# Root Pose Decomposition Towards Generic Non-rigid 3D Reconstruction with Monocular Videos (Appendix)

Yikai Wang[1] Yinpeng Dong[1,2] Fuchun Sun[1]✉ Xiao Yang[1]

[1]Beijing National Research Center for Information Science and Technology (BNRist),
State Key Lab on Intelligent Technology and Systems,
Department of Computer Science and Technology, Tsinghua University [2]RealAI

{yikaiw,dongyinpeng,fcsun}@tsinghua.edu.cn, yangxiao19@mails.tsinghua.edu.cn

## A. More Details and Discussions

In this part, we provide additional implementation details and discussions for our method and experiment.

**Root poses during training.** We provide Fig. 8 to compare initial and final root poses, taking two fish scenarios as examples. It is observed that the final root poses could reflect motion patterns given input video sequences. To kickstart with a reasonable initial pose, we follow ViSER [6] that adopts optical flow for learning approximate pixel-surface embeddings (which are not category-specific). But due to our proposed designs, our performance significantly outperforms ViSER as compared in Fig. 4 (main paper).

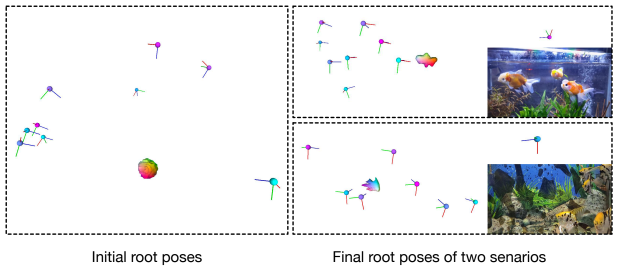

Figure 8. Illustration of initial root poses and final root poses with corresponding canonical spaces. Best view in color and zoom in.

**Visualization of canonical space.** As mentioned in Sec. 4.1 (main paper), we adopt 25 control points when optimizing linear skinning weights. Fig. 9 provides two examples of the learned models in the canonical space on OVIS (fish), with also control points. By illustration, we observe that these canonical models well capture the geometric shapes of target objects. Besides, mostly, we find that using 25/30 control points gets similar results. As a result, we follow BANMo [7]'s default setting (25 control points) for fair comparison.

**Point sampling for registration.** As mentioned in the main paper, for registration, we sample points in the camera space based on their $\tau^n$ values. Specifically, we keep a per-frame buffer that contains a maximum of $10^4$ points. Old points are removed from the buffer if the maximum volume is exceeded. During training, we discard a ray if its all $\tau^n$ values are lower than $10^{-3}$. We then sample points by the probability of a $\mathrm{Softmax}$ output over $\tau^n$ values, with a temperature of $0.01$.

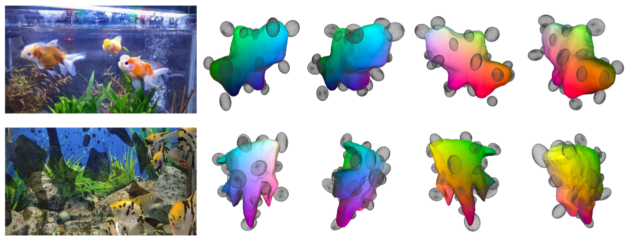

Figure 9. Illustration of models in the canonical space on the OVIS dataset with learned control points.

**Decomposition into $\mathrm{Sim}(3)$ or $\mathrm{SO}(3)$.** In Sec. 3.2 (main paper), we leverage dense $\mathrm{Sim}(3)$ with scaling factors $s_t$ to deal with the scale change between different shapes. If we substitute the dense field with $\mathrm{SO}(3)$ by disregarding $s_t$, we find an accelerated convergence process when learning the canonical space, but the root pose might be inaccurate when encountering individual differences, such as the height difference.

**Why using both decomposed poses and the root pose.** The deformation field introduces ambiguities that make optimization more challenging, especially when learning skinning weights. We address this issue by maintaining the global transformation, as described in our main paper.

**Object occlusions.** Compared with the multi-view 3D reconstruction, the issue of object occlusion is less explored when only given monocular videos. We demonstrate that the framework could handle object occlusions. For a point $\mathbf{x}_*$ on the object surface of the canonical space, denote $\mathbf{p}_t \in \mathbb{R}^2$ as the projected 2D pixel that corresponds to $\mathbf{x}_*$ given

✉ Corresponding author: Fuchun Sun.

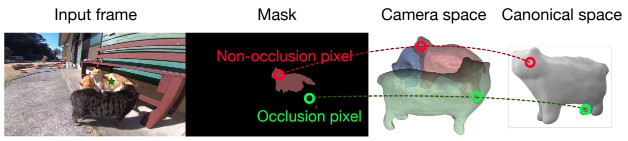

Figure 10. Illustration of an occlusion pixel that occludes a given target object, in this case an orange and white cat.

the transformation $\mathbf{G}_t$, attained by,

$$\mathbf{p}_t = \Pi_t \mathbf{G}_t^{-1} \mathcal{M}_{*\to t}(\mathbf{x}_*), \tag{12}$$

where $\Pi_t$ is the video-specific projection matrix of a pinhole camera.

We call $\mathbf{p}_t$ an occlusion pixel if it is outside the annotated 2D mask, as illustrated in Fig. 10. In summary, an occlusion pixel is the pixel that occludes a given target object, which comes from a surface point $\mathbf{x}_*$ by Eq. (12) yet locates outside the silhouette/mask. Given a target object, denote $\mathbb{U}$ as a set that contains all occlusion pixels. We design an anti-occlusion silhouette reconstruction loss[1] defined by,

$$\mathcal{L}_{\text{sil}} = \sum_{\mathbf{p}_t} \mathbb{A}_{\mathbf{p}_t \notin \mathbb{U}} \left\| \sum_n \tau^n_{\mathbf{p}_t} - \mathbb{I}_{\mathbf{p}_t} \right\|^2, \tag{13}$$

where $\sum_n \tau^n_{\mathbf{p}_t}$ sums over sampled points along the camera ray that emanates from the pixel $\mathbf{p}_t$, and $\mathbb{I}_{\mathbf{p}_t} \in \{1, 0\}$ is an indicator function implying whether $\mathbf{p}_t$ belongs to the segmentation mask of the target object. $\mathbb{A}_{\mathbf{p}_t \notin \mathbb{U}} \in \{1, \alpha\}$ is another indicator function that equals to 1 if $\mathbf{p}_t \notin \mathbb{U}$ otherwise $\alpha$, where $\alpha$ is annealing parameter that is initialized to 1 and decays during the training process.

Similarly, the method is relatively robust to out-of-frame pixels since they are not mistakenly penalized by Eq. (13).

## B. Loss Functions

In our main paper, we basically describe loss functions that are specifically designed/adopted for our method. Here, we detail other loss functions and the formulation of summing all loss functions together.

Following the standard pipeline [3, 8], we adopt a color reconstruction loss $\mathcal{L}_{\text{rgb}}$, computed as

$$\mathcal{L}_{\text{rgb}} = \sum_{\mathbf{p}_t} \left\| \sum_n \tau^n \mathbf{c}_t\big(\mathcal{W}_{t\to *}(\mathbf{x}^n_t)\big) - \hat{\mathbf{c}}_t|_{\mathbf{p}_t} \right\|^2, \tag{14}$$

where $\mathbf{x}^n_t$ is the $n$-th point emanates from the pixel $\mathbf{p}_t$; the color $\mathbf{c}_t(\cdot)$ is defined by Eq. (2) in our main paper; and $\hat{\mathbf{c}}_t|_{\mathbf{p}_t}$ denotes the observed color at the pixel $\mathbf{p}_t$.

We further calculate an optical flow loss $\mathcal{L}_{\text{flow}}$ with a similar formulation with existing methods [1, 7],

$$\mathcal{L}_{\text{flow}} = \sum_{\mathbf{p}_t,(t,t')} \left\| \mathcal{F}\big(\mathbf{p}_t, t \to t'\big) - \hat{\mathcal{F}}\big(\mathbf{p}_t, t \to t'\big) \right\|^2, \tag{15}$$

[1]Note that $\tau^n$ is only optimized by Eq. (13) while is temporally frozen when minimizing other terms such as Eq. (5) in main paper and Eq. (17).

where the computed optical flow $\mathcal{F}(\mathbf{p}_t, t \to t') = \mathbf{p}_{t'} - \mathbf{p}_t$, and the observed optical flow $\hat{\mathcal{F}}(\mathbf{p}_t, t \to t')$ is estimated by an off-the-shelf flow network, VCN-robust [5]. Following BANMo [7], the pixel $\mathbf{p}_{t'}$ at time $t'$ is obtained by,

$$\mathbf{p}_{t'} = \sum_n \tau_n \Pi_{t'} \Big( \mathcal{W}_{*\to t'}\big(\mathcal{W}_{t\to *}(\mathbf{x}^n_t)\big)\Big), \tag{16}$$

where $\Pi_{t'}$ is the video-specific projection matrix (at time $t'$) of a pinhole camera.

For optimization, we adopt a color reconstruction loss [3, 8] and an optical flow loss [7]. We optimize $\tau^n$ by applying an anti-occlusion silhouette reconstruction loss by Eq. (13). Similar to NSFF [1], we maintain the cycle consistency between deformed frames for the monocular reconstruction with a 3D consistency loss given by,

$$\mathcal{L}_{\text{3D-cyc}} = \sum_{i,n} \tau^n \left\| \mathcal{W}_{*\to t}\big(\mathcal{W}_{t\to *}(\mathbf{x}^n_t)\big) - \mathbf{x}^n_t \right\|^2_2. \tag{17}$$

The consistency loss $\mathcal{L}_{\text{cyc}}$ is composed by the 2D part ($\mathcal{L}_{\text{rgb}}$, $\mathcal{L}_{\text{flow}}$, $\mathcal{L}_{\text{sil}}$) and the 3D part ($\mathcal{L}_{\text{3D-cyc}}$), namely,

$$\mathcal{L}_{\text{cyc}} = \mathcal{L}_{\text{rgb}} + \mathcal{L}_{\text{flow}} + \mathcal{L}_{\text{sil}} + \mathcal{L}_{\text{3D-cyc}}. \tag{18}$$

As mentioned in the main paper, the total loss function $\mathcal{L}$ is summarized as

$$\mathcal{L} = \sum_{k=1}^{K} \mathcal{L}^{(k)}_{\text{cd}} + \mathcal{L}_{\text{ela}} + \mathcal{L}_{\text{cyc}}, \tag{19}$$

where $\mathcal{L}^{(k)}_{\text{cd}}$ is the chamfer distance loss function at the $k$-th pyramid level, and $\mathcal{L}_{\text{ela}}$ denotes the penalty loss for as-rigid-as-possible movement regularization, both of which have been introduced in the main paper.

## C. More Visualizations

We provide Fig. 11 to evaluate RPD on reconstructing ducks. The experiment is performed by jointly using a 4-second video and a 15-second complicated video with heavy occlusions.

As mentioned in our main paper, for ease of optimization, we let $\tilde{\mathbf{T}}_t \equiv \mathbf{T}_t$ for all points but learn the per-point rotation matrix $\tilde{\mathbf{R}}_t$. Setting $\tilde{\mathbf{T}}_t \equiv \mathbf{T}_t$ assumes the camera is at the roughly same distance to the object, which might lead to failure cases when objects quickly running towards the camera, especially for multi-object cases. To examine the performance, we provided Fig. 12 which reconstructs chickens. We observe that when a target object rapidly changes its distance to the camera, the reconstruction becomes coarse and the performance is barely acceptable.

A failure case is depicted in Fig. 13, where the root pose encounters a rapid change in the 3rd second, leading to ambiguous pose estimation and confusion in distinguishing the head and tail in the camera space.

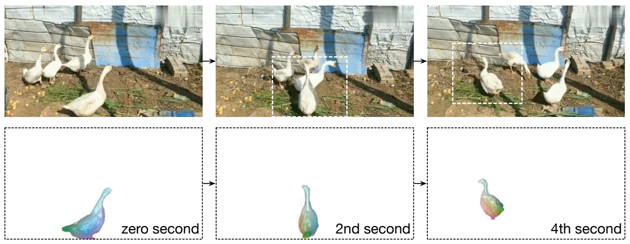

Figure 11. Illustration of reconstructing a duck.

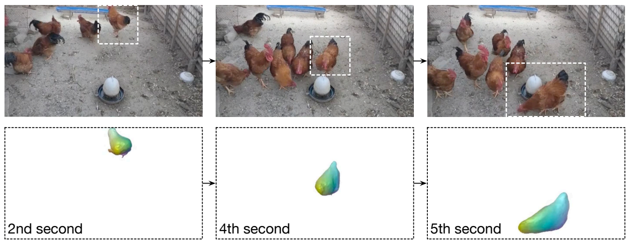

Figure 12. Illustration of reconstructing a chicken which quickly changes its distance to the camera.

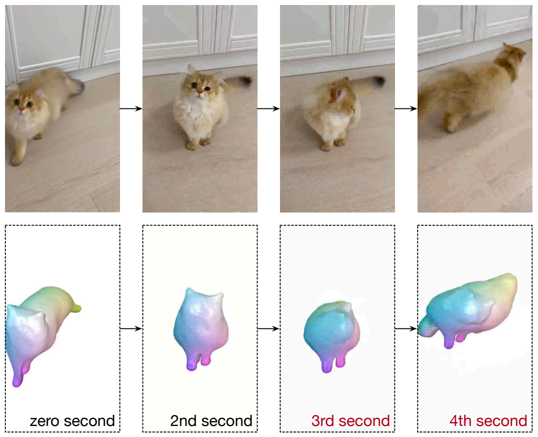

Figure 13. Failure case: illustration of reconstructing a cat which has a rapid pose change in the 3rd second.

## References

[1] Hongrui Cai, Wanquan Feng, Xuetao Feng, Yan Wang, and Juyong Zhang. Neural surface reconstruction of dynamic scenes with monocular RGB-D camera. In *NeurIPS*, 2022. 2

[2] Yang Li and Tatsuya Harada. Non-rigid point cloud registration with neural deformation pyramid. In *NeurIPS*, 2022.

[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. In *ECCV*, 2020. 2

[4] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *ICCV*, 2021.

[5] Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. In *NeurIPS*, 2019. 2

[6] Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Ce Liu, and Deva Ramanan. Viser: Video-specific surface embeddings for articulated 3d shape reconstruction. In *NeurIPS*, 2021. 1

[7] Gengshan Yang, Minh Vo, Natalia Neverova, Deva Ramanan, Andrea Vedaldi, and Hanbyul Joo. Banmo: Building animatable 3d neural models from many casual videos. In *CVPR*, 2022. 1, 2

[8] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *NeurIPS*, 2020. 2