NeRF-HuGS: Improved Neural Radiance Fields in Non-static Scenes Using Heuristics-Guided Segmentation

Jiahao Chen[1] Yipeng Qin[2] Lingjie Liu[3] Jiangbo Lu[4] Guanbin Li[1*]

[1]Sun Yat-sen University [2]Cardiff University [3]University of Pennsylvania [4]SmartMore Corporation

chenjh328@mail2.sysu.edu.cn, qiny16@cardiff.ac.uk, lingjie.liu@seas.upenn.edu

jiangbo.lu@gmail.com, liguanbin@mail.sysu.edu.cn

Abstract

Neural Radiance Field (NeRF) has been widely recognized for its excellence in novel view synthesis and 3D scene reconstruction. However, their effectiveness is inherently tied to the assumption of static scenes, rendering them susceptible to undesirable artifacts when confronted with transient distractors such as moving objects or shadows. In this work, we propose a novel paradigm, namely "Heuristics-Guided Segmentation" (HuGS), which significantly enhances the separation of static scenes from transient distractors by harmoniously combining the strengths of hand-crafted heuristics and state-of-the-art segmentation models, thus significantly transcending the limitations of previous solutions. Furthermore, we delve into the meticulous design of heuristics, introducing a seamless fusion of Structure-from-Motion (SfM)-based heuristics and color residual heuristics, catering to a diverse range of texture profiles. Extensive experiments demonstrate the superiority and robustness of our method in mitigating transient distractors for NeRFs trained in non-static scenes. Project page: https://cnhaox.github.io/NeRF-HuGS/

1. Introduction

Neural Radiance Fields (NeRF) [29] have garnered significant attention for their remarkable achievements in novel view synthesis. Utilizing multiple-view images, NeRF conceptualizes the 3D scene as a neural field [54] and produces highly realistic renderings through advanced volume rendering techniques. This capability has opened the door to a wide array of downstream applications including 3D reconstruction [22, 43, 48], content generation [23, 33, 36], semantic understanding [14, 42, 58], etc.

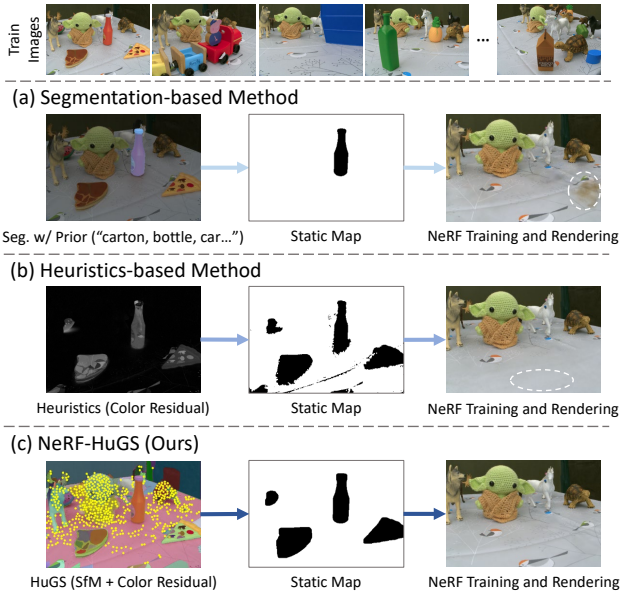


Figure 1. **Comparison between previous methods and the proposed Heuristics-Guided Segmentation (HuGS) paradigm.** When training NeRF with static scenes disturbed by transient distractors, (a) segmentation-based methods rely on prior knowledge and cannot identify unexpected transient objects (*e.g.*, pizza); (b) heuristics-based methods are more generalizable but inaccurate (*e.g.*, tablecloth textures); (c) our method combines their strengths and produces highly accurate transient *vs.* static separations, thereby significantly improving NeRF results.

However, the images used as NeRF training data must meet several strict conditions, one of which is the requirement for content consistency and stability. In other words, the native NeRF model operates under the assumption of a static scene. Any elements that exhibit motion or inconsistency throughout the entire data capture session, which we refer to as "*transient distractors*", can introduce undesirable artifacts into the reconstructed 3D geometry. However, the presence of transient distractors is nearly inevitable in real-world scenarios. For instance, in outdoor settings, random appearances of pedestrians and vehicles may occur during

*Corresponding author is Guanbin Li. This work was supported in part by the National Natural Science Foundation of China (NO. 62322608), in part by the CAAI-MindSpore Open Fund, developed on OpenI Community, in part by the Open Project Program of State Key Laboratory of Virtual Reality Technology and Systems, Beihang University (No.VRLAB2023A01), in part by Shenzhen Science and Technology Program KQTD20210811090149095 and also the Pearl River Talent Recruitment Program 2019QN01X226.

image acquisition, while indoor shooting may be affected by shadows cast by the photographer. Furthermore, manually removing these transient distractors from a substantial number of images is a challenging and time-consuming task, often necessitating pixel-by-pixel labeling.

To mitigate the effects of transient distractors, previous research has explored two main paradigms. One paradigm involves leveraging pre-trained segmentation models to detect transient distractors [12, 26, 38, 43, 45, 47]. Although it can produce accurate results, this approach exhibits limited generality as it relies on additional prerequisites of prior knowledge (*e.g.*, semantic classes of transient objects). The other strategy aims to separate transient distractors from static scenes through the application of hand-crafted heuristics [7, 15, 17, 18, 28, 40]. Nonetheless, this approach often yields imprecise or erroneous results, primarily attributed to the intricate nature of heuristic design and the inherent ill-posedness of existing heuristics.

In this work, we take the best of both worlds and propose a novel paradigm called "*Heuristics-Guided Segmentation*" (HuGS) to maximize the accuracy of static *vs.* transient object identification for NeRF in non-static scenes (Fig. 1). The rationale of our approach lies in the principle embodied in the British idiom "horses for courses", emphasizing the alignment of talents with tasks. Specifically, our paradigm harnesses the collective strengths of i) hand-crafted heuristics, adept at discerning rough indicators of static elements, and ii) contemporary segmentation models, like the Segment Anything Model (SAM) [16] renowned for their ability to delineate precise object boundaries. Furthermore, we delve deeply into the design of heuristics and suggest a seamless fusion of i) our newly devised Structure-from-Motion (SfM)-based heuristics, which efficiently identify static objects characterized by high-frequency texture patterns, with ii) the color residual heuristics derived from a partially trained Nerfacto [46], which excel at detecting static elements marked by low-frequency textures. This tailored integration of heuristics empowers our method to robustly encompass the full spectrum of static scene elements across a diverse range of texture profiles. Extensive experiments have demonstrated the superiority of our method. Our contributions can be summarized as follows:

- We propose a novel paradigm called "Heuristics-Guided Segmentation" for improving NeRF trained in non-static scenes, which takes the best of both hand-crafted heuristics and state-of-the-art segmentation models to accurately distinguish static scenes from transient distractors.
- We delve into heuristic design and propose the seamless fusion of SfM-based heuristics and color residual ones to capture a wide range of static scene elements across various texture profiles, offering robust performance and superior results in mitigating transient distractors.
- Extensive experimental results show that our method produces sharp and accurate static *vs.* transient separation results close to the ground truth, and significantly improves NeRFs trained in non-static scenarios.

2. Related Work

NeRF [29] has recently emerged as a promising solution to synthesizing novel photo-realistic views from multiple images, which is a long-standing problem in computer vision. Although numerous methods [1–3, 32, 56] have been proposed to improve its synthesis quality and training efficiency, most of them assume that the scenes to be reconstructed are static and are therefore not suitable for many real-world scenes (*e.g.*, tourist attractions).

NeRF in Non-static Scenes. In general, there are two major types of non-static scenes that present challenges for NeRF: i) Dynamic scenes that change over time, where the model needs to render consistent novel views of the scene as it evolves [10, 19, 21, 26, 34, 35, 53], *e.g.*, scenes with moving objects or environmental effects like lighting or weather changes. ii) Static scenes disturbed by transient distractors, where the model should exclude dynamic objects like tourists walking through static attractions as background scenes. Our work focuses on ii), where existing solutions can be roughly grouped into two main paradigms:

- *Segmentation-based methods* [12, 26, 38, 43, 45, 47] use pre-trained semantic or video segmentation models to identify transient distractors *vs.* static scenes and use the information obtained to facilitate NeRF training. These models can produce accurate results but have some key limitations: i) They require additional priors like the semantic class of transient distractors or the temporal relationships of the images as video frames, which are hard to satisfy in practice as it is intractable to enumerate all possible distractor classes, and images may be unordered. ii) Semantic segmentation cannot distinguish between static and transient objects of the same class.
- *Heuristics-based methods* [7, 15, 17, 18, 28, 40] use hand-crafted heuristics to separate transient distractors from static scenes during NeRF training, making themselves more generalizable as they require no prior. However, heuristics that enable accurate separation are difficult to design. For example, NeRF-W [28] observes that the density of transient objects is usually small and uses this to regularize NeRF training. However, it can easily produce foggy residuals with small densities that are not transient objects. RobustNeRF [40] distinguishes transient pixels from static ones through color residuals as transient pixels are more difficult to fit during NeRF training. However, high-frequency details of static objects are also difficult to fit, causing RobustNeRF to easily ignore them when dealing with transient distractors.

In our work, we propose a new paradigm called HuGS that

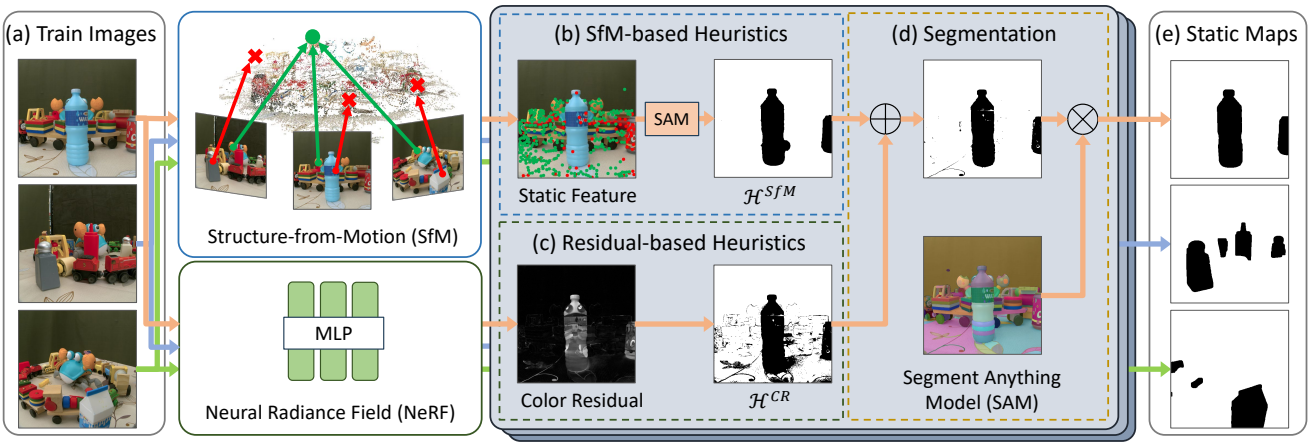


Figure 2. **Pipeline of HuGS.** (a) Given unordered images of a static scene disturbed by transient distractors as input, we first obtain two types of heuristics. (b) SfM-based heuristics use SfM to distinguish between static (green) and transient features (red). The static features are then employed as point prompts to generate dense masks using SAM. (c) Residual-based heuristics are based on a partially trained NeRF (*i.e.*, trained for several thousands of iterations) that can provide reasonable color residuals. (d) Their combination finally guides SAM again to generate (e) the static map for each input image.

takes the best of both worlds. In short, we match talents to tasks and propose to use heuristics only as rough cues to guide the segmentation and produce highly accurate transient *vs.* static separations that are close to the ground truth. We also investigate heuristics design and propose to use a combination of heuristics based on color residuals and SfM.

SfM in NeRF. SfM is a technique for reconstructing the corresponding 3D geometry from a set of 2D images. In NeRF, SfM is typically used to estimate the camera pose of an image. Recent works have also used it to estimate the scene depth [43], locate target objects [49, 55] or initialize the set of 3D Gaussians [13]. In addition to estimating camera poses, our method also uses SfM to design novel heuristics for static *vs.* transient object identification. Specifically, we leverage the insight that only feature points belonging to static scene elements can be reliably matched and triangulated across multiple views in the SfM pipeline. To the best of our knowledge, we are the first to exploit this property of SfM for NeRF in non-static scenes.

3. Preliminaries

Let $\mathcal{I} = \{I_i \mid i = 1, 2, \ldots, N_I\}$ be a set of multi-view input images with transient objects, we have:

Structure-from-Motion (SfM). SfM first extracts a set of 2D local feature points $\mathcal{F}_i$ for each I_i:

$$\mathcal{F}_i = \left\{ \left(\mathbf{x}_i^j, \mathbf{f}_i^j \right) \mid j = 1, 2, \ldots, N_{F_i} \right\}, \tag{1}$$

where $\mathbf{f}_i^j$ is an appearance descriptor and $\mathbf{x}_i^j \in \mathbb{R}^2$ denotes its coordinates in I_i. Then, SfM uses the $\mathcal{F}_i$ of all images to reconstruct a sparse point cloud $\mathcal{C}$ representing the 3D structure of the target scene, where the correspondence between 2D feature points (*i.e.*, *matching* points) in different images is determined by whether or not they correspond to the same 3D point in $\mathcal{C}$. For each 2D feature point $\left(\mathbf{x}_i^j, \mathbf{f}_i^j \right)$, we denote the number of its *matching* points in $\mathcal{I}$ as n_i^j.

Neural Radiance Field (NeRF). In short, NeRF represents a static scene with a multi-layer perceptron (MLP) parameterized by $\boldsymbol{\theta}$. Specifically, given a 3D position $\mathbf{p} \in \mathbb{R}^3$ and its viewing direction $\mathbf{d} \in \mathbb{S}^2$, NeRF outputs its corresponding color $\mathbf{c} \in \mathbb{R}^3$ and density $\sigma \in \mathbb{R}$ as:

$$(\mathbf{c}, \sigma) = \text{MLP}(\mathbf{p}, \mathbf{d}; \boldsymbol{\theta}). \tag{2}$$

This allows NeRF to render each pixel color $\hat{\mathbf{C}}(\mathbf{r})$ in a 2D projection by applying volume rendering along its corresponding camera ray $\mathbf{r}$ with multiple sample points. During training, the parameters $\boldsymbol{\theta}$ are optimized by minimizing the error between $\hat{\mathbf{C}}(\mathbf{r})$ and the ground truth color $\mathbf{C}(\mathbf{r})$ in input images using loss function:

$$\mathcal{L}(\mathbf{r}) = \mathcal{L}_{\text{recon}}(\hat{\mathbf{C}}(\mathbf{r}), \mathbf{C}(\mathbf{r})), \tag{3}$$

where $\mathcal{L}_{\text{recon}}$ is a reconstruction loss whose popular choices include the MSE loss and the Charbonnier loss [5].

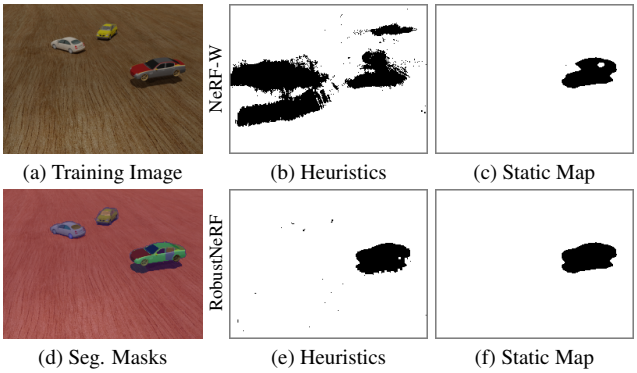
NeRF in Static Scenes. Let M_i be the static map corresponding to image I_i which labels pixels of transient objects with 0 and pixels of static objects with 1, we modify Eq. 3 in a straightforward way by using M_i as the loss weight to avoid the interference of transient pixels:

$$\mathcal{L}(\mathbf{r}) = M(\mathbf{r})\mathcal{L}_{\text{recon}}(\hat{\mathbf{C}}(\mathbf{r}), \mathbf{C}(\mathbf{r})), \tag{4}$$

where we omit i and use $\mathbf{r}$ instead for simplicity.

4. Method

As Eq. 4 implies, the more accurate the static maps M_i are, the higher the quality of the trained NeRF. To max-



(a) Training Image (b) Heuristics (c) Static Map

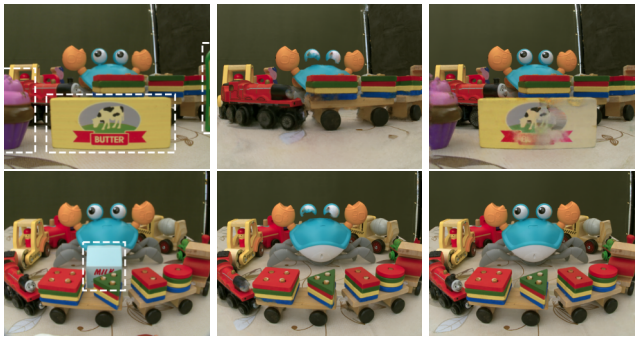
(d) Seg. Masks (e) Heuristics (f) Static Map

Figure 3. **Performance of HuGS using existing heuristics.** (a) is an example training image with a moving red car, and (d) is its segmentation result using SAM. (b, e) are heuristic maps obtained from different partially trained models. (c, f) are static maps produced by our method, where inaccurate heuristics may lead to incorrect results (NeRF-W).

imize the accuracy of M_i, we follow the British idiom "horses for courses", which suggests matching talents to tasks, and approach the problem through a novel framework called *Heuristics-Guided Segmentation (HuGS)* (Sec. 4.1). As Fig. 2 shows, HuGS combines the strengths of both hand-crafted heuristics in identifying coarse cues of static objects and the capabilities of state-of-the-art segmentation models in producing sharp and accurate object boundaries. Furthermore, we conduct an in-depth analysis of the choice of heuristics (Sec. 4.2). Our solution combines novel SfM-based heuristics, which effectively identify static objects with high-frequency texture patterns, with the color residual heuristics from a partially trained Nerfacto [46], which excel at detecting static objects characterized by low-frequency textures. This tailored integration of heuristics allows our method to robustly capture the full range of static scene elements across diverse texture profiles.

4.1. Heuristics-Guided Segmentation (HuGS)

While humans can easily distinguish between transient and static objects, providing a rigorous mathematical definition of this distinction has proven elusive thus far due to the high diversity of real-world scenes. To this end, the most effective existing solutions rely heavily on hand-crafted heuristics to make this distinction. For example, NeRF-W [28] employs the heuristics that transient objects usually have lower density than their static counterparts and incorporates this as a regularization term during NeRF training; RobustNeRF [40] leverages the observation that transient objects are typically harder to fit during optimization and uses it to produce the static maps used in Eq. 4. However, despite their success, these methods implicitly make the strong assumption that distinguishing between transient and static rays/pixels can be determined *solely* based on simple hand-crafted heuristics, which does not hold up while handling the diverse shapes and appearances of real-world objects. As a result, these methods are prone to produce errors and/or ambiguous object boundaries (Figs. 3b and 3e).



(a) Training Image (b) Quantile $= 0.7$ (c) Quantile $= 0.9$

Figure 4. **Performance of RobustNeRF *vs*. transient objects of different sizes.** Transient distractors in the training images are framed in white. A lower quantile (threshold) causes the model to miss small-sized static objects, while a higher quantile prevents the removal of large-sized transient objects.

To address this limitation, we propose a novel framework HuGS that avoids fully relying on hand-crafted heuristics to differentiate between transient and static objects. Instead, our approach leverages heuristics to provide *only* rough hints $\mathcal{H}_i$ about potential static objects in each image I_i, and then refines those imprecise cues into accurate static maps M_i using the segmentation masks of I_i provided by model S. Specifically, let $S(I_i) = \{m_i^1, m_i^2, ..., m_i^{N_{M_i}}\}$, where m_i^j denotes the segmentation mask of the j-th object (instance) and N_{M_i} is the number of masks, we have:

$$M_i = \bigcup m_i^j, \ \forall \frac{m_i^j \cap \mathcal{H}_i}{m_i^j} \geq \mathcal{T}_m, \tag{5}$$

where $\mathcal{T}_m$ is a user-specified threshold and we implement S using the state-of-the-art SAM [16]. As shown in Fig. 3, our framework can produce static maps with sharp object boundaries even when using partially trained (10%) models of previous methods [28, 40] as heuristics (Figs. 3b and 3e). However, despite the relaxation, the success of our framework is based on the assumption that rough but accurate $\mathcal{H}_i$ about static objects are available (Figs. 3c and 3f).

4.2. Heuristics Development

To provide rough but accurate heuristics $\mathcal{H}_i$ of static objects, we use a combination of two complementary heuristics, *i.e.* our novel SfM-based heuristics and the color residual heuristics from a partially trained Nerfacto [46], which excel in detecting statics objects with high-frequency and low-frequency textures respectively.

SfM-based Heuristics. As mentioned above in Sec. 3, SfM reconstruction relies on matching distinct, identifiable features across images. This makes it well-suited for detect-

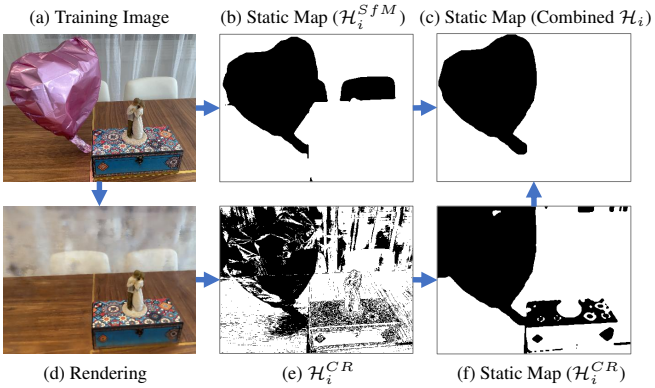


Figure 5. **Heuristics combination.** (b) The SfM-based heuristics $\mathcal{H}_i^{SfM}$ alone captures high-frequency static details (*e.g.*, box textures) well but misses smooth ones (*e.g.*, white chairs). This could be complemented by incorporating (e) residual-based heuristics $\mathcal{H}_i^{CR}$ from a (d) Nerfacto with $5k$ training iterations which does the opposite (f). Their combination (c) covers the full spectrum of static scenes and identifies transient objects (*e.g.*, pink balloon).

ing objects characterized by high-frequency textures, since these distinctive textures provide abundant unique features to match. To distinguish between static and transient objects, our SfM-based heuristics share a similar high-level intuition to previous methods in that transient objects are considered a minority compared to static ones and their positions are constantly changing. However, ours has a different interpretation of "minority". Specifically, our method defines it as the *frequency of occurrence* across input images, which aligns well with the temporal meaning of "transient". In contrast, NeRF-W [28] and RobustNeRF [40] interpret "minority" in terms of total density or quantile of color residual respectively, which relate more to *spatial area coverage*. As a result, their methods struggle with transient objects of varying sizes (Fig. 4), since the area-based definitions do not fully capture the temporal aspect of identifying transient objects. Recalling the definition of n_i^j (the number of matching points) and N_I (the number of input images) in Sec. 3, we have the following observation:

Observation 1. *The SfM features of static objects have much larger n_i^j than those of transient ones in image I_i.*

Note that the already smaller n_i^j of transient objects could be further reduced by their constantly changing positions (*i.e.*, less likely to get matched during SfM reconstruction), making them easier to distinguish. Accordingly, we set a threshold $\mathcal{T}_{SfM}$ to obtain set $\mathcal{X}_i$ of the coordinates of static feature points for each image I_i:

$$\mathcal{X}_i = \left\{ \mathbf{x}_i^j \,\middle|\, \mathbf{x}_i^j \in \mathcal{F}_i \text{ and } \frac{n_i^j}{N_I} \geq \mathcal{T}_{SfM} \right\}. \tag{6}$$

where we set $\mathcal{T}_{SfM}$ based on $\frac{n_i^j}{N_I}$ rather than n_i^j as the former better represents the *frequency of occurrence* across the whole scene. However, $\mathcal{X}_i$ is a relatively sparse point set, so we need to convert it to a pixel-wise static map for NeRF training. Fortunately, SAM [16] is a promptable segmentation model that can accept points as prompts and output their corresponding segmentation masks. Therefore, we feed $\mathcal{X}_i$ into SAM to obtain heuristics $\mathcal{H}_i^{SfM}$, which is a static map where a pixel value of 1 means that it belongs to a static object and 0 means that it is a transient one.

Combined Heuristics. Although effective, our SfM-based heuristics $\mathcal{H}^{SfM}$ may neglect low-frequency static objects due to their lack of distinctive features (Fig. 5). To address this limitation, we propose an integrated approach that incorporates the complementary strength of another heuristic: the color residual of a partially trained Nerfacto [46], which effectively identifies smooth transient objects but struggles with textured objects. Specifically, we first train a Nerfacto for several thousands of iterations and construct its color residual map $\mathcal{R}_i$ using the color residual for each ray $\mathbf{r}$ as $\epsilon(\mathbf{r}) = \|\hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})\|_2$. We then combine $\mathcal{H}_i^{SfM}$ with residual-based heuristics $\mathcal{H}_i^{CR}$ to get heuristics $\hat{\mathcal{H}}_i$:

$$\hat{\mathcal{H}}_i = \mathcal{H}_i^{SfM} \cup \mathcal{H}_i^{CR}, \tag{7}$$

where $\mathcal{H}_i^{CR} = \mathcal{R}_i \leq \text{mean}(\mathcal{R}_i)$. While in practice, our $\mathcal{H}_i^{SfM}$ may occasionally incorrectly include some transient objects due to misclassification of feature points or SAM segmentation errors. To eliminate them, we apply an upper bound defined by $\hat{\mathcal{H}}_i^{CR}$ to Eq. 7 as additional insurance:

$$\mathcal{H}_i = \hat{\mathcal{H}}_i \cap \hat{\mathcal{H}}_i^{CR}, \tag{8}$$

where $\hat{\mathcal{H}}_i^{CR} = \mathcal{R}_i \leq \text{quantile}(\mathcal{R}_i, \mathcal{T}_{CR})$. $\mathcal{T}_{CR}$ is a high threshold that ensures $\hat{\mathcal{H}}_i^{CR}$ include all static objects.

Remark. We use Nerfacto [46] to generate residual maps as it can be trained quickly with much fewer computational resources and still producing reasonable results (Fig. 5). Although relatively low, this level of performance is sufficient to satisfy our requirement for rough heuristic cues, which further demonstrates the superiority of our paradigm.

5. Experiments

5.1. Experimental Setup

Datasets. We use three datasets in our experiments:

- *Kubric Dataset [53].* Generated by Kubric [11], this synthetic dataset contains five scenes with simple geometries placed in an empty room. The frames have temporal relationships and a subset of geometries serves as transient distractors that move between frames.
- *Distractor Dataset [40].* This real-world dataset has four controlled indoor scenes with 1-150 distractors per scene.
- *Phototourism Dataset [28].* This real-world dataset has scenes of four cultural landmarks, each with photos collected online containing various transient distractors. The

Method	Car			Cars			Bag			Chairs			Pillow			Avg.		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Nerfacto [46]	30.71	.862	.227	29.50	.809	.316	31.32	.917	.103	27.41	.811	.258	29.86	.906	.148	29.76	.861	.210
Mip-NeRF 360 [2]	26.67	.846	.238	28.88	.822	.281	32.81	.948	.053	27.07	.839	.179	29.66	.919	.123	29.02	.875	.175
NeRF-W [28]	29.44	.901	.124	28.34	.867	.186	34.49	.946	.045	22.75	.826	.187	29.04	.915	.142	28.81	.891	.137
HA-NeRF [7]	28.69	.915	.124	31.95	.903	.143	38.48	.969	**.021**	33.48	.922	.071	31.66	.946	.083	32.85	**.931**	.089
D^2NeRF [53]	34.03	.874	.099	33.67	.844	.123	33.77	.889	.118	32.77	.875	.113	29.49	.907	.139	32.75	.878	.118
RobustNeRF [40]	**37.31**	**.968**	**.040**	**40.52**	**.963**	**.047**	**40.50**	**.976**	.026	**38.56**	**.958**	**.037**	**41.31**	**.980**	**.028**	**39.64**	**.969**	**.036**
Ours (Nerfacto)	**39.49**	**.964**	**.042**	**39.95**	**.958**	**.045**	**41.39**	**.980**	**.017**	**38.48**	**.962**	**.036**	**42.70**	**.982**	**.025**	**40.40**	**.969**	**.033**
Ours (Mip-NeRF 360)	**39.75**	**.972**	**.036**	**40.74**	**.966**	**.046**	**42.32**	**.983**	**.019**	**39.32**	**.968**	**.033**	**43.90**	**.986**	**.023**	**41.21**	**.975**	**.032**

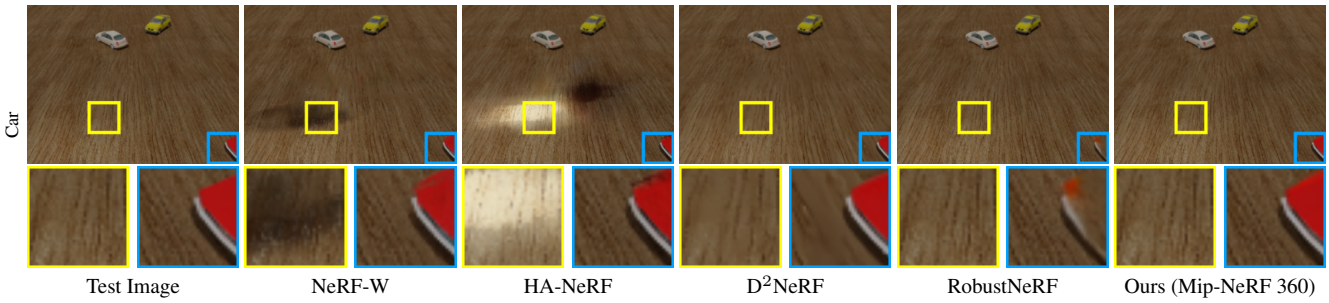


Figure 6. **Quantitative and qualitative results on the Kubric dataset.** The **1st**, **2nd** and **3rd** best results are highlighted. Quantitatively, our method not only significantly improves the performance of Nerfacto and Mip-NeRF 360, but also helps Mip-NeRF 360 outperform the previous methods and become the SOTA. Qualitatively, our method can better preserve static details while ignoring transient distractors.

Method	Statue			Android			Crab			BabyYoda			Avg.		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Nerfacto [46]	18.21	.591	.336	21.34	.617	.226	26.62	.864	.135	22.06	.697	.267	22.06	.692	.241
Mip-NeRF 360 [2]	**19.86**	.690	.233	21.81	.695	.176	29.25	.918	.086	23.75	.770	.216	23.67	.768	.178
NeRF-W [28]	18.91	.616	.369	20.62	.664	.258	26.91	.866	.157	28.64	.752	.260	23.77	.725	.261
HA-NeRF [7]	18.67	.616	.367	22.03	.706	.203	28.58	.901	.116	**29.28**	.779	.208	24.64	.750	.224
RobustNeRF [40]	**20.60**	**.758**	**.154**	**23.28**	**.755**	**.126**	**32.22**	**.945**	**.060**	**29.78**	**.821**	**.155**	**26.47**	**.820**	**.124**
Ours (Nerfacto)	19.18	**.703**	**.183**	**22.59**	**.720**	**.120**	**32.11**	**.939**	**.033**	28.77	**.802**	**.087**	**25.66**	**.791**	**.106**
Ours (Mip-NeRF 360)	**21.00**	**.774**	**.135**	**23.32**	**.763**	**.123**	**34.16**	**.956**	**.032**	**30.70**	**.834**	**.124**	**27.29**	**.832**	**.103**

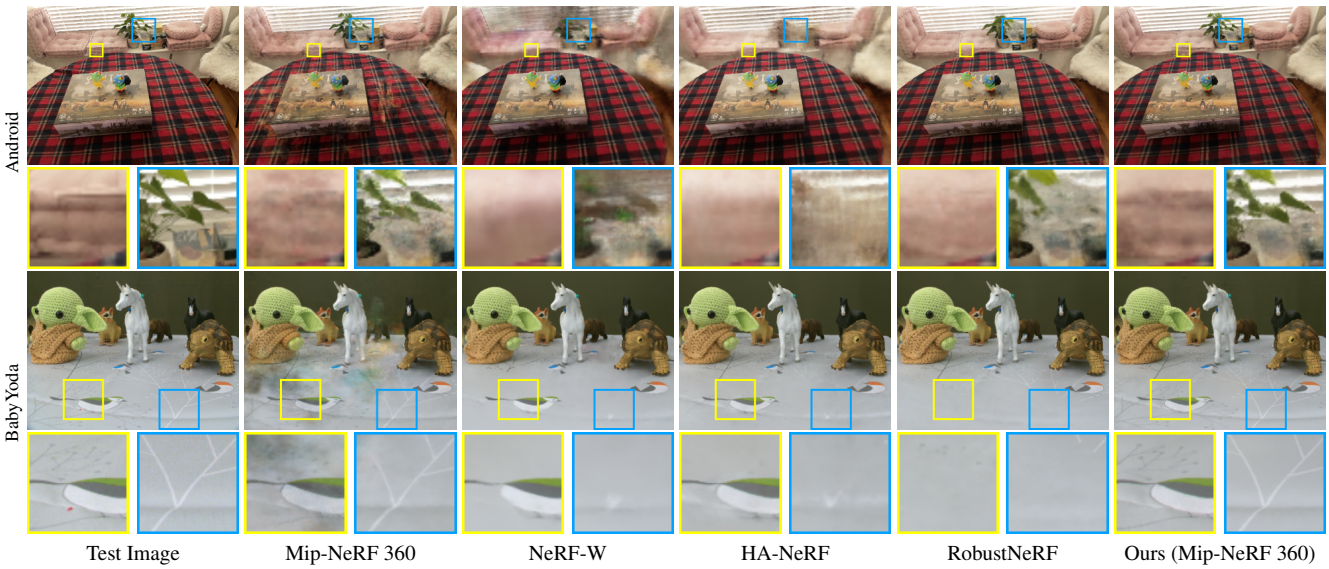


Figure 7. **Quantitative and qualitative results on the Distractor dataset.** The **1st**, **2nd** and **3rd** best results are highlighted. Our method applied to Mip-NeRF 360 is the best in most quantitative results, with our method applied to Nerfacto leading the rest. Qualitatively, our method captures scene details better compared to other baselines, which suffer from missing or disturbed details.

Method	Brandenburg Gate			Sacre Coeur			Taj Mahal			Trevi Fountain			Avg.		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Nerfacto [46]	24.69	.890	.132	21.09	.819	.169	22.80	.804	.237	22.63	.748	.195	22.80	.815	.183
Mip-NeRF 360 [2]	25.59	.904	.121	**21.46**	.818	.175	**24.33**	.828	.202	**23.25**	.767	.189	23.66	.829	.172
NeRF-W [28]	23.62	.851	.171	19.75	.749	.233	22.23	.772	.317	20.80	.664	.299	21.60	.759	.255
HA-NeRF [7]	23.93	.881	.140	19.85	.808	.175	20.70	.811	.234	20.07	.713	.223	21.14	.803	.193
RobustNeRF [40]	**25.79**	**.923**	**.094**	20.94	**.852**	**.137**	**24.64**	**.859**	**.173**	**23.58**	**.785**	**.170**	**23.73**	**.855**	**.144**
Ours (Nerfacto)	**25.99**	**.919**	**.087**	**22.03**	**.856**	**.132**	24.06	**.836**	**.198**	22.90	**.770**	**.173**	**23.74**	**.845**	**.147**
Ours (Mip-NeRF 360)	**27.17**	**.929**	**.083**	**22.23**	**.862**	**.124**	**24.92**	**.857**	**.176**	**23.41**	**.788**	**.165**	**24.43**	**.859**	**.137**

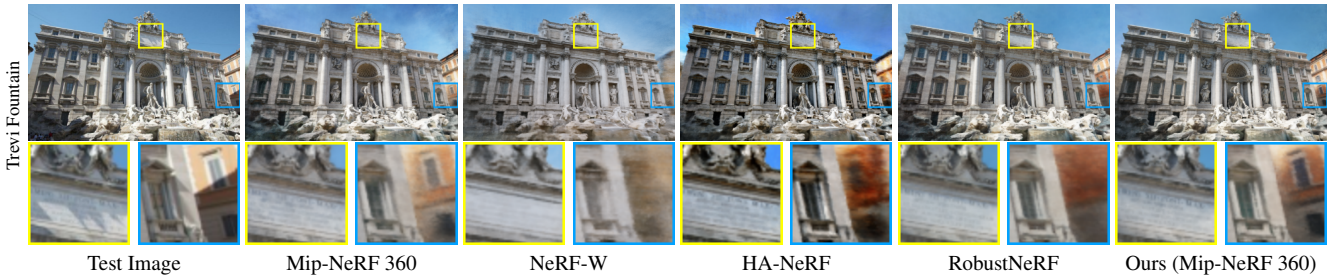


Figure 8. **Quantitative and qualitative results on the Phototourism dataset.** The **1st**, **2nd** and **3rd** best results are highlighted. Note that the main content of test images in this dataset is the upper part of the building, which is less affected by transient distractors (*e.g.*, tourists). Thus, our method brings less improvement but still yields competitive results against the SOTA.

landmark and distractor appearances vary across images due to shooting differences.

Implementation Details. Please see the supplementary materials for more details.

- *HuGS.* We use COLMAP [41] for SfM reconstruction and SAM [16] as the segmentation model. COLMAP uses SIFT [27] to extract image features, and we set COLMAP's parameters to default values. We set $\mathcal{T}_m$ to a common 0.5. The values of $\mathcal{T}_{SfM}$ and $\mathcal{T}_{CR}$ depend on the complexity of the scene, so we empirically set them to 0.2 and 0.9 for Kubric, 0.01 and 0.95 for Distractor, 0.01 and 0.97 for Phototourism datasets.
- *NeRF Training.* We apply our method to two baseline NeRF models, Nerfacto [46] and Mip-NeRF 360 [2], to show its generalizability. We did not test it on the vanilla NeRF [29] because the vanilla NeRF has difficulty handling the unbounded scenes in the Distractor dataset [2].

5.2. Evaluation on View Synthesis

Baselines. In addition to baseline models, we compare our method to three other state-of-the-art heuristics-based methods: NeRF-W [28], HA-NeRF [7] and RobustNeRF [40], which design heuristics based on scene density, pixel visible possibility, and color residuals, respectively. We also compare our method to D^{2}NeRF [53] on the Kubric dataset, which is a dynamic NeRF that works well on monocular videos. Segmentation-based methods are not included in our comparison because they rely on priors of transient distractors, which cannot be satisfied in most scenes.

Comparisons. Both the above models and ours are trained on images disturbed by transient distractors and evaluated on images with only static scenes. We report image synthesis qualities based on PSNR, SSIM [50] and LPIPS [57].

- *Kubric dataset* (Fig. 6). Compared to the native ones, applying our method leads to substantial PSNR improvements of 8.78 to 12.84dB for Nerfacto, and 9.51 to 14.24dB for Mip-NeRF 360. Our method achieves this by generating high-quality static maps that effectively shield the native models from pixels disturbed by transient distractors. Compared to the other baselines, our method achieves the highest quantitative results and maintains a good balance between ignoring transient distractors and preserving static details. Specifically, NeRF-W and HA-NeRF fail due to incorrect decoupling of transient distractors from the static scenes; D^{2}NeRF and RobustNeRF achieve better decoupling but lose static details such as ground textures and the red car.
- *Distractor dataset* (Fig. 7). The results and conclusions are similar to those on the Kubric dataset.
- *Phototourism dataset* (Fig. 8). Its training and test sets share a unique feature: the landmark *main bodies* are not deeply disturbed by transient distractors (*e.g.* nearby tourists) and can be well reconstructed even without the removal of transient distractors. Thus, the improvement from our method mainly focus on the landmark *boundaries* and is relatively less compared to the above datasets. Nonetheless, our results remain quantitatively competitive and qualitatively recover more details compared to prior works. Please see the supplement for more details.

5.3. Evaluation on Segmentation

Baselines. We perform the comparison on the Kubric dataset, which is synthetic and has ground truth segmen-

Method	Seg. Type	Require Prior	Car		Cars		Bag		Chairs		Pillow		Avg.	
			mIoU↑	F1↑	mIoU↑	F1↑	mIoU↑	F1↑	mIoU↑	F1↑	mIoU↑	F1↑	mIoU↑	F1↑
DeepLabv3+ [6]	*Semantic*	✓	.604	.378	.578	.293	.501	.048	.564	.239	.535	.149	.556	.221
Mask2Former [8]	*Semantic*	✓	.664	.520	.622	.376	.513	.071	.707	.561	.653	.377	.632	.381
Grounded-SAM [16, 25]	*Open-Set*	✓	.888	.877	.640	.406	.755	.671	.603	.373	**.851**	**.828**	.747	.631
DINO [4]	*Video*	✓	**.947**	**.947**	.720	.557	.591	.367	**.777**	**.703**	**.911**	**.904**	.789	.695
NeRF-W [28]	/	✗	.682	.575	.584	.328	.526	.099	.547	.298	.557	.296	.579	.319
HA-NeRF [7]	/	✗	.869	.852	**.823**	**.724**	**.813**	**.771**	.729	.595	.819	.766	**.811**	**.742**
D^{2}NeRF [53]	/	✗	**.912**	**.909**	**.895**	**.867**	**.794**	**.727**	.660	.507	.800	.760	**.812**	**.754**
RobustNeRF [40]	/	✗	.813	.784	.718	.547	.731	.633	**.731**	**.638**	.724	.633	.743	.647
HuGS (Ours)	/	✗	**.963**	**.964**	**.940**	**.907**	**.939**	**.935**	**.937**	**.927**	**.940**	**.937**	**.944**	**.934**



Figure 9. **Quantitative and qualitative segmentation results on the Kubric dataset.** The **1st**, **2nd** and **3rd** best results are highlighted.

tation data. We compare our method with various existing segmentation models, including semantic segmentation models [6, 8], open-set segmentation models [16, 25] and video segmentation models [4]. The baseline NeRF models mentioned above are also compared by using the static maps generated after they are fully trained.

Comparisons (Fig. 9). We report segmentation qualities based on the mIoU and F1 score. Interestingly, we observe that even when prior knowledge is provided, the performance of existing segmentation models is limited because they are not designed for this specific task. On the other hand, heuristics-based methods can roughly localize transient distractors but cannot provide accurate segmentation results. By combining heuristics and the segmentation model together, our method takes the best of both worlds and can accurately segment transient distractors from static scenes without any prior knowledge.

Verification of Observation 1. Please see the supplementary materials for additional experiments in which we verify the correctness of Observation 1.

5.4. Ablation Study

Based on Nerfacto, we remove different components of our method to study their effects on two different datasets. As shown in Fig. 10, method (a) without any static maps, *i.e.*, the native Nerfacto, performs the worst. Using SfM-based heuristics or residual-based heuristics alone has limited improvement because the former cannot capture smooth surfaces and the latter has difficulty handling high-frequency details. The complete method (f), which combines them with the segmentation model, achieves the best results.

	$\mathcal{H}^{SfM}$	$\mathcal{H}^{CR}$	SAM	Kubric (Avg.)			Distractor (Avg.)		
				PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
(a)	-	-	-	29.76	.861	.210	22.06	.692	.241
(b)	✓	-	-	38.14	.962	.054	**25.67**	**.788**	**.108**
(c)	-	✓	-	39.86	**.967**	**.036**	24.95	.767	.146
(d)	-	✓	✓	**40.12**	**.968**	**.033**	24.58	.779	.126
(e)	✓	✓	-	**40.11**	**.968**	**.035**	**25.40**	**.786**	**.116**
(f)	✓	✓	✓	**40.40**	**.969**	**.033**	**25.66**	**.791**	**.106**

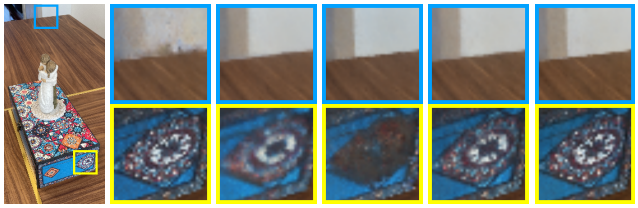


Figure 10. **Ablation results.** The patches in blue frames denote the smooth wall, and those in yellow frames denote complex textures. The **1st**, **2nd** and **3rd** best results are highlighted.

6. Conclusions

In this work, we propose a novel heuristics-guided segmentation paradigm that effectively addresses the prevalent issue of transient distractors in real-world NeRF training. By strategically combining the complementary strengths of hand-crafted heuristics and state-of-the-art semantic segmentation models, our method achieves highly accurate segmentation of transient distractors across diverse scenes without any prior knowledge. Through meticulous heuristic design, our method can capture both high and low-frequency static scene elements robustly. Extensive experiments demonstrate the superiority of our approach over existing methods. Please see the supplementary details for **limitations and future work**.

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. 6, 7, 1, 2

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *arXiv preprint arXiv:2304.06706*, 2023. 2

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 8, 3

[5] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of 1st international conference on image processing*, pages 168–172. IEEE, 1994. 3

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 8, 2

[7] Xingyu Chen, Qi Zhang, Xiaoyu Li, Yue Chen, Ying Feng, Xuan Wang, and Jue Wang. Hallucinated neural radiance fields in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12943–12952, 2022. 2, 6, 7, 8, 1, 3

[8] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 8, 2

[9] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark, 2020. 2

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

[11] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3749–3761, 2022. 5

[12] Mert Asim Karaoglu, Hannah Schieber, Nicolas Schischka, Melih Görgülü, Florian Grötzner, Alexander Ladikos, Daniel Roth, Nassir Navab, and Benjamin Busam. Dynamon: Motion-aware fast and robust camera localization for dynamic nerf, 2023. 2

[13] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4):1–14, 2023. 3

[14] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. *arXiv preprint arXiv:2303.09553*, 2023. 1

[15] Injae Kim, Minhyuk Choi, and Hyunwoo J. Kim. Up-nerf: Unconstrained pose-prior-free neural radiance fields. *arXiv:2311.03784*, 2023. 2

[16] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 2, 4, 5, 7, 8, 1, 3

[17] Jaewon Lee, Injae Kim, Hwan Heo, and Hyunwoo J Kim. Semantic-aware occlusion filtering neural radiance fields in the wild. *arXiv preprint arXiv:2303.03966*, 2023. 2, 3

[18] Peihao Li, Shaohui Wang, Chen Yang, Bingbing Liu, Weichao Qiu, and Haoqian Wang. Nerf-ms: Neural radiance fields with multi-sequence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18591–18600, 2023. 2

[19] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5521–5531, 2022. 2

[20] Yuan Li, Zhi-Hao Lin, David Forsyth, Jia-Bin Huang, and Shenlong Wang. Climatenerf: Extreme weather synthesis in neural radiance field. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3227–3238, 2023. 3

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

[22] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8456–8465, 2023. 1

[23] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023. 1

[24] Haotong Lin, Qianqian Wang, Ruojin Cai, Sida Peng, Hadar Averbuch-Elor, Xiaowei Zhou, and Noah Snavely. Neural scene chronology. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20752–20761, 2023. 3

[25] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 8, 3

[26] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13–23, 2023. 2

[27] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004. 7

[28] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021. 2, 4, 5, 6, 7, 8, 1

[29] 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 *European Conference on Computer Vision*, pages 405–421. Springer, 2020. 1, 2, 7

[30] Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, Peter Hedman, Ricardo Martin-Brualla, and Jonathan T. Barron. MultiNeRF: A Code Release for Mip-NeRF 360, Ref-NeRF, and RawNeRF, 2022. 2

[31] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20669–20679, 2023. 3, 5

[32] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 2

[33] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021. 1

[34] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 2

[35] Sungheon Park, Minjung Son, Seokhwan Jang, Young Chun Ahn, Ji-Yeon Kim, and Nahyup Kang. Temporal interpolation is all you need for dynamic neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4212–4221, 2023. 2

[36] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2022. 1

[37] Chen Quei-An. Nerf_pl: a pytorch-lightning implementation of nerf, 2020. 2

[38] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12932–12942, 2022. 2

[39] Viktor Rudnev, Mohamed Elgharib, William Smith, Lingjie Liu, Vladislav Golyanik, and Christian Theobalt. Nerf for outdoor scene relighting. In *European Conference on Computer Vision*, pages 615–631. Springer, 2022. 3

[40] Sara Sabour, Suhani Vora, Daniel Duckworth, Ivan Krasin, David J Fleet, and Andrea Tagliasacchi. Robustnerf: Ignoring distractors with robust losses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20626–20636, 2023. 2, 4, 5, 6, 7, 8, 1

[41] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 7

[42] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kontschieder. Panoptic lifting for 3d scene understanding with neural fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9043–9052, 2023. 1

[43] Jiaming Sun, Xi Chen, Qianqian Wang, Zhengqi Li, Hadar Averbuch-Elor, Xiaowei Zhou, and Noah Snavely. Neural 3d reconstruction in the wild. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 1, 2, 3

[44] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2149–2159, 2022. 5

[45] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8258, 2022. 2

[46] 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 2023 Conference Proceedings*, pages 1–12, 2023. 2, 4, 5, 6, 7, 1

[47] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12922–12931, 2022. 2

[48] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view recon-

struction. *Advances in Neural Information Processing Systems*, 34:27171–27183, 2021. 1

[49] Yuang Wang, Xingyi He, Sida Peng, Haotong Lin, Hujun Bao, and Xiaowei Zhou. Autorecon: Automated 3d object discovery and reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21382–21391, 2023. 3

[50] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 7

[51] Zian Wang, Tianchang Shen, Jun Gao, Shengyu Huang, Jacob Munkberg, Jon Hasselgren, Zan Gojcic, Wenzheng Chen, and Sanja Fidler. Neural fields meet explicit geometric representations for inverse rendering of urban scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8370–8380, 2023. 3

[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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16528–16538, 2023. 5

[53] Tianhao Wu, Fangcheng Zhong, Andrea Tagliasacchi, Forrester Cole, and Cengiz Oztireli. D2nerf: Self-supervised decoupling of dynamic and static objects from a monocular video. *Advances in Neural Information Processing Systems*, 35:32653–32666, 2022. 2, 5, 6, 7, 8, 1

[54] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, pages 641–676. Wiley Online Library, 2022. 1

[55] Youtan Yin, Zhoujie Fu, Fan Yang, and Guosheng Lin. Ornerf: Object removing from 3d scenes guided by multiview segmentation with neural radiance fields. *arXiv preprint arXiv:2305.10503*, 2023. 3, 5

[56] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2

[57] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 7

[58] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15838–15847, 2021. 1

[59] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2019. 3

NeRF-HuGS: Improved Neural Radiance Fields in Non-static Scenes Using Heuristics-Guided Segmentation

Supplementary Material

Method	Cor.	Cars PSNR↑	Cars SSIM↑	Cars LPIPS↓	Chairs PSNR↑	Chairs SSIM↑	Chairs LPIPS↓
D^{2}NeRF	-	26.01	.816	.140	24.55	.536	.171
	✓	**33.67**	**.844**	**.123**	**32.77**	**.875**	**.113**
RobustNeRF	-	26.60	.946	.064	24.84	.517	.117
	✓	**40.52**	**.963**	**.047**	**38.56**	**.958**	**.037**
Ours (Mip-NeRF 360)	-	26.54	.948	.064	24.84	.518	.116
	✓	**40.74**	**.966**	**.046**	**39.32**	**.968**	**.033**

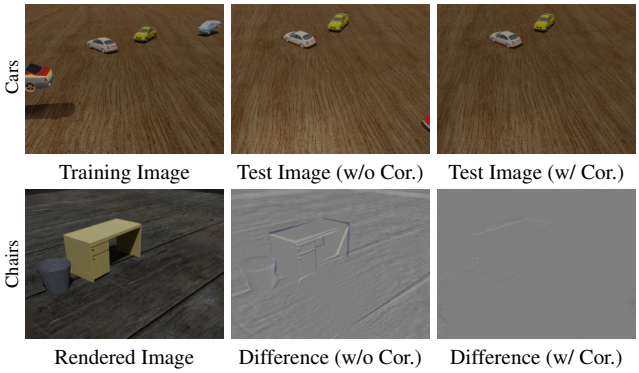


Figure 11. **Quantitative results and visualization of the correction on the Kubric dataset. Better results** are highlighted. After correction, the resynthesized test images keep the same illumination as the training images in the Cars scene and are aligned with the camera poses in the Chairs scene, resulting in smaller difference between the test images and the rendered images from NeRF. "Cor.": Correction.

7. Implementation Details

7.1. Datasets

Kubric Dataset [7]. Each scene of this dataset contains 200 noisy images for training and 100 clean images for evaluation, and we downsample these images by 2x following the prior works. As mentioned in Sec. 5.1 of the main paper, images of this dataset are ordered in time, which is a requirement for dynamic NeRFs and video segmentation models to work. In addition, we uncovered some inconsistencies in the test set of the released Kubric dataset [53] that affect the evaluation. Specifically, i) in the Cars scene, the global illumination of the test set is inconsistent with that of the training set, and a red car that acts as a transient distractor is not removed from the test images; ii) in the Chairs scene, there is no alignment between the camera poses and corresponding images of the test set. Therefore, we use the originally released script files to resynthesize the above scenes and correct the test images. After correction, the quantitative results of different models have improved and are now consistent in magnitude with the results of other scenes (Fig. 11).

Distractor Dataset [40]. Each scene of this dataset contains 72-255 noisy images for training and 19-202 clean images for evaluation. Images of this dataset are unordered, and we downsample them by 8x following [40].

Phototourism Dataset [28]. Each scene of this dataset contains 859-1716 unordered noisy images for training and 10-27 nearly clean images for evaluation, and we downsample these images by 2x. Different from other datasets, images of this dataset vary greatly in appearance (*e.g.*, scene illumination, image style, sky and building color) due to diverse shooting environments and shooting equipment. To address such appearance variations, we follow the prior works [2, 28] and assign a 48-dimensional *appearance embedding* (a.k.a., *GLO vector*) to each image when training baseline models and ours. Except for HA-NeRF, other models for comparison require finetuning appearance embeddings on new images. We therefore optimize the embeddings with $5k$ iterations using the left half of the test images after training and report metrics on the right half.

7.2. Segment Anything Model (SAM)

The Segment Anything Model (SAM) [16] is a promptable segmentation model that unifies the 2D segmentation task by introducing a prompt-based segmentation paradigm. Given an image I and a set of prompts $\mathcal{P}$ as input, SAM can output the corresponding 2D segmentation binary mask M:

$$M = \text{SAM}\left(I, \mathcal{P}\right), \tag{9}$$

where $\mathcal{P}$ can be points, boxes or masks. Point and box prompts are the most common forms, while mask prompts are used to assist them.

In our method, we use SAM twice for each image:

1. The first is to convert static SfM feature points (as input prompt $\mathcal{P}$) into their corresponding static map $\mathcal{H}_i^{SfM}$ (Sec. 4.2 of the main paper). Specifically, SAM converts each of them (or with its nearest static neighbors) to a sub-map and unifies them into $\mathcal{H}_i^{SfM}$.
2. The second is to generate instance segmentation masks $S(I_i)$ (Sec. 4.1 of the main paper). SAM achieves this by taking a regular point grid as input prompt $\mathcal{P}$, and the default grid resolution of our method is 64×64.

7.3. Model Architecture, Training and Evaluation

Nerfacto [46]. Nerfacto-related models in our experiments have the same architecture as Nerfacto-huge from the official Nerfstudio codebase. The model consists of two den-

Quantile (Threshold)	Statue			Android			Crab			BabyYoda			Avg.		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
0.5	16.80	.670	.303	19.46	.695	.212	16.48	.751	.329	17.59	.630	.395	17.58	.687	.310
0.6	18.38	.707	.243	20.52	.722	.187	21.09	.857	.177	20.54	.736	.259	20.13	.755	.216
0.7	20.06	.750	.175	21.45	.738	.156	26.17	.922	.092	25.58	.798	.186	23.32	.802	.152
0.8	**20.60**	**.758**	**.154**	23.28	**.755**	**.126**	**32.22**	**.945**	**.060**	29.78	.821	.155	26.47	**.820**	**.124**
0.9	20.54	.745	.171	**23.31**	.753	.128	32.16	.944	**.060**	**30.35**	**.828**	**.145**	**26.59**	.817	.126
1.0	19.86	.690	.233	21.81	.695	.176	29.25	.918	.086	23.75	.770	.216	23.67	.768	.178



Figure 12. **Quantitative and qualitative results of RobustNeRF with different thresholds on the Distractor dataset.** The **best results** are highlighted.

sity fields for proposal sampling and one nerfacto field for final sampling, and each field consists of a hash encoding [32] with a base resolution of 16 and a tiny MLP with a width of 256. The maximum hash encoding resolution is 512/2048 for the first/second density field, and 8192 for the nerfacto field. We sample 512/256/128 points per ray in the first/second/third sampling. We train these models for $25k$ iterations ($\sim$1 hour on one NVIDIA RTX 4090) through the Adam optimizer with a batch size of $16,384$ rays. The learning rate is exponentially decayed from 10^{-2} to 10^{-3}. When experimenting on the Kubric dataset, we remove the first density field because we find that multiple proposal sampling will result in biased geometry and degraded rendering quality. For the partially trained models used in HuGS, We reduce the number of training iterations to at most $5k$.

Mip-NeRF 360 [2]. For all models related to Mip-NeRF 360, we use the reference implementation of the MultiNeRF codebase [30]. The model architecture comprises a proposal MLP with 4 hidden layers and 256 units, and a NeRF MLP with 8 hidden layers and 1,024 units. Following the default settings, we train Mip-NeRF 360 for $250k$ iterations ($\sim$1 day on four NVIDIA RTX 3090) through the Adam optimizer with a batch size of 16,384 rays. The learning rate is exponentially decayed from 2×10^{-3} to 2×10^{-5}.

NeRF-W [28]. Because no official NeRF-W code is available, we refer to another popular open-source work [37]. The model architecture and training settings are the same as [37], except that the number of training iterations is unified to $250k$ (*vs.* $200k$-$900k$ in [37]) and the batch size is expanded to 4,096 (*vs.* 1,024 in [37]). To obtain heuristics or binary masks related to transient distractors, we follow [53] and apply a threshold of 0.1 on the accumulated weights of the transient component.

HA-NeRF [7]. The model architecture and training settings are the same as the official codebase. When experimenting on the Kubric and Distractor datasets, we remove the appearance hallucination module and unify the number of training iterations to $250k$, because the shooting environment of these datasets is consistent and the image style does not need to be transferred. The way to obtain heuristics or binary masks is the same as that of NeRF-W.

Scene	Semantic Label	Text Prompt
Car	*car*	*"Moving red car and its shadow. "*
Cars	*car*	*"Moving red car and its shadow. Moving blue car and its shadow. Moving green car and its shadow. "*
Bag	*bag*	*"Moving bag and its shadow. "*
Chairs	*chair*	*"Moving chairs and their shadows. "*
Pillow	*pillow, cushion*	*"Moving pillow and its shadow. "*

Table 1. **Prior knowledge that semantic and open-set segmentation models use on the Kubric dataset.**

D^{2}NeRF [53]. The model architecture, training settings and the way to obtain heuristics or binary masks are the same as the official codebase.

RobustNeRF [40]. Since the authors only provide the loss function as the official code and claim that RobustNeRF is implemented based on Mip-NeRF 360, we apply the loss function to MultiNeRF codebase [30]. RobustNeRF shares the same model architecture and training settings as Mip-NeRF 360. During the experiment, we find that the color residual threshold (Quantile $= 0.5$) provided in [40] is too low to reconstruct most static objects, so we experiment with different thresholds on the Distractor dataset and find that Quantile $= 0.8$ performs the best (Fig. 12). We therefore apply this threshold uniformly in other experiments. To obtain heuristics or binary masks, we follow the procedure in [40] to process the color residual maps.

7.4. Settings on the Baseline Segmentation Models

DeepLabv3+ [6] and Mask2Former [8]. For the two semantic segmentation models, we use the implementation of MMSegmentation [9] that has been pre-trained on the



(a) Training Images

(b) Test Images

Figure 13. **Examples of training and test images from the Phototourism dataset.** Note that most of the transient distractors (*e.g.*, tourists and trees) in the training images do not block the landmark main bodies (*e.g.*, the attic storey of the Brandenburg Gate and its top sculptural group), which are the focus of the test images.

ADE20K dataset [59]. When evaluating on the Kubric dataset, we segment transient objects instead of static ones because the former have fewer types and are easier to identify. The semantic labels used are shown in Tab. 1.

Grounded-SAM [16, 25]. This open-set segmentation model uses Grounding DINO [25] to locate target objects through natural language and then uses SAM [16] to obtain related segmentation masks. The text prompts used are shown in Tab. 1.

DINO [4]. This model is a pre-trained backbone model that can be used for video segmentation [31]. To segment the target object, DINO requires the object mask in the first video frame as initialization, which we provide in the form of the first ground-truth transient mask.

8. Additional Experiments

8.1. Comparison on the Phototourism Dataset

As shown in Fig. 8 of the main paper, our method brings less improvement on Phototourism compared to other datasets. Similar to [17], we ascribe this to: i) most of the transient distractors in its training set do not disturb the landmark main bodies and are located at the boundary of the landmarks or images (Fig. 13); ii) the content of almost all test images concentrates on the landmark main bodies, meaning that models without any processing of transient distractors can also generate acceptable results when evaluating on the test set (see the rendered images of Nerfacto and Mip-NeRF 360 in Figs. 8 and 24). Therefore, the improvements achieved on this dataset are mostly from addressing the variable appearances to improve the model's rendering quality [7, 20, 24, 39, 51], which is orthogonal to our work.

8.2. Verification of Observation 1

Observation 1 is the basis of our SfM-based heuristics. To verify its correctness, we conduct experiments on the Kubric dataset since it has ground-truth masks that indicate which image regions are transient.

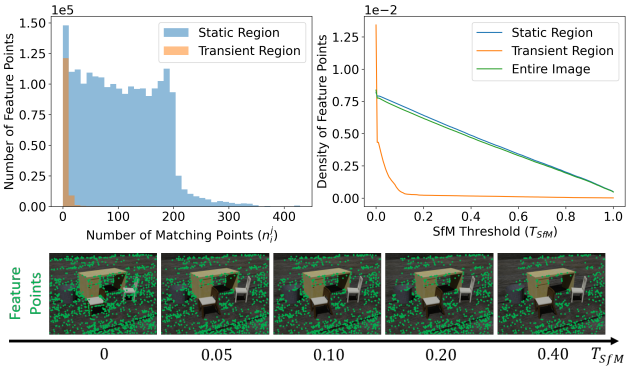


Figure 14. **Verification of Observation 1 on the Kubric dataset.** (top left) The histogram of n_i^j for static and transient regions. (top right) As the threshold $\mathcal{T}_{SfM}$ increases, the density of feature points from transient regions decreases much faster than that in static regions. (bottom) As a result, feature points in transient regions (*e.g.*, chairs) are gradually removed and the static regions are identified at the same time.

Distribution of $\mathbf{n_i^j}$. As Fig. 14 (top left) shows, the n_i^j distributions of static and transient regions are significantly different. Specifically, the n_i^j distribution of the static region is relatively uniform and concentrated in $[0, 200]$; in contrast, the n_i^j distribution of the transient region is significantly biased towards 0 and concentrated in $[0, 10]$. These validate our Observation 1: most SfM features of static objects have much larger n_i^j than those of transient ones.

Effectiveness of $\mathcal{T}_{\mathbf{SfM}}$. Based on Observation 1, we applied a simple yet effective threshold $\mathcal{T}_{SfM}$ to distinguish between static and transient regions. As shown in Fig. 14 (top right), the density of feature points decreases *linearly* as $\mathcal{T}_{SfM}$ increases for both the entire image and static regions, while it decreases *exponentially* for transient regions. When $\mathcal{T}_{SfM} \geq 0.2$, the density of feature points in transient regions is almost 0, meaning that there are almost no remaining feature points in transient regions (Fig. 14, bottom), *i.e.*, the static regions can be well-segmented from the remaining feature points.

8.3. Sensitivity to Hyperparameters

Our method has two hyperparameters: i) $\mathcal{T}_{SfM}$ which is used to filter transient feature points and ii) $\mathcal{T}_{CR}$ which is used as additional insurance to our combined heuristics (Sec. 4.2 of the main paper). Following the main paper, we conduct experiments based on Nerfacto and investigate how the values of $\mathcal{T}_{SfM}$ and $\mathcal{T}_{CR}$ affect the performance of our method.

Sensitivity to $\mathcal{T}_{\mathbf{SfM}}$. As shown in Fig. 15 (left), the average PSNRs first slightly increase and then slightly decrease as $\mathcal{T}_{SfM}$ gradually increases on both the Kubric and Distractor datasets. This indicates that the best $\mathcal{T}_{SfM}$ is achieved when striking a balance between removing transient fea-

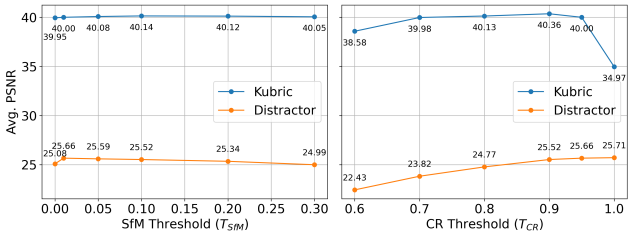


Figure 15. **Sensitivity to $\mathcal{T}_{SfM}$ and $\mathcal{T}_{CR}$ on different datasets.** (left) $\mathcal{T}_{CR}$ is fixed at 0.95 and (right) $\mathcal{T}_{SfM}$ is fixed at 0.01.

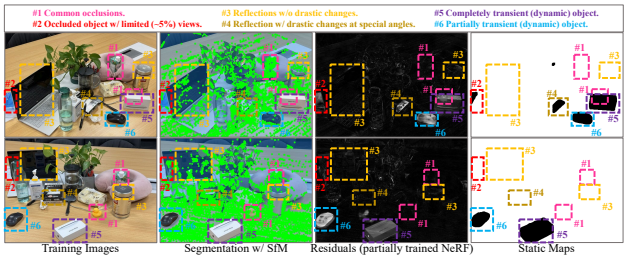


Figure 16. **Visualizations of HuGS in the real-world scene *Room***. Zoom in for a better view.

tures and retaining static features. Nevertheless, the improvement brought about by the choice of $\mathcal{T}_{SfM}$ is up to 0.19dB on the Kubric and 0.58dB on the Distractor dataset, which suggests that even without careful choice of $\mathcal{T}_{SfM}$, our method achieves a performance superior to the state-of-the-art methods.

Sensitivity to $\mathcal{T}_{CR}$. As shown in Fig. 15 (right), consistent with the results of RobustNeRF (Fig. 4 of the main paper and Fig. 12), using a smaller $\mathcal{T}_{CR}$ usually causes the performance to decline as it unnecessarily removes some static objects while using a overly large $\mathcal{T}_{CR}$ (*e.g.*, $\mathcal{T}_{CR} = 1$) makes it ineffective by losing the capability of removing "obvious" transient objects as insurance. These validate our choice of using a relatively large $\mathcal{T}_{CR} = 0.95$ as additional insurance.

8.4. Performance in Challenging Cases

To enable a more in-depth analysis of our method's performance, we introduce a novel real-world scene *Room* with a variety of different materials (*e.g.*, glass, liquid and metal), featuring challenging cases of occlusion, reflection and movement. It contains 64 images. A white box acts as transient distractors that changes its position in every image, and a black mouse is partially dynamic that keep moving in half of the images and static in the other half. As Fig. 16 shows:

1. **Occlusion**. Our method excels in dealing with common occlusions (#1) but struggles with extreme cases (#2) that are rarely visible as they provide few features or supervision signals for SfM matching or NeRF fitting.
2. **Reflection.** Our method handles reflections well (#3) if there are no drastic appearance changes. However, limited views of highly reflective objects may induce such changes (#4), leading to misclassification as transient/static due to larger residuals from unfitted NeRF even with successful SfM matching.
3. **Movement.** Partially dynamic/transient objects are inherently ambiguous, leading to relatively large residuals from unfitted NeRF. For half-dynamic objects (static in 50% views), our method tends to treat them (#6) as transient distractors. The specific outcome, however, depends on the degree of "partially dynamic".

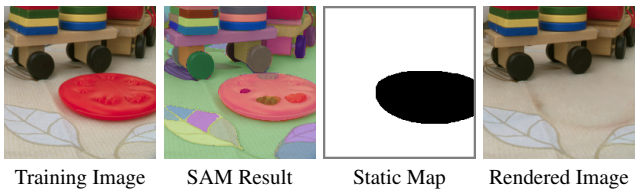


Figure 17. **A failure case.** SAM sometimes fails to segment the shadow around transient distractors (*e.g.*, the red plate in the image) and incorrectly counts it in the static map, resulting in artifacts in the rendered image.

	Training Data	Kubric (Avg.) PSNR↑	SSIM↑	LPIPS↓
(a)	Noisy Images w/ Our Static Maps	40.40	.969	.033
(b)	Noisy Images w/ GT static Maps	40.49	.969	.032
(c)	Clean Images	**40.95**	**.971**	**.031**

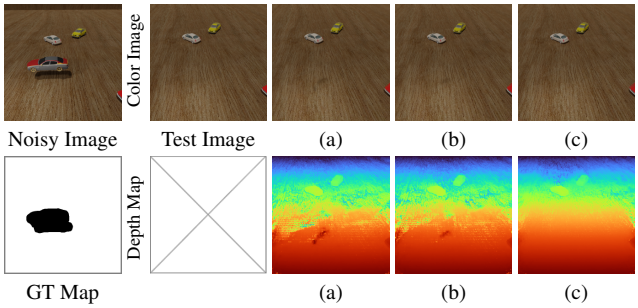


Figure 18. **Performance of Nerfacto with different kinds of training data.** The **best results** are highlighted. Even if ground-truth static maps are provided, simply removing large areas of transient pixels during training may lead to a lack of supervision in some spatial regions and viewing directions, resulting in incorrect densities or biased colors (*e.g.*, floater in the middle).

9. Limitations and Future Work

Our method relies on the effective design of heuristics and the power of the segmentation model used. Despite this, our method is still limited when facing extreme cases in real-world scenes, such as rarely visible static objects and highly reflective materials (Fig. 16). In our work, we used a naive SfM algorithm, which could be further improved to handle cases of sparse views or scenes dominated by tran-

sient distractors. Additionally, although SAM is one of the most advanced segmentation models, it still cannot handle very slight shadows. This limitation results in reduced rendering quality in some cases (Fig. 17). Therefore, utilizing more robust SfM algorithms and/or more capable segmentation models represents a potential path to further improve our method. Furthermore, our method removes the transient pixels using Eq. 4 without additional processing of them. Excessive pixel removal can result in insufficient supervisory signals, potentially degrading the model's performance relative to training on clean images (Fig. 18). Therefore, another possible improvement is to combine our method with 2D/3D inpainting [31, 44, 52, 55], which can predict the missing content caused by removing transient pixels to assist NeRF training.

10. Additional Qualitative Results

Visualization of HuGS. We additionally show the results of our HuGS, including intermediate results and final static maps.

- *Kubric dataset* (Fig. 19). Although there are no SfM features remaining on the shadows of moving objects after filtering, other static features on the ground still guide SAM to segment the entire ground, causing the SfM-based heuristics $\mathcal{H}^{SfM}$ to incorrectly include shadows. As discussed in Sec. 4.2 of the main paper and Sec. 8.3, the presence of $\mathcal{T}_{CR}$ can be additional insurance to remove shadows from combined $\mathcal{H}$ and the final static map.
- *Distractor dataset* (Fig. 20). Consistent with Sec. 4.2 and Fig. 5 of the main paper, the SfM-based heuristics $\mathcal{H}^{SfM}$ alone may miss static objects that only have smooth textures (*e.g.*, white chairs of the Statue scene), while the residual-based heuristics $\mathcal{H}^{CR}$ alone may struggle with high-frequency static details (*e.g.*, box textures of the Statue scene). Their combination $\mathcal{H}$ can roughly identify transient objects from static scenes, and final static maps are generated by SAM to provide accurate static *vs.* transient separation results.
- *Phototourism dataset* (Fig. 21). The results and conclusions are similar to those on the Distractor dataset.

Evaluation on View Synthesis. We show more qualitative results of view synthesis on different datasets.

- *Kubric dataset* (Fig. 22). Existing baseline models, namely Nerfacto and Mip-NeRF 360, exhibit limitations in managing transient distractors, resulting in significant artifacts in their rendered images. Similarly, NeRF-W and HA-NeRF demonstrate inadequacies in segregating transient objects from static scenes, leading to artifact-ridden outputs. D^{2}NeRF shows improved accuracy in separating elements, yet some transient distractors persist within the static scenes. RobustNeRF achieves complete removal of transient distractors using a color residual threshold; however, this results in the inadvertent loss of certain static details. In contrast, our HuGS, applicable across various baseline models, effectively balances the exclusion of transient distractors and retention of static scene details, thereby enhancing the rendering quality.
- *Distractor dataset* (Fig. 23). The results and conclusions are similar to those on the Kubric dataset.
- *Phototourism dataset* (Fig. 24). As discussed in Sec. 8.1, transient distractor management is less critical in the test set evaluation of this dataset. Nevertheless, our approach demonstrates a notable reduction of artifacts (*e.g.*, at the bottom of the Brandenburg Gate) and an enhanced recovery of architectural details (*e.g.*, the fence of the Sacre Coeur), surpassing previous methodologies. This underscores the superiority of our method.

Evaluation on Segmentation. Enhanced qualitative segmentation results are presented in Fig. 25, paralleling the discussions in Sec. 5.3 of the main paper. Existing segmentation methodologies that leverage prior knowledge still exhibit limited efficacy, even with the provision of such information. Conversely, heuristics-based approaches, while broader in applicability, fall short in achieving precise segmentation. Our proposed framework merges heuristics with the segmentation model, thus extracting the strengths of both techniques. This integration allows for the effective separation of transient distractors within static scenes, independently of any prior knowledge.

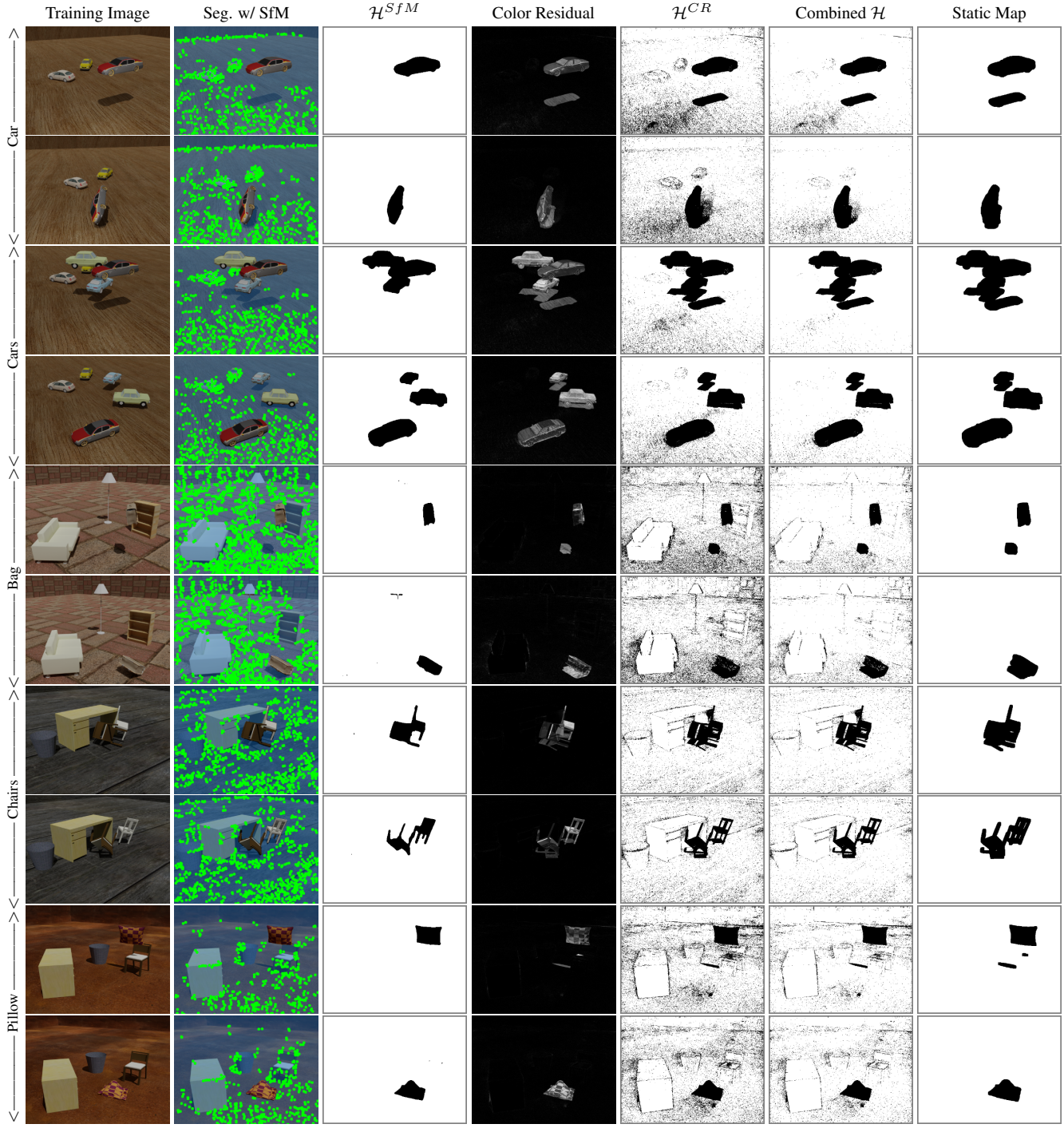


Figure 19. **Visualization of HuGS on the Kubric dataset.**

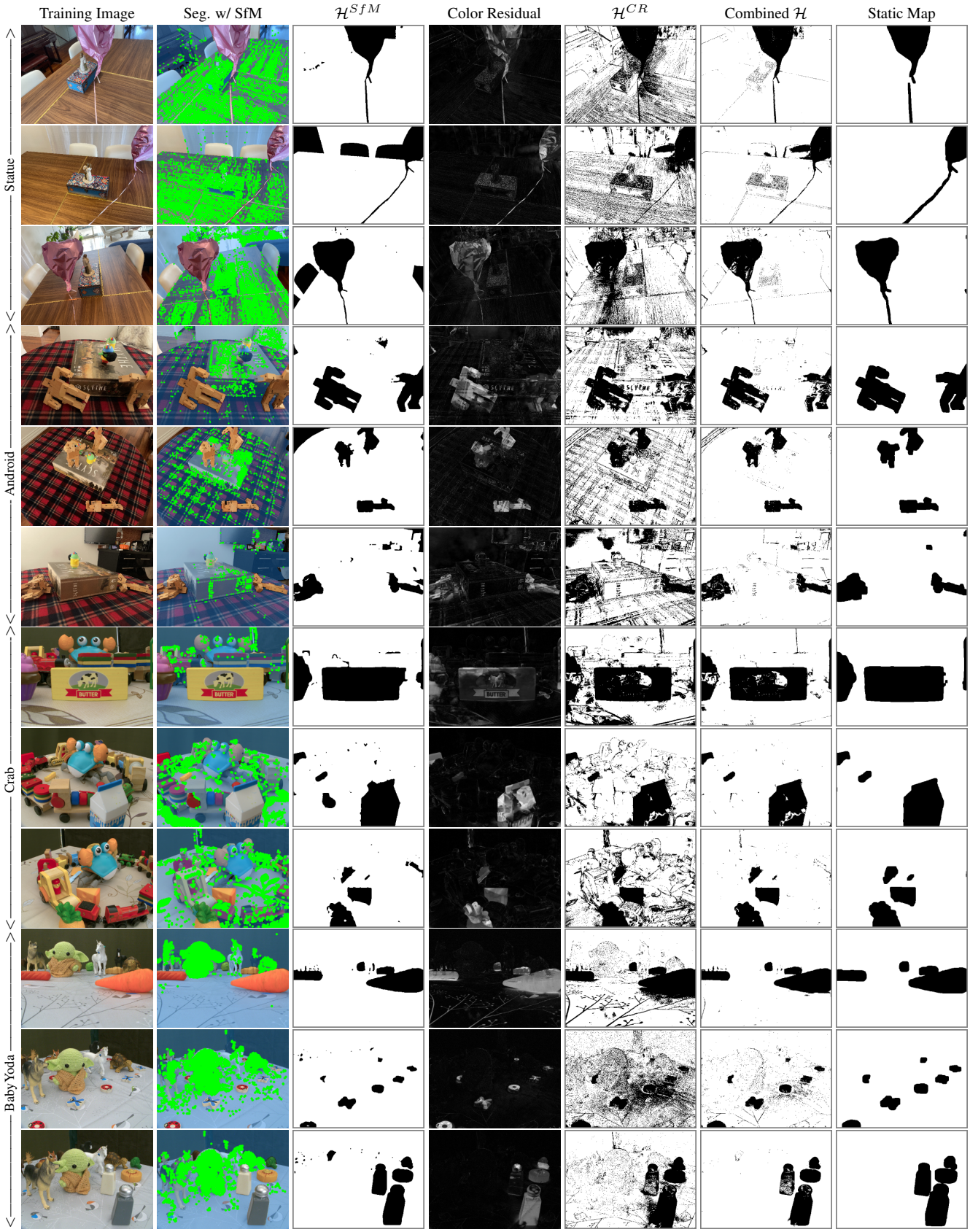


Figure 20. **Visualization of HuGS on the Distractor dataset.**

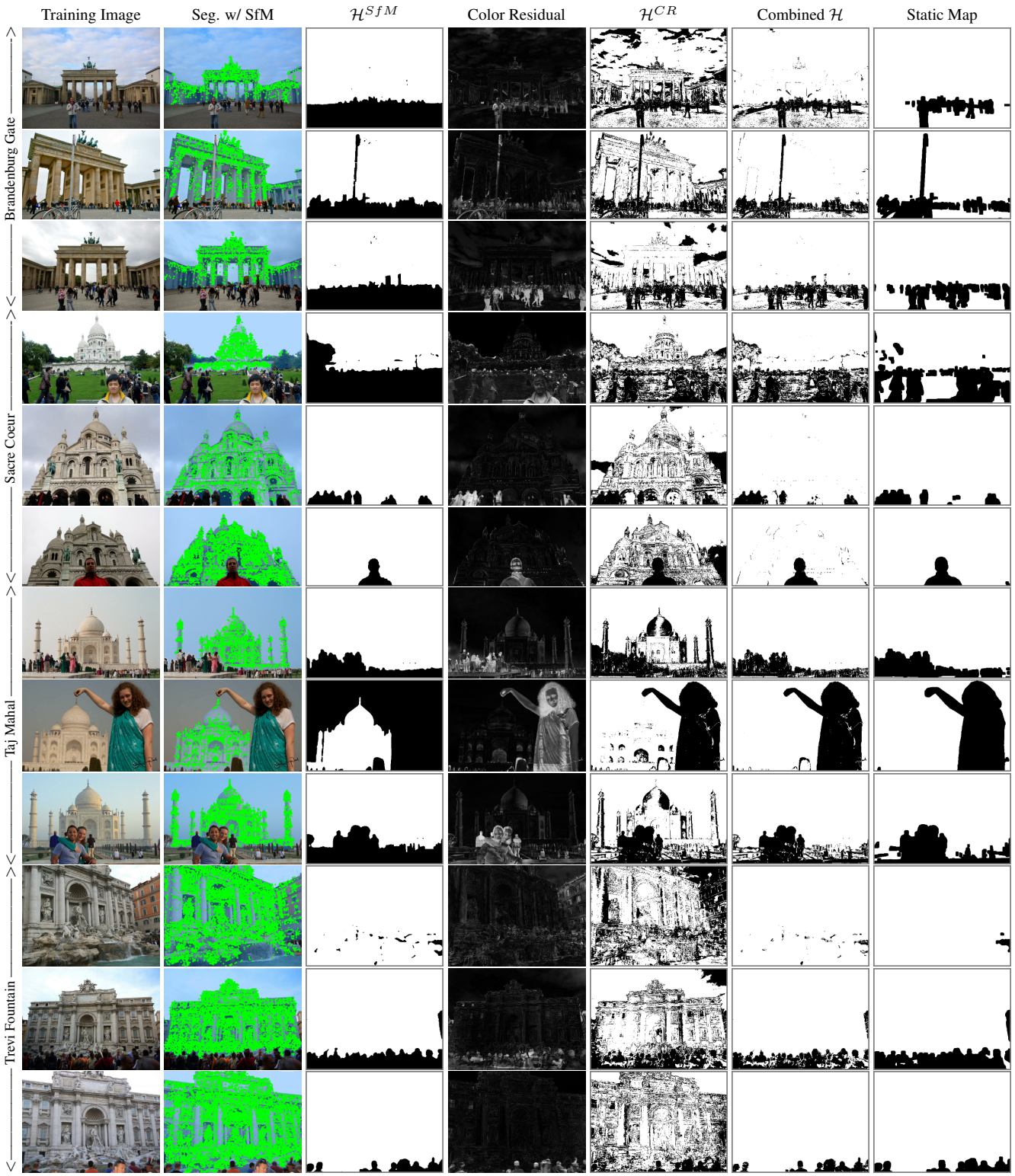


Figure 21. **Visualization of HuGS on the Phototourism dataset.**



Figure 22. **Qualitative results of view synthesis on the Kubric dataset.** Our method can eliminate artifacts from the native baseline models (Nerfacto and Mip-NeRF 360) and preserve more static details (*e.g.*, the ground textures) than those of existing methods.



Figure 23. **Qualitative results of view synthesis on the Distractor dataset.** Our method can better preserve static details (*e.g.*, plants of the Statue and Android scene, table-cloth textures of the Crab and BabyYoda scene) while ignoring transient distractors.

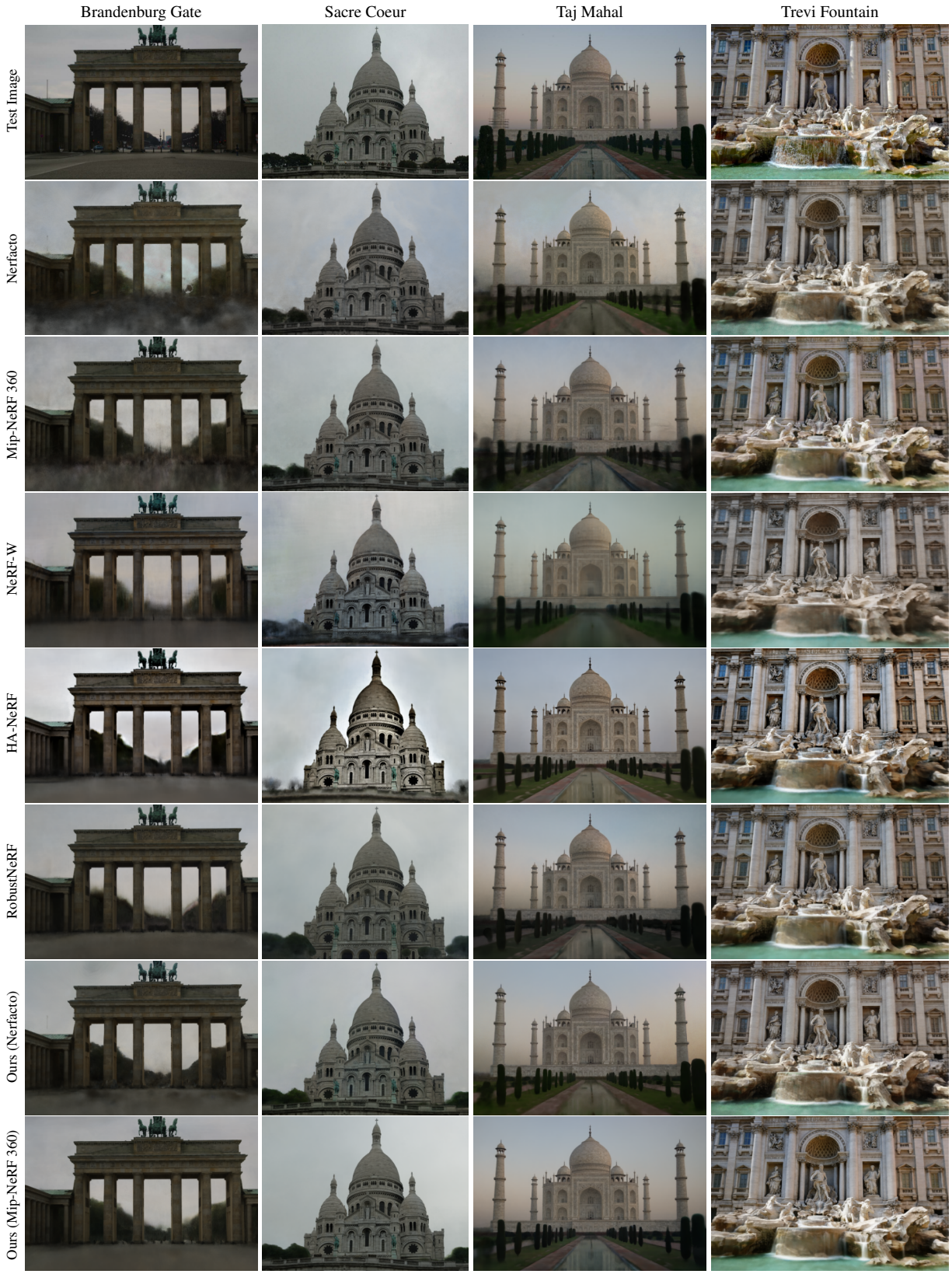


Figure 24. **Qualitative results of view synthesis on the Phototourism dataset.** The native baseline models (Nerfacto and Mip-NeRF 360) can generate acceptable results without any processing of transient distractors. Even so, our method achieves competitive results by eliminating artifacts (*e.g.*, the bottom of the Brandenburg Gate) and recovering more static details (*e.g.*, the fence of the Sacre Coeur).

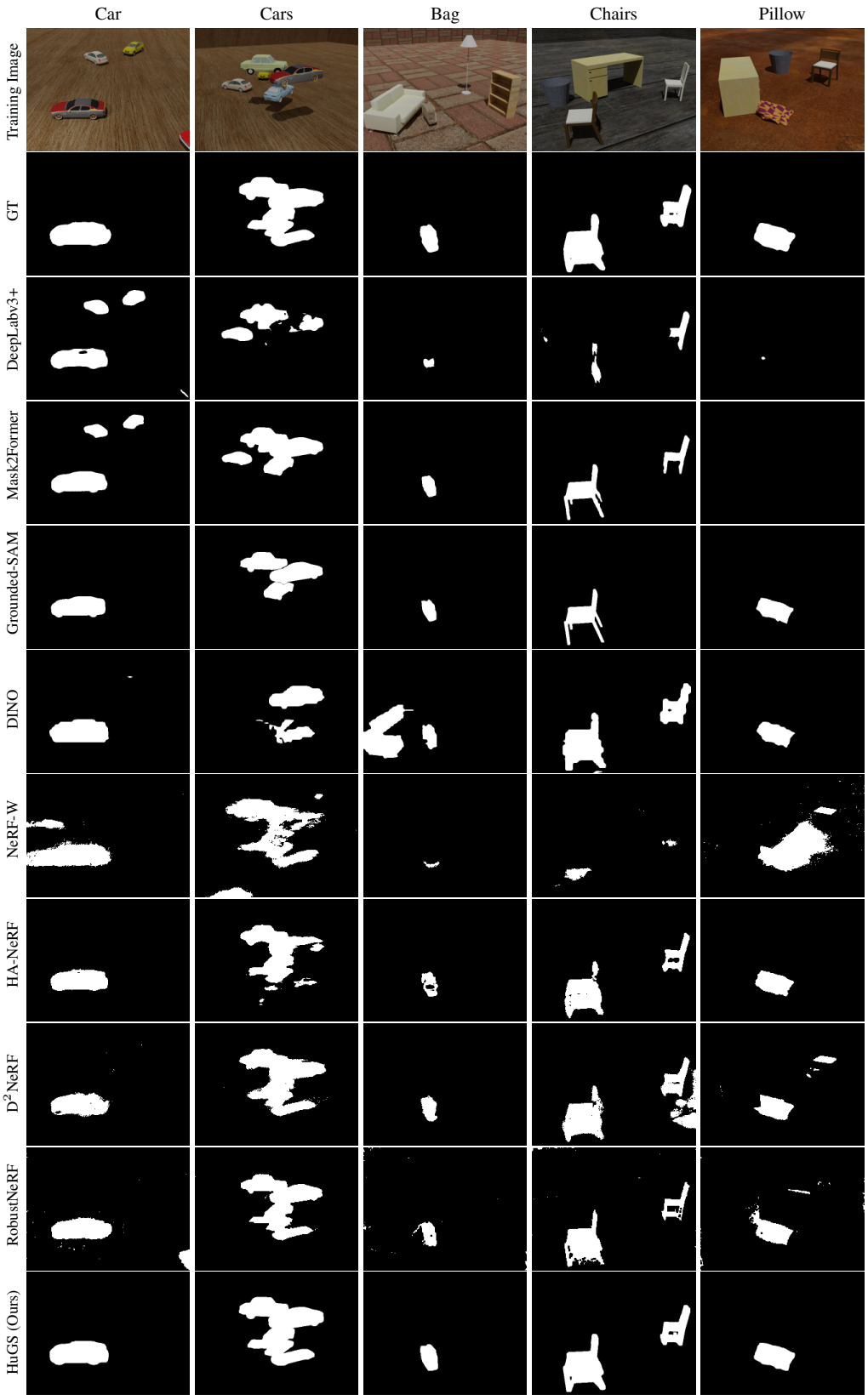


Figure 25. **Qualitative segmentation results on the Kubric dataset.** Compared to existing methods, our method can segment transient objects from static scenes more accurately.