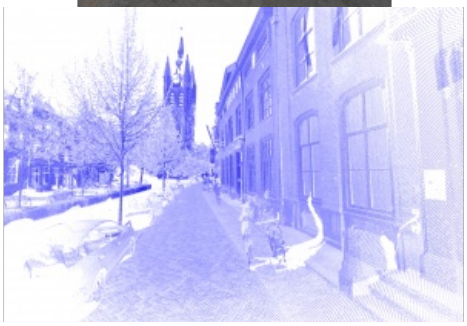
FILE AND DATABASE STORAGE METHODS FOR HUGE POINT CLOUDS

Future-proof Data Storage



Point cloud datasets have proven to be useful for many applications, ranging from engineering design to asset management. While point clouds are becoming denser and more accurate, new software is allowing an ever-broader user group for these datasets. However, due to their size and dependence on specialised tools, data management of point clouds is still complicated. A major consideration for data managers is the choice of point cloud storage format, as several different formats are available.

Point cloud data can be collected in numerous ways, with laser scanning being the most common method. In terms of data management, it is useful to distinguish between dynamic and static acquisition. Examples of dynamic acquisition are airborne and mobile Lidar, while static acquisition can be achieved with terrestrial laser scanning. Static datasets result in 'organised point clouds', which means that the interval between subsequent points is constant. This knowledge can be employed by storing the scan data in a raster where each cell corresponds to a laser return. Rasters can be stored and queried efficiently, thus simplifying the point cloud storage problem. Manufacturers of terrestrial scanners have introduced their own formats for storing points in this way. However, this method of storage is no longer applicable when registering multiple point clouds or when using data from dynamic acquisition. Instead, the real 3D coordinates for each point need to be saved individually. It is this type of data that poses the biggest challenges in terms of storage size and performance.

File or database

In the world of traditional vector GIS data, there has been a shift from file-based storage to databases. This is not yet the case for point cloud data; datasets are most commonly stored as a set of files on a local drive or a shared network location. The two major spatial databases currently on the market, Oracle Spatial and PostGIS, do provide point cloud support but their functionality is still limited and does not yet scale very well with data size. Database providers are actively working on improving point cloud support. Until such support is sufficiently reliable, file-based storage is recommended. Therefore, in practice, organisations store their point clouds subdivided into files with tiles or strips.

Figure 1, A terrestrial scan inside a factory coloured by intensity. Each pixel represents one laser return.

Data model

In the field of GIS it is common to separate the semantics of data from the actual storage, and this is also a relevant consideration for point clouds. The geometrical part of a point cloud is clear: each point is defined by a set of 3 coordinates. In addition, each point can be enriched by attributes such as point colour or intensity. There is no official semantic definition available for point clouds, but the LAS file standard does implicitly define a data model. The LAS file format was defined by the American Society for Photogrammetry and Remote Sensing (ASPRS) in 2003 and has grown to be the de facto standard in practice. The specifications of this file format define which attributes can be stored for each point, and these include class code, colour, time, flight line and pulse count.

Standards

The naive way to store a point cloud would be to generate a regular text file, providing one point per row with coordinates separated by a pre-defined character. This is a convenient format that can easily be read by many applications. However, the resulting files will be large, and data exchange can be unpredictable due to misunderstandings in the meaning of the fields in the file. Hence, various organisations have tried to standardise the storage of point clouds efficiently. The abovementioned LAS format was developed by users from the

airborne Lidar community, which resulted in a file format that was well designed for such datasets. Over time, the file format also found its use for mobile Lidar, terrestrial laser scanning and point clouds from photogrammetric dense matching. Since LAS is a binary format, it results in smaller file sizes than simple ASCII storage.

Another open standard for interchanging and archiving point cloud data is the E57 format. The development for E57 was initiated by a group of data users, scanner manufacturers and scientists who observed a need for a general-purpose point cloud storage format. E57 files are written in XML with embedded binary data to efficiently store large volumes of data such as point clouds. While the LAS format originated from airborne Lidar, the E57 format is intended to be generic to the type of scanning system. In addition to point clouds, the format supports additional meta data as well as associated 2D imagery. It also supports a wide range of attributes to be stored with each point. The real advantage of E57 is its versatility: it can be used to store terrestrial scans using the raster-based storage, but also unordered point clouds from an airborne or mobile system. However, in practice, software support for E57 is very limited. Many applications prefer domain-specific or manufacturer-specific file formats which give better performance.

Compression of Lidar data

Both the LAS and E57 file format use binary data storage to achieve a considerable reduction in storage size when compared to plain ASCII files. Further compression was achieved with the introduction of the LAZ file format by Martin Isenburg from the company rapidlasso. His method is based on entropy encoding. The principle is that the points in the file are chunked into blocks of 50,000 points. For each block, the first point is stored, together with the predicted difference with the next point. Then, for each subsequent point, only the error in the predicted difference is written to the file. Since Lidar point clouds are quite regular by nature, the predicted difference will be accurate. This means that only very small error values need to be written to the file, and storing small values takes up less space than storing large values. This principle is applied to both the coordinates as well as all attributes in the file. In practice, a compression to 10% of the original LAS file size can be achieved if the points are ordered in the file.


Figure 2, Millions of points representing part of the city of Rotterdam.

Indexing and optimisation

Reducing the size of point cloud data by compression has been a game changer. It has simplified shipment and processing of data considerably. However, fast query and display of points requires yet another feature: indexing. An index is used by the software when querying points from a file. It tells the software where to find the points inside the file. Thanks to the index, the software does not have to read each point in the file but can instead skip directly to the required position. There are many different ways to index point cloud data, and the choice of an index has a huge impact on query time and thus on the speed of the software.

Further optimisation of query speed can be obtained by re-ordering the points in the file. This is based on the fact that it will be faster to query a set of points that are located close to each other on the hard drive. Hence, query speed will be improved by storing all points that are likely to be queried at once close together.

Vendor-specific formats

Since optimisation is possible by indexing and optionally by re-ordering points as described above, software vendors often explore these options as they try to improve software performance. This has resulted in a number of proprietary vendor-specific file formats such as LizardTech's MrSID MG4, Bentley's POD and Euclideon's Unlimited Detail. A recent addition to this list is Esri's zLAS format. This format was introduced by Esri at the beginning of 2014 as its preferred storage format for point cloud data. The data model of this file is based on the standard LAS format. However, when points are stored in zLAS they are optionally reorganised and indexed, resulting in improved query and display times.


Figure 3, A mobile mapping scan in the city of Delft.

For end users, the presence of multiple file formats complicates the decision about which file format to use. As the files are very big, converting them to another format is undesirable. If a user relies fully on a single software stack, opting for the related vendor-specific format is a sensible choice provided that conversion to LAS is available. If a user relies on multiple applications, as is often the case right now, it might be more beneficial to stick to the open LAZ format, which can be read by most software and can easily be converted to LAS, thus making it adaptable to future requirements. If the point cloud is only used for visualisation and not for analysis or vector mapping, a dedicated visualisation file format such as POD, Euclideon or zLAS could be considered since this will give better performance.

Future developments

Standardisation has always been key to the geomatics field in allowing efficient exchange of data between software packages and organisations. Because point cloud acquisition software and systems are still developing fast, the formats we use will keep changing in the years to come. This in turn will make it hard to arrive at a unified standard that will serve all applications. Meanwhile, database providers are working on improving the storage of point clouds in their systems. New database paradigms, such as NoSQL or column storage databases, will also play a role.

Ultimately, file formats should not matter for the end user: points should simply stream from any source to the user application. Akin to the OGC Web Feature Service for GIS vector data, a Web Point Cloud Service might be needed. This is a topic of ongoing research by a consortium headed by Delft University, The Netherlands.

Acknowledgements

Thanks to Peter Becker, product manager at Esri, for his feedback on and improvements to this article.