OmniLocalRF: Omnidirectional Local Radiance Fields from Dynamic Videos

Dongyoung Choi Hyeonjoong Jang Min H. Kim

KAIST

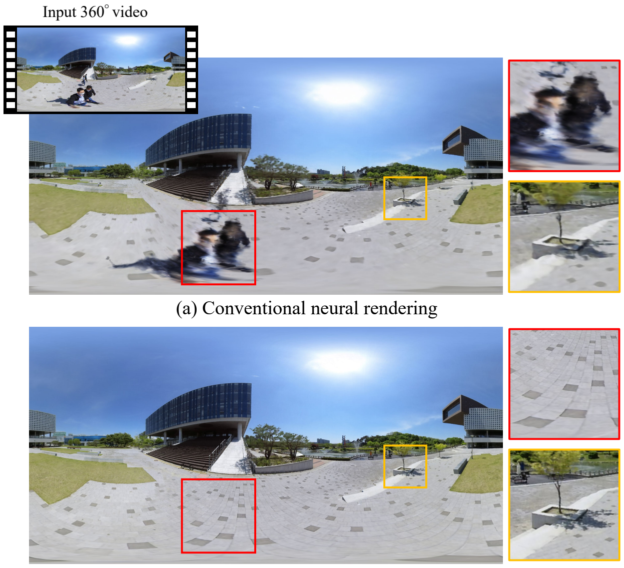
Abstract

Omnidirectional cameras are extensively used in various applications to provide a wide field of vision. However, they face a challenge in synthesizing novel views due to the inevitable presence of dynamic objects, including the photographer, in their wide field of view. In this paper, we introduce a new approach called Omnidirectional Local Radiance Fields (OmniLocalRF) that can render static-only scene views, removing and inpainting dynamic objects simultaneously. Our approach combines the principles of local radiance fields with the bidirectional optimization of omnidirectional rays. Our input is an omnidirectional video, and we evaluate the mutual observations of the entire angle between the previous and current frames. To reduce ghosting artifacts of dynamic objects and inpaint occlusions, we devise a multi-resolution motion mask prediction module. Unlike existing methods that primarily separate dynamic components through the temporal domain, our method uses multi-resolution neural feature planes for precise segmentation, which is more suitable for long 360° videos. Our experiments validate that OmniLocalRF outperforms existing methods in both qualitative and quantitative metrics, especially in scenarios with complex real-world scenes. In particular, our approach eliminates the need for manual interaction, such as drawing motion masks by hand and additional pose estimation, making it a highly effective and efficient solution.

1. Introduction

Omnidirectional cameras such as Ricoh Theta or Insta360 allow capturing panoramic 360° views in a single shot. Various applications with omnidirectional images such as spherical depth estimation [53, 58, 59], novel view synthesis [2, 3, 5–7, 11, 18, 31, 35] and geometry reconstruction [3, 18] aiming at large-scale static scenes have recently been explored. In particular, synthesizing 360° novel views can provide continuous views from unobserved camera angles while maintaining its details.

However, recent novel view synthesis methods struggle to apply to omnidirectional input for the following reasons.



(a) Conventional neural rendering

(b) OmniLocalRF (ours)

Figure 1. We introduce omnidirectional local radiance fields for photorealistic view synthesis of static scenery from 360° videos. Our method effectively removes dynamic objects (including the photographer) without manual interaction. Also, it achieves high-resolution details in the inpainted regions by means of bidirectional observations of omnidirectional local radiance fields. Refer to the supplemental video for more results.

When capturing omnidirectional videos to record static environments, dynamic objects are prone to be captured as an extension of the field of view, and capturing a photographer is inevitable unless employing a dedicated hardware or remote controller. When synthesizing novel views, these captured objects are represented as ghosting artifacts onto the rendered results [37]. Despite these problems, existing methods have achieved 360° view synthesis, relying on constrained capturing conditions, where it minimizes the advent of dynamic objects [3, 11] or requiring dedicated hardware [5–7, 31, 35], which are not suitable for casual 360° photography.

Low-rank decomposition through robust principal component analysis [15, 16, 57] and existing optimization-based methods [13, 14] effectively eliminate dynamic objects on the image domain. However, their applicability is limited to scenarios involving multiple images captured from the same viewpoints. Recent view synthesis works [32, 33, 46, 54] detach dynamic objects during static view synthesis by modeling dynamic objects along the temporal domain. However, they are inappropriate for long 360° videos because of the neural model's capacities.

In this paper, we propose omnidirectional local radiance fields (OmniLocalRF) that can render novel views of static scene environments from casual dynamic 360° videos. We formulate local radiance fields (LocalRF) [26] into a novel bidirectional training scheme designed explicitly for omnidirectional video input. We develop a module that uses multi-resolution neural feature planes to predict motion masks of every frame and segment frame-dependent components. Our LocalRF-based approach enables us to synthesize views and estimate camera poses from long videos. We automatically remove and inpaint dynamic objects while performing large-scale omnidirectional view synthesis. See Figure 1.

It is worth noting that different from conventional perspective cameras, omnidirectional cameras capture continuous scene information across multiple frames. This unique characteristic allows us to design a novel optimization approach that involves the bidirectional evaluation of samples taken from distant frames through an omnidirectional contraction scheme. This method enables us to effectively remove and inpaint broken dynamic objects through backward refinement, demonstrating omnidirectional reconstruction and decoupling of dynamic objects across complex real-world scenes. In summary, our contributions are:

- A view synthesis method based on bidirectional optimization through distant frames while conserving locality across radiance fields,
- A new motion mask prediction module that accurately segments dynamic objects, even in 360° videos without requiring a pretrained model, and
- A camera pose estimation technique based on local view synthesis of 360° videos.

Our code is freely available for research purposes[1].

2. Related work

Omnidirectional view synthesis. OmniPhotos [3] and Jang et al. [18] reconstruct geometry from 360° videos using mesh representation. Several approaches [5–7, 31, 35] use a spherical camera rig to render images at unobserved camera positions. MatryODShka [1] proposes a multi-sphere image representation for omnidirectional view synthesis. However, these methods do not work for 360° videos casually taken with dynamic objects.

Recent advancements for rendering 360° images, e.g., Mip-NeRF360 [2] and EgoNeRF [11], use volume rendering to create omnidirectional images. Mip-NeRF360 extends the capabilities of Neural Radiance Fields (NeRF) [27] to unbounded scenes, while EgoNeRF uses the Yin-Yang grid to obtain a balanced polar representation of neural features. However, these methods limit use in scenarios with dynamic objects in the training dataset, as they are designed for egocentric scenes captured using a selfie stick or by excluding dynamic objects with a motion mask.

Removing dynamic objects. Novel view synthesis methods generally aim to reconstruct static objects through the multi-view stereo [2, 27, 43, 47]. In existing works, separating dynamic and static objects by pretrained model [9, 17] on input images and modeling them respectively enables rendering static geometry if multi-view of dynamic objects are sufficiently provided in a dataset [21, 25, 32, 33]. In 360° videos that do not provide enough multi-view cues of dynamic objects, modeling them across the 3D space is challenging due to geometric inconsistency over time.

To reduce ghosting artifacts caused by dynamic objects in view synthesis, several models [36, 43] exclude dynamic objects using an external segmentation model [9, 10]. However, these models fail to mask out objects that are not labeled, such as shadows. Recent works, such as D^{2}NeRF [54] and Neuraldiff [46], segment dynamic objects by reconstructing them in 3D space across the temporal domain, but require an extensive parameterization, making them less scalable for long 360° videos. OmnimatteRF [24] estimates delicate motion masks from rough masks provided by Mask R-CNN [17] with optical flow [45], but still needs a pretrained segmentation model. RobustNeRF [37] discriminates between inliers and outliers based on photometric error and down-weights the outliers to decrease the effect of dynamic objects during training. Nevertheless, using a limited number of ray samples driven by down-weighting outliers slows down the training procedure, and these models are not scalable for long 360° videos.

Pose estimation. Conventional view synthesis relies on Structure from Motion (SfM) methods, like COLMAP [38] and OpenMVG [29], for camera poses, but these can be challenging to compute for large-scale data, leading to poor view synthesis quality. Several NeRF variants [4, 19, 23, 51] optimize radiance fields and poses jointly, but struggle to estimate camera poses for 360° videos with dynamic objects [26]. RoDynRF [25] uses Mask R-CNN to separate dynamic and static components, but it is not scalable and requires a pretrained segmentation model. LocalRF [26] succeeds in long trajectory pose calibration but is susceptible to artifacts caused by dynamic objects.

3. Omnidirectional Local Radiance Fields

Overview. Our goal is to generate photorealistic static scenes from unobserved viewpoints using long 360° videos,

[1] https://vclab.kaist.ac.kr/cvpr2024p1/

including dynamic objects as input. We optimize multiple radiance fields with continuous local frames while sliding frame windows and produce high-quality view synthesis. However, training blocks solely with local frames can result in ghosting artifacts from dynamic objects. To address this, we use an omnidirectional local radiance fields approach and a motion mask module to separate dynamic and static objects. We also propose a novel bidirectional optimization method to enhance the stability of static structural representations and eliminate residual artifacts. Our method produces superior rendering results than existing methods.

3.1. Preliminaries

We render the color $\hat{\mathbf{C}}(\mathbf{r})$ by volume rendering samples $\mathbf{x}_i$ whose density and color are $(\sigma_i, \mathbf{c}_i)$ along a ray $\mathbf{r}$ and train the radiance fields $\mathbf{RF}_\Theta$ to predict $(\sigma_i, \mathbf{c}_i)$ from the L1 photometric error between $\hat{\mathbf{C}}(\mathbf{r})$ and input image $\mathbf{C}(\mathbf{r})$ as it is more robust against outliers than MSE:

$$\mathcal{L}_{\text{rgb}} = \left\| \hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r}) \right\|_1 . \tag{1}$$

To extend NeRF to cover a large scale, we also use the contraction equation [2] over sample points:

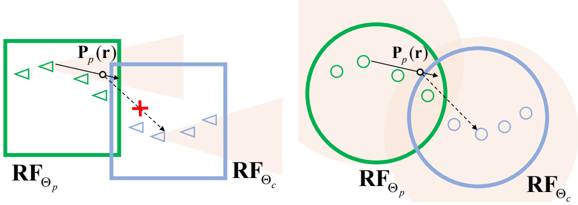
$$\text{contract}(\mathbf{x}) = \begin{cases} \mathbf{x} & \|\mathbf{x}\| \leq 1 \\ \left(2 - \frac{1}{\|\mathbf{x}\|}\right)\left(\frac{\mathbf{x}}{\|\mathbf{x}\|}\right) & \|\mathbf{x}\| > 1 \end{cases} , \tag{2}$$

Using the contraction function in radiance fields, which maps world coordinates onto a contracted space, aids in large-scale view synthesis by focusing nearby regions while representing distant components [26, 44]. However, they face challenges in long camera trajectories due to static model allocation.

Our approach uses multiple TensoRFs [8] to perform view synthesis and camera registration from videos, following LocalRF [26]. We allocate a NeRF block $\mathbf{RF}_{\Theta_m}$, where $m \in \{1, \cdots, M\}$, and insert input frames $\mathbf{C}_k$ with the corresponding camera poses $[R \mid t]_k$, $k \in \{1, \cdots, K\}$ into a temporal window $\mathcal{W}_m = \{(\mathbf{C}_{w(m,1)}, [R \mid t]_{w(m,1)}), \cdots, (\mathbf{C}_{w(m,N)}, [R \mid t]_{w(m,N)})\}$, where $w(m,n)$ denotes the frame number of m-th windows' n-th frame. If the distance between the camera $[R \mid t]_{w(m,N)}$ and the block center becomes too large, we stop inserting frames and optimize the block $\mathbf{RF}_{\Theta_m}$ with the camera poses $[R \mid t]_{w(m,n)}$ where $n \in \{1, \cdots, N\}$ using $\mathcal{W}_m$. After optimizing a $\mathbf{RF}_{\Theta_m}$ block, we create a new block $\mathbf{RF}_{\Theta_{m+1}}$ with its window $\mathcal{W}_{m+1}$ and insert frames until the end of the videos.

3.2. Bidirectional Optimization by Distant Frames

Previous reconstruction approaches [2, 11, 37, 48] predefine the reconstruction range before optimization and train a single NeRF-based module with registered poses. LocalRF [26] progressively allocates NeRF modules and performs large-scale reconstruction in a local manner without requiring camera priors. However, the vanilla LocalRF has limitations in leveraging global information and is susceptible to slowly moving objects that appear stationary from a local perspective. To overcome these issues, we propose a novel solution that leverages the omnidirectional nature of 360° view synthesis while taking advantage of locality. This improves overall reconstruction quality.



(a) Perspective camera (b) Omnidirectional camera

Figure 2. In the perspective video of marching forward, optimized radiance blocks $\mathbf{RF}_{\Theta_p}$ may not be visible in the frame used to train current radiance fields $\mathbf{RF}_{\Theta_c}$. However, in omnidirectional video, every uncontracted space of the optimized blocks can be seen, enabling effective bidirectional optimization. The boundary indicates the radiance fields' focusing region uncontracted.

As shown in Figure 2, in perspective videos of marching forward, distant frames contain only a small amount of information needed to refine poses and radiance fields. This is inefficient and can lead to instability. However, in 360° videos, distant frames contain a substantial amount of data to refine radiance fields. As a result, we propose a bidirectional optimization by distant frames as part of progressive optimization, which can globally refine radiance fields and poses for omnidirectional view synthesis.

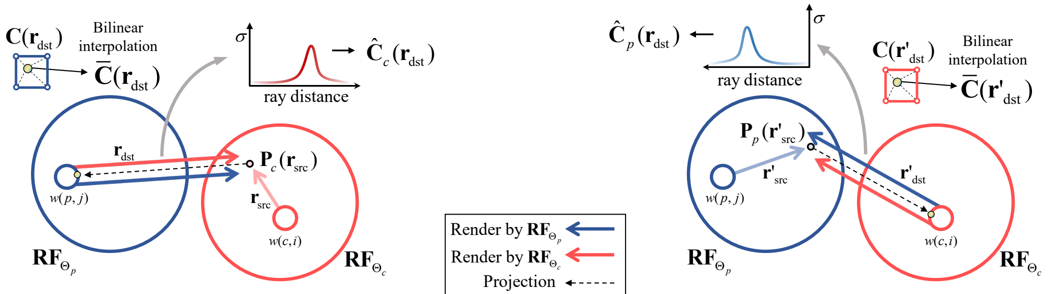
Forward step. As we progressively optimize multiple radiance fields $\mathbf{RF}_{\Theta_m}$, there are $\mathbf{RF}_{\Theta_c}$ that OmniLocalRF currently optimizes and a group of $\mathbf{RF}_{\Theta_p}, p \in \{1, \cdots, c-1\}$, which are already converged. In LocalRF, $\mathbf{RF}_{\Theta_c}$ is trained only with its temporal window $\mathcal{W}_c$ using the photometric loss defined as:

$$\mathcal{L}_{\text{rgb, s}}^{\text{for}} = \sum_{\mathbf{r}_{\text{src}} \in \mathcal{R}} \left\| \hat{\mathbf{C}}_c(\mathbf{r}_{\text{src}}) - \mathbf{C}(\mathbf{r}_{\text{src}}) \right\|_1 , \tag{3}$$

where $\mathbf{r}_{\text{src}}$ represents a ray from the current window $\mathcal{W}_c$, limiting the use of global frames. Here we additionally refine $\mathbf{RF}_{\Theta_c}$ using prior frames outside $\mathcal{W}_c$ and refer to it as a forward step since it proceeds in the same direction as window sliding.

As shown in Figure 3(a), to concentrate on the areas where $\mathbf{RF}_{\Theta_c}$ occupies, we render a depth value through $\mathbf{RF}_{\Theta_c}$ at the pixel $\mathbf{p}_{\text{src}}$, the origin of $\mathbf{r}_{\text{src}}$, on $w(c,i)$-th frame, $w(c,i) \in \mathcal{W}_c$, and obtain projected pixels $\mathbf{p}_{\text{dst}}$ to the $w(p,j)$-th frame which is a randomly selected member in $\mathcal{W}_p$ but not included in $\mathcal{W}_c$ by

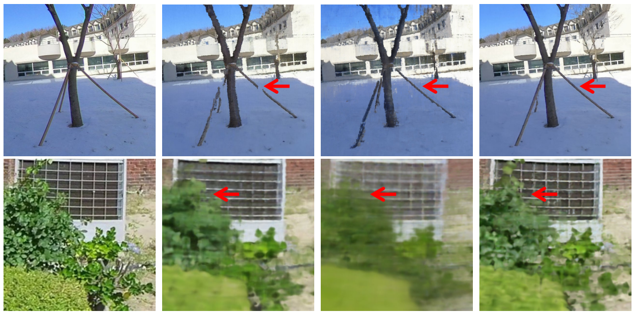
$$\mathbf{p}_{\text{dst}} = \Pi\left([R \mid t]_{w(c,i) \rightarrow w(p,j)} \Pi^{-1}\left(\mathbf{p}_{\text{src}}, \hat{D}_c(\mathbf{r}_{\text{src}})\right)\right). \tag{4}$$


$\mathbf{C}(\mathbf{r}_\text{dst})$
Bilinear interpolation
$\bar{\mathbf{C}}(\mathbf{r}_\text{dst})$
σ
ray distance
$\hat{\mathbf{C}}_c(\mathbf{r}_\text{dst})$
$\mathbf{r}_\text{dst}$
$\mathbf{P}_c(\mathbf{r}_\text{src})$
$\mathbf{r}_\text{src}$
$w(p,j)$
$w(c,i)$
$\mathbf{RF}_{\Theta_p}$
$\mathbf{RF}_{\Theta_c}$
Render by $\mathbf{RF}_{\Theta_p}$
Render by $\mathbf{RF}_{\Theta_c}$
Projection
$\hat{\mathbf{C}}_p(\mathbf{r}_\text{dst})$
σ
ray distance
$\mathbf{C}(\mathbf{r}'_\text{dst})$
Bilinear interpolation
$\bar{\mathbf{C}}(\mathbf{r}'_\text{dst})$
$\mathbf{P}_p(\mathbf{r}'_\text{src})$
$\mathbf{r}'_\text{src}$
$\mathbf{r}'_\text{dst}$
$w(p,j)$
$w(c,i)$
$\mathbf{RF}_{\Theta_p}$
$\mathbf{RF}_{\Theta_c}$

(a) Forward step

(b) Backward step

Figure 3. Our bidirectional optimization for omnidirectional videos. (a) In the forward step, we project the point $\mathbf{P}_c(\mathbf{r}_\text{src})$ rendered by $\mathbf{RF}_{\Theta_c}$ to the destination frame $w(p,j)$, used to train the previous radiance block $\mathbf{RF}_{\Theta_p}$. We then render the color and depth through $\mathbf{RF}_{\Theta_c}$ and $\mathbf{RF}_{\Theta_p}$, respectively, and use the L1 photometric error between the fully rendered color $\hat{\mathbf{C}}_c(\mathbf{r}_\text{dst})$ and the bilinearly interpolated input image $\bar{\mathbf{C}}(\mathbf{r}_\text{dst})$ (Eq. (6)) to update $\mathbf{RF}_{\Theta_c}$ and a mask module. (b) In the backward step, we switch the source and destination frames and refine $\mathbf{RF}_{\Theta_p}$ through the valid rays from static areas that meet $\mathcal{R}_\text{P}$.



(a) Test view (b) No backward step (c) Backward step w/o $\mathcal{L}^\text{back}_\text{rgb, s}$ (d) Complete optimization

Figure 4. Ablation study on the impact of the backward step. (b) Solely employing the forward step results in a blurred image. (c) Omitting the utilization of Eq. (7) leads to overfitting on distant frames. (d) Our bidirectional optimization shows great quality in representing details.

Here $\hat{D}_c$ is a rendered depth value using $\mathbf{RF}_{\Theta_c}$, and $[R \mid t]_{w(c,i)\to w(p,j)}$ is the relative camera matrix from $w(c,i)$-th to $w(p,j)$-th frame. $\Pi()$ denotes the equirectangular projection operator, and $\Pi^{-1}()$ backprojects a pixel on an equirectangular image to the world space.

We then render a depth value and a color value along $\mathbf{r}_\text{dst}$ which casts from $\mathbf{p}_\text{dst}$ through $\mathbf{RF}_{\Theta_p}$ and $\mathbf{RF}_{\Theta_c}$, respectively. In order to exclusively use the reliable samples, we choose valid rays which satisfy

$$\mathcal{R}_\text{C} = \{\mathbf{r}\,|(1-\mathcal{T})\hat{D}_p(\mathbf{r}) \le \hat{D}_c(\mathbf{r}) \le (1+\mathcal{T})\hat{D}_p(\mathbf{r}), \hat{D}_c(\mathbf{r}) \le 1\}, \tag{5}$$

where $\hat{D}_p$ is a rendered depth by $\mathbf{RF}_{\Theta_p}$, and $\mathcal{T}$ is the valid margin of photometric refinement, constant at 0.05. The regions satisfying Eq. (5) are co-visible from two RF blocks. In cases where static geometry exists, causing occlusion, the rendering depths between the two RF blocks differ significantly, preventing the execution of bidirectional optimization. We update $\mathbf{RF}_{\Theta_c}$ and a mask module from the conventional photometric loss in Eq. (3) with

$$\mathcal{L}^\text{for}_\text{rgb, d} = \sum_{\mathbf{r}_\text{dst}\in\mathcal{R}_\text{C}} \left\| \left(1-\hat{M}(\mathbf{r}_\text{dst})\right)\left(\hat{\mathbf{C}}_c(\mathbf{r}_\text{dst}) - \bar{\mathbf{C}}(\mathbf{r}_\text{dst})\right)\right\|_1, \tag{6}$$

where $\hat{\mathbf{C}}_c$ denotes a rendered color by $\mathbf{RF}_{\Theta_c}$ within $\mathcal{R}_\text{C}$. We bilinearly interpolate input color based on the projected pixel's coordinates to estimate the color $\bar{\mathbf{C}}$. $\hat{M}(\mathbf{r})$ is the estimated motion mask of $\mathbf{r}$ using the global mask module, detailed in Section 3.3. $\mathbf{RF}_{\Theta_c}$ additionally uses the static regions of frames beyond the temporal window to maintain locality and incorporate richer view information.

Backward step. We illustrate the backward process in Figure 3(b). We cast a ray from $\mathbf{r}'_\text{src}$ on $w(p,j)$-th frames, which is used as the destination frame in the forward step, and render a depth value with a color value through $\mathbf{RF}_{\Theta_p}$. We then supervise the rendered color by the color of the input frame as:

$$\mathcal{L}^\text{back}_\text{rgb, s} = \sum_{\mathbf{r'}_\text{src}\in\mathcal{R}} \left\| \left(1-\hat{M}(\mathbf{r'}_\text{src})\right)\left(\hat{\mathbf{C}}_p(\mathbf{r'}_\text{src}) - \mathbf{C}(\mathbf{r'}_\text{src})\right)\right\|_1, \tag{7}$$

where $\hat{\mathbf{C}}_p$ indicates a rendered color by $\mathbf{RF}_{\Theta_p}$. Unless we use Eq. (7), $\mathbf{RF}_{\Theta_p}$ overfits to render distant frames while distorting adjacent views as shown in Figure 4(c).

We get the projected pixel $\mathbf{p}'_\text{dst}$ on $w(c,i)$-th frame as follows:

$$\mathbf{p}'_\text{dst} = \Pi\Big([R\mid t]_{w(p,j)\to w(c,i)}\Pi^{-1}\left(\mathbf{p}'_\text{src}, \hat{D}_p(\mathbf{r}'_\text{src})\right)\Big), \tag{8}$$

where the source and destination frame are reversed in Eq. (4). We render a color with a depth through $\mathbf{RF}_{\Theta_p}$ and also render a depth value using $\mathbf{RF}_{\Theta_c}$ at $\mathbf{r}'_\text{dst}$. The photometric loss for backward refinement supervises the color rendered by $\mathbf{RF}_{\Theta_p}$:

$$\mathcal{L}^\text{back}_\text{rgb, d} = \sum_{\mathbf{r}'_\text{dst}\in\mathcal{R}_\text{P}} \left\|\left(1-\hat{M}(\mathbf{r}'_\text{dst})\right)\left(\hat{\mathbf{C}}_p(\mathbf{r}'_\text{dst}) - \bar{\mathbf{C}}(\mathbf{r}'_\text{dst})\right)\right\|_1. \tag{9}$$

$\mathcal{R}_\text{P}$ denotes the valid ray bundles where the terms $\hat{D}_c(\mathbf{r})$ and $\hat{D}_p(\mathbf{r})$ are reversed in Eq. (5), and $\bar{\mathbf{C}}(\mathbf{r}'_\text{dst})$ is a bilinear interpolation of $\mathbf{C}(\mathbf{r}'_\text{dst})$ based on $\mathbf{p}'_\text{dst}$, a pixel origin of $\mathbf{r}'_\text{dst}$.

During the forward and backward steps, we use rays from frames close to the target RF block (source frame) and from frames far away from the block (destination frame). In this manner, we utilize four photometric loss functions, Eqs. (3), (6), (7), and (9). Then, Eq. (3) is used to train

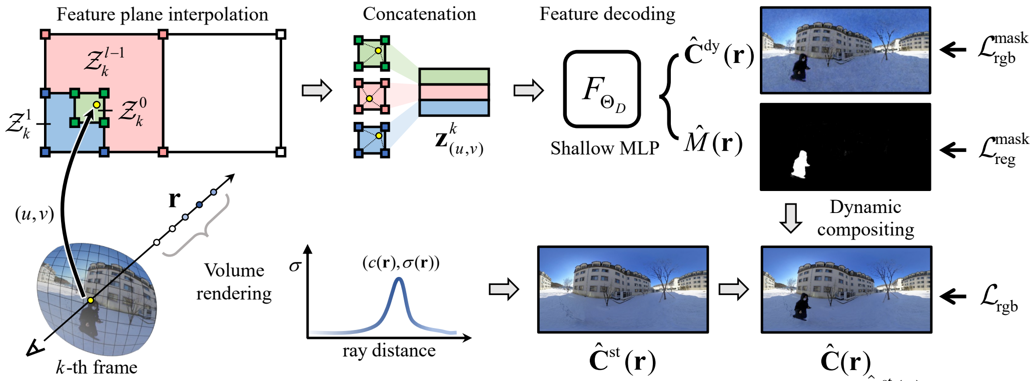

Feature plane interpolation
Concatenation
Feature decoding
$\mathcal{Z}_k^{l-1}$
$\mathcal{Z}_k^1$
$\mathcal{Z}_k^0$
$\mathbf{z}_{(u,v)}^k$
F_{Θ_D}
Shallow MLP
$\hat{\mathbf{C}}^{\text{dy}}(\mathbf{r})$
$\hat{M}(\mathbf{r})$
$\mathcal{L}_{\text{rgb}}^{\text{mask}}$
$\mathcal{L}_{\text{reg}}^{\text{mask}}$
Dynamic compositing
(u,v)
$\mathbf{r}$
Volume rendering
σ
$(c(\mathbf{r}),\sigma(\mathbf{r}))$
ray distance
k-th frame
$\hat{\mathbf{C}}^{\text{st}}(\mathbf{r})$
$\hat{\mathbf{C}}(\mathbf{r})$
$\mathcal{L}_{\text{rgb}}$

Figure 5. For motion mask prediction, we cast a ray $\mathbf{r}$ from the k-th frame and render the static structure $\hat{\mathbf{C}}^{\text{st}}(\mathbf{r})$ through volume rendering using radiance fields $\mathbf{RF}_\Theta$. We extract multi-resolution features of normalized (u, v) by traversing feature plane set $\mathcal{Z}_k$ and concatenate them into a single code $\mathbf{z}_{(u,v)}^k$. We estimate dynamic color $\hat{\mathbf{C}}^{\text{dy}}(\mathbf{r})$ and motion mask $\hat{M}(\mathbf{r})$, and render the final results $\hat{\mathbf{C}}(\mathbf{r})$ through dynamic compositing (Eq. (10)). We jointly update the mask module $(\mathcal{Z}_t, F_{\Theta_D})$ with radiance fields $\mathbf{RF}_\Theta$ using L1 photometric loss. We supervise $\hat{\mathbf{C}}^{\text{dy}}(\mathbf{r})$ by $\tilde{\mathbf{C}}(\mathbf{r})$ for unique factorization (Eq. (11)) and regularize the alpha of the mask (Eq. (12)).

$\mathbf{RF}_{\Theta_c}$ through both dynamic and static regions of the close frame. In this process, geometrically inconsistent areas are mapped to the mask module as dynamic objects. Eq. (6) refines $\mathbf{RF}_{\Theta_c}$ by far frame that have been used to train $\mathbf{RF}_{\Theta_p}$. As we already obtained the motion mask, we use only information from static regions to minimize the intervention of dynamic objects while refining. We also apply this approach in the backward step, exclusively updating static regions during additional refinement of $\mathbf{RF}_{\Theta_p}$ (Eqs. (7), (9)). Through this way, the overlapped regions in $\mathbf{RF}_{\Theta_c}$ and $\mathbf{RF}_{\Theta_p}$ are refined concurrently using distant frames.

3.3. Motion Mask Prediction

We use a set of neural feature planes for each frame to estimate a motion mask. To do so, we decode a feature code from a multi-resolution feature plane. Our mask module is compatible with a pixel-wise ray marching NeRF setup and can be optimized jointly while rendering static objects. Figure 5 provides an overview of our motion mask prediction.

Mask module architecture. We leverage a Bayesian learning framework for segmenting dynamic components inspired by DynIBaR [22] that combines IBRNet [49] with 2D CNN. We create a low-resolution equirectangular feature plane set $\mathcal{Z}_k$ for each frame to handle delicate frame-dependent components. The feature plane set comprises multi-resolution planes, denoted by $\mathcal{Z}_k = \{\mathcal{Z}_k^1, \mathcal{Z}_k^2, \cdots, \mathcal{Z}_k^L\}$, where the height of each plane is $h_k^l = h_k^0/2^{l-1}$, with $l \in \{1, \cdots, L\}$. This approach follows previous works [12, 30, 42, 47] that have shown better spatial context-aware inference under a multi-resolution manner. We use 4 feature channels, $h_k^0 = 128$, and $L = 4$ for all our experiments.

To render an omnidirectional image, a camera ray $\mathbf{r}$ is generated by multiplying the camera matrix with an equirectangular backprojected ray from pixel $\mathbf{p}(u, v)$ on the k-th frame. Before casting a ray from $\mathbf{p}$, we interpolate multi-resolution features at normalized (u, v) by traversing $\mathcal{Z}_k$ and concatenate L levels of features into a single code. We use a global, shallow, multi-layer perceptron (MLP) F_{Θ_D} to decode the feature code and obtain the color of dynamic objects $\hat{\mathbf{C}}^{\text{dy}}(\mathbf{r})$ with the alpha value of the motion mask $\hat{M}(\mathbf{r})$ at the ray $\mathbf{r}$.

Motion mask optimization. We compute the final color $\hat{\mathbf{C}}_m(\mathbf{r})$ by compositing dynamic results $\hat{\mathbf{C}}^{\text{dy}}(\mathbf{r})$ with static results $\hat{\mathbf{C}}_m^{\text{st}}(\mathbf{r})$, rendered by $\mathbf{RF}_{\Theta_m}$:

$$\hat{\mathbf{C}}_m(\mathbf{r}) = \hat{M}(\mathbf{r})\hat{\mathbf{C}}^{\text{dy}}(\mathbf{r}) + (1 - \hat{M}(\mathbf{r}))\hat{\mathbf{C}}_m^{\text{st}}(\mathbf{r}). \tag{10}$$

We update the mask module, consisting of the feature plane set and MLP, with radiance fields by propagating the L1 photometric loss of rendered colors (Section 3.2).

The mask blended dynamic color in Eq. (10) has a variety of combinations of a mask and a dynamic color. When transient components have an intermediate alpha value due to the ambiguity in factorization, radiance fields try to compensate residuals and leave floating artifacts. Therefore, we supervise a dynamic color by Gaussian noise added input color $\tilde{\mathbf{C}}(\mathbf{r})$:

$$\mathcal{L}_{\text{rgb}}^{\text{mask}} = \sum_{\mathbf{r} \in \mathcal{R}} \left\| \hat{\mathbf{C}}^{\text{dy}}(\mathbf{r}) - \tilde{\mathbf{C}}(\mathbf{r}) \right\|_1, \tag{11}$$

which ensures a unique factorization while preventing our model from relying on the mask module to express fine details. We regularize the motion mask by the total variation (TV) loss and the binary loss for forcing the mask converged into a binary value while smoothing it as:

$$\mathcal{L}_{\text{reg}}^{\text{mask}} = \mathcal{L}_{\text{TV}}^{\text{mask}} + \mathcal{L}_{\text{bin}}^{\text{mask}}. \tag{12}$$

Refer to the supplementary material for details on the mask regularizers.

3.4. Progressive Optimization

We optimize $\mathbf{RF}_\Theta$ blocks, camera poses, feature plane sets, and mask MLP by sliding a window over the input video.

We insert frames into the window and optimize poses using $\mathbf{RF}_{\Theta_c}$ (Eq. (3)). If the poses fall outside the contraction range of $\mathbf{RF}_{\Theta_c}$, currently targeted radiance field, we move to the refining step. In the refining step, we perform bidirectional optimization by simultaneously updating the $\mathbf{RF}_{\Theta_c}$ and the randomly selected $\mathbf{RF}_{\Theta_p}$ that has been previously optimized. We use LocalRF's optical flow loss and normalized monocular depth supervision for robust pose estimation. To render novel views, we search for the nearest frame from the given viewpoints. If a frame is used to train two adjacent radiance blocks, we blend the results based on their position among overlapped frames.

4. Experimental Results

Our method takes 12 hours for 125 frames on a machine equipped with a single NVIDIA A6000 GPU and an Intel Xeon Silver 4214R 2.40 GHz CPU with 256 GB RAM. Refer to the supplemental material for further implementation details. To evaluate OmniLocalRF in large-scale view synthesis from 360° videos, we compare our approach against Mip-NeRF360 [2] and EgoNeRF [11], known for strong performance in omnidirectional view synthesis. Additionally, we also compare our approach with D^{2}NeRF [54] and RobustNeRF [37], both of which aim to reconstruct static structures from dynamic videos. Since these methods require precomputed camera pose, we utilize the pose estimated by OpenVSLAM [40] as camera priors. We also compare ours with LocalRF [26] that self-calibrate poses during view synthesis. For a comprehensive comparison, we evaluate both LocalRF and our method under two conditions: one with camera priors provided for pose refinement and the other without camera priors, starting from scratch and estimating the poses during view synthesis.

4.1. Dataset and Metrics

We capture and provide a new dataset from six outdoor scenes captured with an Insta360 camera, consisting of 5760×2880 resolution 30 fps 360° videos each. These are casual videos designed to capture backgrounds that include a photographer and dynamic objects like pedestrians. Considering reasonable training time and memory size, we use half-resolution input images. However, in the case of D^{2}NeRF, we have to use a quarter of the spatial resolution for rendering due to GPU memory constraints of 48 GB. We use a total of 125 images, taking every fourth frame from the first 500 frames. Test frames are selected for every tenth frame among 125 images, including the first image, resulting in 112 training images with 13 test images. For all test images in every scene, we manually create motion masks for validation. We then compare the rendered results with the dynamic objects masked ground truth and report PSNR, SSIM [50], and LPIPS [55]. We also measure weighted-to-spherically uniform PSNR and SSIM [41] (PSNR$^{\text{WS}}$ and SSIM$^{\text{WS}}$), taking into account the distortion near the pole in equirectangular images.

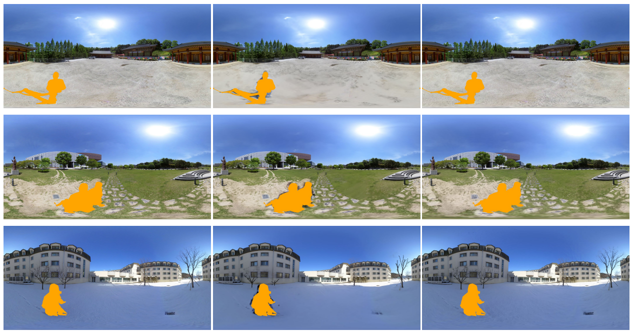


Figure 6. Masked results for evaluation in the real dataset. We manually create motion masks for the test views and compute the metrics after inpainting masked regions with gray. While our method masks the dynamic areas robustly, conventional neural rendering methods still exhibit residual artifacts due to temporal and spatial inconsistency.

We generate three synthetic 360° videos, each containing dynamic objects floating within the scene for evaluating pose estimation accuracy by absolute trajectory error (ATE) and relative pose error (RPE) between ground truth poses as standard visual odometry metrics [20, 39, 56]. The synthetic videos have a resolution of 2880×1440, consisting of 125 images each. Out of these, 13 images are allocated for testing, following the procedure used with the real dataset. Additionally, we provide view synthesis results of existing methods and our method for comprehensive comparisons.

4.2. Quantitative Evaluation

Dynamic objects in test images are excluded using manually created masks when computing metrics. As both LocalRF and our method can estimate camera poses during view synthesis, we present the metrics for models trained with and without preprocessed camera poses by OpenVSLAM [40]. According to Tables 1 and 2, scalable models like Mip-NeRF360, EgoNeRF, and LocalRF showed relatively high performance. D^{2}NeRF and RobustNeRF, which target reconstructing static structures from dynamic videos, achieve lower scores due to a limited model capacity. To ensure fair comparisons, we exclude the transient data of test views. However, dynamic objects frequently appear outside the masks as artifacts, leading to a decrease in the metrics, as illustrated in Figure 6. As a result, our method outperforms existing methods thanks to its effective dynamic object removal and locality even without given camera poses.

4.3. Qualitative Evaluation

In Figures 7 and 8, we conduct a qualitative comparison of our OmniLocalRF with the baseline methods. Our method effectively mitigates ghosting artifacts while preserving locality, allowing us to represent details in large-scale real-

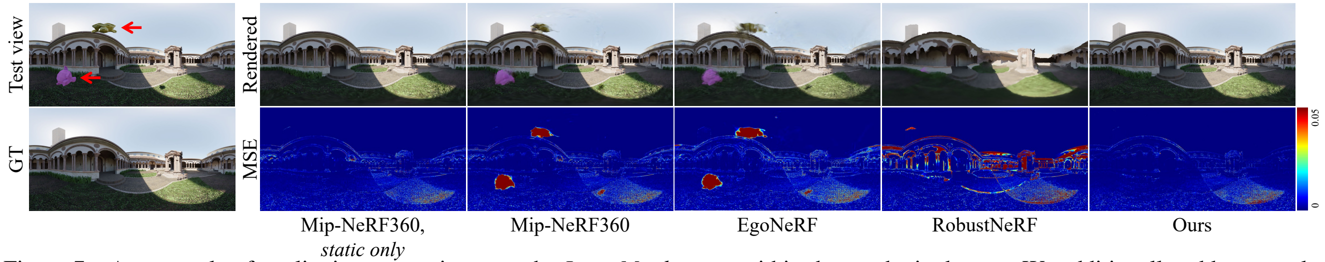


Figure 7. An example of qualitative comparisons on the *Lone Monk* scene within the synthetic dataset. We additionally address results of Mip-NeRF360 from static-only videos (*static only*) that do not include dynamic objects at the first column. Our results demonstrate the effective removal of dynamic objects while preserving details in the expansive scene.

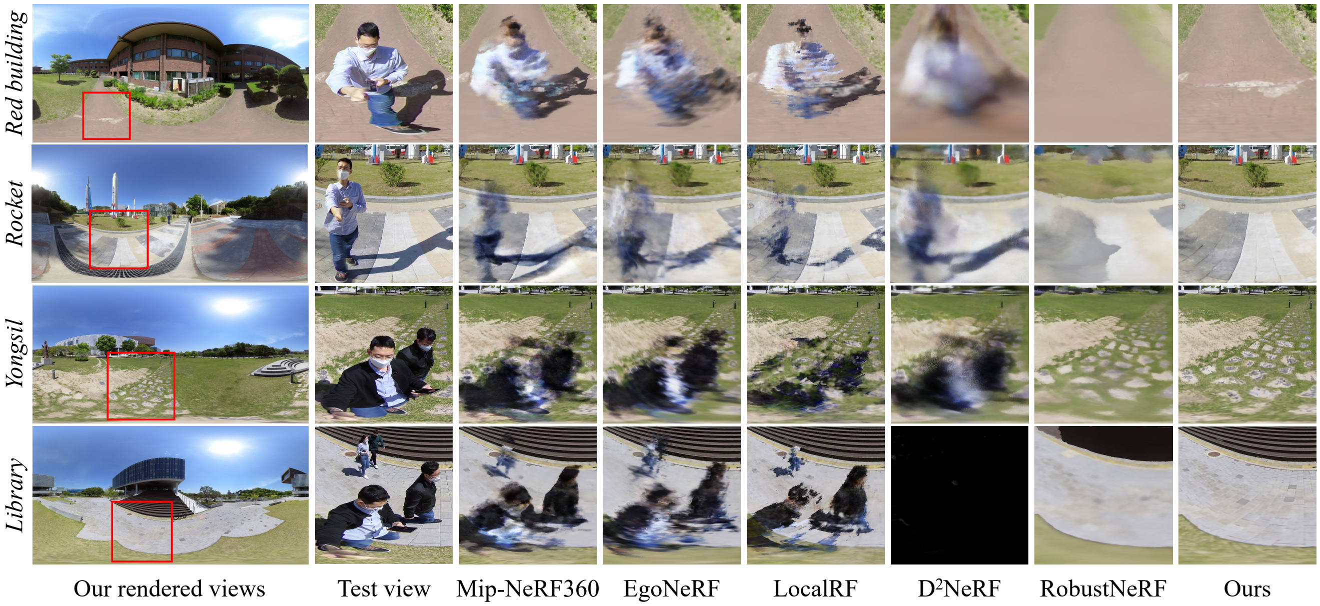


Figure 8. Qualitative comparisons on the real dataset. Our method can render locally unobserved areas with fine details throughout the bidirectional optimization with a self-supervised mask module. D^{2}NeRF sometimes encounters challenges in distinguishing between static backgrounds and dynamic objects, which results in the omission of certain regions in the *Library* scene. Refer to the supplemental video.

Table 1. Quantitative comparisons of the synthetic dataset. The averages of the metrics are measured across three synthetic scenes. Refer to Figure 7 for qualitative comparisons.

	PSNR ↑	PSNR$^{\text{WS}}$ ↑	SSIM ↑	SSIM$^{\text{WS}}$ ↑	LPIPS ↓
Mip-NeRF360 [2], *static only*	29.48	29.85	0.8625	0.8628	0.2699
Ours wo/ pose, *static only*	30.29	30.24	0.8679	0.8681	0.2561
Mip-NeRF360 [2]	25.59	25.32	0.8455	0.8464	0.2973
EgoNeRF [11]	23.99	23.67	0.8044	0.7951	0.3949
LocalRF [26] w/ pose	25.50	25.31	0.8454	0.8427	0.2897
D^{2}NeRF [54]	19.91	19.43	0.6212	0.5929	0.6298
RobustNeRF [37]	20.59	19.79	0.7326	0.7096	0.4734
Ours w/ pose	29.76	29.68	0.8633	0.8628	0.2624
LocalRF [26] wo/ pose	25.22	25.02	0.8389	0.8354	0.2949
Ours wo/ pose	29.93	29.85	0.8648	0.8648	0.2610

Table 2. Quantitative comparisons of the real dataset. We report the averages of the metrics measured across six real scenes. Refer to Figure 8 for qualitative comparisons.

	PSNR ↑	PSNR$^{\text{WS}}$ ↑	SSIM ↑	SSIM$^{\text{WS}}$ ↑	LPIPS ↓
Mip-NeRF360 [2]	26.88	26.44	0.8094	0.7977	0.3585
EgoNeRF [11]	25.95	25.38	0.7609	0.7424	0.4383
LocalRF [26] w/ pose	26.56	26.22	0.8041	0.7966	0.3471
D^{2}NeRF [54]	20.95	20.34	0.6105	0.5829	0.5100
RobustNeRF [37]	20.78	19.56	0.7093	0.6679	0.4864
Ours w/ pose	27.72	27.09	0.8171	0.8085	0.3299
LocalRF [26] wo/ pose	26.56	26.23	0.8034	0.7984	0.3410
Ours wo/ pose	27.73	27.13	0.8165	0.8088	0.3297

world data, even in locally occluded regions. Methods like EgoNeRF, Mip-NeRF360, and LocalRF, which lack dynamic handling, face challenges when reconstructing static scenes, relying solely on geometric consistency across input data. Even though their masked metrics indicate high performance, these methods suffer from ghosting artifacts, making them less suitable for practical applications that demand high-quality static view synthesis.

Since D^{2}NeRF parameterizes radiance along spatial and temporal domains with a single NeRF module, it often fails to separate dynamic components. It renders blurry results because of the large spatial and time complexity of input videos. RobustNeRF employs the iteratively reweighted least squares (IRLS) approach for patch-wise outlier down-weighting, which decouples transient artifacts from geometrically consistent structures through adaptive loss reweighting. However, in the omnidirectional videos, the overall geometry reconstruction quality is compromised due to the large scene scale, making it challenging to distinguish static from dynamic objects based on the photometric error over patches. For a fair comparison, we additionally train RobustNeRF for twice the number of iterations com-

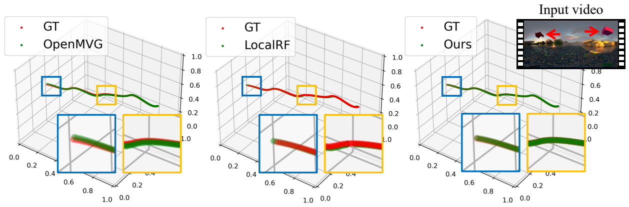


Figure 9. An example of pose comparisons on the *Pavillion* scene including dynamic objects. Our method shows robustness in estimating camera pose in the presence of dynamic objects.

Table 3. Comparisons of pose accuracy on the synthetic dataset. We average the results from three scenes.

	RPE$_r$ ↓	RPE$_t$ ↓	ATE ↓
OpenMVG [29]	0.10761	0.01799	0.00218
LocalRF [26]	0.10404	0.00096	0.00376
Ours	0.10398	0.00081	0.00165
OpenMVG [29], *static only*	0.10706	0.01796	0.00187
LocalRF [26], *static only*	0.10398	0.00071	0.00208
Ours, *static only*	0.10399	0.00074	0.00165

pared to Mip-NeRF360, taking into account the slowdown caused by the IRLS approach, but still fail to represent fine details effectively.

4.4. Pose Estimation Comparison

We compare our method with OpenMVG [29], an SfM-based pose estimator, and LocalRF in Figure 9. We report the relative rotation error (RPE$_r$), relative translation error (RPE$_t$), and ATE. We also estimate the camera trajectories from the ground truth videos to eliminate the influence of dynamic objects. We align the estimated pose with the ground truth, addressing scale factor and rotation discrepancies, before computing ATE. As shown in Table 3, the pose optimized through our approach is robust in the presence of dynamic objects, showing similar results in the static-only videos.

4.5. Ablation Study

We conduct an ablation study for our proposed mask module and bidirectional optimization: (a) Removing Eq. (6) degrades overall quality as the local blocks use fewer input images for training. (b) The omission of source rays' photometric supervision during the backward step (Eq. (7)) significantly degrades the model as it tends to converge on reconstructing distant regions as described in Figure 4(c). (c) We observe that incorporating distant frames through the backward step enables the model to capture and refine detailed information. (d) Geometrical inconsistencies in the temporal domain of dynamic objects often lead to the introduction of larger artifacts compared to the regions masked out during test views, as shown in Figure 6. Consequently, the absence of the mask module adversely affects both the qualitative and quantitative aspects of the results. (e) The exclusion of Eq. (3) during mask optimization results in intermediate alpha values within the motion mask due to the ambiguity in factorization and leaves floating artifacts, which yields lower metrics than the entire model. Table 4 quantitatively compares results.

Table 4. Ablation study results of our model. Reported metrics are averaged over the six scenes on our real dataset (Section 4.5).

	PSNR ↑	SSIM ↑	LPIPS ↓
(a) No forward step	27.64	0.8127	0.3356
(b) Backward step w/o $\mathcal{L}_{\text{rgb, s}}^{\text{back}}$	26.23	0.7800	0.3866
(c) No backward step	27.63	0.8086	0.3447
(d) No mask module	26.08	0.7876	0.3630
(e) No mask photometric supervision	25.27	0.7894	0.3593
Complete model	27.73	0.8165	0.3297

5. Discussion and Conclusions

We have presented OmniLocalRF, a novel method for omnidirectional view synthesis in dynamic 360° videos. Our approach integrates LocalRF with a mask module and bidirectionally refines distant NeRF blocks to remove dynamic artifacts and fill in occluded regions, resulting in accurate static structure reconstruction while preserving fine details within large scenes. Our method also accurately estimates camera trajectories during view synthesis, making it suitable for various applications such as street viewers and augmented reality environments.

Our model can synthesize static structures from 360° videos without motion masks and camera priors. However, it faces the usual challenges associated with neural rendering-based view synthesis. For instance, it is unable to inpaint regions that are completely occluded in the videos because NeRF-based models are trained using photometric loss between input images. To overcome this limitation, incorporating perceptual loss [28] or generative models [34, 52], such as stable diffusion, can be helpful.

We use linear interpolation in equirectangular space, which has grids with the same size of zenith and azimuth angles, to predict motion masks. Operating in this space can mitigate data redundancy compared to utilizing an undistorted cube map. However, this space can lead to inefficient oversampling near polar regions in mask predictions. To address this issue, we could use uniformly sampled spherical grids in future works.

Even though our model globally refines the local blocks based on photometric error, we do not deal with the global bundle adjustment and loop closure for pose estimation, which are used in completed SLAM systems. Adding these components to our approach would enable more robust and accurate pose estimation. While our model is capable of generating static structures from 360° videos, it faces several challenges that require further refinement.

Acknowledgements

Min H. Kim acknowledges the MSIT/IITP of Korea (RS-2022-00155620, 2022-0-00058, and 2017-0-00072), LIG, and Samsung Electronics.

References

[1] Benjamin Attal, Selena Ling, Aaron Gokaslan, Christian Richardt, and James Tompkin. MatryODShka: Real-Time 6DoF Video View Synthesis using Multi-Sphere Images. In *ECCV*, 2020. 2

[2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. In *CVPR*, pages 5470–5479, 2022. 1, 2, 3, 6, 7

[3] Tobias Bertel, Mingze Yuan, Reuben Lindroos, and Christian Richardt. OmniPhotos: Casual 360° VR Photography. *TOG*, 39(6):266:1–12, 2020. 1, 2

[4] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. NoPe-NeRF: Optimising Neural Radiance Field with No Pose Prior. In *CVPR*, pages 4160–4169, 2023. 2

[5] Michael Broxton, Jay Busch, Jason Dourgarian, Matthew DuVall, Daniel Erickson, Dan Evangelakos, John Flynn, Ryan Overbeck, Matt Whalen, and Paul Debevec. A Low Cost Multi-Camera Array for Panoramic Light Field Video Capture. In *SIGGRAPH Asia Posters*, New York, NY, USA, 2019. Association for Computing Machinery. 1, 2

[6] Michael Broxton, Jay Busch, Jason Dourgarian, Matthew DuVall, Daniel Erickson, Dan Evangelakos, John Flynn, Peter Hedman, Ryan Overbeck, Matt Whalen, and Paul Debevec. DeepView Immersive Light Field Video. In *ACM SIGGRAPH Immersive Pavilion*. Association for Computing Machinery, 2020.

[7] Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew Duvall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. Immersive Light Field Video with a Layered Mesh Representation. *TOG*, 39 (4), 2020. 1, 2

[8] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. TensoRF: Tensorial Radiance Fields. In *ECCV*, pages 333–350. Springer, 2022. 3

[9] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *ECCV*, pages 801–818, 2018. 2

[10] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-Deeplab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation. In *CVPR*, pages 12475–12485, 2020. 2

[11] Changwoon Choi, Sang Min Kim, and Young Min Kim. Balanced Spherical Grid for Egocentric View Synthesis. In *CVPR*, pages 16590–16599, 2023. 1, 2, 3, 6, 7

[12] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit Radiance Fields in Space, Time, and Appearance. In *CVPR*, pages 12479–12488, 2023. 5

[13] Miguel Granados, Hans-Peter Seidel, and Hendrik PA Lensch. Background Estimation from Non-Time Sequence Images. In *Proceedings of Graphics Interface 2008*, pages 33–40, 2008. 1

[14] Miguel Granados, James Tompkin, Kwang In Kim, Oliver Grau, Jan Kautz, and Christian Theobalt. How Not to Be Seen—Object Removal from Videos of Crowded Scenes. In *CGF*, pages 219–228. Wiley Online Library, 2012. 1

[15] Charles Guyon, Thierry Bouwmans, and El-Hadi Zahzah. Foreground Detection via Robust Low Rank Matrix Decomposition Including Spatio-Temporal Constraint. In *ACCV*, pages 315–320. Springer, 2012. 1

[16] Charles Guyon, Thierry Bouwmans, and El-Hadi Zahzah. Moving Object Detection via Robust Low Rank Matrix Decomposition with IRLS Scheme. In *Int. Symp. Visual Computing*, pages 665–674. Springer, 2012. 1

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, pages 2961–2969, 2017. 2

[18] Hyeonjoong Jang, Andréas Meuleman, Dahyun Kang, Donggun Kim, Christian Richardt, and Min H. Kim. Egocentric Scene Reconstruction from an Omnidirectional Video. *TOG*, 41(4), 2022. 1, 2

[19] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-Calibrating Neural Radiance Fields. In *ICCV*, pages 5846–5854, 2021. 2

[20] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust Consistent Video Depth Estimation. In *CVPR*, pages 1611–1621, 2021. 6

[21] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. In *CVPR*, 2021. 2

[22] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. DynIBaR: Neural Dynamic Image-Based Rendering. In *CVPR*, 2023. 5

[23] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. BARF: Bundle-Adjusting Neural Radiance Fields. In *CVPR*, pages 5741–5751, 2021. 2

[24] Geng Lin, Chen Gao, Jia-Bin Huang, Changil Kim, Yipeng Wang, Matthias Zwicker, and Ayush Saraf. OmnimatteRF: Robust Omnimatte with 3D Background Modeling. In *ICCV*, 2023. 2

[25] Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. Robust Dynamic Radiance Fields. In *CVPR*, 2023. 2

[26] Andreas Meuleman, Yu-Lun Liu, Chen Gao, Jia-Bin Huang, Changil Kim, Min H Kim, and Johannes Kopf. Progressively Optimized Local Radiance Fields for Robust View Synthesis. In *CVPR*, pages 16539–16548, 2023. 2, 3, 6, 7, 8

[27] 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*, pages 405–421. Springer, 2020. 2

[28] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Konstantinos G Derpanis, Jonathan Kelly, Marcus A Brubaker, Igor Gilitschenski, and Alex Levinshtein. SPIn-NeRF: Multiview Segmentation and Perceptual Inpainting with Neural Radiance Fields. In *CVPR*, pages 20669–20679, 2023. 8

[29] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. OpenMVG: Open Multiple View Geometry. In

Int. Workshop on Reproducible Research in Pattern Recognition, pages 60–74. Springer, 2017. 2, 8

[30] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *TOG*, 41(4):1–15, 2022. 5

[31] Ryan S. Overbeck, Daniel Erickson, Daniel Evangelakos, Matt Pharr, and Paul Debevec. A System for Acquiring, Processing, and Rendering Panoramic Light Field Stills for Virtual Reality. *TOG*, 37(6), 2018. 1, 2

[32] 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*, pages 5865–5874, 2021. 2

[33] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. HyperNeRF: a Higher-Dimensional Representation for Topologically Varying Neural Radiance Fields. *TOG*, 40(6):1–12, 2021. 2

[34] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D Diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 8

[35] Albert Parra Pozo, Michael Toksvig, Terry Filiba Schrager, Joyce Hsu, Uday Mathur, Alexander Sorkine-Hornung, Rick Szeliski, and Brian Cabral. An Integrated 6DoF Video Camera and System Design. *TOG*, 38(6), 2019. 1, 2

[36] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban Radiance Fields. In *CVPR*, pages 12932–12942, 2022. 2

[37] Sara Sabour, Suhani Vora, Daniel Duckworth, Ivan Krasin, David J Fleet, and Andrea Tagliasacchi. RobustNeRF: Ignoring Distractors with Robust Losses. In *CVPR*, pages 20626–20636, 2023. 1, 2, 3, 6, 7

[38] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *CVPR*, 2016. 2

[39] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A Benchmark for the Evaluation of RGB-D SLAM Systems. In *IROS*, pages 573–580. IEEE, 2012. 6

[40] Shinya Sumikura, Mikiya Shibuya, and Ken Sakurada. OpenVSLAM: A Versatile Visual SLAM Framework. In *ACMMM*, pages 2292–2295, 2019. 6

[41] Yule Sun, Ang Lu, and Lu Yu. Weighted-to-Spherically-Uniform Quality Evaluation for Omnidirectional Video. *SPL*, 24(9):1408–1412, 2017. 6

[42] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural Geometric Level of Detail: Real-Time Rendering with Implicit 3D Shapes. In *CVPR*, pages 11358–11367, 2021. 5

[43] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-NeRF: Scalable Large Scene Neural View Synthesis. In *CVPR*, pages 8248–8258, 2022. 2

[44] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A Modular Framework for Neural Radiance Field Development. In *ACM SIGGRAPH Conference Proceedings*, pages 1–12, 2023. 3

[45] Zachary Teed and Jia Deng. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In *ECCV*, 2020. 2

[46] Vadim Tschernezki, Diane Larlus, and Andrea Vedaldi. NeuralDiff: Segmenting 3D Objects that Move in Egocentric Videos. In *3DV*, 2021. 2

[47] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-NeRF: Scalable Construction of Large-Scale NeRFs for Virtual Fly-Throughs. In *CVPR*, pages 12922–12931, 2022. 2, 5

[48] Peng Wang, Yuan Liu, Zhaoxi Chen, Lingjie Liu, Ziwei Liu, Taku Komura, Christian Theobalt, and Wenping Wang. F2-NeRF: Fast Neural Radiance Field Training with Free Camera Trajectories. In *CVPR*, pages 4150–4159, 2023. 3

[49] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. IBRNet: Learning Multi-View Image-Based Rendering. In *CVPR*, pages 4690–4699, 2021. 5

[50] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *TIP*, 13(4):600–612, 2004. 6

[51] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. NeRF–: Neural Radiance Fields without Known Camera Parameters. *arXiv preprint arXiv:2102.07064*, 2021. 2

[52] Silvan Weder, Guillermo Garcia-Hernando, Aron Monszpart, Marc Pollefeys, Gabriel J Brostow, Michael Firman, and Sara Vicente. Removing Objects from Neural Radiance Fields. In *CVPR*, pages 16528–16538, 2023. 8

[53] Changhee Won, Jongbin Ryu, and Jongwoo Lim. SweepNet: Wide-baseline Omnidirectional Depth Estimation. In *ICRA*, pages 6073–6079. IEEE, 2019. 1

[54] Tianhao Wu, Fangcheng Zhong, Andrea Tagliasacchi, Forrester Cole, and Cengiz Oztireli. D2NeRF: Self-Supervised Decoupling of Dynamic and Static Objects from a Monocular Video. *NIPS*, 35:32653–32666, 2022. 2, 6, 7

[55] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*, pages 586–595, 2018. 6

[56] Zichao Zhang and Davide Scaramuzza. A Tutorial on Quantitative Trajectory Evaluation for Visual (-Inertial) Odometry. In *IROS*, pages 7244–725. IEEE, 2018. 6

[57] Xiaowei Zhou, Can Yang, and Weichuan Yu. Moving Object Detection by Detecting Contiguous Outliers in the Low-Rank Representation. *TPAMI*, 35(3):597–610, 2012. 1

[58] Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, and Petros Daras. Omnidepth: Dense Depth Estimation for Indoors Spherical Panoramas. In *ECCV*, pages 448–465, 2018. 1

[59] Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, Federico Alvarez, and Petros Daras. Spherical View Synthesis for Self-Supervised 360° Depth Estimation. In *3DV*, pages 690–699, 2019. 1