nerf2nerf: Pairwise Registration of Neural Radiance Fields

(Supplementary Material)

A.1 Fast Global Registration

To compare against Fast Global Registration (FGR) [32], we first extract a pointcloud from each NeRF. This is computed by using expected ray termination information from the trained NeRF and on rays of the training set. To avoid having excessive noise while covering the complete scene, $\sim 10-20$ cameras are sampled from the training set, using farthest point sampling on the training set camera origins. The pointclouds are then computed from the expected ray termination of the images corresponding to the sampled cameras. Since FGR fails completely on registering the full scenes, and to have a fair comparison with the usage of keypoints $\mathcal{Q}$, the object of interest is approximately cropped out of the point clouds using $\mathcal{Q}$. To perform the crop, only points closer to the keypoints than half the maximum distance between keypoints are kept and the rest are discarded. This crop operation is performed on both scenes. The final point clouds were then further manually inspected to confirm both contain the object of interest completely. The FGR implementation from Open3D [66] is used and voxel size hyper-parameter is set to 0.001. While we chose FGR as a representative of robust pointcloud registration, we do not over-index on nerf2nerf with other pointcloud registration algorithms, as nerf2nerf is introduced to operate on a different representation.

A.2 Equal Scale Assumption

In case of scale difference between the two scenes, the fields can be easily relatively scaled so that $\mathcal{S}^{\nu}(x) = \mathcal{S}(\frac{x}{\nu})$. An approximate scale factor ν is calculated as the mean ratio of the keypoint pairs distances. This is empirically shown to suffice for the purpose of optimizing the translation and rotation variables, however for finding a more precise estimate of the scale factor, ν can be added as a parameter to our optimization process so that 6 is changed to:

$$\begin{aligned}\underset{\mathbf{R},\mathbf{t},\nu}{\arg\min}\ & (1-\lambda^{(t)})\cdot\mathcal{L}_{\text{match}}(\mathcal{S}_a, \mathcal{S}_b^{\nu}; \mathbf{T}) \\ & + \lambda^{(t)}\cdot\mathcal{L}_{\text{key}}(\mathcal{Q}_a, \mathcal{Q}_b^{\nu}; \mathbf{T}) + \ ||(1-\nu)||_2^2\end{aligned} \tag{19}$$

where surface field $\mathcal{S}_b^{\nu}$ is scaled with scale factor ν. Similarly keypoints $\mathcal{Q}_b^{\nu} = \nu \cdot \mathcal{Q}_b$ are the scaled version of the original keypoints. The added last part is a regularizer that pushes scale factor near identity.

A.3 Performance Boost

The registration time complexity can be further improved by using the more recent hybrid NeRF representations [23, 67]. These representations significantly reduce NeRF query time, and therefore impact both the queries made in optimization loop in nerf2nerf and also queries done in the surface distillation step. Additionally, a concurrent work [68, eq10] shows that surface fields can be computed on-the-fly, which reduces the distillation time to zero. This renders our robust registration method applicable to real-world robotic tasks where fast execution is required.