1 Point clouds

Point clouds are data structures that represent multidimensional space as an array of coordinates. 3-dimensional (3D) point clouds represent 3D space as an array of x, y, and z coordinates. More coordinates can add to point clouds to represent other dimensions of data such as time, color, or value. 3D point clouds can be generated algorithmically or captured by sensing technologies like lidar, structured light, time-of-flight, or structure from motion. Point clouds captured by these sensing technologies represent sampling points on the sensed surface. Point clouds are used in diverse applications including surveying, spatial modeling, digital fabrication, robotics, and human-computer interaction. Point clouds are used often in tangible interfaces, a type of human-computer interaction that aims to physically manifest digital data – to digitally model physical objects, coupling them with their digital counterparts.

Tangible Landscape – a tangible interface for geographic information systems (GIS) – uses a 3D sensor to capture a physical model of a landscape as a point cloud and import it into GIS. Tangible Landscape's predecessor, the Tangible Geospatial Modeling System, used a terrestrial lidar scanner (Tateosian et al., 2010), the first generation of Tangible Landscape used the first generation Kinect with structured light sensing (Petrasova et al., 2014), and the second generation of Tangible Landscape used the second generation Kinect with time-of-flight sensing (Petrasova et al., 2015). Tangible Landscape couples the physical model with a digital model through a near real-time cycle of 3D scanning, point cloud processing, geospatial modeling and simulation, and projection. This enables users to tangibly interact with digital models and simulations either by shaping topography with their hands or by placing markers that are identified through object detection. As the digital models and simulations update the results are projected back onto the model for the user to see. Tangible Landscape also uses 3D data derived from airborne lidar point clouds as reference data. In either case the point clouds are converted into raster data for geospatial modeling, analysis, and simulation.

1.1 Acquisition

Lidar In lidar sensing the time a pulse of light takes to travel from the sensor and reflect back is used to measure 3D space. A laser, a pulse of light is emitted from the sensor, travels until it hits an object and is reflected back to the sensor. The distance travelled by the pulse is computed from its travel time. The position of the scanned object is computed from the distance travelled by the pulse and the position and orientation of the sensor. The position of the scanned object is recorded as a 3D point. A lidar survey generates a set of 3D points – a point cloud.

Terrestrial lidar can be used to scan small objects or survey landscapes. Turntables are often used to precisely 3D scan all around small objects and capture all of the geometry. Terrestrial lidar surveys can precisely capture features in a landscape at centimeter to millimeter resolution – Starek et al., for example, surveyed gullies at 1 cm resolution from 12 m range (2013). Terrestrial surveys of entire landscapes, however, tend to have highly spatially variable densities with sparse and dense regions due to obstacles and the dispersion of points throughout the field of view.

Entire landscapes can be precisely 3D scanned as point clouds with airborne lidar at meter resolution. As pulses of light are emitted from the aircraft they hit and reflect back from vegetation, structures, and topography. When a pulse of light hits a tree part of the light is reflected and recorded as the first return, while the rest penetrates the outer canopy. The residual pulse, recorded as a series of returns, reflects off of leaves, branches, shrubs, and the ground. The waveform of the pulse can also be recorded to model the structure of the canopy.

Structured light Structured light – the deformation of a known pattern of light over a surface – can be used to reconstructed 3D depth. Structured light depth sensors like the first generation Kinect project a known pattern of infrared (IR) light onto a scene or object while scanning the scene with an IR sensor to capture the deformation of the pattern of IR light. The depth of the scene can be interpolated from the deformation of the pattern. Structured light depth sensing suffers from data loss – holes in the reconstruction – at the boundary of objects and with transparent or specular materials (Hansard et al., 2013). The first generation Kinect combines an IR projector, an IR sensor, and a color sensor to generate IR amplitude, depth, and color images (Smisek et al., 2011; Hansard et al., 2013). The maximum resolution of the depth image is 640×480 pixels. The first generation Kinect can capture depth images at 0.8 - 3.5 m range with 3 mm xy resolution and 10 mm z resolution at a 2 m distance.

Time-of-flight In time-of-flight depth sensing the phase delay between projected and reflected IR light is used compute depth (Hansard et al., 2013). The second generation Kinect uses a time-of-flight depth sensor with a range of 0.8 - 4.2 m, a maximum depth resolution of 512 × 424 pixels, and a frame rate of 30 frames per second (Bamji et al., 2015). Time-of-flight depth sensing suffers from data drift – offsets in depth values – at the boundary of objects and with transparent materials and from data loss with specular materials. The geometry, color, and material of the scene or object cause variations in the phase and amplitude of reflected IR light giving rise to errors. Systematic errors include noise caused by short integration times and ambiguity caused by variations in amplitude due to color. Non-systematic errors include motion blur and scattering caused by IR saturation, ambiguous borders, and overlapping reflected IR signals (Hansard et al., 2013).

Structure from motion In structure from motion 3D surfaces are algorithmically reconstructed by matching features captured from multiple viewpoints in overlapping, offset images. Image matching algorithms match features in random, unstructured sets of photographs with enough overlapping images. Bundle adjustment is then used to optimize the camera parameters and the geometry of the surface (Westoby et al., 2012; Fonstad et al., 2013; Snavely et al., 2008).

Structure from motion can be used to reconstruct DEMs from aerial photography captured by manned or unmanned aerial vehicles (UAVs). Structure from motion for aerial photography generates a 3D point cloud from matched features that can be georeferenced using ground control points or the recorded position of the camera (Fonstad et al., 2013).

The resolution of the point cloud depends upon the height of the aircraft, the resolution of the camera, and the percent overlap of the photographs. Imagery from low altitude aircraft like UAVs can be used to generate high resolution point clouds – for example imagery captured by a UAV flying at an altitude of 120 m with a 2456 x 1632 pixel resolution camera can generate a 10 cm resolution point cloud.

1.2 Modeling

Once a point cloud has been captured it can be used to generate other 3D representations such as meshes, parametric surfaces and volumes, 2.5D rasters, and 3D rasters. 2.5D rasters are xy arrays of cells with a z value representing elevation, whereas 3D rasters are xyz arrays of volumetric cells that can have a value representing a 4th dimension of data.

Raster representations of landscapes such as digital elevation models (DEMs) – rasters representing bare earth topography – and digital surface models (DSMs) – rasters representing topography, vegetation, and structures – are commonly used in GIS to model and analyze terrain because they can easily be mathematically transformed. The partial derivatives of a function approximating a DEM or DSM, for example, can be used to compute topographic parameters such as slope, aspect, and curvature.

Point clouds representing landscapes captured by 3D sensing technologies like structure from motion and time-of-flight require processing before they can be modeled as DEMs or DSMs. Outlying points should be filtered and the data should be georeferenced. Sometimes edge detection can be used to identify boundaries and extract a region.

Lidar point clouds stored in the las format can be filtered by returns and, if classified, by classes. They can be filtered by the first return, middle return, or last return. They can also be filtered by the following classes: 0 for created,1 for unclassified, 2 for ground, 3 for low vegetation 4 for medium vegetation, 5 for high vegetation, 6 for building, 7 for low point, 8 for model key-point, 9 for water, 10 for reserved, 11 for reserved. 12 for overlap, and 13-31 for reserved (ASPRS, 2013).

Unclassified point clouds – whether from lidar or other 3D sensing technologies like structure from motion and time-of-flight – can be automatically classified. Edge detection can be used to identify and classify features like buildings. Ground points can be automatically classified using the multi-scale curvature classification algorithm, ‘an iterative multiscale algorithm for classifying LiDAR returns that exceed positive surface curvature thresholds’ (Evans and Hudak, 2007).

Point clouds should be filtered before being used to generate DEMs or DSMs. Only points classified as ground should be used to model bare earth topography as a DEM. Only the first return and classes 1,2,3,4,5,6,9 should be used to model topography, vegetation, and buildings as a DSM.

Raster surfaces like DEMs and DSMs can be computed from point clouds using either binning or interpolation. In binning values are assigned to cells based on a univariate statistical analysis of the z coordinates of the set of points contained in the cell. Statistical methods for computing raster values from a set of points include sample size (n), minimum, maximum, range, sum, mean, standard deviation, variance, the coefficient of variance, median, and percentile. Binning is a fast method, but can result in rough surfaces with holes in regions without enough points.

In interpolation the values between points are estimated mathematically or statistically. Interpolation can be processing intensive, but can fill holes in data and – with the right choice of method and parameters – can smoothly, precisely model complex surfaces. Poorly parameterized methods can cause distortions like overfitting (with overshooting and pitting) or over smoothing. Methods include kriging, radial basis function, natural neighbor, inverse weighted distance, bicubic spline, and regularized spline with tension interpolation.

References

ASPRS. ASPRS LAS 1.4 Format Specification - R13, July 15, 2013. Technical report, 2013. URL `http://www.asprs.org/a/society/committees/standards/LAS{_}1{_}4{_}r13.pdf`.

Cyrus S. Bamji, Patrick O'Connor, Tamer Elkhatib, Swati Mehta, Barry Thompson, Lawrence A. Prather, Dane Snow, Onur Can Akkaya, Andy Daniel, Andrew D. Payne, Travis Perry, Mike Fenton, and Vei Han Chan. A 0.13 μm CMOS System-on-Chip for a 512 x 424 Time-of-Flight Image Sensor with Multi-Frequency Photo-Demodulation up to 130 MHz and 2 GS/s ADC. *IEEE Journal of Solid-State Circuits*, 50(1):303–319, 2015. ISSN 00189200. doi: 10.1109/JSSC.2014.2364270.

Jeffrey S. Evans and Andrew T. Hudak. A multiscale curvature algorithm for classifying discrete return LiDAR in forested environments. *IEEE Transactions on Geoscience and Remote Sensing*, 45(4):1029–1038, 2007. ISSN 01962892. doi: 10.1109/TGRS.2006.890412.

Mark A. Fonstad, James T. Dietrich, Brittany C. Courville, Jennifer L. Jensen, and Patrice E. Carbonneau. Topographic structure from motion: A new development in photogrammetric measurement. *Earth Surface Processes and Landforms*, 38(4):421–430, 2013. ISSN 01979337. doi: 10.1002/esp.3366.

Miles Hansard, Seungkyu Lee, Ouk Choi, and Radu Horaud. *Time-of-Flight Cameras: Principles, Methods and Applications*. Springer London, London, 2013. ISBN 978-1-4471-4658-2. doi: 10.1007/978-1-4471-4658-2.

Anna Petrasova, Brendan A Harmon, Vaclav Petras, and Helena Mitasova. GIS-based environmental modeling with tangible interaction and dynamic visualization. In D.P. Ames and N. Quinn, editors, *Proceedings of the 7th International Congress on Environmental Modelling and Software*, San Diego, California, USA, 2014. International Environmental Modelling and Software Society. URL `http://www.iemss.org/society/index.php/iemss-2014-proceedings`.

Anna Petrasova, Brendan Harmon, Vaclav Petras, and Helena Mitasova. *Tangible Modeling with Open Source GIS*. Springer, 2015. ISBN 978-3-319-25775-4.

Jan Smisek, Michal Jancosek, and Tomas Pajdla. 3D with Kinect. *Proceedings of the IEEE International Conference on Computer Vision*, pages 1154–1160, 2011. ISSN 978-1-4673-0063-6. doi: 10.1109/ICCVW.2011.6130380.

Noah Snavely, Steven M. Seitz, and Richard Szeliski. Modeling the world from Internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008. ISSN 09205691. doi: 10.1007/s11263-007-0107-3.

M. J. Starek, H. Mitasova, K. W. Wegmann, and N. Lyons. Space-time cube representation of stream bank evolution mapped by terrestrial laser scanning. *IEEE Geoscience and Remote Sensing Letters*, 10(6):1369–1373, 2013. ISSN 1545598X. doi: 10.1109/LGRS.2013.2241730.

Laura Tateosian, Helena Mitasova, Brendan A Harmon, Brent Fogleman, Katherine Weaver, and Russell S Harmon. TanGeoMS: tangible geospatial modeling system. *IEEE transactions on visualization and computer graphics*, 16(6):1605–12, 2010. ISSN 1077-2626. doi: 10.1109/TVCG.2010.202. URL `http://www.ncbi.nlm.nih.gov/pubmed/20975203`.

M. J. Westoby, J. Brasington, N. F. Glasser, M. J. Hambrey, and J. M. Reynolds. 'Structure-from-Motion' photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, 179:300–314, 2012. ISSN 0169555X. doi: 10.1016/j.geomorph.2012.08.021. URL http://dx.doi.org/10.1016/j.geomorph.2012.08.021.