CORE-MPI: Consistency Object Removal with Embedding MultiPlane Image

Donggeun Yoon[1,2] and Donghyeon Cho[3]*
[1]Department of Electronics Engineering, Chungnam National University, Daejeon, South Korea
[2]Korea Electronics Technology Institute (KETI), Seongnam, South Korea
[3]Department of Computer Science, Hanyang University, Seoul, South Korea
ehdrms903@gmail.com, doncho@hanyang.ac.kr

Abstract

Novel view synthesis is attractive for social media, but it often contains unwanted details such as personal information that needs to be edited out for a better experience. Multiplane image (MPI) is desirable for social media because of its generality but it is complex and computationally expensive, making object removal challenging. To address these challenges, we propose CORE-MPI, which employs embedding images to improve the consistency and accessibility of MPI object removal. CORE-MPI allows for real-time transmission and interaction with embedding images on social media, facilitating object removal with a single mask. However, recovering the geometric information hidden in the embedding images is a significant challenge. Therefore, we propose a dual-network approach, where one network focuses on color restoration and the other on inpainting the embedding image including geometric information. For the training of CORE-MPI, we introduce a pseudo-reference loss aimed at proficient color recovery, even in complex scenes or with large masks. Furthermore, we present a disparity consistency loss to preserve the geometric consistency of the inpainted region. We demonstrate the effectiveness of CORE-MPI on RealEstate10K and UCSD datasets.

1. Introduction

The importance of visual content in applications such as social media has grown exponentially, highlighting the need for innovative visual features to enhance user engagement and experience on these platforms. In this digital age, novel view synthesis (NVS) has emerged as a key technology [10, 45] that enables users to experience immersive changes in visual content from different perspectives. However, the presence of unwanted objects, such as personal information, in synthesized images can significantly degrade the user experience, thus object removal techniques are important in the pursuit of smooth and realistic NVS. Recent advancements in deep learning, such as the neural radiance field (NeRF) [26] and multiplane image (MPI) [25, 54], have shown remarkable achievements in NVS. Studies are currently being conducted to improve these techniques for fast [7, 12, 35, 43], and efficient [8] rendering aiming to expand their practical applications. As these methods become more user-friendly, there is a growing interest in removing objects within these 3D representations [27, 48].

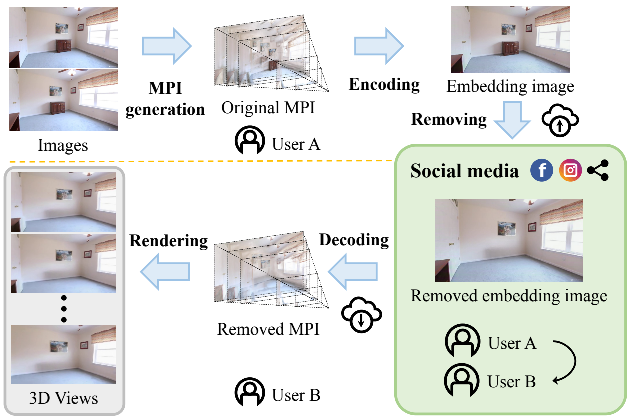

Figure 1. An overview of CORE-MPI: We create an embedding image that encodes MPI data into a commonly used RGB format. The embedding image is transmitted over the Internet, allowing users to interact with it to create a mask for object removal. The embedding image, devoid of these objects, is then converted back into MPI to render novel views without the unwanted objects.

SPIn-NeRF [27] performs object removal from NeRF by generating a 3D object mask based on a viewpoint derived from the trained NeRF model. However, this approach is time-consuming because it requires propagation of the object mask to the other viewpoints to create a 3D mask and requires training the NeRF model twice. To address these issues, recent research [48] has focused on accelerating mask generation. Despite these efforts, they have not

* Corresponding author.

Method	# of views	Time	Results	Requires scene-specific training
SPIn-NeRF [27]	100-200	20-45 minutes	Good	O
Pre-inpaint	2 or more	<1 second	Unstable	X
Post-inpaint	2 or more	<1 second	Poor	X
Layer-inpaint	2 or more	<1 minutes	Unstable	X
CORE-MPI	2 or more	<1 second	Good	X

Table 1. Comparison of object removal methods in novel view synthesis. Pre-inpaint and Layer-inpaint are unstable due to inconsistencies caused by multiple iterations of inpainting.

overcome the intrinsic challenges of NeRF, which requires scene-specific training with large amounts of training data.

MPI, on the other hand, offers a distinct advantage in that it does not require scene-specific training and can represent a 3D scene with few images. Since object removal in MPI remains an unexplored field, we perform preliminary experiments to observe the challenges of naively implemented inpainting methods for MPI, which are summarized in Table 1. First, Pre-inpaint, which removes objects from each input to MPI generator, requires an object mask for each input image. Also, inconsistencies between filled regions prevent the generation of proper MPIs, resulting in broken geometry. Next, Post-inpaint, which uses a modified version of the inpainting network after generating MPI, performs unsatisfactorily due to the high dimensionality of MPI. Finally, Layer-inpaint, which applies the inpainting network to MPI layer by layer, results in inconsistencies between filled layers. Also, it is inefficient because processes should be executed for each MPI depth.

In this paper, we employ steganography, a technique for encoding large amounts of data [31, 45] or hiding information [2], to create embedding images that contain MPI. As shown in Figure 1, we propose **C**onsistency **O**bject **R**emoval with **E**mbedding **M**ulti**P**lane **I**mage (CORE-MPI), which removes unwanted objects on embedding images to enhance consistency and accessibility of MPI object removal. First, we render MPI with the center camera parameters to obtain a reference image that is used to generate MPI. Then, we generate an embedding image that contains both the geometric and color information of MPI. Since the embedding image is a 2D image that is commonly used on social media, it can be transmitted in real-time and the user can directly interact with it to create unwanted object masks. Finally, by reversing the embedding process while maintaining the manipulations, we obtain MPI with the objects removed. This method not only allows us to remove objects with a single mask that corresponds to the object in the embedding image, but also eliminates inconsistency issues because it uses only one inpainting. However, image inpainting network is not designed to fill in hidden information, thus removing objects from the embedding image can degrade both scene content and geometric information.

To address the challenges of color consistency and geometric fidelity in embedding image inpainting, we introduce a dual-network approach: one network is dedicated to color restoration, while the other focuses on embedding image inpainting. The color inpainting network restores the reference image, reducing the blurring effect that occurs in complex scenes or with large masks. Furthermore, we introduce a pseudo-reference loss that uses the output of the pretrained inpainting network as an approximation to guide the inpainting process towards the ground truth. In addition, we propose a disparity consistency loss to specifically supervise the preservation of geometric information, by comparing between disparity maps derived from before and after object removal within MPI.

2. Related Work

Novel View Synthesis. Recent advances in novel view synthesis have utilized several methods, including neural radiance fields (NeRF) [26], multiplane image (MPI) [54]. NeRF has achieved impressive results and has been studied for practical applications such as using fewer images [29, 49], faster rendering [7, 12, 35] and efficiency [8]. As NeRF became more accessible, research into object removal manipulation began [27, 42, 48]. In particular, Weder *et al*. [42] proposed an algorithm for selecting plausibly inpainted images to preserve view consistency inpainted NeRF. SPIn-NeRF [27] integrated mask propagation with depth supervision and perceptual loss to make object removal NeRF user-friendly. Yin *et al*. [48] contributed to reducing time by expediting the mask generation process through multi-view segmentation. Despite these advances, NeRF still requires scene-specific training, which is a significant drawback for real-world applications.

On the other hand, MPI, which was first introduced in StereoMag [54] and renders novel views from two stereo images, can use the trained model for other scenes. MPI has been studied to improve its quality by refining its architecture [34] or increasing the number of input images [11, 25]. Srinivasan *et al*. [34] proposed a 3D convolutional network to expand the range of rendered views. DeepView [11] solved issues such as occlusion and thin objects by changing in the optimization method. Mildenhall *et al*. [25] presented guidelines to help users sample views that enable high-quality view interpolation with the algorithm. Moreover, Nex [43] proposed a hybrid implicit-explicit modeling for real-time rendering. While research has made progress in removing artifacts and improving rendering quality, research on manipulation has been limited due to the complex multi-layer dimensions of MPI.

Steganography. Steganography aims to hide secret information within a carrier, such as images or videos. Traditionally, spatial-based methods such as least significant bits

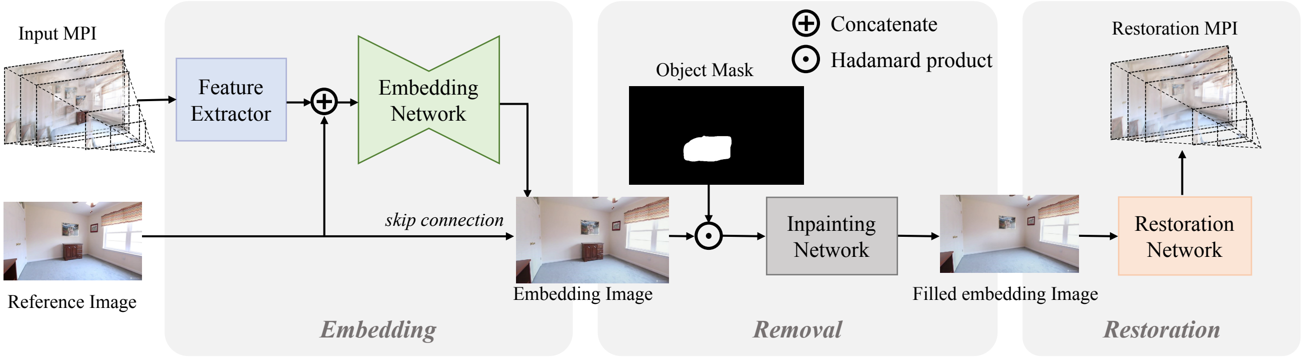

Figure 2. The embedding network takes MPI with a reference image I_r and generates an embedding image I_e which contains geometric information. The inpainting network fills in the removed embedding image with geometric information as well as color. Then, the filled image $\hat{I}_e$ is restored back to MPI by the restoration network.

(LSB) [5, 24], pixel value differences (PDVs) [6, 44], and difference extensions (DE) [16, 38] were commonly used. However, they are susceptible to statistical attacks that can reveal the hidden information. To overcome the shortcomings of traditional steganography, deep learning-based approaches have been developed. These methods strive for imperceptibility and robustness against distortion [47, 56], and increasing the capacity for hidden data [2, 52]. In addition, heavy data such as video [57], high-resolution images [31, 46], and multiplane images [45] are embedded in a light image for transmission and later restored to their original form to facilitate rapid transmission over the platforms such as social media and cloud services. However, these embedding images, which encode a significant amount of information, are sensitive to distortion and manipulation.

Image Inpainting. Image inpainting is a challenging task that reconstructs damaged region in images where information is missing. Traditionally, there are diffusion-based methods [1, 4], which are suitable for small areas by propagating information from adjacent regions, and patch-based methods [3, 17, 36], which work well with repeated patterns but not with unstructured regions. Recently, deep learning approaches, especially generative adversarial networks (GAN) [13], have become mainstream. Pathak *et al*. [30] proposed an inpainting model based on encoder-decoder structure with GAN to recover squared empty regions. Liu *et al*. [20] introduced a partial convolutional layer for irregular random holes, and Yu *et al*. [50] proposed a gated convolutional layer that updates according to the shape of the mask, suitable for arbitrary masks. After that, there have been various attempts to enhance the performance of the inpainting in [21, 22, 28, 32, 37]. Ren *et al*. [32] introduced a two-stage strategy for image inpainting, focusing sequentially on the structure restoration followed by texture. Conversely, Liu *et al*. [22] developed a mutual encoder-decoder framework that performs texture and structure inpainting concurrently in a single step. LaMa [37] utilized Fourier convolutions for large receptive fields, thereby enhancing both the perceptual quality of the inpainting and the efficiency of the model parameters. TransFill [55] employed a clean image for reference image inpainting, while more recently, ViT [9] has been used for image inpainting [18, 39, 51].

However, inpainting for MPI is still an unexplored research area and it is also not designed for use with steganography. In this paper, we combine these two unexplored fields for the first time and propose an inpainting method for the embedding image created from MPI using steganography techniques.

3. Method

As shown in Figure 2, **C**onsistency **O**bject **R**emoval with **E**mbedding **M**ulti**P**lane **I**mage (CORE-MPI) consists of an embedding network, inpainting network, and restoration network. We use an embedding image that encodes MPI into a common RGB image to not only remove objects, but to make MPI more accessible. This section provides background knowledge on MPI in Section 3.1 and introduces a novel method for consistent object removal in MPI through the embedding image space. The overall pipeline of CORE-MPI is structured in three steps: (1) embedding MPI into a single RGB image (Section 3.2), (2) object removal within the embedding image (Section 3.3), and (3) MPI restoration from the embedding image (Section 3.4).

3.1. Preliminary: MPI

MPI consists of a set of fronto-parallel RGBα planes, uniformly sampled in depth within a view of the reference camera frustum. To generate MPI, we adopt StereoMag [54], utilizing two images with calibrated camera parameters.

Mathematically, given the images I_1 and I_2 with camera parameters c_1 and c_2, MPI generation operates as follows:

$$\mathbf{F}(I_1, I_2, c_1, c_2) \rightarrow (C, \alpha), \tag{1}$$

where C and α are color images and alpha maps, respectively, with dimensions $w \times h \times 3 \times n$ and $w \times h \times n$. Here, w and h represent the width and height respectively, while n denotes the number of depth layers, with n set to 32 in our experiments. MPI allows for the synthesis of novel-view images through planar transformation and alpha-composition techniques, as detailed in [54]. The disparity map D is synthesized from the alpha maps using the following formula:

$$D = \sum_{i=1}^{n} (d_i^{-1} \alpha_i \prod_{j=i+1}^{n} (1 - \alpha_j)), \tag{2}$$

where d_i is the inverse depth value of each layer. Note that our method can be integrated not only with StereoMag but also with other MPI generation models.

3.2. Embedding Image Generation

Given a generated MPI, the embedding process starts with the acquisition of a reference image I_r, which is the basis for MPI embedding. By rendering MPI with central camera parameters, we can obtain I_r that has the same viewpoint as the one used to create MPI. It is a suitable candidate for embedding image that encodes MPI information because it has color information of MPI. Then, a feature extractor produces the crucial information from each MPI layer. The extracted features for the color and alpha channels, denoted as f_c and the alpha feature f_α, respectively, are derived from separate convolutional layers. These features are then combined by element-wise multiplication with the alpha values to form MPI feature f_{MPI}, as shown in the equation:

$$f_{MPI} = (\alpha \odot f_C) \oplus f_\alpha, \tag{3}$$

where, $\oplus$ is concatenation and $\odot$ represents the Hadamard product. After that, the embedding network based on the U-Net architecture incorporates this f_{MPI} and I_r to make the embedding image I_e, by facilitating the compression of geometric information in the image residuals through a skip connection. To optimize the embedding network, we use an embedding loss $\mathcal{L}_e$ that includes the mean squared error (MSE) and the perceptual loss for high-level feature similarity, as expressed in the following equation:

$$\mathcal{L}_e = \lambda_{er} \parallel I_r - I_e \parallel^2 + \lambda_{ep} \parallel \phi(I_r) - \phi(I_e) \parallel, \tag{4}$$

where λ_{er} and λ_{ep} are weight terms, and ϕ corresponds to a pretrained VGG-19 [33], promoting the preservation of perceptual features crucial for color reproduction in I_e.

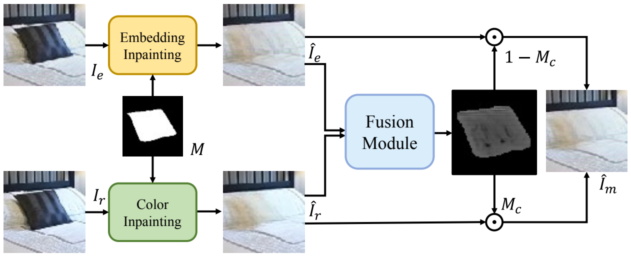

Figure 3. Structure of the fusion process for embedding image inpainting. The fusion module estimates combining ratios to merge the restored reference and embedding images.

3.3. Object Removal from Embedding Image

The embedding image containing geometric information is in the RGB format, which is commonly used in social media. This format supports user-friendly interaction, enabling the creation of masks M over specific areas for the removal of objects within I_e. Once the areas with undesired objects are masked, image inpainting network is utilized to restore the masked regions. However, the image inpainting network is designed to recover the color of the image, not to reconstruct hidden information. As a result, it is limited in its ability to revive images that contain hidden information, especially when masks are large and scenes have complex elements. Therefore, we adopt dual-network approach: a color and embedding image branches. Both branches employ the same inpainting model, LaMa [37]. The color inpainting branch fills in the reference image, while the embedding inpainting branch focuses on the hidden geometric information in the embedding image. To integrate the inpainted reference image $\hat{I}_r$ with the inpainted embedding image $\hat{I}_e$, a fusion module is utilized. The fusion module estimates the combining ratios for the inpainted images, as shown in Figure 3. Then, two images are fused as follows:

$$\hat{I}_m = \hat{I}_r \cdot M_c + \hat{I}_e \cdot (1 - M_c), \tag{5}$$

where $\hat{I}_m$ is the merged image and M_c is the combining ratios with values ranging from 0 to 1. By M_c, $\hat{I}_r$ and $\hat{I}_e$ are smoothly combined, ensuring $\hat{I}_m$ maintains visual coherence and preserves restored color and geometric details.

For color inpainting branch training, we introduce a pseudo-reference loss function. Pseudo-reference loss uses the output of pretrained inpainting network as a pseudo-ground truth for easier color recovery:

$$\mathcal{L}_{pse} = \parallel \hat{I}_r - \hat{I}_{pse} \parallel^2 . \tag{6}$$

Here, $\hat{I}_{pse}$ is the result of applying a pretrained inpainting model to I_r with M. Since estimate $\hat{I}_r$ is made easier in comparison to predicting I_r, it enables stable training for the inpainting network. In parallel, the embedding inpainting branch is trained with combination loss functions as:

$$\mathcal{L}_f = \lambda_{fr} \parallel I_e - \hat{I}_e \parallel^2 + \lambda_{fp} \parallel \Phi(I_e) - \Phi(\hat{I}_e) \parallel, \quad (7)$$

where λ_{fr}, and λ_{fp} are weight terms, and Φ corresponds to segmentation ResNet50 with dilated convolutions used in [37]. To improve the quality of the inpainted textures and the coherence of the geometry, we integrate a discriminator D_ξ that considers the geometry in the scene. The discriminator and generator losses are defined as follows:

$$\begin{aligned} \mathcal{L}_D = -\log D_\xi(I_e, D) - [\log D_\xi(\hat{I}_e, \bar{D}) \odot M] \\ -[\log(1 - D_\xi(\hat{I}_e, \bar{D})) \odot (1 - M)], \end{aligned} \quad (8)$$

$$\mathcal{L}_G = -\log D(\hat{I}_e, \bar{D}), \quad (9)$$

$$\mathcal{L}_{Ad} = \mathcal{L}_D + \mathcal{L}_G. \quad (10)$$

Here, D and $\bar{D}$ are disparity maps synthesized from the original MPI and the restoration MPI, respectively. The function of the discriminator is not limited to discriminating between the original and inpainted embedding images; it also evaluates geometric consistency, certifying that the restored content appears faithful and accurately matches the geometry of the scene. Following [37], we incorporate gradient penalty $\mathcal{L}_{gp}$ [23] and feature matching loss $\mathcal{L}_{fm}$ [40] into final inpainting loss $\mathcal{L}_{inp}$.

$$\mathcal{L}_{gp} = ||\nabla D_\xi(I_e)||^2, \quad (11)$$

$$\mathcal{L}_{fm} = ||D^i_\xi(I_e, D) - D^i_\xi(\hat{I}_e, \bar{D})||, \quad (12)$$

$$\mathcal{L}_{inp} = \mathcal{L}_f + \lambda_{Ad}\mathcal{L}_{Ad} + \lambda_{pse}\mathcal{L}_{pse} + \lambda_{gp}\mathcal{L}_{gp} + \lambda_{fm}\mathcal{L}_{fm}, \quad (13)$$

where D^i_ξ denotes ith layer of D_ξ, and λ_{Ad} λ_{pse}, λ_{gp}, and λ_{fm} are weight terms.

3.4. MPI Restoration from Embedding Image

The restoration network is designed to reverse the embedding process, converting the embedding image back to MPI representation. Importantly, this transformation ensures that manipulations, such as object removal, are preserved; objects removed from the embedding images are not present in the restoration MPI. The network is trained using an adaptive MSE to MPI $\mathcal{L}_{MPI}$, which is expressed as:

$$\mathcal{L}_{MPI} = \lambda_c \parallel C \odot \alpha - \bar{C} \odot \bar{\alpha} \parallel^2 + \lambda_\alpha \parallel \alpha - \bar{\alpha} \parallel^2, \quad (14)$$

where, λ_c, and λ_α are weight terms, and C and α are the colors and alpha maps of original MPI, and $\bar{C}$ and $\bar{\alpha}$ are the colors and alpha maps of restoration MPI. $\mathcal{L}_{MPI}$ is instrumental in ensuring that the color and alpha channels in the restored MPI are consistent with the original, pre-manipulation state. In addition, a loss function for maintaining visual consistency in novel views $\mathcal{L}_{Nov}$ is implemented:

$$\mathcal{L}_{Nov} = \lambda_{nr} \parallel I_n - \bar{I}_n \parallel^2 + \lambda_{np} \parallel \phi(I_n) - \phi(\bar{I}_n) \parallel_1, \quad (15)$$

where, λ_{nr}, and λ_{np} are weight terms, and I_n and $\bar{I}_n$ are rendered images from original MPI and restoration MPI with randomly generated camera parameters.

We also introduce disparity consistency loss, which supervises synthesized disparity maps for improving the geometric restoration for inpainted region.

$$\mathcal{L}_{dr} = \parallel D - \bar{D} \parallel^2, \quad (16)$$

where D and $\bar{D}$ are the disparity maps from the original and restoration MPI. Furthermore, we incorporate an edge-aware smoothness to promote disparity smoothness while respecting image edges:

$$\mathcal{L}_{dsm} = |\partial_x \bar{D}| e^{-|\partial_x I_r|} + |\partial_y \bar{D}| e^{-|\partial_y I_r|}. \quad (17)$$

The final disparity consistency loss is as follows:

$$\mathcal{L}_{dc} = \lambda_{dr}\mathcal{L}_{dr} + \lambda_{dsm}\mathcal{L}_{dsm}, \quad (18)$$

where, λ_{dr}, and λ_{dsm} are weight terms. As a result, the total loss for the restoration network is as follows:

$$\mathcal{L}_R = \mathcal{L}_{MPI} + \mathcal{L}_{Nov} + \mathcal{L}_{dc}. \quad (19)$$

4. Experiments

4.1. Experimental Settings

Datasets. We use the RealEstate10K [54] dataset, which is commonly used for MPI. In addition, we use UCSD [19] with human mask and background pairs that can be used as target masks and ground truth data.

- **RealEstate10K**: About 80,000 video clips, derived from 10,000 YouTube videos featuring various interior and exterior scenes are provided. It provides a substantial volume of data, including 10 million frames with corresponding camera parameters. For our purposes, frames are cropped to a standardized resolution of 512 × 512. Our training set consists of 62,184 clips, while the test set includes 1,500 clips and generated free-form masks [50].
- **UCSD**: There are 96 dynamic multi-view videos captured in outdoor settings with 10 synchronized action cameras focusing on human subjects. It includes human masks and background images, making it ideal for removal within MPI. The dataset is divided into 86 training videos and 10 test videos, each with a frame size of 640 × 360. To avoid dynamic backgrounds in our evaluation, we select 7 test videos and further choose 100 clips from each, amounting to 700 clips for our test set. To mitigate human-related artifacts like shadows, we apply a 5 × 5 kernel dilation two times to refine the human masks.

Methods	RealEstate10K [54]						UCSD [19]					
	Render				Disparity		Render				Disparity	
	PSNR↑	SSIM↑	LPIPS↓	FID↓	L1↓	L2↓	PSNR↑	SSIM↑	LPIPS↓	FID↓	L1↓	L2↓
Pre-inpaint	27.948	0.923	0.0880	77.956	0.1180	0.1670	36.838	0.976	0.0290	46.161	**0.0141**	0.0219
Post-inpaint	26.992	0.911	0.1030	95.039	0.0610	0.0870	35.912	0.973	0.0810	59.681	0.0154	0.0207
Layer-inpaint	28.699	0.936	0.0710	65.873	**0.0410**	**0.0560**	36.676	0.974	0.0313	51.483	0.0201	0.0251
Embedding-Base	28.807	0.935	0.0774	80.254	0.0680	0.0880	37.477	0.981	0.0282	58.810	0.0196	0.0266
Embedding-Disparity	28.821	0.936	0.0746	76.518	0.0467	0.0674	37.431	0.981	0.0279	56.225	0.0159	0.0216
Embedding-Guide	29.756	0.942	0.0632	62.813	0.0504	0.0696	38.310	0.983	0.0223	43.922	0.0167	0.0228
CORE-MPI	**29.790**	**0.943**	**0.0628**	**59.616**	0.0459	0.0648	**38.367**	**0.983**	**0.0220**	**43.829**	0.0149	**0.0205**

Table 2. Quantitative comparison on RealEstate10K and UCSD datasets. Render evaluates the rendered view, while Disparity evaluates the difference in the disparity map for scene consistency. The best performance is highlighted in **bold** and the second best in underline.

Implementation Details. In our implementation, we generate MPI from pairs of images using StereoMag [54]. We then use LaMa [37] as our inpainting network with free-form masks during the training. For stable training, the training procedure is divided into two stages: In the first stage, we focus on training the embedding and restoration networks. This phase involves over 50,000 iterations with a batch size of 15, using the Adam optimizer with a fixed learning rate of $4e^{-5}$. In the second stage, we train the inpainting and restoration networks while keeping the embedding network fixed. This stage also consists of 50,000 iterations. The learning rates are set to $4e^{-4}$ for the inpainting network and $1e^{-4}$ for the discriminator.

Baselines. Since our work is the first attempt for object removal in MPI, we have established several baselines that focus on the method of object removal in MPI scenario. To demonstrate the superior performance of our proposed components, we compare these baselines against various ablations of CORE-MPI.

- **Pre-inpaint**: Objects are removed from each image before MPI is generated. MPI is then generated from the recovered images with the inpainting model.
- **Post-inpaint**: Objects are removed by applying a channel-extended inpainting model directly to MPI.
- **Layer-inpaint**: After MPI is generated, to remove the target object, an inpainting model is applied separately to each layer for both color and alpha of MPI.
- **Embedding-Base**: Embedding-Base conducts object removal by encoding MPI into an embedding image. For training, a combination of embedding, inpainting, novel view, and MPI losses are used.
- **Embedding-Disparity**: Building on Embedding-Base, this version integrates disparity consistency loss to improve scene consistency.
- **Embedding-Guide**: An extension of Embedding-Disparity, this method introduces the pseudo-reference loss to the embedding image inpainting network.
- **CORE-MPI**: Building on Embedding Disparity, CORE-MPI integrates a color inpainting branch and adopts the pseudo-reference loss for its training. A fusion module is then used to combine the recovered images.

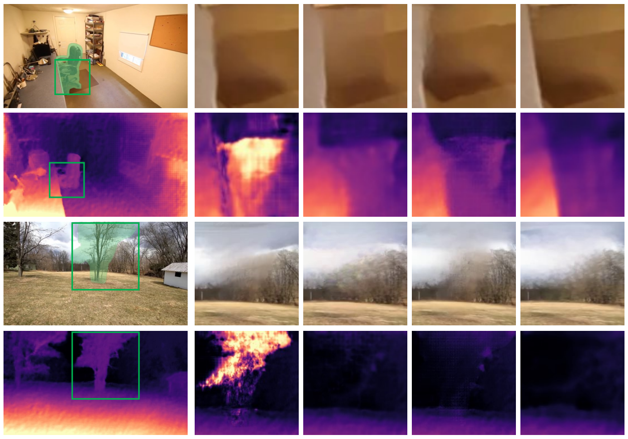

Figure 4. Comparison of CORE-MPI with baselines on the Realestate10K dataset. For each scene, the first row is the rendered view and the second is the disparity map.

Evaluation Metrics. Following [45], we assess the rendering quality by averaging the results from nine rendered views. The evaluation metrics include peak signal-to-noise ratio (PSNR) [15], structural similarity index measure (SSIM) [41], learned perceptual image patch similarity (LPIPS) [53], Frechet inception distance (FID) [14]. In addition, we measure the L1 and L2 errors of the disparity map to evaluate the consistency of the rendered views.

4.2. Experimental Results

Quantitative Comparison. Table 2 presents our comparative analysis, showing quantitative results of the proposed CORE-MPI against established baselines on the RealEstate10K and UCSD datasets. On the RealEstate10K dataset, we observe notable differences between the meth-

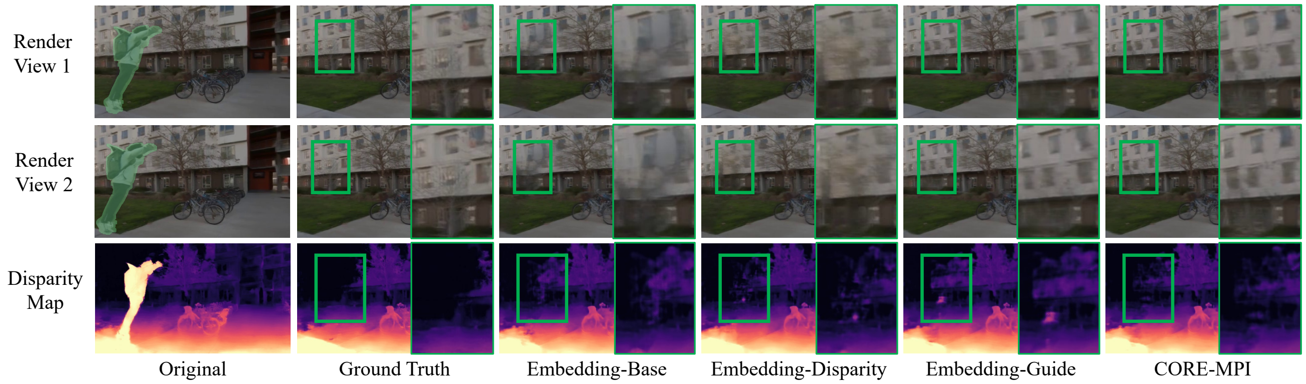

Figure 5. Visualization of ablation experiment results of embedding methods on UCSD. The first and second rows are rendered views from different viewpoints, and the third row shows the disparity map.

ods, particularly in the PSNR of the rendered view. Among the baselines, Post-inpaint has the lowest performance, while Layer-inpaint leads, indicating its effectiveness in rendering. Despite a higher PSNR, Embedding-Base does not perform as well as Layer-inpaint in perceptual metrics such as LPIPS and FID. The Embedding-Disparity shows an improvement in disparity map accuracy, which improves the quality of the rendered view. The Embedding-Guide achieves significant rendering performance improvements, but slightly reduces disparity accuracy compared to Embedding-Disparity. CORE-MPI shows comprehensive improvements in all six evaluation metrics, which can be attributed to the improved disparity map quality.

On the UCSD dataset, the trends mirror those seen on RealEstate10K, with Pre-inpaint performing best on the L1 error of the disparity map. However, Pre-inpaint performs relatively poorly on L2 error, indicating the presence of outliers. In contrast, Post-inpaint shows a better L2 error, showing that it is robust to outliers. Apart from the L1 error, CORE-MPI outperforms all other methods, which confirms the robustness of our approach.

Qualitative Comparison. Figure 4 qualitatively compares results of the baselines and our method on RealEstate10K data. The first example shows the result of removing a chair from a room. Pre-inpaint, which removes the object prior to MPI generation, causes inconsistencies between the inpainted scenes, disrupting the disparity map and ruining the rendered view. Post-inpaint struggles with color restoration, leading to a blurred output where the distinction between the floor and carpet is lost. Layer-inpaint hard to distinguish the line between the desk and the floor, while CORE-MPI recovers the desk and floor with a clear distinction. The second example shows the result of removing a tree from an outdoor scene. Here, the removed region is large and complex, making it difficult for both Pre-inpaint and Layer-inpaint, which require multiple feedforwards of the inpainting model. Pre-inpaint fills the sky differently in the input image, and Layer-inpaint restores the tree background area differently in each layer, resulting in blurry rendering views. These experiments demonstrate the limitations of applying multiple feedforwards of the inpainting model over region and complex scenes. However, CORE-MPI, which requires inpainting only once, produces a plausibly filled result without these inconsistencies.

Furthermore, Figure 5 shows visual comparisons embedding-based methods and CORE-MPI on the UCSD dataset. Embedding-base recovers the removed region as blurred, and poorly recovers the window behind it. With disparity supervision, Embedding-Disparity achieves a noticeable improvement in the reconstructed disparity map. However, the rendering results still appear confused. In contrast, Embedding-Guide succeeds in restoring the color and creating a window where the person has been removed, but it is restored at the same disparity as the adjacent tree, thus the window is incorrectly positioned in the rendered view. Our complete method, CORE-MPI, significantly corrects these geometric inaccuracies, producing results that are consistent across rendered views.

4.3. Ablation Study

LLFF MPI generator. We mainly use StereoMag to generate MPI, but CORE-MPI is not limited to a specific MPI generator. We experiment with LLFF model [25], which uses five images to generate MPIs and performs novel view synthesis using multiple MPIs. Since there are no object-free images in the LLFF dataset, we use free-form masks. Pre-inpaint generates MPI poorly due to inconsistencies in the inpainted images, and Table 3 shows the comparison with other baselines. Post-inpaint shows a significant performance drop due to the accumulation of degradation in a single MPI when rendering a novel view. CORE-MPI performs the best on all four metrics. Furthermore, Fig-

Method	Render PSNR↑	SSIM↑	LPIPS↓	FID↓
Post-inpaint	25.048	0.9190	0.0808	128.4170
Layer-inpaint	34.256	0.9623	0.0293	27.0671
CORE-MPI	**34.813**	**0.9637**	**0.0292**	**25.2438**

Table 3. Quantitative results on LLFF dataset with LLFF model.

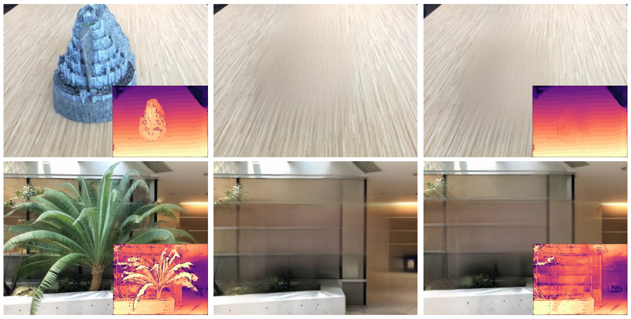
Original Removed View 1 Removed View 2

Figure 6. Visualization of object removal from LLFF using CORE-MPI, showing the rendered views and the disparity map.

ure 6 shows that CORE-MPI produces plausible results for synthesizing novel views after object removal in the LLFF dataset. This experiment demonstrates that CORE-MPI can use a variety of MPIs.

Comparison to SPIn-NeRF. NeRF achieves impressive results in novel view synthesis, and to improve its usability, SPIn-NeRF [27] has explored NeRF-based object removal. To show the strengths of MPI, we compare CORE-MPI with SPIn-NeRF. Table 4 shows the object removal results for each of the 10 scenes in the SPIn-NeRF dataset. CORE-MPI demonstrates its effectiveness by outperforming SPIn-NeRF on all scenes, even though the same inpainting model [37] is used. Figure 7 visualizes these experimental results, providing a clear comparative illustration of the performance. Note that, we use a pretrained model on RealEstate10K without the need for further training. A significant advantage of CORE-MPI is that it does not require per-scene training and offers fast inference speeds, making it practical for real-world applications.

5. Conclusion

In this paper, we present Consistency Object Removal with Embedding MultiPle Image (CORE-MPI) for object removal in multiplane image. CORE-MPI effectively solves the challenges associated with high-dimensional data, including computational cost and inconsistencies within inpainted regions, by operating within the embedding image space. To generate plausible results, we introduce a novel pseudo-reference loss that uses a pretrained inpainting network to create a pseudo-ground truth. Furthermore, we propose a disparity consistency loss to improve consistency. We establish baselines for object removal in MPI and validate CORE-MPI on the RealEstate-10k and UCSD datasets. CORE-MPI is also compared to NeRF-based object removal, demonstrating the performance benefits and the advantages of using MPI. We anticipate that CORE-MPI will provide a cornerstone for novel view synthesis applications, particularly in social media.

Dataset	SPIn-NeRF LPIPS↓	FID↓	CORE-MPI LPIPS↓	FID↓
1	0.2471	377.02	0.1071	296.31
2	0.4379	375.73	0.1635	343.95
3	0.2865	327.83	0.1836	131.33
4	0.2168	178.75	0.1464	163.96
7	0.1733	106.41	0.0961	90.91
9	0.2834	322.60	0.1192	175.24
10	0.2054	88.52	0.0945	57.01
12	0.3269	271.45	0.1798	122.67
book	0.1586	150.30	0.1022	108.94
Trash	0.2152	179.48	0.1055	71.11
Total	0.2551	184.90	**0.1300**	**110.74**

Table 4. Quantitative results of object removal on SPIn-NeRF dataset. For an in-depth evaluation, we provide a dataset-specific comparison with SPIn-NeRF.

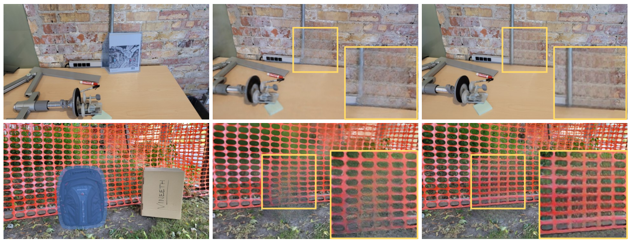
Sample view & Mask SPIn-NeRF CORE-MPI

Figure 7. Qualitative comparison with SPIn-NeRF.

Limitation and Future work Even though CORE-MPI utilizes the dual-network approach and disparity loss for preserving geometric information, it does not explicitly consider disparity within the inpainting network itself. In the future, we plan to develop a novel inpainting model designed to inherently restore geometric information, outperforming the current dual-network strategy for hidden information.

Acknowledgement

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (2022-0-00022, Development of immersive video spatial computing technology for ultra-realistic metaverse services).

References

[1] Coloma Ballester, Marcelo Bertalmio, Vicent Caselles, Guillermo Sapiro, and Joan Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE TIP*, 10(8): 1200–1211, 2001. 3

[2] Shumeet Baluja. Hiding images within images. *IEEE TPAMI*, 42(7):1685–1697, 2019. 2, 3

[3] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM TOG*, 28(3):24, 2009. 3

[4] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *ACM SIGGRAPH*, pages 417–424, 2000. 3

[5] Chi-Kwong Chan and Lee-Ming Cheng. Hiding data in images by simple lsb substitution. *PR*, 37(3):469–474, 2004. 3

[6] Ko-Chin Chang, Chien-Ping Chang, Ping S Huang, and Te-Ming Tu. A novel image steganographic method using tri-way pixel-value differencing. *Journal of multimedia*, 3(2), 2008. 3

[7] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *ECCV*, pages 333–350. Springer, 2022. 1, 2

[8] Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In *CVPR*, pages 16569–16578, 2023. 1, 2

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 3

[10] Zhiwen Fan, Panwang Pan, Peihao Wang, Yifan Jiang, Hanwen Jiang, Dejia Xu, Zehao Zhu, Dilin Wang, and Zhangyang Wang. Drag view: Generalizable novel view synthesis with unposed imagery. *arXiv preprint arXiv:2310.03704*, 2023. 1

[11] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *CVPR*, pages 2367–2376, 2019. 2

[12] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *ICCV*, pages 14346–14355, 2021. 1, 2

[13] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 3

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 30, 2017. 6

[15] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *ICPR*, pages 2366–2369. IEEE, 2010. 6

[16] Yongjian Hu, Heung-Kyu Lee, Kaiying Chen, and Jianwei Li. Difference expansion based reversible data hiding using two embedding directions. *IEEE TMM*, 10(8):1500–1512, 2008. 3

[17] Joo Ho Lee, Inchang Choi, and Min H Kim. Laplacian patch-based image synthesis. In *CVPR*, pages 2727–2735, 2016. 3

[18] Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. Mat: Mask-aware transformer for large hole image inpainting. In *CVPR*, pages 10758–10768, 2022. 3

[19] Kai-En Lin, Lei Xiao, Feng Liu, Guowei Yang, and Ravi Ramamoorthi. Deep 3d mask volume for view synthesis of dynamic scenes. In *ICCV*, pages 1749–1758, 2021. 5, 6

[20] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *ECCV*, pages 85–100, 2018. 3

[21] Hongyu Liu, Bin Jiang, Yi Xiao, and Chao Yang. Coherent semantic attention for image inpainting. In *ICCV*, pages 4170–4179, 2019. 3

[22] Hongyu Liu, Bin Jiang, Yibing Song, Wei Huang, and Chao Yang. Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In *ECCV*, pages 725–741, 2020. 3

[23] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018. 5

[24] Jarno Mielikainen. Lsb matching revisited. *IEEE Sign. Process. Letters*, 13(5):285–287, 2006. 3

[25] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM TOG*, 38 (4):1–14, 2019. 1, 2, 7

[26] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2

[27] 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. 1, 2, 8

[28] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Structure guided image inpainting using edge prediction. In *ICCV workshops*, 2019. 3

[29] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *CVPR*, pages 5480–5490, 2022. 2

[30] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, pages 2536–2544, 2016. 3

[31] Chenyang Qi, Xin Yang, Ka Leong Cheng, Ying-Cong Chen, and Qifeng Chen. Real-time 6k image rescaling with rate-distortion optimization. In *CVPR*, pages 14092–14101, 2023. 2, 3

[32] Yurui Ren, Xiaoming Yu, Ruonan Zhang, Thomas H Li, Shan Liu, and Ge Li. Structureflow: Image inpainting via structure-aware appearance flow. In *ICCV*, pages 181–190, 2019. 3

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4

[34] Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the boundaries of view extrapolation with multiplane images. In *CVPR*, pages 175–184, 2019. 2

[35] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, pages 5459–5469, 2022. 1, 2

[36] Jian Sun, Lu Yuan, Jiaya Jia, and Heung-Yeung Shum. Image completion with structure propagation. *ACM TOG*, 24(3): 861–868, 2005. 3

[37] 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 *WACV*, pages 2149–2159, 2022. 3, 4, 5, 6, 8

[38] Jun Tian. Reversible data embedding using a difference expansion. *IEEE TCSVT*, 13(8):890–896, 2003. 3

[39] Ziyu Wan, Jingbo Zhang, Dongdong Chen, and Jing Liao. High-fidelity pluralistic image completion with transformers. In *ICCV*, pages 4692–4701, 2021. 3

[40] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, pages 8798–8807, 2018. 5

[41] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 6

[42] 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. 2

[43] Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. Nex: Real-time view synthesis with neural basis expansion. In *CVPR*, pages 8534–8543, 2021. 1, 2

[44] Da-Chun Wu and Wen-Hsiang Tsai. A steganographic method for images by pixel-value differencing. *Pattern recognition letters*, 24(9-10):1613–1626, 2003. 3

[45] Yue Wu, Guotao Meng, and Qifeng Chen. Embedding novel views in a single jpeg image. In *ICCV*, pages 14519–14527, 2021. 1, 2, 3, 6

[46] Jinbo Xing, Wenbo Hu, Menghan Xia, and Tien-Tsin Wong. Scale-arbitrary invertible image downscaling. *IEEE TIP*, 2023. 3

[47] Youmin Xu, Chong Mou, Yujie Hu, Jingfen Xie, and Jian Zhang. Robust invertible image steganography. In *CVPR*, pages 7875–7884, 2022. 3

[48] Youtan Yin, Zhoujie Fu, Fan Yang, and Guosheng Lin. Or-nerf: Object removing from 3d scenes guided by multiview segmentation with neural radiance fields. *arXiv preprint arXiv:2305.10503*, 2023. 1, 2

[49] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, pages 4578–4587, 2021. 2

[50] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *ICCV*, pages 4471–4480, 2019. 3, 5

[51] Yingchen Yu, Fangneng Zhan, Rongliang Wu, Jianxiong Pan, Kaiwen Cui, Shijian Lu, Feiying Ma, Xuansong Xie, and Chunyan Miao. Diverse image inpainting with bidirectional and autoregressive transformers. In *ACM MM*, pages 69–78, 2021. 3

[52] Kevin Alex Zhang, Alfredo Cuesta-Infante, Lei Xu, and Kalyan Veeramachaneni. Steganogan: High capacity image steganography with gans. *arXiv preprint arXiv:1901.03892*, 2019. 3

[53] 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

[54] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM TOG*, 37(4):1–12, 2018. 1, 2, 3, 4, 5, 6

[55] Yuqian Zhou, Connelly Barnes, Eli Shechtman, and Sohrab Amirghodsi. Transfill: Reference-guided image inpainting by merging multiple color and spatial transformations. In *CVPR*, pages 2266–2276, 2021. 3

[56] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *ECCV*, pages 657–672, 2018. 3

[57] Qianshu Zhu, Chu Han, Guoqiang Han, Tien-Tsin Wong, and Shengfeng He. Video snapshot: Single image motion expansion via invertible motion embedding. *IEEE TPAMI*, 43(12):4491–4504, 2020. 3