Eurographics Symposium on Geometry Processing (2018)
T. Ju and A. Vaxman (Editors)

Poster

Out-of-core Resampling of Gigantic Point Clouds

Arnaud Bletterer[1], Frédéric Payan[1], Marc Antonini[1] and Anis Meftah[2]

[1]Université Côte d'Azur, CNRS, I3S
[2]Cintoo3D

Abstract

Nowadays, LiDAR scanners are able to capture complex scenes of real life, leading to extremely detailed point clouds. However, the amount of points acquired (several billions) and their distribution raise the problem of sampling a surface optimally. Indeed, these point clouds finely describe the acquired scene, but also exhibit numerous defects in terms of sampling quality, and sometimes contain too many samples to be processed as they are. In this work, we introduce a local graph-based structure that enables to manipulate gigantic point clouds, by taking advantage of their inherent structure. In particular, we show how this structure allows to resample gigantic point clouds efficiently, with good blue-noise properties, whatever their size in a reasonable time.

1. Introduction

Recently, it became quite easy to digitize large and complex 3D scenes via a set of terrestrial LiDAR acquisitions. By merging all these data, we obtain gigantic point clouds (over one billion points) that finely describe the acquired scene. However, these point clouds exhibit numerous defects in terms of sampling quality (overlapping regions, highly non-uniform distributions, etc.), and contain too many samples to be processed as they are. These drawbacks limit their usage, and make some processing very complex, or even impossible (*e.g.*, surface reconstruction). One solution is to resample these point clouds. In our context, resampling aims to reduce the amount of data while improving the quality of the distributions.

To process gigantic point clouds, many prior methods subdivide the 3D space containing them [EBN13]. Thus, without additional processing, these methods are unable to consider efficiently the local behavior of the surface described by the point cloud.

Therefore, we developed on a graph-based approach, similarly to [CTF*18]. The graph provides a discrete representation of the captured surface. However, rather than constructing a single graph on a point cloud, leading to a representation hardly scalable, we chose to create a set of graphs from the connectivity of the acquisitions, *i.e.*, by using depth maps. As a consequence, the memory required for any local processing can be bounded by the number of graphs involved, instead of the whole point cloud.

2. A set of local graphs to describe the captured surface

Construction For each acquisition, a terrestrial LiDAR scanner provides a depth map. A depth map is a 2D image whose intensities represent the distance between the points acquired and the position of the scanner. It is composed of *valid pixels*, *i.e.*, acquired points, and *non-valid pixels*, representing directions with no depth information (intensity equal to zero). A depth map can be seen as a structured representation of the acquired scene, and we consider its connectivity to construct a graph over this part of the point cloud.

For a given depth map, the points belonging to distinct elements of the scene but projected to neighboring pixels in the depth map must not be connected in the graph. For this purpose, we combine morphological gradients [RSB93] with an adaptive thresholding in order to tag as non-valid the pixels around the highest depth variations. From this step, a graph $G(V,E)$ is constructed, where the vertices V are the valid pixels, and where the edges E connect neighboring valid pixels (points belonging to the same element).

Connection between the local graphs To link all the graphs relative to the set of acquisitions, we establish correspondences between the vertices of the different graphs that describe the same part of the surface. More precisely, for each vertex $v \in V_i$ of a given graph $G_i = (V_i, E_i)$, we associate a set of *corresponding vertices* in the other graphs (if such vertices exist). As the acquisitions are registered together, we use the *transition functions* τ which links all the pairs of depth maps.

Occasionally, a single acquisition can capture hundreds of millions of points. In that case, the associated graph may not fit into memory. To tackle this problem, our algorithm splits *a priori* a depth map into a set of *overlapping tiles*, and their respective transition functions τ are determined (they are simply translations of the transition function of the original depth map). Hence, processing the set of original depth maps or these sets of overlapping tiles is equivalent. As a consequence, the local graphs can be constructed and then connected on any computer, whatever the size of the acquisitions, and the memory limitations, by simply controlling the size of the *overlapping tiles*.

DOI: 10.2312/sgp.20181178

Managing the overlapping regions To avoid superfluous computations in the overlapping regions, for a given set of corresponding vertices, only the one belonging to the acquisition with the highest sampling density is considered during the computations. Hence, for each graph, only a subset of vertices $V_i^+ \subseteq V_i$ has to be processed. The other ones will fetch the results from the other graphs.

3. Resampling gigantic point clouds using local graphs

Now we show how the popular *dart throwing* algorithm [Coo86] can be implemented efficiently over a set of local graphs, to get a maximal Poisson-disk sampling of the underlying surface of a gigantic point cloud. Due to its principle, this algorithm is able to process the local graphs in a sequential manner. Let $G = \{G_1, G_2, ..., G_n\}$ be the set of n local graphs constructed using the aforementioned approach. For each graph $G_i \in G$:

- Maximally sample G_i, by considering the vertices V_i^+ as the candidate samples;
- For each other graph $G_j \in G, i \neq j$, the vertices V_j fetch the information of inclusion (or not) in a specific disk from their corresponding vertex in V_i.

4. Experimental results and Discussion

Figure 1 gives an example of point clouds generated with our resampling. On the left a uniform sampling, on the right a curvature-aware sampling to enhance detailed areas. This is one interesting aspect of our graph-based approach: many metrics can be modeled, by simply modifying the weights associated to the edges of the graphs.

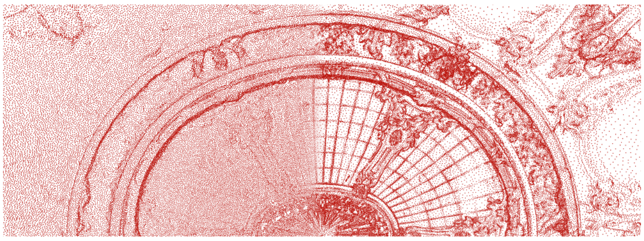

Figure 1: *Example of two distributions obtained with our graph-based approach on* Interior *(0.15% of the 2 billion original points remaining).*

To validate the global sampling quality of our distributions, we also analyzed the resulting point clouds with the tool [WW11] on synthetic models. The results are satisfactory: RAPS and anisotropy of our distributions always exhibit nice blue noise properties, which are typical of Poisson-disk samplings (Figure 2 shows an example).

We also evaluated the performances of our algorithm in term of timings and memory on many gigantic acquisitions of real-life scenes (Table 1 gives two examples). Observe how the size of the tiles allows us to control the peak memory reached, whatever the size of the input data.

Finally, Figure 3 shows that nice reconstructions can also be obtained from our resampled point clouds.

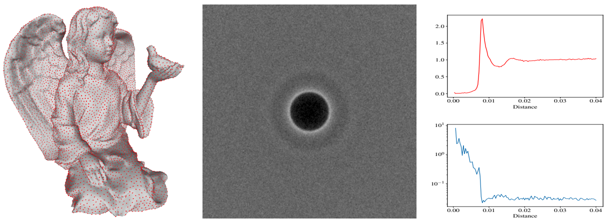

Figure 2: *Quality of the distributions obtained with our graph-based approach. The red and blue curves correspond to the RAPS and the anisotropy, respectively.*

		Time (h:m) / Peak mem. (GB)	
Model	**#pts. (#acq.)**	**8192x8192**	**4096x4096**
Facade	977M (19)	00:59 / 6.1	01:36 / 2.1
Interior	2.0B (35)	02:38 / 7.5	06:55 / 2.0

Table 1: *Computing time and peak memory of our algorithm, w.r.t the size of the overlapping tiles (with the same disk radius).*

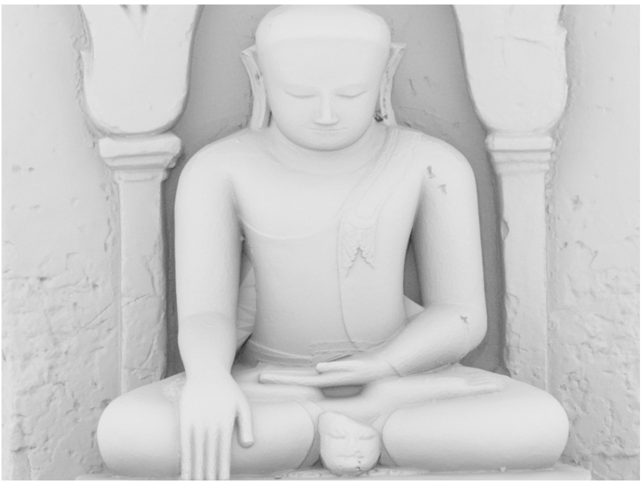

Figure 3: *Part of* Eim Ya Kyaung *resampled (2.9% of the 1.3 billion original points remaining) and reconstructed using [BL17].*

References

[BL17] Boltcheva D., Levy B.: Surface reconstruction by computing restricted voronoi cells in parallel. *Computer-Aided Design 90* (2017), 123–134. 2

[Coo86] Cook R. L.: Stochastic sampling in computer graphics. *ACM Trans. Graph. 5*, 1 (Jan. 1986), 51–72. 2

[CTF*18] Chen S., Tian D., Feng C., Vetro A., Kovačević J.: Fast resampling of three-dimensional point clouds via graphs. *IEEE Transactions on Signal Processing 66*, 3 (2018), 666–681. 1

[EBN13] Elseberg J., Borrmann D., Nüchter A.: One billion points in the cloud–an octree for efficient processing of 3d laser scans. *ISPRS Journal of Photogrammetry and Remote Sensing 76* (2013), 76–88. 1

[RSB93] Rivest J.-F., Soille P., Beucher S.: Morphological gradients. *J. Electronic Imaging 2*, 4 (1993), 326–336. 1

[WW11] Wei L.-Y., Wang R.: Differential domain analysis for non-uniform sampling. *ACM Trans. Graph. 30* (2011), 50:1–50:10. 2