., Meijers M. and Verbree E. (2022) 'Organizing and visualizing point clouds with continuous levels of detail', *ISPRS Journal of Photogrammetry and Remote Sensing*, 194:119–131, doi:https://doi.org/10.1016/j.isprsjprs.2022.10.004.

Watkins D., Bolger M. and Zhang Z. (9 July 2023) 'Workspace and the Mixed Reality Lab: Where scientific workflows meet Industry 4.0', *25th International Congress on Modelling and Simulation*, Darwin, NT, Australia, doi:https://doi.org/10.36334/modsim.2023.watkins.

Watkins D., Thomas D., Hetherton L., Bolger M. and Clear P.W. (3 December 2017) 'Workspace - a Scientific Workflow System for enabling Research Impact', *22nd International Congress on Modelling and Simulation*, Hobart, Tasmania, Australia, https://mssanz.org.au/modsim2017/C3/watkins.pdf.

'Workspace' (2014), doi:http://dx.doi.org/10.4225/08/54D03170101B7.