Point-DynRF: Point-based Dynamic Radiance Fields from a Monocular Video

Byeongjun Park Changick Kim
Korea Advanced Institute of Science and Technology (KAIST)
{pbj3810, changick}@kaist.ac.kr

A. Overview

In this supplementary material, we further demonstrate our experimental setup and provide additional results that the scene geometry is well regressed. First, we explain the total loss formulation in our training process in Sec. B. Then, we describe implementation details with image near-far bound determination by neural points in Sec. C and provide additional results for dynamicsness map of novel views in Sec. D. Finally, we demonstrate failure cases in Sec. E.

B. Losses

Our optimization process involves utilizing the loss functions L_{rec}, L_{geo}, L_{depth}, and L_{mask}. These loss functions are either modifications of those used in DVS [1] or newly introduced in this paper. To train Point-DynRF more stable, we also incorporate with a depth order loss L_{order} introduced in DVS [1] and a sparsity loss L_{sparse} introduced in Point-NeRF [5].

Depth Order Loss While the depth adjust loss helps optimize the overall scene geometry, there are inherent challenges in accurately determining the distance between dynamic objects and the background. Therefore, we use depth order loss L_{order} to allow the dynamic radiance fields to be regularized via a frame-by-frame depth map. Since regularizing the dynamic radiance fields with per-frame depth maps has scale and shift ambiguities as mentioned earlier, we leverage the volume rendering process of Dynamic NeRF to propose L_{order} as:

$$L_{order} = \sum_{i=1}^{N} \sum_{uv} \|\tilde{\mathbf{D}}(\mathbf{r}_{uv}^{i}, i, \mathbb{P}_i) - \tilde{\mathbf{D}}^{d}(\mathbf{r}_{uv}^{i}, i, \mathbb{P}_{i,d})\|_2^2. \quad (1)$$

Sparsity Loss Following the point-based representation, we apply a sparsity loss L_{sparse} on the point-wise rigidness to enforce it to be close to zero or one as:

$$L_{sparse} = \sum_{i} (\log(\gamma_i) + \log(1 - \gamma_i)). \quad (2)$$

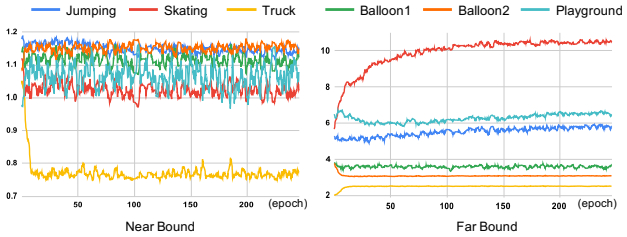

Figure 1. **Image Near-Far Bound Determination.**

Total Training Loss Formulation We formulate a reconstruction loss L_{rec}, a scene geometry loss L_{geo}, a depth adjust loss L_{depth}, a depth order loss L_{order}, a mask adjust loss L_{mask} and a sparsity loss L_{sparse}, to train our Point-DynRF and neural points. Specifically, we define $\lambda_{rec}^{full} = 3$, $\lambda_{rec}^{s} = 1$, $\lambda_{rec}^{d} = 1$ for the reconstruction loss. For the scene geometry loss, we define $\lambda_{flow} = 0.1$, $\lambda_{miss}^{s} = 1$, $\lambda_{miss}^{d} = 1$. Finally, we define $\lambda_{depth} = 0.1$, $\lambda_{order} = 0.1$, $\lambda_{mask} = 0.1$, and $\lambda_{sparse} = 0.0002$ to formulate the final loss as:

$$L_{total} = L_{rec} + L_{geo} + \lambda_{depth} L_{depth} + \lambda_{order} L_{order} + \lambda_{mask} L_{mask} + \lambda_{sparse} L_{order}.$$

C. Implementation Details.

We randomly sampled 1024 rays in a batch, and each ray was assigned up to 32 sampling points. We used COLMAP to estimate the camera poses and resized all images into a resolution of 480×272. Also, we initialized our scale and shift parameters by using near and far bounds from COLMAP. We trained Point-DynRF for $250k$ iterations, and training takes about 20 hours on a single NVIDIA Geforce RTX 3090 GPU.

Near-Far Boundary Determination As our Point-DynRF is built on Point-NeRF [5] representation, dynamic radiance fields are regressed in 3D world coordinates, not in NDC space used by previous methods. Moreover, we need to render the far background as well, so we set the image near-far bound dynamically associated with the neural

Figure 2. **Comparison to baselines on NVIDIA Dynamic Scene Dataset [6].**

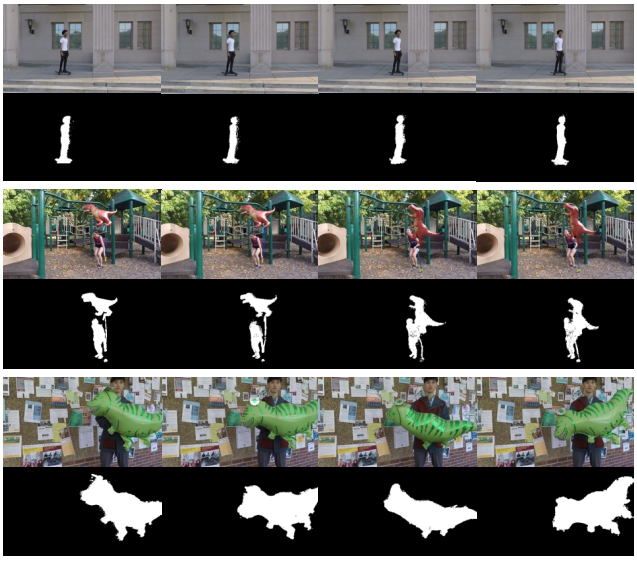

Figure 3. **Dynamicsness Maps for novel views.**

points. Specifically, we set the image near boundary to be the depth for the nearest neural point multiplied by 0.9, and the image far boundary to be the depth for the farthest neural point multiplied by 1.1. Figure 1 shows the convergence of the image near-far boundary of the scenes in the Dynamic Scene Dataset [6] during training. This result confirms that the scene geometry is stably trained and refined the initialized scene geometry well.

D. Additional Results

Additional Qualitative Results. We further provide additional qualitative results on Dynamic Scene Dataset [6] . Point-DynRF generates more realistic images compared to previous methods, and the human face in the third row of Fig. 2 confirms that Point-DynRF produces much sharper images, while other methods either fail to synthesize or produce blurry images. We also provide a video result of a causally captured monocular video that our Point-DynRF generates realistic images while the state-of-the-art method DVS [1] suffers from duplicated dynamic objects when rendering from a fixed viewpoint.

Our foreground masks $(M_1, \ldots, M_N)$ are also optimized during the training, so we provide dynamicsness maps for novel views, as shown in Fig 3. For each novel view, our Point-DynRF can render blending weights by using the volume rendering process. These dynamicsness maps for novel views confirm that our Point-DynRF well represents dynamic regions in the scene, and we can see that the static representation in the center of the person in the Playground Sequence is due to the fact that all the sequences in the input video for that region are learned as dynamic regions and represented as background by the miss ray marching scheme.

E. Failure Cases

Figure 4. **Failure Case.**

While Point-DynRF optimizes well the ambiguous initial geometry and foreground masks, it fails to represent the scene if the neural point clouds are unnaturally initialized. A combination of inaccurate camera pose, depth map, and foreground masks sometimes unnaturally initialize neural point clouds where background points are closer to the camera than dynamic points as shown in Fig. 4. In this failure

case, Point-DynRF falls short of distinguishing background points in front of the dynamic objects even addressing the scale ambiguity, and novel views also contain artifacts on these background points.

References

[1] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5712–5721, 2021. 1, 2

[2] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6508, 2021. 2

[3] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2

[4] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12959–12970, 2021. 2

[5] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5438–5448, 2022. 1

[6] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5336–5345, 2020. 2