Common Pets in 3D:
Dynamic New-View Synthesis of Real-Life Deformable Categories

Supplementary material

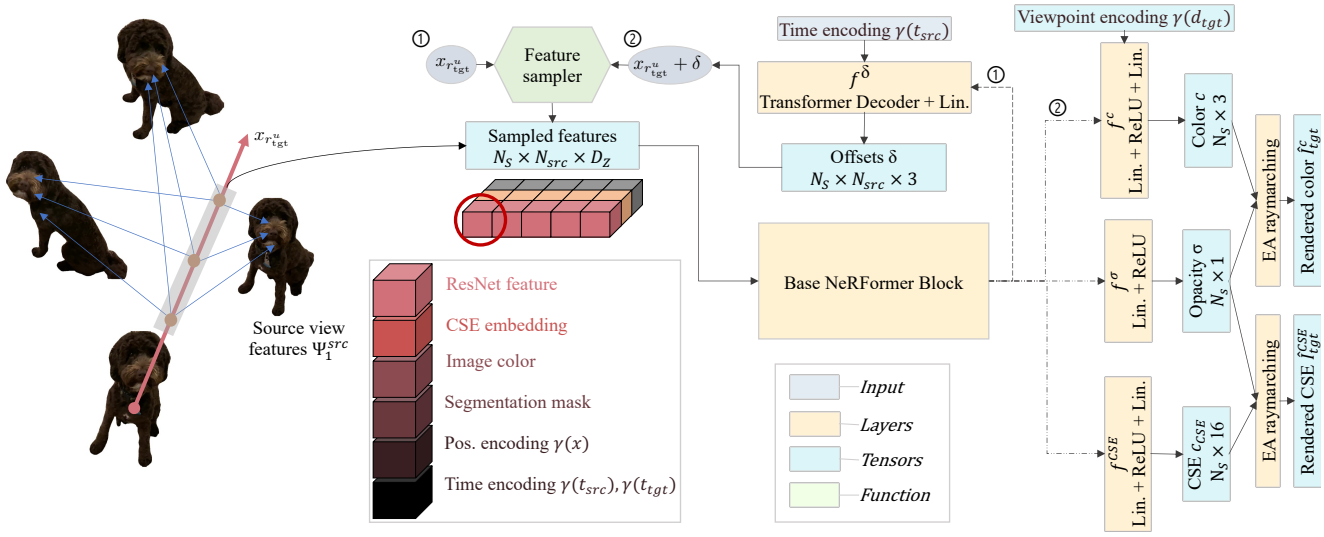

Figure I. **The architecture of Tracker-NeRF** First, source image features are bilinearly sampled given the points on the target ray $\mathbf{r}^{\mathbf{u}}_{\text{tgt}}$. The sampled features are then processed with the Base NeRFormer Block (with the same architecture as in [38]) to generate intermediate features. The latter enters the offset prediction head $\mathcal{D}_{\text{TR}}$ that generates per-point offsets δ (① in the figure). The source images are then sampled again at the locations of projected offset-adjusted ray-points. The resampled features enter Base NeRFormer Block again to predict a new intermediate feature grid which enters 3 final heads that predict colour, opacity, and CSE embedding for each ray point (② in the figure). The final color and CSE render is formed by Emission-Absorption ray marching over the predicted opacities, colors, and CSE embeddings of the ray-points.

A. Network architecture

The network architecture is summarized in Figure I. The sampled grid of TWCE encodings $Z^{\mathbf{r_u}}_{\text{TWCE}}$, or the adjusted tokens $\bar{Z}^{\mathbf{r_u}}_{\text{TWCE}}$, are processed with Base NeRFormer Block which has the same architecture as the vanilla NeRFormer from [39]. The output of the NeRFormer block is a set of intermediate features that are converted to either: 1) scene flow δ with the offset predictor $\mathcal{D}_{\text{TR}}$ during the first pass or, 2) are converted to colors $\mathbf{c}$, opacities σ, or CSE embeddings $\mathbf{C}$ with the final head f'_{TR} during the second pass. The offset predictor $\mathcal{D}_{\text{TR}}$ is implemented with a single-layer transformer decoder block that takes as input the intermediate features from the base NeRFormer block together with the encoding of the source time of each sampled point $\gamma(t^{\text{src}})$, and outputs the per-point offsets.

B. Mask annotations for CoP3D

To generate the object masks M^j_i, we input each frame I^j_i to the PointRend semantic segmentation network [16], extracting N^j_i candidate masks $\hat{M}^j_i \in [0,1]^{N^j_i \times H \times W}$. To track a single foreground object across the video, we use a Hidden Markov Model to generate the set $\{M^t_i | M^t_i \in [0,1]^{H \times W}\}_{j=1}^{N_{\mathcal{V}_i}}$ containing a single mask per frame, using intersection-over-union (IoU) of masks' bounding boxes as a pairwise potential.