High quality and efficient direct rendering of massive real-world point clouds

H. Bouchiba, R. Groscot, J-E. Deschaud, F. Goulette

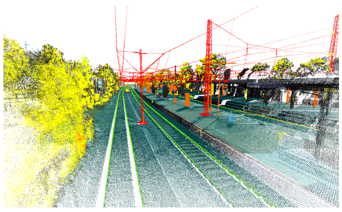
PSL★ RESEARCH UNIVERSITY PARIS
MINES ParisTech

Contribution

New real-time screen-space rendering algorithm for real-world raw 3D scanned datasets

- Scenes with high depth differences
- Datasets with linear patches
- Higher efficiency

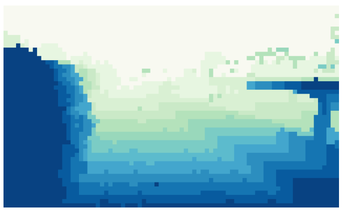
Linear Patches Segmentation



- Optional pass [RDG16]
- Does not add topological information
- Separate the linear parts from the rest
- 143.5s for 35 M points

Adaptive visibility operator

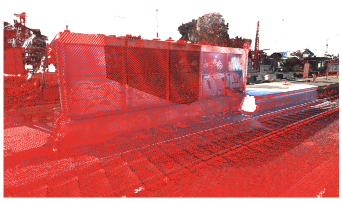
Screen-space adaptive Hidden Point Removal (HPR) operator.



We use an adaptive size of the kernel computed from the coarse depth buffer:

$$s(z) = \text{clamp}(K/z, r_{min}, r_{max})$$

The algorithm labels the holes for the subsequent filling pass:



In red: the points that are not visible

Pyramidal operators

- Down-sampling: push phase (we only keep the closest pixel to the camera)
- Filling: pull phase [MKC08]

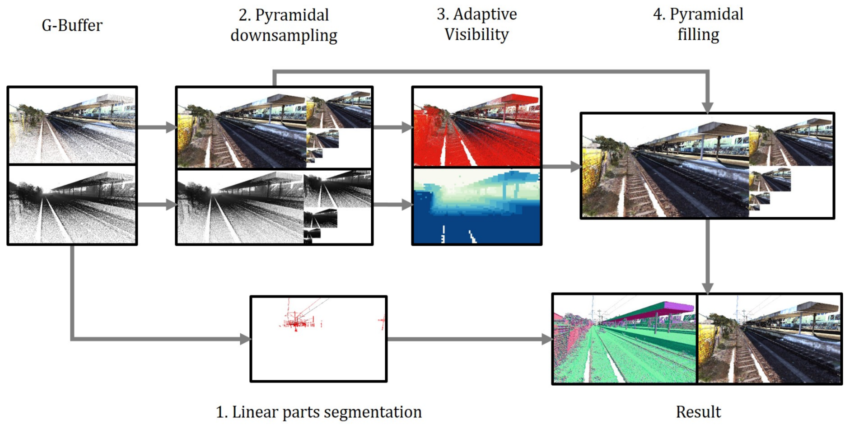
References

[MKC08] Marroquim R., Kraus M., Cavalcanti P. R.: Efficient image reconstruction for point-based and line-based rendering.

[PGA11] Pintus R., Gobbetti E., Agus M.: Real-time rendering of massive unstructured raw point clouds using screen-space operators.

[RDG16] Roynard X., Deschaud J. E., Goulette F.: Fast and Robust Segmentation and Classification for Change Detection in Urban Point Clouds.

Rendering pipeline



Results

We illustrate our work on two real-world laser scanned datasets:

- Railway MLS dataset: 600m long railway, **35 million points**.
- School of Mines TLS dataset: 14 rooms, 9 facades, **2.1 billion points**

All the images have been rendered at 1248 x 768 on a 3.4 Ghz Intel Core i7 with an Nvidia GT 640.

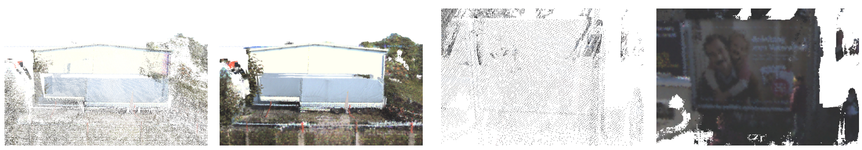
Adaptive HPR reduces the mean computational cost of the algorithm:

HPR kernel size (px)	[PGA11]*	our method*
9×9	17.4	27.4
15×15	9.1	25.6
25×25	4.2	24.4

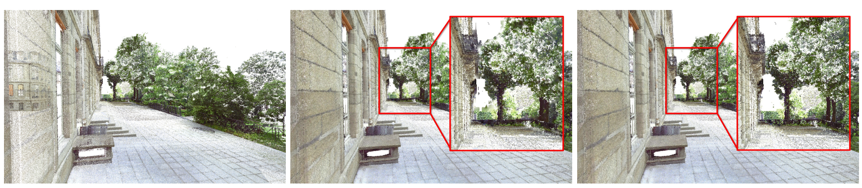
* frame rates in FPS

Railway dataset: **(a)** One pixel per point. **(b)** [PGA11] algorithm. **(c)** Our method.

Railway dataset: **(a)** One pixel per point. **(b)** [PGA11] algorithm. **(c)** our method.
Our adaptive visibility HPR does not over-detect holes in the model: the gap between the ground and the bridge is detected as a hole with [PGA11] whereas it is not with ours.



Railway dataset: Recovering a filled surface in low density regions.



School dataset: **(a)** One pixel per point. **(b)** [PGA11] algorithm. **(c)** our method.
The part of the building seen by transparency on the left is well filled by both algorithms. Our algorithm gives better results on tree edges.