Instant-NVR: Instant Neural Volumetric Rendering for Human-object Interactions from Monocular RGBD Stream

Yuheng Jiang\textsuperscript{1,2}, Kaixin Yao\textsuperscript{1,2}, Zhuo Su\textsuperscript{3}, Zhehao Shen\textsuperscript{1}, Haimin Luo\textsuperscript{1}, Lan Xu\textsuperscript{1}

\textsuperscript{1}ShanghaiTech University  \textsuperscript{2}NeuDim  \textsuperscript{3}Pico IDL, ByteDance

Abstract

Convenient 4D modeling of human-object interactions is essential for numerous applications. However, monocular tracking and rendering of complex interaction scenarios remain challenging. In this paper, we propose Instant-NVR, a neural approach for instant volumetric human-object tracking and rendering using a single RGBD camera. It bridges traditional non-rigid tracking with recent instant radiance field techniques via a multi-thread tracking-rendering mechanism. In the tracking front-end, we adopt a robust human-object capture scheme to provide sufficient motion priors. We further introduce a separated instant neural representation with a novel hybrid deformation module for the interacting scene. We also provide an on-the-fly reconstruction scheme of the dynamic/static radiance fields via efficient motion-prior searching. Moreover, we introduce an online key frame selection scheme and a rendering-aware refinement strategy to significantly improve the appearance details for online novel-view synthesis. Extensive experiments demonstrate the effectiveness and efficiency of our approach for the instant generation of human-object radiance fields on the fly, notably achieving real-time photo-realistic novel view synthesis under complex human-object interactions. Project page: https://nowheretrix.github.io/Instant-NVR/.

1. Introduction

The accurate tracking and photo-realistic rendering for human-object interactions are critical for numerous human-centric applications like telepresence, tele-education or immersive experience in VR/AR. However, a convenient solution from monocular input, especially for on-the-fly setting, remains extremely challenging in the vision community.

Early high-end solutions \cite{6, 9, 13, 18} require dense cameras for high-fidelity reconstruction. Recent approaches \cite{11, 12, 17, 46, 47, 59, 63} need less RGB or RGBD video inputs (from 3 to 8 views) by using volumetric tracking techniques \cite{19, 32}. Yet, the multi-view setting is still undesirable for consumer-level daily usage. Differently, the monocular method with a single handiest commercial RGBD camera is more practical and attractive. For monocular human-object modeling, most approaches \cite{2, 15, 53, 57, 65, 66} track the rigid and skeletal motions of object and human using a pre-scanned template or parametric model. Besides, the monocular volumetric methods \cite{32, 41, 43, 58, 64} obtain detailed geometry through depth fusion, while the recent advance \cite{44} further extends it into the human-object setting. However, they fail to generate realistic appearance results, restricted by the limited geometry resolution.

Recent neural rendering advances, represented by Neural Radiance Fields (NeRF) \cite{29}, have recently enabled photo-realistic rendering with dense-view supervision. Notably, some recent dynamic variants of NeRF \cite{21, 28, 50, 51, 60, 67} obtain the compelling novel-view synthesis of human activities even under monocular capturing. However, they rely on tedious and time-consuming per-scene training to fuse the temporal observations into the canonical space, thus unsuitable for on-the-fly usage like telepresence. Only recently, Instant-NGP \cite{30} enables fast radiance field generation in seconds, bringing the possibility for on-the-fly radiance field modeling. Yet, the original Instant-NGP can only handle static scenes. Few researchers explore the on-the-fly neural rendering strategies for human-object interactions, especially for monocular setting.

\textsuperscript{*}Equal Contribution.
In this paper, we present Instant-NVR – an instant neural volumetric rendering system for human-object interacting scenes using a single RGBD camera. As shown in Fig. 1, Instant-NVR enables instant photo-realistic novel view synthesis via on-the-fly generation of the radiance fields for both the rigid object and dynamic human. Our key idea is to bridge the traditional volumetric non-rigid tracking with instant radiance field techniques. Analogous to the tracking-mapping design in SLAM, we adopt a multi-thread and tracking-rendering mechanism. The tracking front-end provides online motion estimations of both the performer and object, while the rendering back-end reconstructs the radiance fields of the interaction scene to provide instant novel view synthesis with photo-realism.

For the tracking front-end, we first utilize off-the-shelf instant segmentation to distinguish the human and object from the input RGBD stream. Then, we adopt an efficient non-rigid tracking scheme for both the performer and rigid object, where we adopt both embedded deformation [45] and SMPL [27] to model human motions. For the rendering back-end, inspired by Instant-NGP [30] we adopt a separate instant neural representation. Specifically, both the dynamic performer and static object are represented as implicit radiance fields with multi-scale feature hashing in the canonical space and share volumetric rendering for novel view synthesis. For the dynamic human, we further introduce a hybrid deformation module to efficiently utilize the non-rigid motion priors. Then, we modify the training process of radiance fields into a key-frame based setting, so as to enable graduate and on-the-fly optimization of the radiance fields within the rendering thread. For the dynamic one, we further propose to accelerate our hybrid deform module with a hierarchical and GPU-friendly strategy for motion-prior searching. Yet we observe that naively selecting key-frames with fixed time intervals will cause non-evenly distribution of the captured regions of the dynamic scene. It results in unbalanced radiance field optimization and severe appearance artifacts during free-view rendering. To that end, we propose an online key-frame selection scheme with a rendering-aware refinement strategy. It jointly considers the visibility and motion distribution across the selected key-frames, achieving real-time and photo-realistic novel-view synthesis for human-object interactions.

To summarize, our main contributions include:

- We present the first instant neural rendering system under human-object interactions from an RGBD sensor.
- We introduce an on-the-fly reconstruction scheme for dynamic/static radiance fields using the motion priors through a tracking-rendering mechanism.
- We introduce an online key frame selection scheme and a rendering-aware refinement strategy to significantly improve the online novel-view synthesis.

2. Related Work

**Traditional Human Volumetric Capture.** Human volumetric capture and reconstruction have been widely investigated to achieve detailed geometry reconstruction and accurate tracking. A series of works are proposed to make volumetric fusion more robust with SIFT features [16], multi-view systems [11, 12], scene flow [54], human articulated skeleton prior [62, 64], extra IMU sensors [70], data-driven prior [43, 44], learned correspondences [5], neural deformation graph [4, 23] or implicit function [17, 63]. Starting from the pioneering work DynamicFusion [32] which benefits from the GPU solvers, the high-end solutions [11, 12] rely on the multi-view camera system and complex calibration. VolumeDeform [16] combines depth-based correspondences with sparse SIFT features to reduce drift. KillingFusion [41] and SobolevFusion [42] support topology changes via more constraints on the motion fields. Thanks to the human parametric model [27], DoubleFusion [64] proposes the two-layer representation to capture scene more robustly. UnstructuredFusion [59] extends it to an unstructured multi-view setup. RobustFusion [44] further handles the challenging human-object interaction scenarios. Besides, Function4d [63] and NeuralHOFusion [17] marry the non-rigid-tracking with implicit modeling. However, these methods are dedicated to getting detailed geometry without focusing on high-quality texture and most methods can not handle human-object interactions. Comparably, our approach bridges the traditional volumetric capture and neural rendering advances, achieving photo-realistic rendering results under human-object interactions.

**Static Neural Scene Representations.** Coordinates-based neural scene representations in static scenes produce impressive novel view synthesis results and show huge potential. Various data representations are adopted to obtain better performance and characteristics, such as point-clouds [1, 48, 56], voxels [26], textured meshes [25, 49], occupancy [33, 40] or SDF [34, 52]. Meanwhile, Since the vanilla NeRF which requires hours of training is time-consuming, some NeRF extensions [30, 39, 61] are proposed to accelerate both training and rendering. Plenotrees [61] utilizes the octree to skip the empty regions. Plenoxels [39] parameterizes the encoding using spherical harmonics on the explicit 3D volume. Instant-NGP [30] utilizes the multi-scale feature hashing and TCNN to speed up. Though its rendering speed seems possible to train on-the-fly, they do not have a specific design for streaming input and only can recover static scenes. Comparably, our Instant-NVR achieves on-the-fly efficiency based on the Instant-NGP [30].

**Dynamic Neural Scene Representations.** Novel view synthesis in dynamic scenes is an important research problem. D-NeRF [38] and Non-rigid NeRF [50] leverage the displacement field to represent the motion while Ne-
Figure 2. Our approach consists of two stages. The tracking front-end (Sec. 4.1) captures human and object motions, while the rendering back-end (Sec. 4.2) separately reconstructs the human-object radiance fields on-the-fly, for instant novel view synthesis with photo-realism.

Thus, we only transmit the motion priors with the RGBD images to the rendering thread and discard the explicit volumetric geometry prior.

Neural Rendering Back-end. We extend instant radiance fields [30] to the monocular and dynamic human-centric scenes, where we maintain the canonical instant radiance fields for both dynamic human and rigid objects separately. We introduce a lightweight pose-conditioned deformation module to learn the residual motion to refine the initial warping provided by motion priors. To enable on-the-fly radiance field generation and rendering, we adapt the training process into a key frame setting with the aid of efficient motion-prior caching. We introduce a key frame selection method to jointly consider the diversity of capturing view and human pose, visibility maps, and the input image quality. We further adaptively refine the appearance output in the rendering view with more analogous spatial-temporal capturing views. Note that our rendering thread reconstructs the radiance fields online to provide instant novel view synthesis with photo realism.

4. Method

4.1. Tracking Front-end

Human Non-rigid Tracking. We follow the traditional volumetric capture methods [44, 64] to track human non-rigid motions. Specifically, we parameterize human non-rigid motions as an embedded deformation graph $W = \{d_{q_i}, x_i\}$, where $x_i$ is the coordinates of the sampled ED node in canonical space and $d_{q_i}$ is the dual quaternions representing the corresponding rigid transformation in $SE(3)$ space. Each 3D point $v_c$ in the canonical space can be wrapped into the live space using an efficient and accurate motion interpolation method Dual-Quaternion Blend-
Figure 3. Our Neural Rendering back-end adopts a separated neural representation. Left are the input RGBD images with motions. Middle is a separate rendering engine that includes a hybrid deformation module and volumetric rendering. Right are rendering results.

\[ DQB(v_c) = \sum_{i \in \mathcal{N}(v_c)} w(x_i, v_c) dq_i, \]

\[ \tilde{v}_c = SE_3(DQB(v_c))v_c. \]

where \( \mathcal{N}(v_c) \) is a set of neighboring ED nodes of \( v_c \), \( w(x_i, v_c) \) is the influence weight of the \( i \)-th node \( x_i \) to \( v_c \) and formulated as \( w(x_i, v_c) = \exp\left(-\|v_c - x_i\|^2/r^2\right) \). \( r \) is the influence radius (0.1 in our setting). Note that the ED-only-based human tracking is fragile since the non-rigid ICP often fails at fast articulated human motions due to losing correspondence. Therefore, we also introduce the SMPL inner body with shape parameters \( \beta \) and pose parameters \( \theta \) as the skeleton prior and utilize \( \theta \) with the skinning weight to wrap 3D point \( v_c \), which further constrain the ED motion tracking within a reasonable motion scale. Please refer to [64] for details about the ED-sampling and double layer motion representation.

To calculate the final ED-based motion, we jointly optimize the skeleton pose \( \theta \) and ED non-rigid motion field \( W \) as follows:

\[ E(W, \theta) = \lambda_{data} E_{data} + \lambda_{bind} E_{bind} + \lambda_{reg} E_{reg} + \lambda_{prior} E_{prior} + \lambda_{pose} E_{pose} + \lambda_{inter} E_{inter}. \]

The data term \( E_{data} \) measures the point-to-plane distances between the deformed model and the current input depth map:

\[ E_{data} = \sum_{(v_c, u) \in \mathcal{P}} \psi(n_u^T(\tilde{v}_c - u)), \]

where \( u \) is a sampled point in the depth map, \( n_u \) is its normal, and \( \tilde{v}_c \) denotes its closest point on the fused surface. \( \mathcal{P}_i \) is the set of correspondences found via a projective local search [32]. Besides, the binding term \( E_{bind} \) constrains both skeleton and final ED motions to be consistent while the geometry regularity term \( E_{reg} \) produces locally as-rigid-as-possible (ARAP) motions to prevent overfitting to depth inputs. These two terms are detailed in [14, 64]. The pose prior term \( E_{prior} \) from [3] penalizes the unnatural poses. Both the pose term \( E_{pose} \) and interaction term \( E_{inter} \) are form [44] to encourage natural motion capture during human-object interactions. Note that the optimization non-linear least squares problem in Eqn. 2 is solved using LM method with the PCG solver on GPU [12, 14].

Object Rigid Tracking. For rigid tracking of objects, we follow [44] to optimize the rigid motions and transform them to camera pose \( T^i \) under the ICP framework, in which we fuse the depth map to a canonical TSDF volume to maintain the stable correspondence for robust object tracking.

4.2. Neural Rendering Back-end

To enable efficient photo-realistic neural rendering of the interaction scenes, our neural rendering back-end adopts a separated instant neural representation based on the on-the-fly key frame selection strategy.

Separated instant neural representation. We design the instant neural representation to reconstruct the human and object separately. For object branch, given the RGBD image \( I_t \) and \( D_t \) with the camera pose \( T^i \) as the training set,
we leverage the original Instant-NGP [30] to extract the 3D point \( v^h \) features on the hash table \( H^p \) and then feed them into the geometry MLP \( E^g \) and color MLP \( E^c \) to acquire the density and color.

For dynamic human, in contrast to recent approaches [35, 36, 38, 50] which can’t handle long sequences via pure MLP and human NeRFs [37, 55, 69] that heavily rely on SMPL [27] which do not align well with the surface and easily cause artifacts, we introduce a hybrid deformation module to efficiently leverage the motion priors. The explicit non-rigid warping and an implicit pose-conditioned deformation net jointly aggregate the corresponding point information in the canonical space.

Specifically, given this human non-rigid motion \( \{ dq^t \} \), SMPL pose \( \theta^t \) and a sampling point \( v^t \) at frame \( t \), we construct the warping function to map \( v^h \) back to the canonical space \( v^t \). We calculate the deformed ED nodes \( x^t_i = dq^t x_i \), and then the point \( v^h \) in the influence radius \( r \) of these nodes can be warped into canonical surface via neighboring ED nodes weight blending:

\[
v^h = SE3(DQB^{-1}(v^h))v^t. \tag{4}
\]

To reduce the warping error and improve the rendering quality, we further integrate pose-conditioned deformation net here to correct the misalignment, where we concatenate the encoded \( v^h \) via hash-encoding with the human pose \( \theta^t \) and predict the residual displacement \( \delta v^h \) through an MLP. Finally, we feed \( v^h = v^h + \delta v^h \) into the canonical hash-encoding \( H^h \), geometry as well as color MLPs \( E^g_h, E^c_h \).

**On-the-fly Radiance Fields.** To ensure accurate tracking, hundreds of ED-nodes are maintained to query live points neighbors which is time-consuming and lead to bottlenecks. The time consumption is \( O(n) \) even if we query a small number of neighbors for each sampled point, in which \( n \) is the number of ED nodes. To enhance on-the-fly efficiency, we introduce a look-up-table-based fast search strategy here to speed up. Specifically, we only initialize the canonical KNN(k-nearest-neighbors) field in the beginning, whose resolution is 512\(^3\), and each voxel saves \( s \) neighboring ED nodes index(4 in our setting). We then concatenate non-rigid motions in each frame to form a look-up table.

At frame \( t \), for a voxel with index \( k \) and coordinates \( v_k \) in the canonical, we warp it via Eqn. 1 to the live space and obtain its corresponding voxel index \( f \). We save the canonical index \( k \) in live voxel \( f \). Afterward, for each sampling point, we can acquire the live space index \( f \) and obtain the canonical index \( k \). \( k \) links the 4 neighbor ED nodes index. Offsetting the index to frame \( t \) on the look-up-table, we can acquire the corresponding motions and calculate the blending weight as well as warped point via \( DQB \) in \( O(1) \) manner. In addition, we are able to construct the live KNN field for each voxel in \( O(1) \) time by utilizing custom CUDA kernels.

**Online Key Frame Selection.** To achieve online performance and high quality rendering, we choose key frames to organize our neural rendering training dataset. Before choosing, we discard blurry RGB frames caused by fast motion based on the bluriness measure [10]. Besides, we observe that naively selecting key-frames with fixed time intervals brings the time-related details but causes the non-evenly distribution of the captured regions. Inspired by [22, 31], we introduce a key frame selection scheme here to keep the diversity of motion distribution and complement visibility. Specifically, we formulate the visibility map for each ED node \( x^t_i = (x', y', z') \) in frame \( t \) as follows:

\[
s^t_i = \begin{cases} 
1, & \text{if } |z' - D^t(\pi(x'_i))| < \epsilon \\
0, & \text{otherwise}
\end{cases}, \tag{5}
\]

where \( \pi(\cdot) \) denotes the projection matrix, \( D^t(\cdot) \) represents the depth value of the corresponding pixel at frame \( t \), \( \epsilon \) is the visibility degree (0.01 in our setting). we continue to define the similarity for two frames:

\[
E_h(t_1, t_2) = \beta_{\text{pose}} |\theta_{t_1} - \theta_{t_2}|^2 + \beta_{\text{vis}} \sum_i s^t_i + s^t_i + \beta_{\text{visibility}} |t_1 - t_2|^2, \tag{6}
\]

where \( \oplus \) is the xor operation and \( t_1, t_2 \) are frame indexes.

For an object with the pose \( T^t \) which includes rotation \( R^t \) and translation \( d^t \), we define the similarity as follows:

\[
E_o(t_1, t_2) = \beta_{R^t} ||d^t - d^t||_2^2 + \beta_{\text{translation}} |t_1 - t_2|^2, \tag{7}
\]

Furthermore, we define \( \gamma \) to determine the diversity of the spatial pool. At the start time, the pool is empty and imported the first frame. Once the similarity of each two among the latest frame received from tracking and frame(s) in the pool is greater than \( \gamma \), we push this frame to the pool. When the pool capacity reaches its peak (100 in our setting), we will continually update the pool using the new frame by removing the frame with the biggest similarity. In this manner, our spatial pool constantly updates in all frames.

To achieve photo-realistic novel-view synthesis, our Instant-NVR further refines the rendering view via training short iterations on the carefully selected frames in a spatial-temporal pool. The spatial-temporal pool includes \( m \) frames from the spatial pool, which have the most similarity with the rendering view and \( m \) latest frames received from the front-end. Our selection strategy ensures high-quality rendering without losing temporal detail.

**4.3. Implementation Details**

To train the dynamic NeRF under human-object interactions, we first apply the semantic segmentation MIVOS [8] to decouple the scene and obtain the human and object masks separately. To assemble human and object in a novel
Figure 5. The rendering results of Instant-NVR on various interaction sequences, including “driving a balance car”, “shaking a bag”, and “playing a water gun”.

In this section, we compare the state-of-the-art methods and evaluate Instant-NVR on various challenging human-object interaction scenarios. Besides, various rendering results of Instant-NVR are shown in Fig. 5, such as driving a balance car, shaking a bag and playing with a water gun. Please also kindly refer to our video.

5.1. Comparison

We compare Instant-NVR against the fusion-based methods RobustFusion [44], NeuralHOFusion [17] and NeRF-based methods NeuralBody [37], HumanNerf [68], both in efficiency and rendering quality. For comparison with fusion-based methods, as illustrated in Fig. 6 (b), RobustFusion [44] generates blurry appearance results, which are restricted by the limited geometry resolution. For a fair comparison, NeuralHOFusion [17] is modified to a single view setting and suffers from artifacts as shown in Fig. 6 (c). For NeRF-based methods, we employ the RGBD input to estimate the SMPL [27] as their prior and adopt RGBD loss terms. Both NeuralBody [37] and HumanNerf [68] give erroneous and blurry rendering results in the monocular setting (Fig. 6 (d-e)), which rely heavily on SMPL [27] and can not handle human-object interactions. In addition, training in these methods is time-consuming and novel.
view synthesis remains slow. In contrast, our Instant-NVR achieves more detailed and photo-realistic rendering results under human-object interactions, as shown in Fig. 6 (f). The quantitative results in Tab. 1 also demonstrate that our approach can achieve consistently better rendering quality and achieve efficient training as well as rendering speed to support on-the-fly performance. Note that both NeuralBody [37] and HumanNerf [68] take several hours to train, while training for our method is online.

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR↑</th>
<th>SSIM ↑</th>
<th>Rendering Time↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>RobustFusion [44]</td>
<td>20.59</td>
<td>0.935</td>
<td>0.123s</td>
</tr>
<tr>
<td>NeuralHOFusion [17]</td>
<td>21.09</td>
<td>0.942</td>
<td>0.151s</td>
</tr>
<tr>
<td>NeuralBody [37]</td>
<td>19.71</td>
<td>0.928</td>
<td>2.420s</td>
</tr>
<tr>
<td>HumanNerf [68]</td>
<td>18.68</td>
<td>0.892</td>
<td>5.103s</td>
</tr>
<tr>
<td>Ours</td>
<td>27.81</td>
<td>0.976</td>
<td>0.023s</td>
</tr>
</tbody>
</table>

5.2. Evaluation

**Online Human rendering.** As shown in Fig. 7 (b), per-vertex texture extracted from the fused albedo volume [14] is blurry. Naively selecting key-frames with fixed time intervals generates noising rendering results in Fig. 7 (c) due to the non-evenly distribution of the captured regions. In contrast, our online key frame selection strategy based on the diversity of motion distribution and complement visibility can achieve much clearer rendering results, as shown in Fig. 7 (d). To boost the rendering quality, the further refinement scheme can help us to achieve more photo-realistic rendering results, as shown in Fig. 7 (e). As for quantitative analysis, we evaluate the rendering quality in Tab. 2, which highlights the contributions of each component.

**Online Object Rendering.** As for the evaluation of online object rendering in Fig. 8, we can observe that per-vertex texture failed to generate high-quality appearance which is restricted by the limited geometry resolution. Moreover, naively selecting key-frames with fixed time interval brings the noises. Fig. 8 (d) shows that applying our key frame selection strategy without refinement is still unclear. In contrast, we can achieve the best rendering results with our full training pipeline. Moreover, the quantitative evaluation is as demonstrated in Tab. 3, in which our full pipeline with the online key frame selection and rendering refinement achieves the highest accuracy.

**Run-time Evaluation.** In Tab. 4, we list the run-time of each step in our pipeline, including both the tracking front-end and the neural rendering back-end. For tracking front-end, the rigid tracking of the object takes 40ms while the hu-
Figure 8. Quantitative evaluation of Online Object Rendering. (a) Input image; (b) Per-vertex texture; (c) Key frame selection using fixed interval; (c) Key frame selection w/o refinement; (d) Key frame selection w refinement.

Table 3. Quantitative evaluation of Online Object Rendering.

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR</th>
<th>SSIM</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Per-vertex texture</td>
<td>21.253</td>
<td>0.944</td>
<td>4.431</td>
</tr>
<tr>
<td>Key frame selection using fixed interval</td>
<td>25.248</td>
<td>0.954</td>
<td>1.043</td>
</tr>
<tr>
<td>Key frame selection w/o refinement</td>
<td>26.747</td>
<td>0.965</td>
<td>0.931</td>
</tr>
<tr>
<td>Key frame selection w refinement</td>
<td>28.826</td>
<td>0.977</td>
<td>0.615</td>
</tr>
</tbody>
</table>

Table 4. Quantitative evaluation of Run-time

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>rigid tracking</td>
<td>40ms</td>
</tr>
<tr>
<td>non-rigid-tracking</td>
<td>62ms</td>
</tr>
<tr>
<td>deformation net</td>
<td>5ms</td>
</tr>
<tr>
<td>training w/o fast search</td>
<td>205.53ms</td>
</tr>
<tr>
<td>training w fast search</td>
<td>17.95ms</td>
</tr>
<tr>
<td>rendering</td>
<td>23.38ms</td>
</tr>
</tbody>
</table>

Figure 9. Evaluation of the hybrid deformation module. (a), (d) are the reference views. (b), (e) are the results with only deform block. (c),(f) use our deform block with the aid of deform net.

May non-rigid tracking takes 62ms. Besides, the deformation net costs 5ms. For rendering back-end, training without fast search strategy takes 205.53ms while using our fast search scheme, the training time reduces to 17.95ms. Besides, we use 15.38ms for the rendering process.

Hybrid Deformation Module Evaluation. We conduct further evaluation of our hybrid deformation module to demonstrate its advantages. As shown in Fig. 9 (b)(e), employing only the explicit deform block results in misalignment between the ground truth and warped space, leading to blurry images and erroneous silhouettes. Conversely, by utilizing the deform block with the aid of implicit deform net to learn the residual displacement in Fig. 9 (c)(f), the rendering results outcome exhibit superior alignment and significantly enhance texture.

5.3. Limitation

As the first instant neural rendering system from an RGBD sensor that performs real-time and photo-realistic novel-view synthesis under human-object interactions, the proposed Instant-NVR still has some limitations. First, although we adopt the hybrid deformation module to efficiently utilize the non-rigid motion priors since our method is in monocular RGB-D camera setting, non-rigid fusion fails when facing the fast movement and leads to inaccurate priors which affect the on-the-fly rendering. Due to limited resolution and inherent noise of the depth input, our method cannot reconstruct the extremely fine details of the performer, such as the fingers. Data-driven techniques on different human parts will be critical for such problem. Besides, Instant-NVR is committed to rendering photo-realistic results on-the-fly. Therefore, we choose the density field as geometry representation, analogous to Instant-NGP [30]. It is promising to integrate other SDF representations [7, 52], which can generate a more delicate geometry. Furthermore, to ensure efficient transmission between tracking front-end and rendering back-end, we discard the volumetric explicit geometry priors produced by the tracking step. It is an interesting direction to explore more complementary between tracking and rendering.

6. Conclusion

We have presented a practical neural tracking and rendering approach for human-object interaction scenes using a single RGBD camera. By bridging traditional non-rigid tracking with recent instant radiance field techniques, our system achieves a photo-realistic free-viewing experience for human-object scenes on the fly. Our non-rigid tracking robustly provides sufficient motion priors for both the performer and the object. Our separated instant neural representation with hybrid deformation and efficient motion-prior searching enables the on-the-fly reconstruction of both the dynamic and static radiance fields. Our online key frame selection with a rendering-aware refinement strategy further provides a more vivid and detailed novel-view synthesis for our online setting. Our experimental results demonstrate the effectiveness of Instant-NVR for the instant generation of dynamic radiance fields and photo-realistic novel view synthesis of human-object interactions in real time. We believe that our approach is a critical step to virtual but realistic teleport human-object interactions, with many potential applications like consumer-level telepresence in VR/AR.

Acknowledgements. This work was supported by Shanghai YangFan Program (21YF1429500), Shanghai Local college capacity building program (22010502800). We also acknowledge support from Shanghai Frontiers Science Center of Human-centered Artificial Intelligence (ShangHAI).
References


[40] Ruizhi Shao, Hongwen Zhang, He Zhang, Mingjia Chen, Yanpei Cao, Tao Yu, and Yebin Liu. Doublefield: Bridging the neural surface and radiance fields for high-fidelity human reconstruction and rendering. In *CVPR*, 2022. 2


[53] Xi Wang, Gen Li, Yen-Ling Kuo, Muhammed Kocabas, Emre Aksan, and Otmar Hilliges. Reconstructing action-conditioned human-object interactions using commonsense knowledge priors. In International Conference on 3D Vision (3DV), 2022. 1


[58] Lan Xu, Wei Cheng, Kaiwen Guo, Lei Han, Yebin Liu, and Lu Fang. Flyfusion: Realtime dynamic scene reconstruction using a flying depth camera. IEEE transactions on visualization and computer graphics, 27(1):68–82, 2019. 1

[59] Lan Xu, Zhuo Han, Tao Yu, Yebin Liu, and Lu Fang. Unstructuredfusion: realtime 4d geometry and texture reconstruction using commercial rgbd cameras. IEEE transactions on pattern analysis and machine intelligence, 42(10):2508–2522, 2019. 1, 2, 3


