PointNeRF++: A multi-scale, point-based Neural Radiance Field

Supplementary Material

We detail mutl-scale point generation and coordinate system of the global triplane. Furthermore, in the `index.html`, we provide more rendering results.

A. Multi-scale Point Generation via Grid-subsampling

We build multi-scale points from an input point cloud using grid subsampling, which is more robust to varying density as shown in KPConv [56]. Specifically, a new support point at each scale is the barycenter of the original input points contained in a grid cell. Thereby, we control the scale and density at each level via the grid size of cells.

We set the grid size at the level s as $\omega * \gamma^{s-1}$ where ω is the initial grid size at the first level and γ is the stride size. We use the larger grid size for severely incomplete point clouds and a small grid size for the complete point cloud. Specifically, for `KITTI-360` [26], we set ω as 8cm and γ as 2.92. As a result, the grid size at the coarsest point level (i.e., s=4) is 2 meters. For `ScanNet` [8], we set ω as 0.008 and γ as 2.0. For `Nerf Synthetic` [28] where point clouds are relatively complete, we set ω as 0.004 and γ as 1.6.

B. The Coordinate System of Global Triplane

We align world coordinate system and the normalized coordinate system of global triplane. We utilize the principal component analysis (PCA) to calculate the reference coordinate frame of input point cloud. The resultant reference frame consists of rotation, translation and scale, thereby defining the alignment matrix transforming world coordinates to coordinate system of global triplane. In `ScanNet` [8] and `Nerf Synthetic` [28] where points distribute uniformly along three axes and center at the origin, we simply normalize world coordinates using the scale part of reference frame. For `KITTI 360` [26], we use full reference frame instead – i.e., we rotate, translate and scale the world coordinates – because, in this dataset, the car moves along one major direction, leading to the points heavily unbalanced along three axes. With this PCA-based canonicalization, we compactly compress all possible query points into triplane's normalized frame, allowing for fully utilizing the capacity of global triplane.

C. More rendering results

We furthermore provide more rendering results – rendering more frames and forming videos. For more details, please refer to `index.html`.

References

[56] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and Deformable Convolution for Point Clouds. In *ICCV*, 2019. 1