Humans as a Calibration Pattern: Dynamic 3D Scene Reconstruction from Unsynchronized and Uncalibrated Videos

Supplementary Material

A. Dataset Details

CMU Panoptic Studio CMU Panoptic Studio dataset contains dynamic sequences of human movement from 31 cameras surrounding the scene. We take subsequences from sports and office sequence and make new dataset containing five independent scenes Baseball, Tennis, Office1, Office2, and Office3. There is one human in the scenes in Panoptic Studio dataset we used. Each multi-view training video is 270 frames long at 30 FPS and starts from a random global timestamp to make unsynchronized setup. All video sequences are sampled to have at least 150 overlapping frames. Namely, maximum time offset between two videos is 120 frames. We undistort all training images before estimating human motion and training dynamic NeRFs with provided radial and tangential distortion parameters.

Mobile-Stage Mobile-Stage dataset from 4K4D contains three dancers captured by multiple smartphone cameras with frontal and side views. Some viewpoints have significant occlusion, and not all people are visible in certain viewpoints. The video lengths, FPS, and random global timestamp sampling strategy are identical to the setup in CMU Panoptic Studio dataset. We use 20 cameras for training and one camera for evaluation.

EgoBody EgoBody dataset contains dynamic interaction between two humans that are captured from four (S22-S21-02) or five (S32-S31-01 and S32-S31-02) static Kinect cameras and one moving head-mounted HoloLens2 camera. HoloLens2 camera has different camera intrinsic from Kinect cameras and it has some missing frames. Also, there are extreme motion blurred frames in HoloLens2 camera videos. We estimate human pose for existing frame with SLAHMR and estimate human pose for missing frames by linear interpolation of 3D joint positions. Each multi-view training video is 200 frames long. They have at least 90 overlapping frames, in other words, maximum time offset between two videos is 110 frames.

B. Implementation Details

B.1. Training K-Planes

We use L=5 spatial grid resolutions $[24, 48, 96, 192, 384]$ for Mobile-Stage dataset and Office1, Office2, Office3, and Tennis scenes of CMU Panoptic Studio dataset, and we use L=4 grid resolutions $[48, 96, 192, 384]$ for Baseball scene of CMU Panoptic Studio dataset. We observe that Baseball scene converges well starting from the resolution of 48. We use a single resolution, 240, for the temporal grid in all scenes.

In addition to the weight scheduling described in the main manuscript, we also schedule the weights of regularization terms. We apply cosine scheduling that decreases weights to $1/100$ of its initial weights at the end of the scheduling. We use weights 0.01 for distortion loss, 0.001 for L1 loss in time planes, 0.001 for total variance loss in spatial planes, 0.01 for time smoothness loss, and 0.01 for density L1 loss. We start scheduling of regularization from 100k steps for Mobile-Stage dataset and Office1, Office2, Office3, and Tennis scenes of Panoptic Studio dataset, and 50k steps for Baseball scene of Panoptic Studio dataset, and end scheduling at 150k steps.

For efficiency, we initialize feature values of finer grids by bilinear interpolation of values from coarser grids. Namely, we initialize finer grids $\mathbf{P}^c_l, (l > 1)$ at $\alpha = l - 1$ with values interpolated from $\mathbf{P}^c_{l-1}$, where $\alpha = L(e^\eta - 1)/(e - 1)$, $\eta \in [0, 1]$ is a normalized training step.

B.2. Global Alignment Algorithm

We provide detailed pseudocode of the global sequence alignment of whole human motions for time offset estimation in Algorithm 1.

B.3. Procrustes Alignment

As we describe in Eq. (7) in the main manuscript, we estimate similarity transform between two 3D joint positions. We first estimate scale, translation, and rotation that align target joint positions $(\mathbf{J}^i_{\text{global},t+\Delta t^i}; t)$ to the reference joint positions of anchor index α, $(\mathbf{J}^\alpha_{\text{global},t+\Delta t^\alpha}; t)$ with Procrustes analysis,

$$s_i, s_\alpha, \mathbf{t}_i, \mathbf{t}_\alpha, R = \textsc{Procrustes}((\mathbf{J}^i_{\text{global},t+\Delta t^i}; t), (\mathbf{J}^\alpha_{\text{global},t+\Delta t^\alpha}; t)). \quad (1)$$

We describe details of the Procrustes analysis in Algorithm 2. Then we can obtain camera poses in the global coordinate (i.e., camera coordinate of anchor index) by applying estimated transformation to the camera poses in the ith camera's coordinate, R^i, τ^i similar to line 3-7 in Algorithm 3.

B.4. Evaluation Details

In this section, we provide additional details for evaluation of our method. Since we only optimize camera poses

Algorithm 1: Global time offset alignment

```
Function GLOBAL ALIGN($(C, \Delta T)$):
    Input  : Cost matrix $C \in \mathbb{R}^{N \times N}$,
             time offset matrix $\Delta T \in \mathbb{Z}^{N \times N}$
    Output: Globally aligning time offsets $\Delta t \in \mathbb{Z}^N$
    Globally aligning time offsets $\Delta t = \mathbf{0} \in \mathbb{Z}^N$
    Globally aligned index group $\mathbf{G} = \varnothing$
    Locally aligned index group list $\mathbf{G}_l = \varnothing$
    Index list $\mathbf{I} = \{(i, j) | \forall i < j\}$
    $\mathbf{I} \leftarrow$ SORT($\mathbf{I}$)          ▷ increasing order w.r.t. $C_{ij}$
    $(i, j) \leftarrow \mathbf{I}[0]$                ▷ anchor indices
    $\Delta t[i] \leftarrow 0, \Delta t[j] \leftarrow \Delta T_{ij}$
    Insert $(i, j)$ to $\mathbf{G}$
    for $k = \{1, \cdots, N(N-1)/2 - 1\}$ do
        $(i, j) \leftarrow \mathbf{I}[k]$
        if $i \in \mathbf{G}, j \in \mathbf{G}$ then
            continue
        else if $i \in \mathbf{G}, j \in \mathbf{G}_l[k], \exists k$ then
            Pop $\mathbf{G}_l[k]$ and add to $\mathbf{G}$ after shift $\Delta T_{ij}$
        else if $i \in \mathbf{G}, j \notin \mathbf{G}, j \notin \mathbf{G}_l[k], \forall k$ then
            Add $j$ to $\mathbf{G}$, $\Delta t[j] \leftarrow \Delta t[i] + \Delta T_{ij}$
        else if $i \in \mathbf{G}_l[k], j \notin \mathbf{G}, j \notin \mathbf{G}[l], \forall l$ then
            Add $j$ to $\mathbf{G}_l[k]$, $\Delta t[j] \leftarrow \Delta t[i] + \Delta T_{ij}$
        else if $i, j \notin \mathbf{G}, \notin \mathbf{G}_l[k], \forall k$ then
            Add $(i, j)$ to new group in $\mathbf{G}_l$
            $\Delta t[i] \leftarrow 0, \Delta t[j] \leftarrow \Delta \mathbf{T}_{ij}$
        else if $i \in \mathbf{G}_l[k], j \in \mathbf{G}_l[l]$ then
            Pop $\mathbf{G}_l[l]$ and add to $\mathbf{G}_l[k]$ after shift $\Delta T_{ij}$
        else if $i, j \in \mathbf{G}_l[k]$ then
            continue
        else
            vice versa for reverse case of $(i, j)$
    return $\Delta t$
```

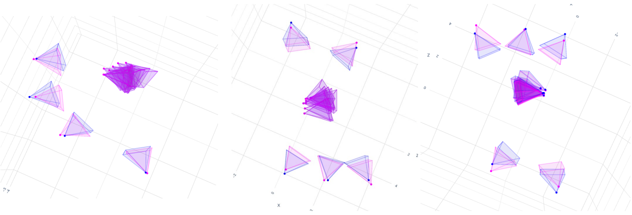
Algorithm 2: Procrustes analysis

```
Function PROCRUSTES($(X, Y)$):
    Input  : Point set to align $X = \{\mathbf{x}_i | \mathbf{x}_i \in \mathbb{R}^3\}_{i=1}^N$,
             Reference point set $Y = \{\mathbf{y}_i | \mathbf{y}_i \in \mathbb{R}^3\}_{i=1}^N$
    Output: scale $s_x, s_y$, translation $\mathbf{t}_x, \mathbf{t}_y$, rotation R
    $\mathbf{t}_x \leftarrow \sum \mathbf{x}_i / N, \mathbf{t}_y \leftarrow \sum \mathbf{y}_i / N$
    $s_x \leftarrow \sqrt{\sum \|\mathbf{x}_i - \mathbf{t}_x\|_2^2 / N}$
    $s_y \leftarrow \sqrt{\sum \|\mathbf{y}_i - \mathbf{t}_y\|_2^2 / N}$
    $\hat{X} \leftarrow \frac{1}{s_x}([\mathbf{x}_i] - \mathbf{t}_x)$
    $\hat{Y} \leftarrow \frac{1}{s_y}([\mathbf{y}_i] - \mathbf{t}_y)$
    $U, \Sigma, V^* \leftarrow \text{SVD}(\hat{Y} \hat{X}^\top)$
    $R \leftarrow UV^*$
    return $s_x, s_y, \mathbf{t}_x, \mathbf{t}_y, R$
```

and time offsets of training videos, we do not have accurate poses and time offsets of test videos in the coordinate system that we are optimizing training camera poses and time offsets. Therefore, we first transform ground-truth test camera poses by aligning the ground-truth training camera poses to the estimated training camera poses. Starting from the transformed test camera poses, we further optimize camera poses while freezing NeRF parameters with supervision of test view video frames before measuring errors of rendered images.

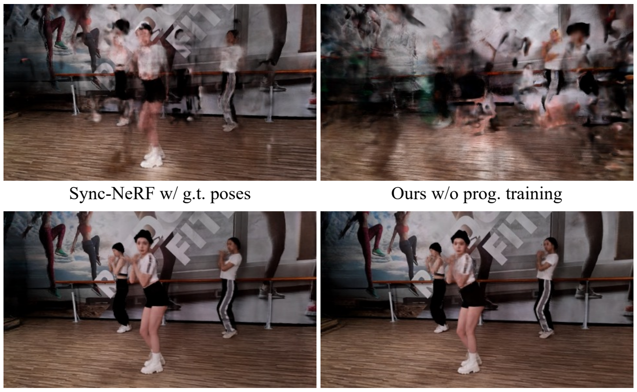
Algorithm 3: Align cameras

```
Function ALIGN CAM($(\{R_{est}^i, \tau_{est}^i\}, \{R_{ref}^i, \tau_{ref}^i\})$):
    Input  : Estimated camera poses $\{R_{\text{est}}^i, \tau_{\text{est}}^i\}$,
             Reference camera poses $\{R_{\text{ref}}^i, \tau_{\text{ref}}^i\}$
    Output: Aligned estimated camera poses $\{\tilde{R}_{\text{est}}^i, \tilde{\tau}_{\text{est}}^i\}$
    Estimated camera centers $\mathbf{o}_{\text{est}}^i \leftarrow -R_{\text{est}}^{i\top} \tau^i$
    Reference camera centers $\mathbf{o}_{\text{ref}}^i \leftarrow -R_{\text{ref}}^{i\top} \tau^i$
    $s_{\text{est}}, s_{\text{ref}}, \mathbf{t}_{\text{est}}, \mathbf{t}_{\text{ref}}, R \leftarrow$ PROCRUSTES($(\{\mathbf{o}_{\text{est}}^i\}, \{\mathbf{o}_{\text{ref}}^i\})$)
    $\tilde{\mathbf{o}}_{\text{est}}^i \leftarrow s_{\text{ref}} R(\frac{1}{s_{\text{est}}}(\mathbf{o}_{\text{est}}^i - \mathbf{t}_{\text{est}})) + \mathbf{t}_{\text{ref}}^i$
    $\tilde{R}_{\text{est}}^i \leftarrow R_{\text{est}}^i R^\top$
    $\tilde{\tau}_{\text{est}}^i \leftarrow -\tilde{R}_{\text{est}}^{i\top} \tilde{\mathbf{o}}_{\text{est}}^i$
    return $\tilde{R}_{\text{est}}^i, \tilde{\tau}_{\text{est}}^i$
```



S22-S21-02 S32-S31-01 S32-S31-02

Figure 1. Visual illustration of estimated camera poses from our initialization stage on EgoBody dataset. Red and blue frustums are the ground-truth and estimated camera poses, respectively.



Sync-NeRF w/ g.t. poses

Ours w/o prog. training

Ours

Oracle (g.t. poses & time)

Figure 2. Additional qualitative results on Mobile-Stage dataset.

Since the estimated camera poses are up to 3D similarity transformation (scale, rotation, and translation), we align our estimated camera poses to the ground-truth training camera poses before measuring pose errors. Detailed description of camera alignment procedures used in both novel-view synthesis performance measurement and camera pose accuracy can be found in Algorithm 3.

C. Additional Results

We additionally visualize both initialized and refined camera poses in all of the scenes in EgoBody dataset in Fig. 1 and Panoptic Studio in Fig. 3. We can observe that our initialization step produces good initial points, and our joint optimization with dynamic NeRF produces near-perfect pose alignments across all scenes.

Furthermore, we show additional qualitative comparisons in Panoptic Studio dataset in Fig. 4 and Mobile-Stage dataset in Fig. 2. We also provide videos rendered at the test viewpoint in the supplementary material. We recommend the readers to see the videos.

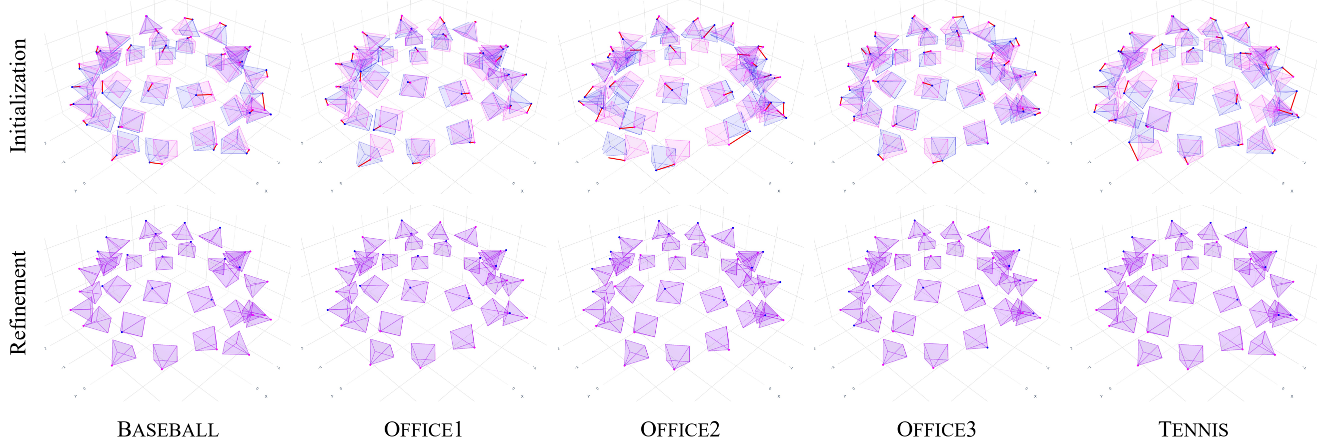


Figure 3. We demonstrate camera pose estimation results of the initialization stage on Panoptic Studio datsaet at the top row (Initialization) and the final results of the joint optimization with K-Planes at the bottom row (Refinement). Red frustums are the ground-truth camera poses and blue frustums are the estimated camera poses.

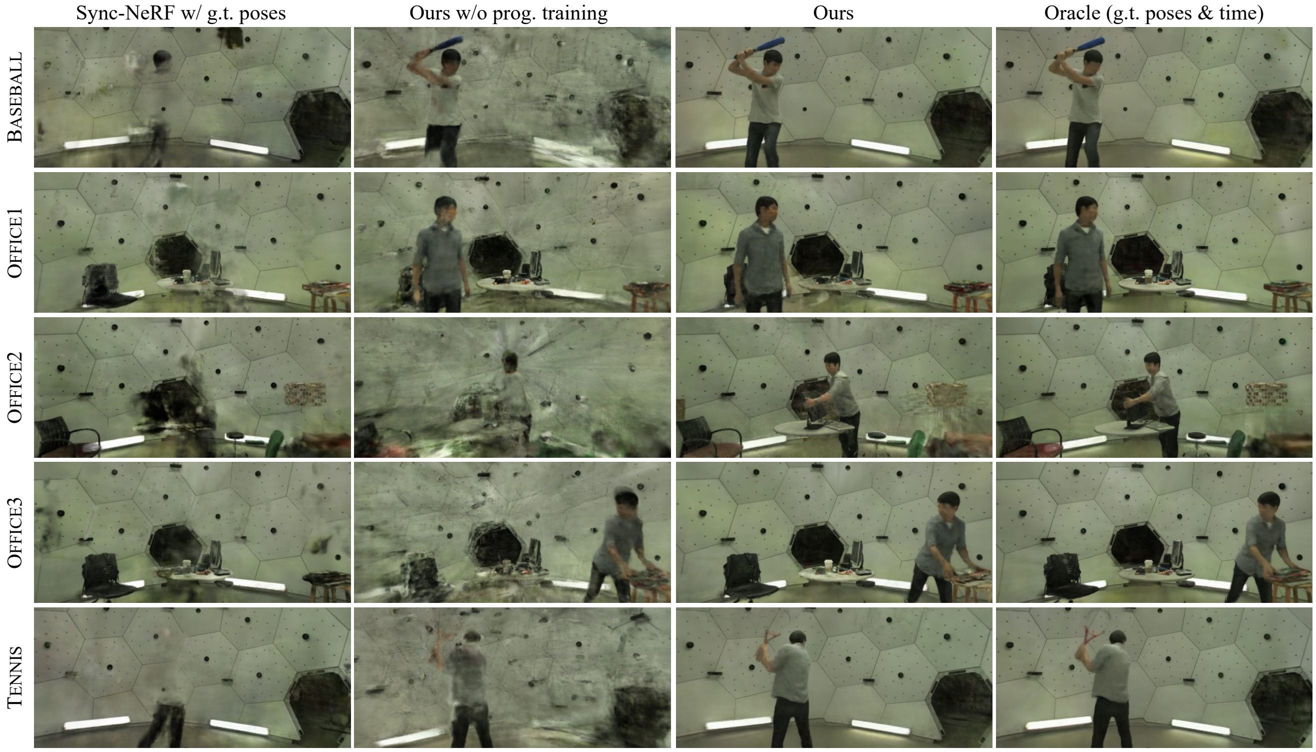


Figure 4. Additional qualitative comparison of novel view synthesis performance.