Proceedings of the International Association for Shell and Spatial Structures (IASS)
Symposium 2015, Amsterdam
Future Visions
17 - 20 August 2015, Amsterdam, The Netherlands

Rapid Processing of 3D Colour Point Clouds for 3D Printing

Paul SHEPHERD*, John TREDDINICK[a]

*Department of Architecture & Civil Engineering
University of Bath, Bath BA2 7AY, UK
p.shepherd@bath.ac.uk

[a] Centre for Digital Entertainment, Department of Computer Science, University of Bath

Abstract

With recent developments in laser scanning technology, and the parallel increase in computer processing power and storage capacity, 3D scanning to produce large point-cloud datasets is becoming an integral part of the modern building design process. And whilst new and efficient software algorithms are also available to process and display such large datasets quickly enough to facilitate interactive design, similar algorithms to allow the physical rendering of point clouds using 3D printing are not so readily available. Many different methods attempt to create a surface mesh from point-clouds, with varying degrees of success. However these approaches require large processing times and do not scale well to the millions of points present in a typical laser scan datasets.

Here the authors take a radically different approach to producing a 3D printable model from an unstructured point-cloud. Instead of trying to create a single water-tight mesh, each individual point is converted into a separate voxel consisting of six quadrilateral faces. The width, depth and height of each voxel is carefully chosen in line with sub-sampling density and cloud registration information to result in an overlapping set of cuboids, which can be 3D printed using standard sintering techniques. Removing the computationally intensive requirement for extraction of a single manifold mesh, complex geometries can still be generated very quickly. And whilst not necessarily being useful for solid-modelling, are perfectly suited to 3D printing.

This paper outlines in detail the process developed by the authors, and demonstrates its effectiveness on part of a 100GB colour laser scan of a historic building and landscape in the UK, where a stunning colour point-cloud is presented alongside a detailed colour 3D print of the same building. A number of avenues for further refinement of the algorithm are also identified, which are the subject of ongoing research by the authors.

Keywords: 3D Printing, Colour, Laser Scan, Point Cloud, Fast, Efficient, Voxel.

1. Introduction

With recent developments in laser scanning technology, and the parallel increase in computer processing power and storage capacity, 3D scanning to produce large point-cloud datasets is becoming an integral part of the modern building design process. Whether scanning existing building interiors for refurbishment projects and as-built surveys, or capturing outdoor site data and surrounding landscapes to supply context, the ability to capture vast amounts of 3D coordinate data, along with 3D surface normals and RGB colour data, is certainly of great interest to architects, engineers and contractors. However new and efficient software algorithms, such as those found in Recap for Autodesk [1], and Pointools for Microstation [2] and Rhino, are also needed to process and display such large datasets quickly enough to facilitate interactive design. Unfortunately, similar algorithms to allow the physical rendering of point clouds using 3D printing are not so readily available. Many different methods, such as Marching Cubes (Lorensen & Cline [3]), Ball Pivoting (Bernardini et. al. [4]) and those involving Tangent Plane Estimation (Hoppe [5]), attempt to create a surface mesh from point-clouds, with varying degrees of success (Alqudah [6]). However these approaches require large processing times and do not scale well to the many millions of points present in a typical laser scan.

In this paper, the authors take a radically different approach to producing a 3D printable model from an unstructured point-cloud. Instead of trying to create a single water-tight mesh for printing, each individual point is converted into a separate cuboidal voxel consisting of six quadrilateral faces. The three dimensions of each voxel are carefully chosen in tandem with sub-sampling and cloud registration information to result in an overlapping set of (possibly coloured) cuboids which can be 3D printed using standard techniques, where the overlaps ensure the resulting physical model is an accurate, and structurally stable representation of the original point-cloud.

This paper outlines in detail the process developed by the authors, and demonstrates its effectiveness on part of a 100GB colour laser scan of a historic building, where a stunning colour point-cloud is presented alongside a detailed colour 3D print of the same building. A number of avenues for further refinement of the algorithm are also identified, which are the subject of ongoing research by the authors.

2. The Process

2.1. Registration

In order to create a 3D point cloud of any given landscape / building is it general the case that more than one scan is required in order to be able to survey the back-side of objects and to capture any objects shadowed by others. A laser scanner is usually set up at one position, a 360° scan carried out from its view point, and then the scanner is moved elsewhere to capture more points. Depending on the size and complexity of the area to be scanned, a large number of scans will be needed for complete coverage. For example, in the case study below, 74 scan locations were used. A significant overlap between two consecutive scans is necessary, with sufficient points visible in both scans to allow corresponding points in both scans to be identified. This allows the two scans to be *registered* together, applying a rigid body transformation (translation and rotation) to the coordinates of one scan such that they are in the same coordinate system as the other. This means that through a series of sequential registrations, all of the separate scans can be described using a single consistent coordinate system. There are many algorithms available to carry out such registration, the most popular in the context of 3D laser scanning for the build environment is perhaps Iterative Closest Point (ICP) (Besl & McKay [7]). However, scan registration is an integral part of the data capture process pipeline and scans are generally supplied to the client already registered. A detailed review of registration is therefore outside the scope of this paper. However for later refinement, each data point should be attributable to the individual scan it came from (albeit in a single, consistent, registered coordinate system) with the position of the laser identifiable for each scan. One easy way to do this is to keep each scan in a separate data file, with the position of the laser encoded in the file name.

2.1. Subsampling

It is typical for laser scan data to be oversampled for the needs of architectural reconstruction. Returning to a site after a scan has been carried out to collect missing data is expensive, and sometimes impossible if the site has been changed or demolished. So a scan is often performed at maximum resolution, collecting as much data as possible just in case. Whilst allowing the data to be used for many different purposes later on, this often causes a problem since, once a number of scans have been registered, the resulting data becomes so large as to be unmanageable. The 74 laser scans used in this case study, for example, held a total of 1.2 billion points. This not only makes their processing extremely time and computer memory intensive, but results in unnecessarily dense geometry for the eventual goal of 3D printing. Some attempt at sub-sampling is therefore required in order to make the number of data points more manageable.

Although robust and well-developed methods of subsampling are available (Bridson [8]), they are overly complex and unnecessarily slow for the purpose of this research. Two very simple and fast methods of data reduction are therefore suggested, using a sub-set of the available data and introducing a volume of exclusion around each point.

A simple and powerful method of data reduction is to simply only consider one in every ‘n’ points, where ‘n’ is an integer chosen to reduce the number of available points down to a manageable size. A data set with a billion points can be reduced to a more manageable set of 100,000 points simply by extracting one in every 10,000 points from the original set. The most obvious way to achieve this is to simply consider the first data point, the 10,001st data point, the 20,001th data point, etc. This would very quickly reduce the data to a required size and is very simple to implement. However, there are situations where this approach would have undesirable consequences in terms of the sub-set of data it presented. For example, in the much simplified case where each laser performed a rasterized scan of 10 points wide by 10 points high in order (or if the data was at least presented sorted in this order). In this case, a regular sub-sampling increment of n=10 would return points along a straight line, as opposed to a more desirable even sampling across the 100 points covered by the laser, as shown in the left of Figure 1 where each data point is shown as a circle with its position in the data set shown numerically inside. Any features in the right hand side of the area scanned, for example near points with indices 8, 18, 28, 38, etc. shown in blue on Figure 1, will not be captured. Even when the value of ‘n’ is not exactly the same as the width of the sample grid, stripes of subsampled data still result, with potential to miss particular features, nor does choosing prime numbers of ‘n’ alleviate the problem, as shown on the right of Figure 1.

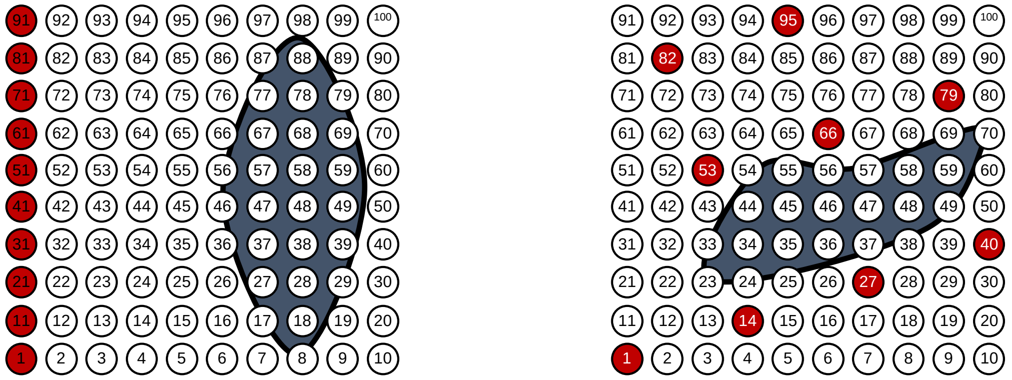

Figure 1: Sequential subsampling of every 10th point (left) and every 13th point (right) shown in red, with potentially missed artefacts shown in blue.

Probabilistic methods of sub-sampling, where each point is considered for the sub-set based on a statistical probability (usually randomly sampled from a uniform distribution), are good ways of removing such anomalies. However, since every single point has the potential to be included in the sub-set, this would involve a random variable being generated for every point in the original set, slowing down the process considerably.

The authors therefore propose a hybrid solution, using the extremely fast "one-in-every-n" method as a first pass data reduction, and combining it with a "radius-of-exclusion". The radius-of-exclusion only adds a candidate point to the sub-sampled set if there are no others already chosen that are within a given distance. Since the sub-sampled vertices are stored using a kd-trie (Orenstein [9]) such a test is efficient and results in an evenly distributed subsampled point cloud.

2.2. Voxel Creation

Once the data has been reduced to a manageable size, each point can then be converted into a voxel by replacing it with a cuboid formed from eight vertices and six quadrilateral faces. The cuboid is located with the scanned point at its centre, and in the simplest case is orientated with its edges aligned with the global coordinate system (see later discussion). The size of the voxel is an important consideration, since large voxels might lead to a poor geometrical representation of an underlying object. However for 3D printing it is crucial that adjacent voxels overlap sufficiently to maintain structural continuity, as shown in the 2D sketch on the left of Figure 2. If the length of each side of a voxel, 'L' say, was smaller than the radius-of-exclusion 'R', say, then two adjacent voxels would not overlap (see Figure 2 right) and the resulting 3D print may fall apart. The situation in 3D is slightly more complex, where a voxel out of plane may connect two other voxels, but certainly a voxel size of at least $R/\sqrt[2]{3}$ is needed.

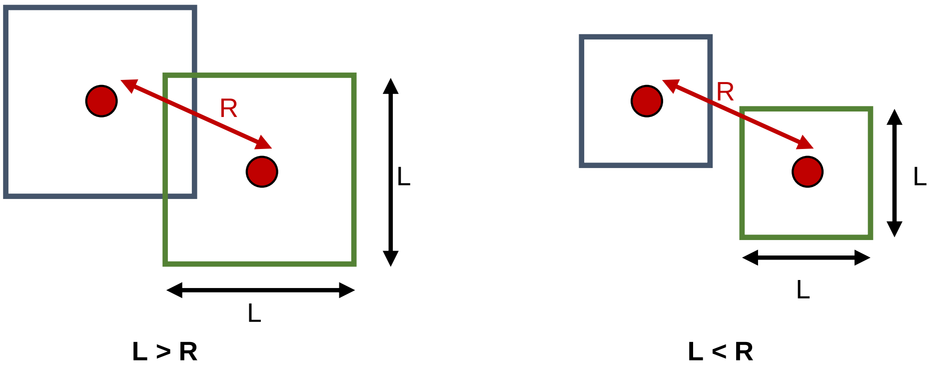

Figure 2: Plan view of two adjacent voxels, with edge 'L' separated by a radius-of-exclusion 'R'.

2.3. Colour

Laser scans often come with RGB colour data associated with each scan point. The method described here is capable of preserving the colour data for use with colour 3D printers. The colour information is simply retained during subsampling, and is applied to all eight vertices of the voxel on creation. Any colour inside overlapping voxels is unseen, but any voxel with an exterior face is visible and exposes its colour. The two voxels on the left of Figure 2 have different colours (blue and green), and when viewed from the very left of the image for example, would expose the relevant blue and green colours to the 3D model.

3. Case Study

In order to demonstrate the effectiveness of the method proposed, a laser scan of Chedworth Roman Villa, UK, was kindly donated by the property's owners, as shown in Figure 3. It contained over 1.2 billion colour data points contained in 74 separate, but consistently registered, text-based data files. In order to result in a meaningful print within the printer's 203x254x203mm build volume at an appropriate scale, a subset of the scan was chosen (highlighted with a red box in Figure 3), cropped by global position into a 15x20x10m region of the site containing a mock-Tudor lodge.

The data was then subsampled as described above, taking one-in-every-100 points for consideration and then applying a radius-of-exclusion filter of 6cm before finally accepting them into the sub-set. This resulted in 141547 points to be used for the model, evenly distributed throughout the area of interest.

Each point was then replaced by a cubic voxel of side-length 5cm, which despite being less than the radius-of-exclusion, was found to give a good overlap between adjacent voxels in 3D (it is certainly greater than the $R/\sqrt[2]{3} = 3.46$cm absolute limit) without having a detrimental effect on the resulting geometry, as can be seen from Figure 4 left. The RGB colour data from the laser scan data was also applied to each of the 8 vertices of the corresponding voxel, leading to the final colour model shown on the right of Figure 4.

Figure 3: Plan view of the entire scan, with the area of the lodge highlighted in red

Figure 4: Colour voxels rendered at varying zoom

The resulting model was exported as a VRML file (.wrl) and printed in colour on a ZCorp Z450 3D printer. A photograph of the real building can be seen in Figure 5 and a photograph of the 3D printed model is shown in Figure 6. Not only is the model extremely detailed, but it is also surprisingly robust, the overlap between adjacent voxels having been sufficiently large and strong enough to allow the leaves and branches of the tree to be printed without physical collapse (see Figure 7).

Figure 5: Photo of the mock-Tudor lodge

Figure 6: Photo of the final 3D colour print

Figure 7: Photo of the final 3D colour print showing the tree (left)

4. Future Development

Whilst the case study above has demonstrated the described technique is remarkably successful at quickly converting large laserscan datasets into physical 3D printed models, there are a number of areas where improvements could be made.

Whilst the 3D printed models were a good geometrical representation of the exterior of the building in question, the inside of the building had not been scanned. This led to a 3D printed model of the exterior shell without internal supporting walls. The model was therefore not very strong, and after handling began to crack in places (see the crack down the corner of the building in Figure 7). A scan of the internal structure of the building would help, but is not common in architectural scans and can also be difficult to register accurately. To strengthen the print without compromising on geometrical accuracy, it is proposed that the size and orientation of the voxels might be more carefully chosen. If the voxels making up the walls were, for example, longer in the direction of the wall thickness, then the exposed quadrilateral face of the voxel would not change, but the resulting structure would be stronger. This could easily be achieved by orientating each voxel along the direction of the laser beam, rather than orientating it along the global axes. Since it is certain that the position of the laser is "outside" of the building, the voxel could be given a cuboidal shape (as opposed to cubic), with longer edges in the direction of the laser, thereby thickening the wall (see Figure 8). Theoretically, extruding the voxels more in the direction of the laser would have a slight warping effect on the exterior surface, where the exterior faces of adjacent voxels on a planar face would no longer remain in plane, as can be seen from the blue voxel outlines in Figure 8. But in practice this is unlikely to be noticed due to the distributed sampling of the voxels and the very large relative distance to the laser scanner resulting in very small differences in angles of incidence between adjacent voxels.

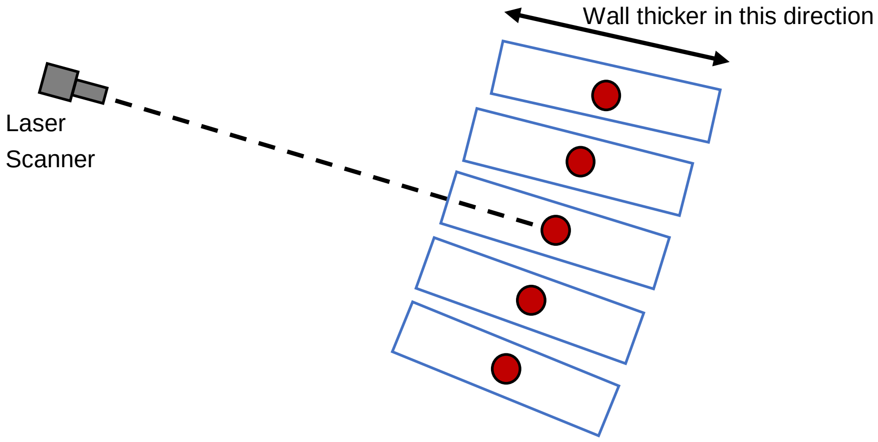

Figure 8: Non-cubic voxels (blue) extruded based on the direction of the laser

This unequal extrusion required the position of the laser scanner to be known for each data point. As mentioned above, encoding the position of each scan into the file name, and keeping data points from each scan in separate files, would allow this to be automated with hardly any overhead in terms of processing time. Determining the position of the laser during each scan may be non-trivial and if it is not documented as part of the scanning process, it could be determined computationally form the scan data itself. The position of the laser during each individual scan can be clearly seen in Figure 3, identified by a circle (nearly) void of data points. This position could therefore be identified automatically using any number of graphical feature recognition techniques (such as iterative moving-window), details of which are beyond the scope of this paper. A detailed investigation into the optimal values of voxel overlap, alignment and wall-thickening is in progress by the authors.

Another area of improvement is the treatment of areas which have not been included in any of the scans. A small hole in the model can be seen in Figure 6, where none of the laser scans picked up parts of the walls around the back (particularly high up) due to shadowing and non-exhaustive coverage of the building. This leads to holes in the 3D printed model, some of which can be critical to its structural stability. Standard hole-detection and surface reconstruction algorithms might need to be used to ensure the print is physically stable. And since the sub-sampling processes outlined would result in a much smaller model, their complexity would not be a hindrance as part of a post-processing workflow prior to printing.

However, despite these shortcomings, the authors believe that by prioritizing simplicity and speed over a desire to generate a single-surface representation of a point cloud, the method outlined above gives an efficient way to produce 3D printed models from the types of dense laser scan point clouds routinely generated in practice.

Acknowledgements

The point cloud used for the case study above was created by the VISTA research group of the University of Birmingham, UK in collaboration with The National Trust, UK.

References

[1] Autodesk, *ReCap*, available online at http://recap.autodesk.com, accessed 26/02/15.

[2] Bentley, *Pointools*, available online at http://pointools.com/, accessed 26/02/15.

[3] Lorensen, W. E. & Cline, H. E., *Marching Cubes: A high resolution 3D surface construction algorithm*, in SIGGRAPH '87 Proceedings of the 14th annual conference on Computer graphics and interactive techniques, pp163-169, 1987. doi:10.1145/37401.37422

[4] Bernardini, F., Mittleman, J., Rushmeir, H., Silva, C. & Taubin, G., The ball pivoting algorithm for surface reconstruction, IEEE Transaction on Visualization and Computer Graphics, 5(4), pp349-359, 1999.

[5] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. & Stuetzle, W., *Surface Reconstruction from Unorganized Points*, in SIGGRAPH'92 Proceedings of the 19th annual conference on computer graphics and interactive techniques, pp71-78, 1992. doi:10.1145/133994.134011

[6] Alqudah, A., *Survey of Surface Reconstruction Algorithms*, Journal of Signal and Information Processing, 5, pp63-79, 2004. doi:10.4236/jsip.2014.53009

[7] Besl, P. J. & McKay, N., D., *A method for registration of 3-D shapes*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(2), 1992. doi:10.1109/34.121791

[8] Bridson, R., Fast Poisson disk sampling in arbitrary dimensions, in ACM SIGGRAPH '07 Sketches, Article 22, 2007.

[9] Orenstein, J.A., "Multidimensional tries used for associative searching", Information Processing Letters, 14(4), pp150-157, 1982.