3D reconstruction and rendering visualization of tunnel point cloud

Haozheng Zhao[1*], Qingwu Hu[2]

[1*]School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China, vettel_5@163.com
[2]School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China, huqw@whu.edu.cn

Commission II, WG II/8

KEY WORDS: 3D reconstruction; point cloud lite; point cloud data rendering; alpha-shape reconstruction; 3D model optimization; kd-tree; 3D mesh model rendering; over-excavation and under-excavation;

ABSTRACT:

This paper presents a method for rapid and automatic visualisation of over-undercutting based on 3D reconstruction and rendering of tunnel point clouds. The method uses model filtering and our improved voxel filtering for point cloud lite to maximise the preservation of the original tunnel point cloud features while streamlining the point cloud. The 3D reconstruction is then completed according to our proposed alpha-shape algorithm with low reconstruction parameters combined with a fast model optimisation model, which achieves high reconstruction accuracy while significantly improving computational efficiency to meet the requirements of the tunneling project. The rendering method is calculated by selecting the RGB values. The darker the colour, the heavier the degree of over-excavation or under-excavation. According to the calculated over-excavation or under-excavation value, the ratio of the point in the over-excavation or under-excavation interval is obtained in turn, and the gradient is rendered by increasing the base colour of each interval proportionally; the kd-tree is then introduced to calculate and visualise the colour information of the reconstructed model. In addition, this paper uses real-world tunnel point cloud data and demonstrates that the method meets expectations through qualitative and quantitative evaluation. Visualisation of the tunnel point cloud is accomplished while maintaining speed, visualising the over-under-excavation situation and providing an outlook for future work.

1. INTRODUCTION

With the rapid development of information technology and safety considerations, the requirements of tunnel construction are becoming more and more demanding, and tunnel over-excavation monitoring is bound to develop in the direction of systematization, real-time, high accuracy and automation [18,19]. Computer and communication technologies improve the collection, transmission, processing and expression of information saving a lot of time and costs; real-time feedback on tunnel excavation to facilitate real-time adjustment and processing of the construction progress and programme to ensure the safety and smoothness of construction [14]; high-precision monitoring data can accurately detect the subtle excavation conditions, complete expression of the cyclical changes in the tunnel; automation The automated processing can break the current problems of low efficiency and accuracy of the complex and extensive data processing by hand, while the automated processing can be more convenient to produce analysis results and trend prediction [5].

As a narrow structure of underground engineering, the conventional way to monitor the tunnel over-excavation is the traditional mode of operation based on latitude and longitude meter, level, total station, GPS receiver, convergence meter, etc. Currently, the monitoring method of this mode consumes a lot of human and financial resources and is difficult to measure [20].

Therefore, it is particularly important to study a new efficient and practical monitoring technology, which must have high accuracy and at the same time be able to complete the monitoring task quickly in real time, with the integration of data acquisition and processing, as well as the ability of three-dimensional spatial analysis. The application of 3D laser scanning technology has brought a boon to the monitoring of over-excavation in tunnels, and a great deal of exploration and analysis has been done in the application of 3D laser technology in tunnel monitoring, which has verified the feasibility of the technology.

Most of the existing methods are to streamline the huge and redundant point clouds collected, which facilitates the rapid construction of a 3D grid of point clouds. The filtered point cloud is meshed to reconstruct the spatial topology between the point clouds. There are two broad types of reconstruction methods: building a mesh model based on scattered points and reconstructing a point cloud mesh model based on geometry. The former represents the model by constructing a polygonal mesh model, while the latter completes the 3D reconstruction of the object by extracting feature points from the point cloud data [5,14,20,21].

Compared with the above methods, our method takes an improved algorithm in point cloud refinement, which can retain the features of the sampled target well. We combine low-threshold reconstruction and post-optimisation in the reconstruction process, reducing computing time while providing options for users with different 3D reconstruction quality requirements. The point clouds are rendered with colours attached to the calculated over-underexploitation values, normalised and then assigned gradient colours in proportion. Blue to light blue is under-excavation and yellow to red is over-excavation, with darker colours representing a greater degree of over-excavation. The model cannot currently find an existing method, so we have created an original rendering of the model. The rendered point cloud for the nearest neighbour of each vertex of the model is found via the kd-tree and its mean value is used as the colour of that vertex for visualisation.

The rest of the paper is organised as follows: in section 2 our method for 3D reconstruction and rendering visualisation of the

tunnel point cloud is detailed, and in section 3 the results obtained by our method are analysed using real-world collected point cloud data. Section 4 points out the limitations of the study and looks at future research directions.

2. PROPOSED METHOD

2.1 point cloud lite

Introduction: When point clouds are collected in tunnels under construction, they are usually scanned for distractions such as scaffolding, tripods, and construction personnel, resulting in noisy point clouds, which can have an impact on subsequent 3D reconstruction work [23]. Therefore, this paper proposes a point cloud lite method. With the help of the tunnel construction design book we can find the conceptual tunnel design and related parameters, transform them into a ground-removed tunnel point cloud model, and use this model as a filter for the model filter to automatically filter the points that are far away from the tunnel model (including ground, construction tools and some anomalies, etc.). This can be used to reduce the number of points and preserve the shape of the point cloud, which is very useful in improving the speed of algorithms for tunnel point cloud registration, surface reconstruction and shape recognition.

2.1.1 Model filters: An example of the original point cloud tunnel capture is provided in Figure 1. The noise points on the ground in between include construction tools, survey instruments, construction vehicles, etc. These points not only have an impact on the accuracy of the modelling, but also create outliers that can have an impact on the subsequent rendering visualisation.

For this reason, a theoretical point cloud model of the tunnel was constructed by finding the relevant ideal tunnel parameters in the tunnel design book. With the help of pcl::ModelOutlierRemoval [15] in the PCL library, an ideal tunnel model was constructed as input based on the tunnel model in the design book, setting a distance threshold of 1.5 times the allowable error value of the tunnel excavation depth. The entire input tunnel acquisition point cloud is traversed, and those with distances greater than the threshold value from the model are removed as noise points. Figure 1 shows the effect after the model filter, it can be clearly seen that the noise on the ground is well removed, effectively avoiding the impact caused by noise [9,22].

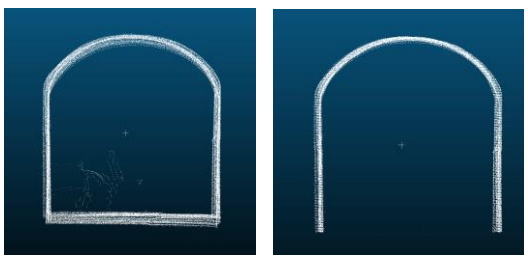


Figure 1. Original tunnel point cloud (left) and model filtered effect (right)

2.1.2 Improved voxel filtration: After the point cloud has been filtered by the model, there is another important step in point cloud refinement - voxel filtering. Due to the specificity of the point cloud geometry. It not only represents the macroscopic geometric shape, but also its microscopic arrangement, e.g. similar dimensions in the horizontal direction and the same distance in the vertical direction. When point clouds are acquired with equipment such as a high-resolution camera like a faro, they are often dense. The excessive number of point clouds can make subsequent rendering and 3D reconstruction difficult, but there is no significant improvement in quality. The voxel grid filter can be used to downsample the point cloud without destroying the geometry of the cloud itself. In contrast to random downsampling, which is more efficient than the voxel filter, the point cloud microstructure is destroyed. The use of the voxelised grid method for downsampling, which reduces the number of points and preserves the shape features of the point cloud, is useful in improving the speed of algorithms for point cloud registration, surface reconstruction, shape recognition, etc [23].

The corresponding algorithm implementation relies on pcl::VoxelGrid, which creates a 3D voxel grid from the input point cloud data. Traditionally, the centre of gravity of all points in a voxel is used to approximate the other points in the voxel, and all points in the voxel are eventually represented by a centre of gravity, but the centre of gravity is not necessarily the point in the original point cloud, and the smallest features of the original point cloud are lost. Therefore, the nearest point to the centre of gravity of the voxel in the original point cloud data is used instead of the centre of gravity of the voxel to improve the accuracy of the representation of the point cloud data. The filtered point cloud is obtained by applying the above processing to all voxels. This improved voxel filtering method is slower than the normal method of approximating voxel centres, but provides a more accurate representation of the surface corresponding to the sampled points.

The calculation process is as follows:

① Find the maximum values X_{max}, Y_{max} and Z_{max} and the minimum values X_{min}, Y_{min} and Z_{min} on the X, Y and Z axes based on the set of coordinates of the point cloud data

② Set the side length r of the voxel grid

③ Find lx, ly and lz from the maximum and minimum values on the x, y and z axes

$$\begin{cases} l_x = x_{max} - x_{min} \\ l_y = y_{max} - y_{min} \\ l_z = z_{max} - z_{min} \end{cases}, \quad (1)$$

where l_x , l_y , l_z= The edge length of the minimum bounding box of the point cloud
X, Y, Z = Coordinates of points

④ Calculate the size of the voxel grid

$$\begin{cases} D_x = \lfloor l_x/r \rfloor \\ D_y = \lfloor l_y/r \rfloor \\ D_z = \lfloor l_z/r \rfloor \end{cases}, \quad (2)$$

where $\lfloor i \rfloor$ = Rounding i down

⑤ Calculate the index of each of the point clouds within the voxel gridlet h

$$\begin{cases} h_x = \lfloor (x - x_{min})/r \rfloor \\ h_y = \lfloor (y - y_{min})/r \rfloor \\ h_x = \lfloor (z - z_{min})/r \rfloor \\ h = h_x + h_y * D_x + h_z * D_x * D_y \end{cases}, \quad (3)$$

⑥ Sort the elements of h in descending order, calculate the centre of gravity of each voxel, and replace all points within the voxel with the centre of gravity

⑦Find the nearest point to the centre of gravity of the voxel and replace it with the centre of the voxel.

Figure 2 shows the results of the improved voxel filtering compared to the uniform sampling filtering. The original point cloud is shown on the left, the improved voxel filtering result is shown in the middle, and the uniform sampling result is shown on the right. It can be clearly seen that the improved voxel filtered point cloud is more evenly distributed and neatly aligned vertically. This shows that our proposed filtering reduces the number of point clouds while retaining the characteristics of the original point cloud in Figure 2, and greatly improves the efficiency of the subsequent 3D reconstruction, without causing the hollowness of the reconstruction due to the varying density of uniform sampling as in Figure 2 (c).

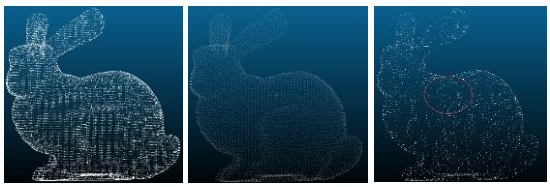


Figure 2. Comparison of the results of different sampling methods

Table 1. Comparison of outcome data for different sampling methods

Sampling method	Original Point Cloud	Improved voxel filtering	Uniform sampling
Number of point cloud	35947	7836	7190

2.2 Tunnel 3D reconstruction

2.2.1 Overview of tunnel 3D model reconstruction: Tunnel 3D model reconstruction techniques are widely used in fields such as reverse engineering, data visualisation, machine vision and virtual reality. Surface reconstruction can be divided into two main categories based on the relationship between the reconstructed surface and the data point cloud: interpolation and approximation. The interpolation method results in a reconstructed surface that is derived entirely from the original data points, while the approximation method uses piecewise linear surfaces or other forms of surfaces to approximate the original data points, resulting in a reconstructed surface that is an approximation of the original data point set [3,7,16]. According to the representation of reconstructed surfaces, they can be classified as: subdivisional surface reconstruction, parametric surface reconstruction, deformation surface reconstruction, piecewise linear surface reconstruction and implicit surface reconstruction. Due to the need for efficiency in reconstructing the model, it is necessary to recreate a realistic tunnel scene in a short period of time to allow for subsequent rendering procedures. Therefore, this paper uses a combination of alpha-shape reconstruction and model optimisation to quickly reconstruct a high-quality 3D model of the tunnel [19].

2.2.2 Principle of alpha-shape reconstruction: On a three-dimensional level, the algorithm can be imagined as a ball rolling through a set of points, where the three points that meet the condition will form a polygon, a "null sphere rule" (similar to the null circle rule), meaning that the ball will contain no points other than the three basic points. The α value is the radius of the sphere, as three points alone do not make a sphere, but need to be combined with a custom α parameter [1,7,12,16].

2.2.3 Steps for alpha-shape reconstruction: The specific steps are shown in Figure 3: firstly, the Delaunay triangulation is performed on the scattered point set, and then the value interval of each simplex (tetrahedron, triangular face, edge, vertex) belonging to the α-form in the result of the dissection is calculated separately, and the simplex is retained if the parameter α value lies within the value interval, and deleted if the α value does not lie within the value interval [2,7,8].

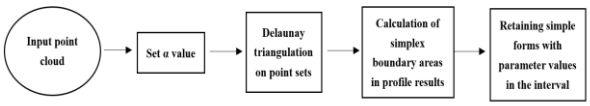


Figure 3. Schematic diagram of the algorithm flow

Suppose there is a point set P in the two-dimensional plane whose α-form is a unique polygon determined by the point set P and the radius parameter α. As shown in Figure 4, it can be envisaged that there exists a circle of radius α that starts rolling from a point outside the point set P. If α is large enough, then the circle will not roll to the interior of the point set P. The trace of the circle rolling is the boundary line of the point set P. If alpha tends to infinity ($\alpha \rightarrow \infty$), then the detected boundary line is the convex envelope of the point set P. Similarly in 3D space, the alpha-shape algorithm determines the boundary point by making a ball of radius α at three points and creates a triangular slice at the resulting boundary point, thus reconstructing the tunnel surface [7].

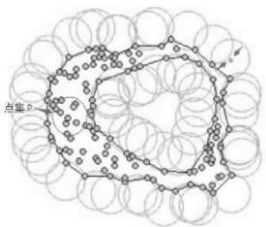


Figure 4. Schematic diagram of the alpha shape algorithm

2.2.4 alpha-shape reconstruction effects: Figure 5 shows the original point cloud of a section of the tunnel and the result of reconstructing the tunnel using the alpha-shape method. In the example below, the mesh parameter alpha is set to 0.05, 0.12, 0.2 and 0.3 from left to right, and the results are shown in Figure 5 for different values of alpha. The smaller the value of alpha, the smoother and more angular the tunnel surface is, the fewer the voids and the fewer the overlapping triangles. However, the time required to construct the mesh increases accordingly, so 0.12 and 0.2 are chosen for further analysis

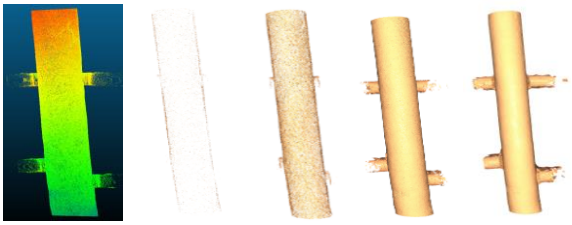


Figure 5. Schematic diagram of the configuration of the network with different parameter values

2.2.5 Principle of 3D model optimisation algorithm: The purpose of the 3D model optimisation is to update the alpha-shape mesh results with a more detailed triangulation of the mesh structure for the generation of high quality triangular and polyhedral decompositions in the general plane, surface and volume domains. It automatically fills in the 'holes' in the tunnel mesh model constructed in the previous step, repairs them, optimises the triangulation mesh to improve the quality of the reconstructed model while maintaining speed, and generates the optimal convex pack mesh for the actual tunnel situation [4]. The optimisation module uses an unstructured mesh generator and the Delaunay Tessellation Library to optimise the reconstructed initial mesh to meet the functional requirements of the software [13,17].

The model optimisation algorithms broadly comprise optimisation-based algorithms for constructing new meshes, existing mesh optimisation drive techniques, and procedures for assembling procedures for Delaunay Tessellation, Voronoi composite and Bauer diagrams, as shown in Figure 6.

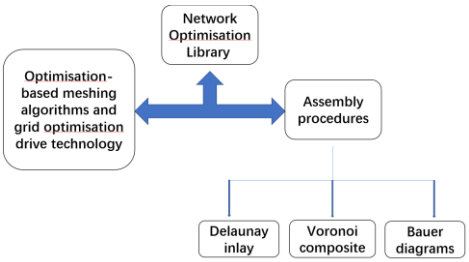


Figure 6. Composition of the optimization library

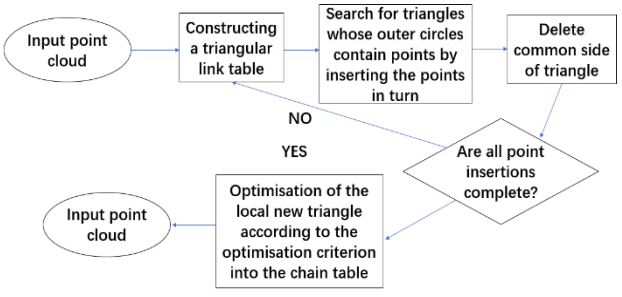
The model optimisation algorithm defines a scalar function that scores the shape quality of each cell based on its geometry. A mesh improvement framework is constructed using various local optimisation predicates, including:①a local vertex smoothing operation based on vertex coordinate perturbations. ② through local updates to the underlying mesh connectivity. ③insertion of new vertices. In this process, additional vertices are introduced into the mesh. It can be shown that locally optimal solutions can be achieved by iterative application of these optimisations, and convergence can be achieved when further improvements in mesh quality are not possible [4,13]. In addition, a new priority-driven optimisation technique is introduced in this paper. In this optimisation technique, the optimisation effort concentrates on triangular meshes that can be improved, effectively filling holes and easily obtaining narrow triangles. In other words, the optimisation algorithm pursues the concept of an 'active set'. In each iteration, optimisation objects are applied to a restricted subset of the mesh entities, which are identified as candidates for further optimisation. The evolution of this active set is explicitly tracked as the optimisation progresses [11]. This approach can significantly improve computational efficiency while maintaining optimisation performance.

The basic principle of Delaunay tessellation is to create a large triangle or polygon, enclose all the data points, insert a point into it which is connected to the three vertices of the triangle containing it to form three new triangles, and then perform null outer circle detection on them one by one, while optimising them using the Rosen-designed local optimisation process, i.e. by swapping diagonals to ensure that the resulting triangle net is a Delaunay triangular mesh [6,11].

The 3D model optimisation algorithm is theoretically rigorous and unique, and the mesh satisfies the null-circle property, which is more desirable. The mesh can be constructed by deleting and adjusting non-Delaunay edges as they are encountered. When new points are added after completion of the construction, it is not necessary to re-network all the points, only the triangular area of influence of the new points, and the local networking method is simple and easy to implement. Similarly, the deletion and movement of points can be done quickly and dynamically [4]. However, in practice, this networking algorithm is also slower when the point set is large and produces triangles that do not satisfy the conditions if the point set range is a non-convex region or if there are inner loops. The aforementioned point cloud refinement is therefore particularly important to ensure a uniform distribution of the point cloud while greatly improving the efficiency of the algorithm operation [17].

The basic steps of the 3D model optimisation algorithm are shown in Figure 7-9:
(1) Construct a mesh improvement framework, smoothing the coordinates of the perturbed vertices and topologically transforming the underlying mesh
(2) Construct a super triangle containing all the scattered points and the newly inserted vertices from the previous step, and place it in the triangle chain table.
(3) Insert the scattered points in the point set in turn, find the triangle in the triangle chain table whose outer circle contains the inserted point (called the influence triangle of the point), delete the common edges of the influence triangle and connect the inserted point with all the vertices of the influence triangle to complete the insertion of a point in the Delaunay triangle chain table.
(4) According to the mesh quality function to determine its convergence, if it does not converge means that the triangle mesh still needs to be optimised, cycle through step 3 above until all scattered points are inserted.

Figure 7. Tessellation optimization process based on



Delaunay's triangle

Figure 8. Discrete point set

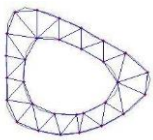


Figure 9. Delaunay triangulation

Triangular mesh improvement is often seen as an optimisation problem and the resulting mesh improvement process can only optimise cell shape quality, further limiting the choice of quality metrics to area and volume length metrics. In essence optimisation is about finding the right geometric and topological configuration for the triangular mesh and therefore a criterion is needed to judge the effectiveness of the optimisation. We have therefore devised a mesh quality function which is designed to score each element configuration based on the shape quality of each element according to a set of user-defined criteria. In contrast to other metrics (such as the Delaunay criterion or dihedral angle-based metrics), this formulation can robustly detect all types of low quality triangular and tetrahedral element types. The use of volumetric length measures typically leads to optimal mesh improvement results for a wide range of tetrahedral tests. By operating on a given vertex a,b, such that each dihedral is shared by a pair of tetrahedra adjacent to vertices a, b. Given such a configuration, the operation attempts to replace the set of tetrahedra adjacent to a subset of sandwich faces by re-triangulating the torus cavity associated with the edge ab such that min(Q(M)) > min(Q(N)). This re-triangulation step is the exact inverse of the edge removal operation described earlier. The polyhedral removal process is shown in Figure 10.

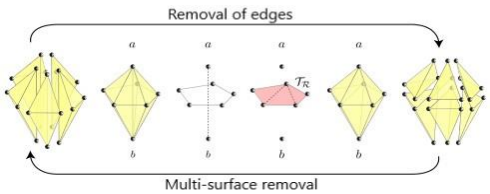


Figure 10. Illustration of edge and polyhedral removal of a tetrahedron

Figure 11 shows an illustration of the polyhedral re-triangulation operation for a tetrahedral complex shape, showing the sequence of inverse operations required to convert from one state to another. As with the edge and polyhedral removal operations, polyhedral re-triangulation is performed on non-planar polygons seeking improved triangulation.

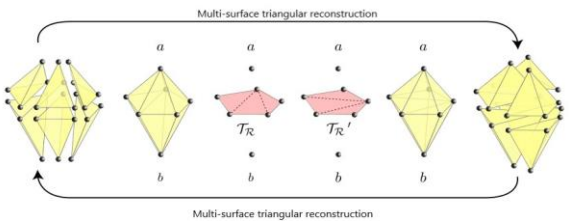


Figure 11. Illustration of a triangular reconstruction of a polyhedron

The polyhedral re-triangulation operation is a topological transformation of tetrahedral meshes in which tetrahedra adjacent to a dihedral ring and sandwiched between a pair of vertices are re-triangulated, preserving the region sandwiched in between, thus optimising more detailed small triangles and making the model smoother [6,11].

2.2.6 Example of 3D model optimisation: As shown in Figure 12 (left), the initial 3D model of the scene was reconstructed using the point cloud generated for a section of the tunnel during the data acquisition process, and the results of the 3D model optimisation algorithm are now invoked as shown in Figure 12 (right). It can be seen that the multifaceted rearrangement operation leads to a moderate improvement in mesh quality compared to the comparative results generated using a combination of edge removal and multifaceted removal operations alone.

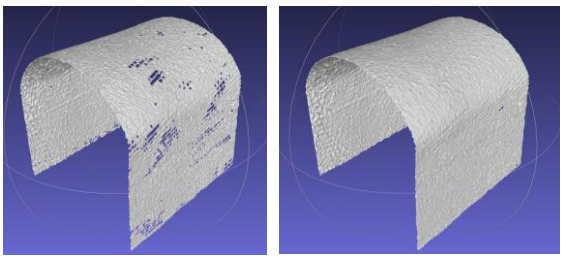


Figure 12. Illustration of the effect of grid optimisation

2.3 Point cloud rendering

Visualisation of over-under-excavation is an important purpose of point cloud rendering and 3D reconstruction of tunnels. Traditional measurement methods have a significant impact on the workload and judgement still relies on manual inspection. This uncertainty of error greatly increases the threat of over-under-excavation to tunnel construction safety and leads to a waste of tunnel excavation resources. In order to accurately reflect the extent of tunnel excavation in reality, we combined the collected point cloud data with the calculated over- and under-excavation values and visualised them using gradient colours.

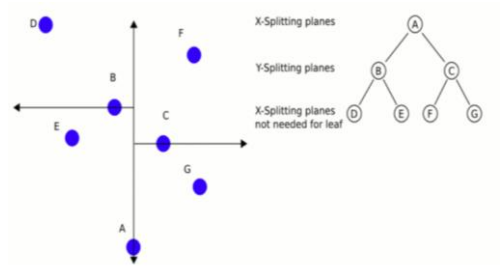
Specifically, we first combine the calculated underdigging values with the original point cloud coordinates to form a new point cloud data format, named PointXYZRGBD, from which we find the extreme values of the underdigging values (including elevation maxima and minima); we then select a rendering colour palette: from the bottom up, the under-excavation is in a cool colour palette, with the minimum values in blue (RGB values 0,0,255) and the maximum values is represented by light blue (RGB values of 0,255,255). The over-excavation values are expressed in warm tones, fading from yellow (RGB values of 255,255,0) to red (RGB values of 255,0,0). It is specified that when the over-excavation value is exactly 0 it is rendered in yellow. The overall gradient is blue → light blue → yellow → red. In order to achieve the gradient, each point cloud RGB value is normalised to the over- and under-dig values, and the ratio of the over- and under-dig values of each laser point in turn to the over- and under-dig intervals where the median and minimum values are located. The under-excavation is done in blue (0,0,255) and the calculated ratio is added to the corresponding RGB G-value for each under-excavation point; similarly, the over-

excavation is done in yellow (0,255,255) and the calculated ratio is reduced to the corresponding RGB G-value for each over-excavation point. Follow this procedure to cycle through each point in the point cloud in turn to complete the rendering of the point cloud.

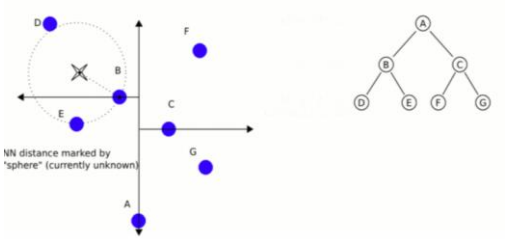
2.4 Rendering of the tunnel 3D model

2.4.1 3D kd-trees for point clouds: A kd-tree (k-dimensional tree) is a data structure used in computer science to establish relationships between points in a k-dimensional space. It is a binary search tree with specific constraints. kd-tree is useful for range search and nearest neighbour search. We usually only deal with point clouds in three dimensions, so all our k-d trees are in three dimensions [10].

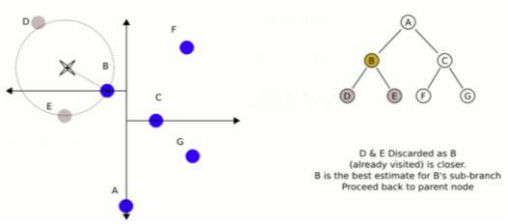
The idea based on the kd-tree uses Euclidean distance to find just the nearest neighbour point between a given kd-tree and a node, which is the nearest neighbour search. Figure 13 below is a demonstration of the steps involved: at each level a split is made in the x-y direction for points other than the leaf nodes. Then a depth-first search is performed on the binary tree, assuming that the point marked as a star is the test point. Starting from A, a circle is drawn with A as the centre and the current nearest distance as the radius; this circle is called the candidate hypersphere. If the circle intersects the axis of the retracing point, all the nodes on the other side of the axis need to be put inside the retracing queue (the nodes on the right in Figure 13 (d) do not intersect and are therefore excluded). Set the best estimate to the distance from test point to A, then check the left child node B, compare the value of the split dimension of the node to be queried and the split node, if it is less than or equal to the left child tree branch, if it is equal to the right child tree branch until the leaf node, follow the "search path" to quickly find the nearest point, that is, the leaf node in the same subspace as the point to be queried, then backtrack search path, and determine whether there may be data points closer to the query point in the space of other sub-nodes of the node on the search path, if possible, you need to jump to the space of other sub-nodes to search (add other sub-nodes to the search path). Repeat this process until the search path is empty.



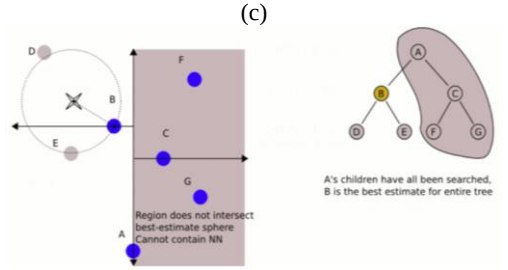
(a)



(b)



(c)



(d)

Figure 13. Nearest Neighbour Search Steps

Figure 14 shows a partial diagram of the zoomed-in point cloud (pink) and the 3D reconstructed mesh vertices (green), which are not overlapping but rather are nearest neighbours. alpha-shape reconstruction of the tunnel model itself has no colour information, so we need to add our own rendering to achieve the result. This paper therefore uses the kd-tree nearest-neighbour search to create a search index for the subsequent colour assignment of the 3D model.

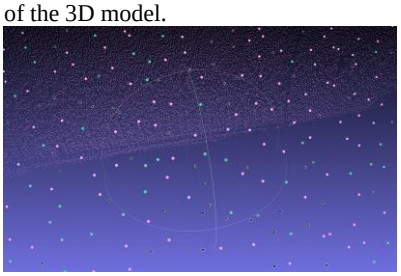


Figure 14. Point cloud with triangular grid vertices

2.4.2 Colour distribution: The first step is to reconstruct the face elements from the alpha-shape 3D reconstruction into a Mesh. The Mesh contains the vertices and triangulated mesh faces of the 3D reconstruction, and in the colour distribution algorithm each vertex in the Mesh is searched to find its nearest neighbour on the rendered point cloud (the nearest neighbour in the algorithm is set to 3), i.e. the search index is the kd-tree established in the previous step for the rendered point cloud. The RGB colour of each point on the mesh is then assigned to the mean of the RGB values of the neighbouring points, and this step is repeated for the entire mesh to render the 3D reconstructed tunnel model.

3. EXPERIMENTAL RESULTS

3.1 Point cloud lite

The data in this paper was collected using a phase-based laser scanner, the FARO Focus 3D, on a section of real tunnel data under construction, which was used to evaluate the method of 3D reconstruction and rendering proposed in this paper. This point cloud data collection contains 585,955 points and is a continuous segment of the tunnel with a length of 53.5m. As shown in Figure 15, there are various stray points on the ground containing construction scaffolding, measuring instruments, associated

personnel, etc., which can cause serious disturbance to the subsequent rendering.

Therefore, a tunnel model was created based on the parameters in the tunnel design plan in the design book as input to the model filter, and then the allowable error in the design book was used as a distance threshold to filter out points that were greater than the threshold from the model. This acquisition process is automatically generated except for the input parameters, and the results are shown in Figure 15 is shown. Compared to the original point cloud, it can be seen that the ground and other noisy point clouds that do not need to be rendered are effectively removed.

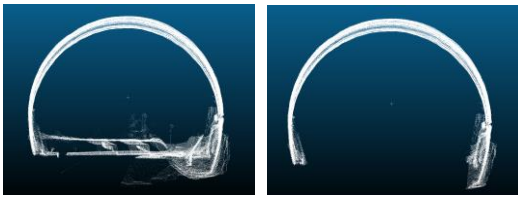
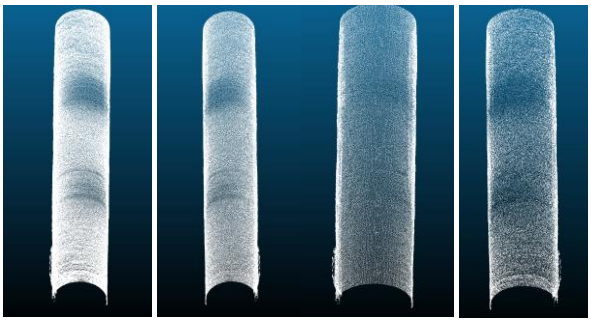


Figure 15. Effect of model filtering

Improved voxel filtering is the unique approach we propose in point cloud refinement. For sampled point clouds, the significant reduction in number can effectively increase the speed of rendering and 3D reconstruction. Implementing downsampling using the voxelised mesh method reduces the number of point clouds while maintaining their shape characteristics; in improving the speed of algorithms for point cloud registration, surface reconstruction, shape recognition, etc. Figure 16 (a)(b)(c)(d) shows the effect of downsampling for different raster voxel sizes, from left to right, (a) original tunnel point cloud; (b) voxel filtering at raster voxel 0.06f; (c) voxel filtering at raster voxel 0.1f; (d) fast uniform sampling. The number of point clouds after thinning is shown in Table 2 is shown. In contrast to the uniform sampling method, although the number of point clouds sampled is similar to that for a raster voxel size of 0.1f*0.1f*0.1f, the voxel sampling preserves the nature of the tunneling point cloud and the point cloud distribution is more uniform and regular. The point cloud from fast uniform sampling is more cluttered, especially in the middle part where the point cloud density is high, which can create voids in the subsequent 3D reconstruction. Although the number of point clouds differs greatly between the two parameter voxels, they both retain the characteristics of the tunnels well and the overall density is uniform, so both can be used as input for 3D reconstruction. Note, however, that the large difference in the number of point clouds can lead to large differences in subsequent rendering and 3D reconstruction times, which has a significant impact on the practical use of the tunnelling project, given that it is a much longer sample than the one used in this paper. We will therefore compare the results after using both samples in the rendering process.

Table 2. Number of post-sampling point clouds

Dilution methods	Number of point clouds
Original tunnel point cloud	467297
Improved voxel filtering (raster voxel 0.06f)	290453
Improved post-voxel filtering (raster voxel 0.1f)	142106
Fast uniform sampling (average of five points to retain one)	155766



a b c d

Figure 16. Effect of different filters

3.2 Tunnel point cloud rendering and 3D reconstruction

The two sampled point clouds from section 3.1 were combined with the calculated values of the super undercut to render and reconstruct them in 3D. The Figure 17 below shows the rendered point cloud, and the advantages of the point cloud refinement and rendering in this paper over other existing methods are evident. It is worth noting that the two point clouds with different levels of point cloud refinement are more than twice as different, but both retain the characteristics of the original tunnel point cloud well in our design, so that both visualise the over-under-excavation of the tunnel. Light blue to blue represents under-excavation and yellow to red represents over-excavation, with darker the colour the greater the degree of over-excavation. Nevertheless, real-world point cloud data can be challenging. For example, the point cloud cavities that appear in this image are escape routes or firefighting equipment storage areas that are being excavated in the tunnel. These voids can cause discontinuities in the rendering and consequently some visual misalignment, and can also create voids in subsequent 3D reconstructions. There are also outliers such as the red areas in the diagram, which are phases of construction that are not reflected in the design document and therefore cannot be filtered out by point cloud refinement. However, it affects the normalisation process of the over-undercut values and therefore makes the rendering inaccurate. This is something we will subsequently improve. However, we still achieved good visualisation renderings, and in this way also proved the feasibility of our approach.

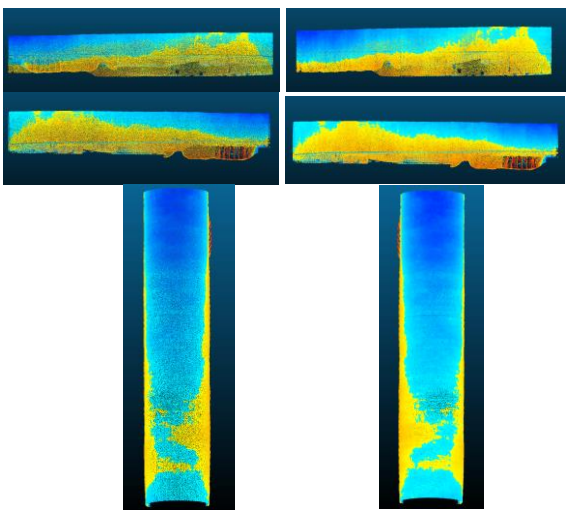


Figure 17. Tunnel point cloud over-undercut visualization (left input is a raster voxel sampled at 0.06f, right is 0.1f)

The next step is the 3D reconstruction of the point cloud. The rendered point cloud is used as input to the surface reconstruction algorithm. The Poisson surface reconstruction algorithm was considered, but it was ultimately rejected as more suitable for closed 3D surfaces with minimal sharpness. In this paper, the ground point cloud was removed for rendering purposes to reduce its interference with the over-undercut visualisation, as this part was not required for rendering. Delaunay retains the angles and is better suited to more planar surfaces, as mentioned earlier in this paper, the complexities of real tunnels under construction are not ideal for smooth, cylindrical-like surfaces. In turn, we chose alpha-shape for the 3D reconstruction, whose parameter α allows us to control the fineness of the triangulated mesh generation, which is in line with our actual needs and can be used in conjunction with the model optimisation algorithm in the next section. There is not much research in the literature on the rendering of triangular meshes for clouds. We therefore explored our own method of rendering 3D meshes. Using a kd-tree, a search index is created on the original RGB point cloud, and each point on the mesh is searched for three proximal points corresponding to the original RGB point cloud. The RGB channel mean of each vertex is then obtained as the RGB mean of the vertex on the mesh. The following Figure 18 shows the reconstructed rendering model after sampling two different raster voxels, with α set to 0.12. Table 3 lists the corresponding efficiencies.

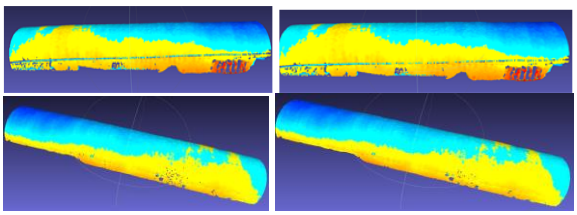


Figure 18. 3D reconstruction effect
(Left is0.1f, Right is 0.06f)

Table 3. Comparison of data from different point cloud reconstructions

Voxel grid (size)	Rendering (time/sec)	Reconstruction (time/sec)	Construction (quality)
0.1f	3.605	38.63.	Fair (needs improvement where high precision is required)
0.06f	7.161	66.589	Better (acceptable)

Overall both reconstructions work well and the over- and under-excavation values can be visualised well in the 3D model. The colours are evenly distributed and give a reasonable representation of the true condition of the over-under-excavation. However, they are still limited by images of real-world data, such as the linear cavities running through the figure, which are pipes laid in the tunnel, and the cavities used to dig escape routes. However, it is undeniable that the quality of the results for a voxel raster size of 0.1f is only average under engineering evaluation, a situation that is clearly unacceptable if higher precision results are required, even though it runs about twice as fast as the latter. So we have an extended idea of whether the 0.1f results can be optimised again to run at a moderate speed to achieve better visualisation. This will be shown and discussed in the next subsection.

3.3 3D model optimisation

The model optimisation workflow is an optimisation operation based on the original mesh and is therefore significantly faster than the last small node cloud reconstruction. Figure 19 shows the detail of the model after optimisation. When the model optimisation finds a cavity, it is good to segment it more finely and generate a denser triangular mesh to fill it. Typically, a denser triangular mesh will result in a smoother surface for the tunnel model, reducing the number of erroneous holes that occur in the 3D reconstruction.

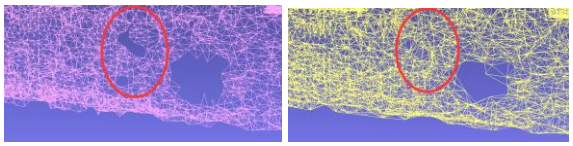


Figure 19. Detail of the model before and after optimization
(left is before optimization, right is after optimization)

In order to quantify this process, we have chosen to compare the effect of an optimised reconfiguration result of 0.1f with a reconfiguration of α = 0.3. Figure 20 shows the results obtained. Both have very good results, fixing the original error holes, tunnel lines and making the tunnel surface look smoother and closer to the real situation. The data in the table shows that the number of mesh faces is around one million, thus achieving a high standard of 3D reconstruction. What is even more striking is that the run times for the selected tunnel segments are optimised to be somewhat less than if high alpha values had been applied directly, so that the application of our method to the whole tunnel would be significantly faster and more efficient.

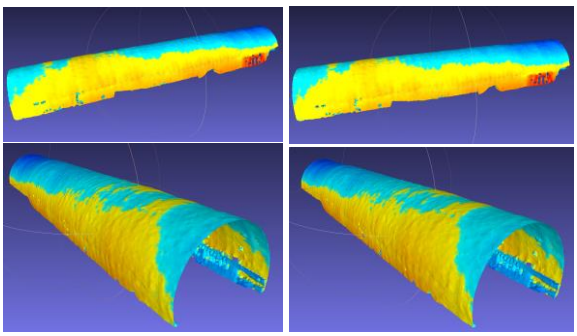


Figure 20. Comparison of optimized results(left is α=0.12 and optimized, right is α=0.3 reconstructed only)

Table 4.Comparison of optimization efficiency

Reconfiguration methods	Operating time/sec	Mesh face/number
Reconfiguration optimization	2.459	989151
Raster size 0.1f (α=0.3) reconfiguration	41.52	740449
Raster size 0.1f (α=0.12) reconstruction	36.707	1081621

However, it is undeniable that the existence of gaps in scenarios with complex real data can introduce ambiguity into the optimisation results. The optimisation has the problem of filling in what would otherwise be gaps in reality, which are still coloured in during the rendering process. It is therefore unreasonable to plan the construction of these areas in accordance with this result, as the filling of seemingly reasonable

voids is not possible in the real world, and we have considered ignoring the realistic voids to avoid such problems.

4. CONCLUSION AND FUTURE WORK

With the advent of high performance laser scanner equipment, there is an interest and demand for SLAM in many industries. But to achieve easy automation and optimised simulation models is still a considerable challenge, as complex environments like tunnels can add many difficult noise points to point clouds, and the need for efficiency requires us to upgrade and improve our method.

Compared to other methods, our method's simplified point cloud better preserves the characteristics of the original point cloud, is more efficient in its simplification, and the uniform density reduces the appearance of voids when reconstructing. The 3D reconstruction takes a combination of low threshold fitting plus optimisation, improving efficiency while providing options for users with different accuracy requirements. The rendering uses our original method of using the RGB mean of the coloured point cloud around a triangular grid vertex as the RGB value for that vertex, allowing for good visualisation of the tunnel over-undercut condition. Compared to traditional tunnel over-under-excavation assessment, our method is much simpler and more automated, with significant labour savings and rendering accuracy that meets engineering requirements.

Future work will focus on the study of noisy point clouds, filtering the scanned construction pipelines, people and equipment through features that differ from the target point cloud to improve quality. Improvements will also be made to the rendering methods of the 3D reconstruction to render the reconstruction results with more detailed colours and more accurate visualisation results.

REFERENCES

[1] Chen Ting-Wang, Wang Qing. A triangular meshing algorithm for SfM-based point cloud reconstruction[J]. Computer Application Research, 2011, 28(2):4.
[2] Chen Jinrui. Research on 3D reconstruction of point cloud data [D]. Wuhan University of Technology, 2011.
[3] Chen D, Yang S-P, Zhuang Y, et al. 3D laser point cloud rendering and depth map construction based on visual information[C]// Proceedings of the 29th Chinese Control Conference. 2010.
[4] Engwirda D . Locally optimal Delaunay-refinement and optimisation-based mesh generation[J]. University of sydneyfaculty of scienceschool of mathematics & statistics, 2014.
[5] Guo XJ, Min FL, Zhong SC, et al. Analysis of the difficulties and key technologies of Nanjing Yangtze River tunnel project[J]. Journal of Rock Mechanics and Engineering, 2012, 31(10):7.
[6] Han J, Rong M, Jiang H, et al. Vectorized indoor surface reconstruction from 3D point cloud with multistep 2D optimization[J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2021, 177: 57-74.
[7] Li Q, Gao XW, Fei XY, et al. Tree canopy 3D model construction using Alpha-shape algorithm[J]. Mapping Bulletin, 2018(12):5.
[8] Lier. 3D point cloud reconstruction and model processing based on feature analysis[J]. 2012.
[9] Lin YD. Research on key technologies of tunnel point cloud data processing and visualization [D]. Donghua University of Technology, 2017.
[10] Moore A . An introductory tutorial on kd-trees[C]// IEEE Colloquium on Quantum Computing: Theory, Applications & Implications. iet, 1991.
[11] Pan R, Skala V. Continuous global optimization in surface reconstruction from an oriented point cloud[J]. Computer-Aided Design, 2011, 43(8): 896-901.
[12] Qiu Qiang. Research on point cloud-based 3D model rendering technology[D]. Harbin Institute of Technology.
[13] Qian Guiping. Scattered point cloud mesh reconstruction and repair[D]. Zhejiang University, 2008.
[14] Song Q, Xu D, Fang S. 3D Tunnel Surface Reconstruction Method in Complex Scenarios[C]//2021 China Automation Congress (CAC). IEEE, 2021: 1369-1374.
[15] Sun Yangxing. Research on point cloud model filtering algorithm for structural feature preservation.
[16] Shi Zuxu, Zeng An, Vincent Ricordel, et al. A point cloud rendering algorithm based on vector quantization of tree-structured meshes[J]. Computer Applications Research, 2019, 36(7):5.
[17] Wang Songbai. Design and implementation of a WebGL-based rendering system for architectural point cloud models [D]. South China University of Technology.
[18] Wei Z, Yao T, Shi C. Research on the Construction of 3D Laser Scanning Tunnel Point Cloud Based on B-spline Interpolation[C]//Civil Infrastructures Confronting Severe Weathers and Climate Changes Conference. Springer, Cham, 2021: 111-118.
[19] Xue Y, Zhang S, Zhou M, et al. Novel SfM-DLT method for metro tunnel 3D reconstruction and Visualization[J]. Underground Space, 2021, 6(2): 134-141.
[20] Yi C, Lu D, Xie Q, et al. Hierarchical tunnel modeling from 3D raw LiDAR point cloud[J]. Computer-Aided Design, 2019, 114: 143-154.
[21] Yi C, Lu D, Xie Q, et al. Tunnel deformation inspection via global spatial axis extraction from 3D raw point cloud[J]. Sensors, 2020, 20(23): 6815.
[22] Zhu N, Jiaa Y, Luo L. Tunnel point cloud filtering method based on elliptic cylindrical model[J]. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 2016, 41: 735.
[23] Zhang B. Research on automatic point cloud alignment based on key point matching[D]. Minnan Normal University, 2020.