

# DeepDR: Deep Structure-Aware RGB-D Inpainting for Diminished Reality

Christina Gsaxner<sup>1</sup> (gsaxner@tugraz.at), Shohei Mori<sup>1</sup>, Dieter Schmalstieg<sup>1,2</sup>, Jan Egger<sup>1,3</sup>,  
Gerhard Paar<sup>4</sup>, Werner Bailer<sup>4</sup> and Denis Kalkofen<sup>1,5</sup>

<sup>1</sup>Graz University of Technology, <sup>2</sup>University of Stuttgart, <sup>3</sup>University of Duisburg-Essen,  
<sup>4</sup>Joanneum Research, <sup>5</sup>Flinders University

## Abstract

*Diminished reality (DR) refers to the removal of real objects from the environment by virtually replacing them with their background. Modern DR frameworks use inpainting to hallucinate unobserved regions. While recent deep learning-based inpainting is promising, the DR use case is complicated by the need to generate coherent structure and 3D geometry (i.e., depth), in particular for advanced applications, such as 3D scene editing. In this paper, we propose DeepDR, a first RGB-D inpainting framework fulfilling all requirements of DR: Plausible image and geometry inpainting with coherent structure, running at real-time frame rates, with minimal temporal artifacts. Our structure-aware generative network allows us to explicitly condition color and depth outputs on the scene semantics, overcoming the difficulty of reconstructing sharp and consistent boundaries in regions with complex backgrounds. Experimental results show that the proposed framework can outperform related work qualitatively and quantitatively.*

## 1. Introduction

Diminished reality (DR) seeks to remove real objects from the environment by replacing them with their background [51], as illustrated in Fig. 1a. While *multi-observational* approaches [38, 49, 50] can utilize existing information about the scene, *inpainting* fabricates unseen background information and is, thus, more flexible.

Inpainting using generative adversarial networks (GANs) is nowadays successfully used in image space, e.g., to remove unwanted items during image and video editing [39, 86]. Contrary to conventional image or video inpainting, DR focuses on modifying a 3D scene rather than solely the image space, e.g., for removing 3D objects that distract from the immersive experience [17, 28, 33], or for replacing existing 3D objects with virtual ones in re-design [25, 61, 67, 87]. In these scenarios, it is important to *consider the underlying 3D geometry* of the scene for realistic rendering of virtual content and interactions with

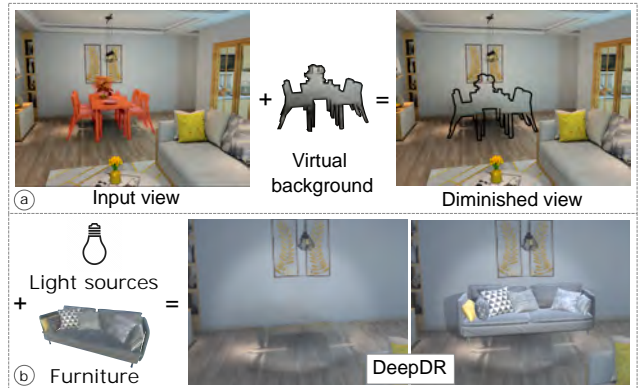


Figure 1. DR aims at replacing objects with their virtual background (a). DeepDR supports structure-aware RGB-D inpainting for DR experiences, enabling 3D scene editing by, e.g., adding light sources and replacing furniture (b).

the hallucinated background, e.g., regarding occlusion and lighting (see Fig. 1b). Thus, image space inpainting is not sufficient for DR applications – depth information needs to be coherently inpainted as well [27, 52]. Further, DR has strict requirements in *adhering to the structural boundaries of the underlying scene* [16, 58]. This is conflicting with the tendency towards producing blurry results at ambiguous object boundaries and regions with mixed semantics, which is commonly seen in image inpainting CNNs [40, 53, 70]. Lastly, unlike ordinary video inpainting [29, 39, 45, 85], DR needs the ability to run in real-time, *avoiding dissonance and flickering between consecutive frames*, without using future frame information.

In this paper, we propose DeepDR: The first approach to inpainting RGB-D frames with support for all aforementioned criteria of DR applications (see Tab. 1 for a structured comparison to the state-of-the-art). DeepDR has been designed as an end-to-end GAN, which performs inpainting of color images and their corresponding depth maps simultaneously. To enforce sharp structures with coherent semantics, we explicitly condition our model on the segmentation of the scene. To this end, we propose a novel structure-aware RGB-D decoder, which ensures adherence

to the underlying structural boundaries. To further limit temporal artifacts over a series of frames, we adopt a simple, yet effective, recurrent strategy based on convolutional long short-term memory (ConvLSTM) [34]. Thus, our framework produces temporally and structurally coherent inpainting. We emphasize inpainting the depth channel to ensure preserving a coherent 3D structure of the scene, which enables a realistic user experience in DR. Compared to related approaches, which rely on completing the various inpainting tasks in a sequential manner, our system processes inputs simultaneously, allowing each sub-task to benefit from the others. This allows to learn a comprehensive scene understanding, leading to a more plausible and consistent inpainting. We evaluate DeepDR in the context of interior redesign, a quintessential DR application, and we show that it can outperform previous methods qualitatively and quantitatively. In summary, we make the following contributions:

- We propose the first GAN for inpainting the color and depth channels of a DR system, which is capable of maintaining temporal and structural consistency.
- We introduce a novel structure-aware RGB-D decoder that supports generating sharp and plausible structures.
- We qualitatively and quantitatively evaluate DeepDR for indoor and outdoor DR applications, by applying it to synthetic and real data.

## 2. Related work

**Image inpainting.** Data-driven inpainting using deep learning leverages information from large databases. By implicitly or explicitly learning about the semantics of the scene, deep learning can produce high-quality results, spatially consistent with the image content. The seminal work of Context Encoders [57] first demonstrated the potential of a generative adversarial network (GAN) for image inpainting. Subsequent methods improve this approach, *e.g.*, using coarse-to-fine nets [22], attention [43, 80, 82], iterative refinement [36, 79, 86] or feature fusion [44, 84]. Partial or gated convolutions [42, 83] enable the handling of irregular masks without introducing artifacts, an important capability that we utilize in our work. Recently, diffusion-based inpainting delivers visually impressive results [47, 60]. However, their inference times of several seconds up to hours prohibit an application for real-time video, such as DR [47].

**RGB-D inpainting.** Depth inpainting literature largely focuses on filling missing depth values in regions *visible* in RGB images, for compensating failures of common depth sensors, *e.g.*, at transparent, reflective, or distant surfaces [21, 46, 78, 89, 90]. Depth inpainting of *hidden* structures, *e.g.*, in diminished parts of a scene, has been considered in only few works so far [3, 11, 12, 58]. Earlier works [11, 12] explore different fusion strategies of RGB and depth information but do not leverage structural guidance or temporal consistency. DynaFill [3] relies on a se-

Table 1. Overview of current deep inpainting works for DR.

	Color	Depth	Structure	Temporal
TransfoMR [25]	✓	✗	✗	✓
DynaFill [3]	✓	✓	✗	✓
PanoDR [16]	✓	✗	✓	✗
Pintore et al. [58]	✓	✓	✗	✗
DeepDR (Ours)	✓	✓	✓	✓

quential approach, where the color domain is coarsely inpainted and a separate depth completion network obtains geometry. For maintaining temporal consistency, it requires odometry, *i.e.*, camera poses. This has many pitfalls, as each sub-task relies on the results from the previous step, comes with significant computational overhead and is difficult to deploy. Pintore et al. [58] focus on the arguably simpler task of completely emptying rooms, while we also want to inpaint regions with complex and mixed semantics. Further, they do not deal with frame-to-frame consistency.

**Structural priors.** An ongoing challenge in inpainting is the reconstruction of sharp boundaries and structures consistent with the surrounding context, especially in regions with mixed semantics, where object boundaries are ambiguous. These structures are particularly important in DR, where interactivity with the scene is desired [16, 58]. Structural priors, such as edges [53], contours [75] or semantic segmentation [1, 16, 40, 70] can guide the inpainting of images. Amongst them, PanoDR [16] also targets an application in DR. However, their framework does not consider temporal coherence and 3D geometry. Sequential frameworks, which first complete the structural image, and feed it to the image generation network, are common. However, recent advances in image-to-image translation show that semantic information at the input of a generator may vanish through multiple downsampling and normalization stages [55]. Hence, simultaneous frameworks for completing structure and texture at the same time have become popular [1, 16, 40]. Inspired by these successes, we incorporate explicit structural guidance via intermediate semantic segmentation and extend it to the depth domain.

**Temporal consistency.** Video inpainting attempts to extend image inpainting to the temporal domain to ensure frame-to-frame consistency. Several approaches use 3D convolutions [7, 8, 29, 71], attention [35, 45, 85] TransfoMR [25] shows that deployment and real-time performance of some of these methods are feasible on mobile devices for DR. However, depth is not considered, which leads to shadow and occlusion artifacts. Diffusion-based techniques tend to be computationally expensive, requiring pre-processing and fine-tuning, which renders them impractical for real-time applications [5]. Another direction in video inpainting are optical flow-based methods, which emerge as most promising [13, 34, 39, 77]. Albeit flow-based meth-

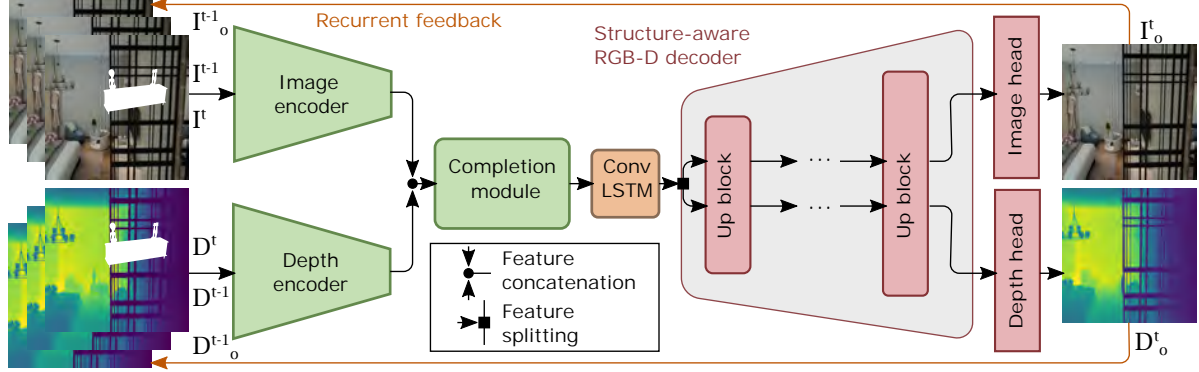


Figure 2. Overview of the proposed network. Image  $I$  and depth  $D$  inputs are encoded separately, before fusing features on a higher dimension and jointly completing them. Our semantics-aware decoder consists of a series of up blocks in which features are conditioned on semantic information. Finally, the outputs  $I_o^t$  and  $D_o^t$  are fed as auxiliary inputs to the next time step in a recurrent feedback loop.

ods show impressive results for tasks such as video editing, most cannot be applied directly to DR. They rely on forward and backward flow, requiring knowledge about past and future frames, which is not available in online scenarios. Furthermore, optical flow is expensive to compute. In our framework, we utilize optical flow only during training and rely on a recurrent network to reduce temporal artifacts.

### 3. Method

This section outlines the architecture of DeepDR. Our dual-stream encoder (Sec. 3.1) extracts contextual features from masked images masked depth separately at shallow layers, then fuses and jointly completes them. A structure-aware decoder (Sec. 3.2) uses two task-specific feature streams for RGB and depth with shared parameters. It estimates a semantic segmentation map from deep features and uses this map to modulate the RGB and depth feature generation. Thus, it is able to produce high-quality images and depths with a coherent semantic structure, which is persistent over domains and contexts. Finally, to reduce temporal artifacts between consecutive frames, we use a recurrent feedback loop with a ConvLSTM layer (Sec. 3.3). An overview of our model is given in Fig. 2. In the following, we explain our core components. Further architectural details are given in the supplementary material Sec. 6.1.

#### 3.1. Dual-stream encoder and completion module

Recent findings in image inpainting suggest that deep features in a CNN contain the majority of structural information, while shallow layers contain textural information [44]. Since RGB and depth inputs are texturally different, but represent the same underlying structure, we encode RGB and depth in two separate but parallel streams (illustrated by the green trapezoid in Fig. 2). The encoders use a coarse-to-fine architecture, which has proven to be highly effective for inpainting tasks [43, 80, 82]. After extracting features

through  $l \in [1, \dots, L]$  fine layers, we place a completion module to fuse them, such that the network can complete RGB and depth inputs simultaneously and thus, coherently. The completion module consists of a series of dilated convolutions [81] to expand the receptive field of the network and efficiently utilize global information. We use gated convolution layers [82] throughout our encoder and completion module, which dynamically learn to select appropriate features from masked and unmasked regions.

#### 3.2. Structure-aware RGB-D decoder

Our structure-aware RGB-D decoder is inspired by the spatially-adaptive normalization (SPADE) principle [55]. SPADE aims to overcome the problem of vanishing semantics, where sequential convolution, non-linearity, and normalization operations in a traditional CNN “wash away” structural information. It conditions generated features directly on semantic priors, by modulating them in normalization layers using a learned transformation. The same principle can be applied to other image-to-image translation tasks, such as inpainting. Our approach further exploits the fact that the RGB and depth inputs share the same underlying semantics – thus, we extend SPADE for RGB-D inpainting. Our decoder consists of a series of  $L$  up-sampling blocks based on residual learning (ResNet) [18], which consist of two RGB-D SPADE layers with intermediate convolutions and a skip connection (see Fig. 3a). Each up block receives an upsampled RGB and depth feature,  $i_l$  and  $d_l$ , from the previous layer. From the RGB feature, we explicitly model the underlying scene semantics by predicting a segmentation map on the current feature scale,  $S_l$ , using a pyramid pooling module [91] (see supplementary Sec. 8.5 for examples). This map, together with  $up(i_l)$  and  $up(d_l)$ , are forwarded to the RGB-D SPADE (Fig. 3b). Within the RGB-D SPADE, the segmentation map is embedded into feature space, and convolved to obtain learned, spatial modulation

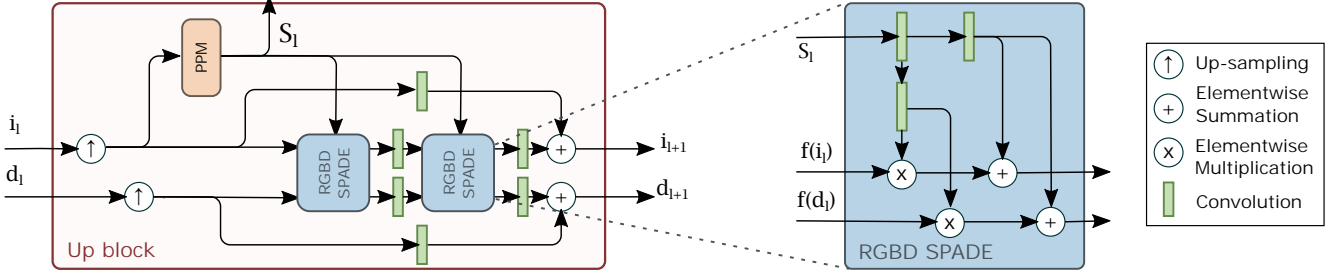


Figure 3. Our up blocks contain a residual architecture. From the up-sampled image feature  $i_l$ , a semantic segmentation map  $S_l$  is predicted using a pyramid pooling module (PPM). This segmentation is fed into the RGB-D SPADE, together with  $i_l$  and  $d_l$ . Within the RGB-D SPADE, the semantic map is embedded to feature space and convolved to obtain a learned, spatial scaling and bias map, with which inputs are transformed. Thus, the output features are consistently modulated by semantic segmentation.

parameters. The RGB and depth features are transformed by these parameters, conditioning them on the semantic information. The parameters of RGB-D SPADE layers are shared between the RGB and depth streams, ensuring consistent semantic structures in both of them.

### 3.3. Maintaining temporal consistency

As an alternative to expensive video inpainting techniques, we use a simple, yet effective method based on a recurrent network and ConvLSTM [64], originally proposed for blind video temporal consistency [34], and adapt it for RGB-D inpainting. Compared to video inpainting, which usually uses past and future frames, it allows our network to process frames in a sequential, online manner, *e.g.*, from  $t = 1$  to  $T$ . At every time step  $t$ , our network additionally receives the previous input image  $I^{t-1}$  and depth  $D^{t-1}$ , as well as their corresponding outputs,  $I_o^{t-1}$  and  $D_o^{t-1}$ , as auxiliary information. A ConvLSTM layer at the end of our completion module captures spatio-temporal correlations between consecutive frames in the feature space. While we use the optical flow between frames during training (see Sec. 3.4 for details), our method does not require flow at inference time. Thus, it is very efficient. Furthermore, we can process inputs of arbitrary length – be it single frames (in which case we set  $I^{t-1} = I^t$ ,  $D^{t-1} = D^t$ ) or long video sequences.

### 3.4. Training objectives

Our generator  $\mathcal{G}$  is trained with a combined loss function, which contains terms for supervising image inpainting, depth inpainting, semantic segmentation and temporal coherence (see Fig. 4):

$$\mathcal{L}_G = \mathcal{L}_I + \mathcal{L}_D + \mathcal{L}_{seg} + \mathcal{L}_{temp}. \quad (1)$$

**Adversarial learning.** On top of our generator, we use two global PatchGAN discriminators [23],  $\mathcal{D}_I$  and  $\mathcal{D}_D$ , to distinguish between real and inpainted RGB and depth patches. Thus, our network is trained in an adversarial fashion. We use Hinge loss [41] to compute our losses for train-

ing the discriminator,  $\mathcal{L}_{\mathcal{D},I}$  and  $\mathcal{L}_{\mathcal{D},D}$ , as well as adversarial generator loss terms  $\mathcal{L}_{adv,I}^G$  and  $\mathcal{L}_{adv,D}^G$ .

**Image inpainting.** We use the  $\ell_1$ -reconstruction loss  $\mathcal{L}_{rec,I}$  between synthesized pixels and the ground truth to ensure pixel-level reconstruction for image inpainting. Further, we use the perceptual loss  $\mathcal{L}_{per}$  [24] and style loss  $\mathcal{L}_{sty}$  [14] to encourage the network to produce RGB images perceptually similar to the ground truth. These data-driven losses enforce similarity in the feature space. Perceptual loss penalizes differences in features directly, while style loss minimizes the difference between feature distributions, de-localizing the feature information. Thus, image inpainting is supervised by the objective

$$\mathcal{L}_I = \lambda_{rec}\mathcal{L}_{rec,I} + \lambda_{per}\mathcal{L}_{per} + \lambda_{sty}\mathcal{L}_{sty} + \mathcal{L}_{adv,I}^G. \quad (2)$$

**Depth inpainting.** For depth inpainting, we again use the  $\ell_1$ -reconstruction loss  $\mathcal{L}_{rec,D}$  to penalize individual pixel errors. However, this loss does not take the local pixel neighborhood into account, which can lead to blurry edges and discontinuous surfaces in reconstructed depth images. Hence, to encourage smooth depth predictions with sharp steps, we use a gradient-based loss term

$$\mathcal{L}_{grad} = \|\nabla D - \nabla D_o\|_1, \quad (3)$$

where  $\nabla$  is the Sobel operator. Thus, depth loss is

$$\mathcal{L}_D = \lambda_{rec}\mathcal{L}_{rec,D} + \lambda_{grad}\mathcal{L}_{grad} + \mathcal{L}_{adv,D}^G. \quad (4)$$

**Semantic segmentation.** By predicting intermediate semantic segmentations in our structure-aware decoder, we ensure that our model explicitly learns semantic information from inputs. For supervising this prediction, we compute the cross-entropy loss for each intermediate segmentation map  $S_l$ , upsampled to the original input resolution, with the ground truth segmentation  $S$ ,

$$\mathcal{L}_{seg} = -\lambda_{seg} \frac{1}{L} \sum_{l \in L} \sum_{j \in S} S^j \log(\text{up}(S_l^j)), \quad (5)$$



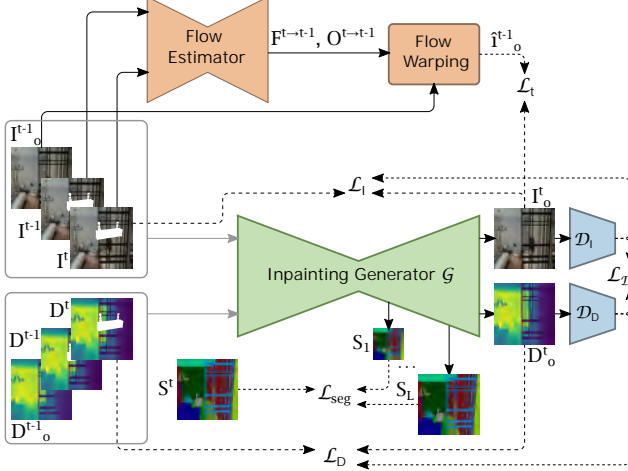


Figure 4. Illustration of training approach. We use appropriate loss terms and adversarial learning to promote accuracy and realism of inpainted RGB and depth ( $\mathcal{L}_I$ ,  $\mathcal{L}_D$ ). Via flow estimation and warping, we compute a temporal loss  $\mathcal{L}_t$  to ensure temporal consistency. Finally, we enforce our model to learn structural information by additionally providing semantic supervision via  $\mathcal{L}_{seg}$ .

where  $j$  denotes the classes of the segmentation map.

**Temporal coherence.** Similar to Lai *et al.* [34], we enforce temporal coherence by training our model with short-term ( $st$ ) and long-term ( $lt$ ) temporal losses between current and previous, and current and first frame, respectively:

$$\mathcal{L}_{st} = \sum_{t=2}^T O^{t \rightarrow t-1} \|I_o^t - \hat{I}_o^{t-1}\|_1, \quad (6)$$

$$\mathcal{L}_{lt} = \sum_{t=3}^T O^{t \rightarrow 1} \|I_o^t - \hat{I}_o^1\|_1.$$

Here,  $\hat{I}_o^x$  is the output image at time  $x$ , warped to the current time point using the optical flow  $F^{t \rightarrow x}$  between  $I^t$  and  $I^x$ , and  $O^{t \rightarrow x}$  is the corresponding occlusion mask (see Fig. 4). In our framework, during training,  $F$  and  $O$  are computed using MaskFlowNet [92]. The temporal loss term is

$$\mathcal{L}_{temp} = \lambda_t (\mathcal{L}_{lt} + \mathcal{L}_{st}). \quad (7)$$

A full description of objectives and training details are found in the supplementary material Sec. 6.2 and Sec. 6.3.

## 4. Experiments and results

### 4.1. Datasets

While several benchmarks exist for RGB inpainting [26, 94], there are few datasets suitable for RGB-D object removal and DR (see supplementary Sec. 7.1). To the best of our knowledge, DynaFill [3] is the sole dataset that offers ground truth by presenting scenes both with and without individual objects that need to be removed. Same as in

the original paper, we extract masks from dynamic objects and use the default training and validation split. Since DynaFill only covers outdoor driving scenarios with limited variability, we additionally evaluate our method on InteriorNet [37], were, similar to other works [29, 39, 45], we simulate the object removal task by inpainting random object-like masks during training and testing (see supplementary Fig. 14, Fig. 15). Specifically, those masks are generated from instance segmentations belonging to non-background classes (*i.e.*, excluding walls, ceilings, floors, windows, and doors). We split the 618 layouts in InteriorNet into 494 for training and 62 for testing and validation. To show the generalizability of our model and demonstrate its performance in-the-wild, we further use 100 layouts from ScanNet [10] for testing the models trained on InteriorNet.

### 4.2. Comparison with other methods

As already mentioned, only few works about RGB-D inpainting of hidden structures are known to us [3, 11, 12, 58]. Only DynaFill [3] is accessible, although training and testing code are not provided. Therefore, we re-compute results and metrics on their dataset using the publicly available demo model. Sequential frameworks, where RGB information is completed first, and missing depth information is filled based on the reconstructed image using depth completion, are an alternative for DR [25, 58]. Hence, we build our baselines on top of recent RGB inpainting methods, and use state-of-the-art depth completion networks, InDepth [90], DM-LRN [63] and NLSPN [54], to fill missing depth regions from the RGB inpainting. Hereafter, we use the best-performing depth completion method on each dataset for our comparison, which is InDepth for InteriorNet, and NLSPN for ScanNet and DynaFill. A detailed comparison is provided in the supplementary Sec. 7.2. Based on performance and code availability, we compare to DeepFillV2 [83], PanoDR [16] and E2FGVI [39], which represent standard, structure-guided, and video inpainting, respectively. For a fair comparison, we re-train the models on our datasets using their publicly available training code.

### 4.3. Quantitative results

To quantitatively assess the performance of our approach, we use the pixel-level metrics peak signal-to-noise ratio (PSNR) and mean absolute error (MAE) for image, and root mean squared error (RMSE) in meters for depth inpainting. However, these metrics only measure pixel-wise concordance and tend to favor blurry over perceptually similar images, which is problematic for DR. Measures computed on deep features better mirror human perception [88] and are, thus, considered more meaningful for our evaluation. We use learned perceptual image patch similarity (LPIPS) [88] and Fréchet inception distance (FID) [19] for images, and video FID (VFID) [73] for sequences. We further compare

efficiency by measuring inference time, multiply-add operations (MADs) and total parameters.

It can be seen from Tab. 2, Tab. 3 and Tab. 4 that DeepDR outperforms all related methods in the feature-based inpainting metrics, on indoor (InteriorNet), outdoor (DynaFill), as well as real, unseen (ScanNet) data. As mentioned, we consider these metrics to be most significant for DR. Considering depth RMSE, it is evident that our joint framework outperforms both sequential methods, consisting of image and depth-from-image inpainting, as well as DynaFill by a large margin. In pixel-based RGB metrics, DeepDR comes second, after E2FGVI or DynaFill. The lead of E2FGVI is larger on ScanNet – we attribute that to its tendency to produce overly smooth results, which matches the blurry images recurrent in ScanNet. DeepDR achieves leading results in video-based metrics as well, surpassing E2FGVI on InteriorNet and coming second on ScanNet. On DynaFill data, E2FGVI and DynaFill outperform DeepDR in VFID, but the increased temporal smoothness comes at the cost of increased blurriness, as shown by the lower FID and LPIPS, and the qualitative results. Contrary to E2FGVI, our method works for single images or very short sequences. No expensive flow computation is required at inference, making it almost one order of magnitude faster (see Tab. 5), which is a critical factor for DR applications, where real-time frame rates are desired. Only DeepFillV2, which is the least powerful method in our tests, is faster than our model. DynaFill assumes availability of accurate camera poses and intrinsics, which may be difficult to obtain in real-world scenarios.

Table 2. Quantitative comparison of inpainting models trained on InteriorNet [37]. For baselines, we use InDepth [90] to fill missing depth.

Model	RGB				Depth	Video
	LPIPS ↓	FID ↓	PSNR ↑	MAE ↓	RMSE ↓	VFID ↓
DeepFillV2 [83]	0.0150	0.448	41.6	0.0312	0.572	0.0446
PanoDR [16]	0.0128	0.606	41.0	0.0331	0.564	0.0360
E2FGVI [39]	0.0131	<u>0.363</u>	<b>43.2</b>	<b>0.0255</b>	<u>0.563</u>	<u>0.0326</u>
DeepDR (Ours)	<b>0.0104</b>	<b>0.218</b>	<u>41.9</u>	<u>0.0311</u>	<b>0.278</b>	<b>0.0257</b>

Table 3. Quantitative comparison of inpainting models trained on DynaFill [3]. For DeepFillV2 [83], PanoDR [16] and E2FGVI [39], we use NLSPN [54] to fill missing depth.

Model	RGB				Depth	Video
	LPIPS ↓	FID ↓	PSNR ↑	MAE ↓	RMSE ↓	VFID ↓
DeepFillV2 [83]	0.0238	4.122	34.2	<u>0.0062</u>	7.92	1.185
PanoDR [16]	0.0250	5.579	31.8	0.0119	8.12	1.822
E2FGVI [39]	<u>0.0169</u>	2.826	<u>35.2</u>	<b>0.0054</b>	7.83	<u>0.777</u>
DynaFill [3]	0.0197	<u>2.665</u>	<b>38.8</b>	0.0107	<u>7.78</u>	<b>0.636</b>
DeepDR (Ours)	<b>0.0168</b>	<b>2.415</b>	34.2	<u>0.0062</u>	<b>4.51</b>	0.788

Table 4. Generalizability experiment on ScanNet [10] of inpainting models trained on InteriorNet [37]. For baselines, we use NLSPN [54] to fill missing depth.

Model	RGB				Depth	Video
	LPIPS ↓	FID ↓	PSNR ↑	MAE ↓	RMSE ↓	VFID ↓
DeepFillV2 [83]	0.0208	0.693	40.1	0.0400	<u>0.508</u>	0.873
PanoDR [16]	0.0119	0.348	41.5	0.0304	0.536	0.358
E2FGVI [39]	<u>0.0110</u>	<u>0.295</u>	<b>46.7</b>	<b>0.0176</b>	0.512	<b>0.206</b>
DeepDR	<b>0.0108</b>	<b>0.292</b>	<u>42.4</u>	<u>0.0280</u>	<b>0.484</b>	<u>0.218</u>

Table 5. Efficiency of DeepDR on an NVIDIA GeForce GTX 1080 Ti GPU in comparison to the baselines.

Model	Time ↓ (ms)	MADs ↓	Params ↓
DeepFillV2 [83]	<b>3.73</b>	<b>25.3 G</b>	<b>4.1 M</b>
PanoDR [16]	7.07	189.6 G	78.8 M
E2FGVI [39]	40.0	309.1 G	41.8 M
DynaFill [3]*	14.3	<u>78.6 G</u>	<u>22.1 M</u>
DeepDR	<u>4.43</u>	184.3 G	69.9 M

\*Measurements do not include camera pose computation.

#### 4.4. Qualitative results

For qualitative analysis, we compare the performance of our method to the baselines for our DR use case. Thus, we diminish objects existing in the scene to show how well the models can hallucinate realistic background textures and structures coherent with the scene semantics. Note that, for this use case, no ground truth data exists on InteriorNet and ScanNet (Fig. 5). Results on DynaFill, which provides ground truth, are shown in Fig. 6. More qualitative examples are given in the supplementary. DeepDR is able to produce high-quality RGB textures while preserving the structure of the scene. Fig. 5 shows that although it was trained on purely synthetic data, it can generalize well to real-world examples from ScanNet. Its abilities are particularly evident for complex and textured backgrounds, where other methods tend to produce artifacts or overly smooth results. The benefits of our explicit structural guidance using RGB-D SPADE are also evident: While the baseline methods have difficulties in reconstructing clean borders and sharp edges, our method can recreate them well. Further, it is evident that the baseline depth completion fails at filling complex depth regions with sharp edges (e.g., between floors and walls), in particular for structures far away from the camera. This observation reveals that sequential approaches suffer from the loss of detail and sharp features in inpainted images.

#### 4.5. User study

We conducted a repeated measures within-subjects user study to demonstrate that our framework can surpass existing works in the task of object removal for DR, and enables advanced 3D scene editing. We used 12 scenes from our InteriorNet testing dataset, in each of which we diminished one object in 50-200 consecutive frames. To illus-

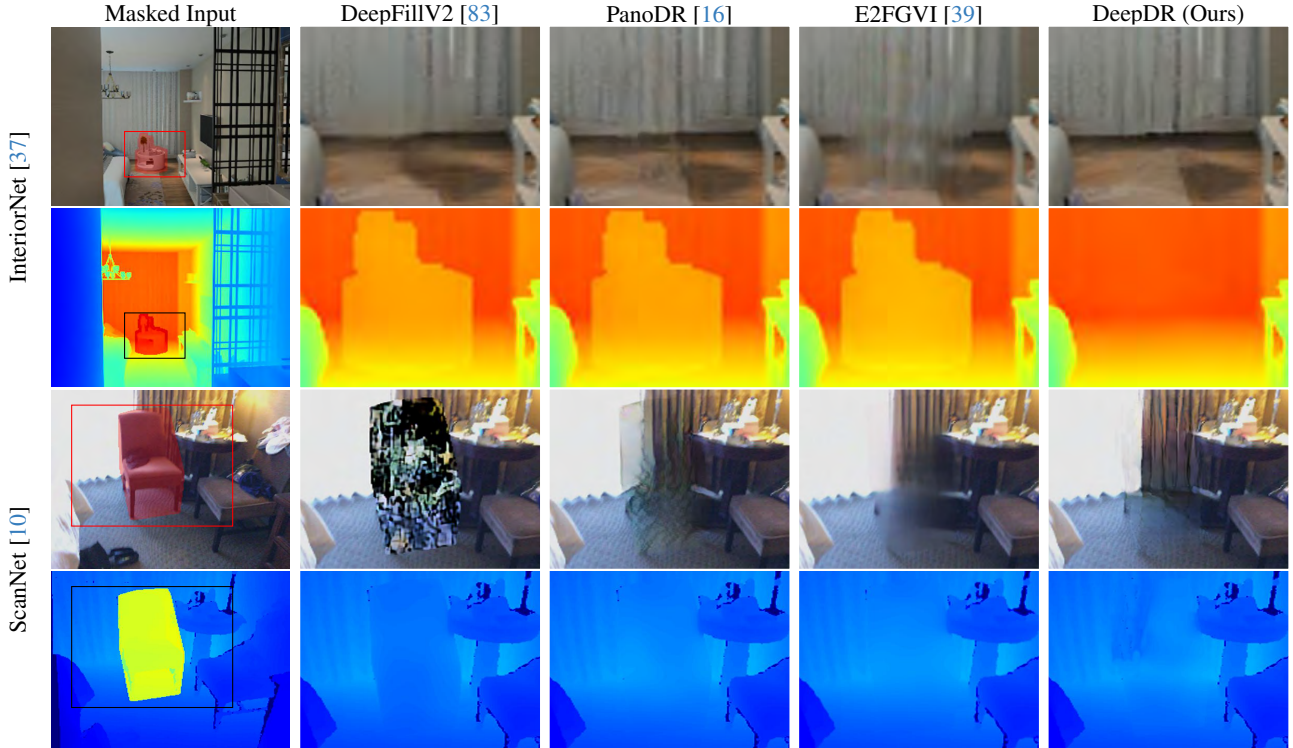


Figure 5. Qualitative comparison for diminishing objects from InteriorNet [37] (synthetic) and ScanNet [10] (real) with models trained on InteriorNet. Result images are zoomed to the red and black outlines in the first column.

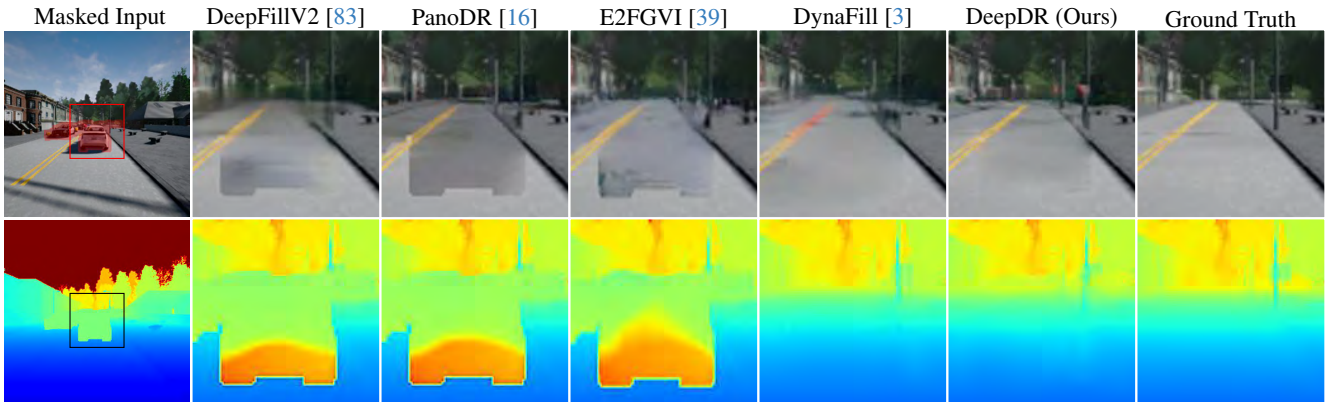


Figure 6. Qualitative comparison for diminishing objects from DynaFill [3].

trate the importance of coherent color, structure, and geometry inpainting, we reconstructed a textured 3D mesh from each inpainted RGB-D pair and augmented the reconstructed scene with additional light sources and furniture objects, as shown in Fig. 1 and the supplementary Fig. 13. The sequences were presented to the participants in random order, side-by-side with the original input sequence, in which the object of interest was highlighted. The participants were asked to rate each item on a 7-point scale from 1 (“very poor”) to 7 (“very well”).

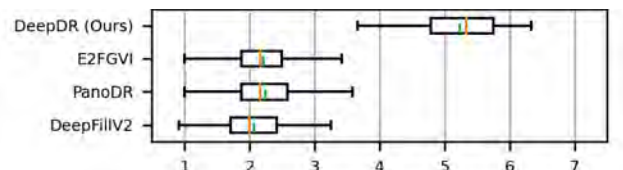


Figure 7. Average user rating over all participants and samples from 1 (“very poor”) to 7 (“very well”).

**Results.** 64 participants (15 female, age  $34.0 \pm 9.4$ ) completed the study. The average experience of users with



AR/DR and inpainting was  $2.4 \pm 1.4$  and  $2.3 \pm 1.2$  on a 5-point scale from “none” to “expert”, respectively. The user ratings are shown in Fig. 7. It is evident that users rate the plausibility and realism of images and geometry inpainted with our method higher than those of others. A Friedman test revealed a significant ( $\chi^2(3) = 133.3$ ,  $W = 0.7$ ,  $p < 0.001$ ) difference in inpainting methods. A post-hoc Wilcoxon signed-rank test indicates that the median rating of our method (5.3) is substantially higher than that of DeepFillV2 (2.0,  $p < 0.001$ ), PanoDR (2.2,  $p < 0.001$ ), and E2GFVI (2.2,  $p < 0.001$ ). No significant differences were found between other methods. We conclude that users prefer our DR results in terms of realism and plausibility.

Table 6. Ablation studies of our model on InteriorNet [37].

Model	RGB				Depth	Video
	LPIPS ↓	FID ↓	PSNR ↑	MAE ↓	RMSE ↓	VFID ↓
no temporal	0.0160	0.567	40.0	0.0336	0.358	0.0487
no RGBD SPADE	0.0143	0.435	40.1	0.0408	0.374	0.0363
joint encoder	0.0121	0.333	40.9	0.0340	0.306	0.0322
DeepDR (Full)	<b>0.0104</b>	<b>0.218</b>	<b>41.9</b>	<b>0.0311</b>	<b>0.278</b>	<b>0.0257</b>

#### 4.6. Ablation study

To demonstrate the effectiveness of the core components of our model, we perform three ablation studies on InteriorNet, see Tab. 6. First, we remove temporal coherence from our model by omitting the auxiliary inputs  $I^{t-1}$ ,  $D^{t-1}$ ,  $I_o^{t-1}$  and  $D_o^{t-1}$ , remove the ConvLSTM layer from our architecture and train without temporal loss  $\mathcal{L}_t$ . As expected, this removal leads to lower perceptual similarity of videos (VFID). Generally, a deteriorated performance is observed, which suggests that our final model is effective in leveraging information from previous frames to fill missing regions. Second, we evaluate a model without structural guidance, by replacing the RGB-D SPADE layers in our up blocks with standard transposed convolutions. No intermediate segmentations are available, therefore segmentation loss  $\mathcal{L}_{seg}$  is omitted. Evidently, the additional supervision via segmentation is beneficial for our model all along the line. The cost of RGB-D SPADE is an almost doubled inference time (see supplementary Tab. 11), which could be a limitation for real-time applications on less capable hardware. Third, we replace our separate encoders with a joint coarse-to-fine encoder. It is apparent that our separate encoder is more effective in extracting appropriate features.

#### 4.7. Limitations

For large diminished areas in front of highly irregularly textured objects (e.g., the carpet in Fig. 8, top), DeepDR may generate structural artifacts, while other methods tend to over-smoothed backgrounds. Furthermore, all methods struggle with completing highly ambiguous object bound-

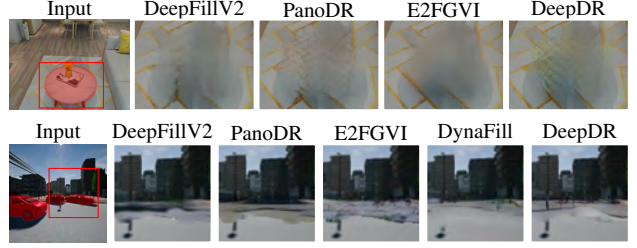


Figure 8. Failure cases. All current methods struggle with irregularly textured objects (e.g., the carpet, top) and highly ambiguous object borders (e.g. the curb, bottom).

aries (see Fig. 8, bottom), which may cause color bleeding artifacts or implausible borders. Lastly, since InteriorNet does not have ground truth for object removal and our mask generation does not include shadows cast by objects of interest, models trained on this dataset do not learn to remove those shadows. Shadow borders are very ambiguous once the object casting them has been removed, in particular, since shaded and un-shaded regions usually belong to the same semantic class, which may lead to artifacts. We analyze this effect in the supplementary Sec. 7.3.

### 5. Conclusion

We introduced DeepDR, the first approach to deep, structure-aware RGB-D inpainting with temporal coherence for DR. Our generative approach uses an RGB-D SPADE decoder to exploit structural priors, consistently conditioning color and depth outputs on them at feature level. To minimize temporal artifacts, we utilize a simple recurrent architecture with a ConvLSTM, which, compared to recent video inpainting, does not require future frame information or expensive optical flow computation at inference time. Quantitative results demonstrate that DeepDR surpasses state-of-the-art inpainting methods in terms of feature-based metrics, while qualitative results show that our method is capable of generating content which is perceptually plausible, realistic in the context of the scene, and blends seamlessly with the surroundings in the image and depth domain of synthetic and real data. DeepDR works better because it effectively leverages information from multiple modalities, in particular, color, depth and structure. Therefore, it has a more comprehensive understanding of the scene, and can inpaint missing regions more plausibly in 3D.

### Acknowledgement

The work was funded by the Austrian Research Promotion Agency (FFG) project “TRIP - Simulation-Based Training for AI-based Interior Planning” (BRIDGE 883658) and the Austrian Science Fund (FWF) project “enFaced 2.0 - Instant AR Tool for Maxillofacial Surgery” (KLI 1044). We thank xCAD Solutions GmbH for their continuous support.



## References

- [1] Pierfrancesco Ardino, Yahui Liu, Elisa Ricci, Bruno Lepri, and Marco De Nadai. Semantic-guided inpainting network for complex urban scenes manipulation. In *ICPR*, pages 9280–9287, 2021. [2](#)
- [2] Amir Atapour-Abarghouei and Toby P Breckon. Depth-comp: real-time depth image completion based on prior semantic scene segmentation. In *BMVC*, 2017. [3](#)
- [3] Borna Bešić and Abhinav Valada. Dynamic object removal and spatio-temporal rgb-d inpainting via geometry-aware adversarial learning. *IEEE trans. intell. veh.*, 7(2):170–185, 2022. [2](#), [5](#), [6](#), [7](#), [3](#), [4](#), [12](#)
- [4] Jinming Cao, Hanchao Leng, Dani Lischinski, Daniel Cohen-Or, Changhe Tu, and Yangyan Li. Shapeconv: Shape-aware convolutional layer for indoor rgb-d semantic segmentation. In *ICCV*, pages 7088–7097, 2021. [4](#)
- [5] Duygu Ceylan, Chun-Hao P Huang, and Niloy J Mitra. Pix2video: Video editing using image diffusion. In *ICCV*, pages 23206–23217, 2023. [2](#)
- [6] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. [2](#)
- [7] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston Hsu. Free-form video inpainting with 3d gated convolution and temporal patchgan. In *ICCV*, pages 9066–9075, 2019. [2](#)
- [8] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston Hsu. Learnable gated temporal shift module for deep video inpainting”. In *BMVC*, 2019. [2](#)
- [9] Xiaokang Chen, Kwan-Yee Lin, Jingbo Wang, Wayne Wu, Chen Qian, Hongsheng Li, and Gang Zeng. Bi-directional cross-modality feature propagation with separation-and-aggregation gate for rgb-d semantic segmentation. In *ECCV*, pages 561–577. Springer, 2020. [4](#)
- [10] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017. [5](#), [6](#), [7](#), [4](#), [10](#), [13](#)
- [11] Helisa Dhamo, Keisuke Tateno, Iro Laina, Nassir Navab, and Federico Tombari. Peeking behind objects: Layered depth prediction from a single image. *Pattern Recognit. Lett.*, 125: 333–340, 2019. [2](#), [5](#)
- [12] Ryo Fujii, Ryo Hachiuma, and Hideo Saito. Rgb-d image inpainting using generative adversarial network with a late fusion approach. In *AVR*, pages 440–451. Springer, 2020. [2](#), [5](#)
- [13] Chen Gao, Ayush Saraf, Jia-Bin Huang, and Johannes Kopf. Flow-edge guided video completion. In *ECCV*, pages 713–729. Springer, 2020. [2](#)
- [14] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, pages 2414–2423, 2016. [4](#)
- [15] Pallabi Ghosh, Vibhav Vineet, Larry S Davis, Abhinav Shrivastava, Sudipta Sinha, and Neel Joshi. Depth completion using a view-constrained deep prior. In *3DV*, pages 723–733. IEEE, 2020. [3](#)
- [16] Vasileios Gkitsas, Vladimiro Sterzentsenko, Nikolaos Zioulis, Georgios Albanis, and Dimitrios Zarpalas. Panodr: Spherical panorama diminished reality for indoor scenes. In *CVPR Workshops*, pages 3716–3726, 2021. [1](#), [2](#), [5](#), [6](#), [7](#), [3](#), [8](#), [9](#), [10](#)
- [17] Satoshi Hashiguchi, Shohei Mori, Miho Tanaka, Fumihisa Shibata, and Asako Kimura. Perceived weight of a rod under augmented and diminished reality visual effects. In *VRST*, 2018. [1](#)
- [18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. [3](#)
- [19] 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. [5](#)
- [20] Heiko Hirschmuller and Daniel Scharstein. Evaluation of cost functions for stereo matching. In *CVPR*, pages 1–8. IEEE, 2007. [2](#)
- [21] Yu-Kai Huang, Tsung-Han Wu, Yueh-Cheng Liu, and Winston H Hsu. Indoor depth completion with boundary consistency and self-attention. In *ICCV Workshops*, 2019. [2](#)
- [22] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM TOG*, 36(4):1–14, 2017. [2](#)
- [23] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017. [4](#)
- [24] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711. Springer, 2016. [4](#)
- [25] Mohamed Kari, Tobias Grosse-Puppenthal, Luis Falconeri Coelho, Andreas Rene Fender, David Bethge, Reinhard Schütte, and Christian Holz. Transformr: Pose-aware object substitution for composing alternate mixed realities. In *IEEE ISMAR*, pages 69–79, 2021. [1](#), [2](#), [5](#)
- [26] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018. [5](#)
- [27] Norihiko Kawai, Tomokazu Sato, and Naokazu Yokoya. Diminished reality based on image inpainting considering background geometry. *IEEE TVCG*, 22(3):1236–1247, 2015. [1](#)
- [28] Norihiko Kawai, Tomokazu Sato, Yuta Nakashima, and Naokazu Yokoya. Augmented reality marker hiding with texture deformation. *IEEE TVCG*, 23(10):2288–2300, 2016. [1](#)
- [29] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Deep video inpainting. In *CVPR*, pages 5792–5801, 2019. [1](#), [2](#), [5](#)
- [30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. [2](#)
- [31] 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](#)

- [32] Lingtong Kong, Chunhua Shen, and Jie Yang. FastflowNet: A lightweight network for fast optical flow estimation. In *IEEE ICRA*, 2021. 2
- [33] Otto Korkalo, Miika Aittala, and Sanni Siltanen. Lightweight marker hiding for augmented reality. In *IEEE ISMAR*, pages 247–248, 2010. 1
- [34] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *ECCV*, pages 170–185. Springer, 2018. 2, 4, 5, 1
- [35] Sungho Lee, Seoung Wug Oh, DaeYeun Won, and Seon Joo Kim. Copy-and-paste networks for deep video inpainting. In *ICCV*, pages 4413–4421, 2019. 2
- [36] Jingyuan Li, Ning Wang, Lefei Zhang, Bo Du, and Dacheng Tao. Recurrent feature reasoning for image inpainting. In *CVPR*, pages 7760–7768, 2020. 2
- [37] Wenbin Li, Sajad Saeedi, John McCormac, Ronald Clark, Dimos Tzoumanikas, Qing Ye, Yuzhong Huang, Rui Tang, and Stefan Leutenegger. Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. In *BMVC*, 2018. 5, 6, 7, 8, 4, 9, 11
- [38] Zhuwen Li, Yuxi Wang, Jiaming Guo, Loong-Fah Cheong, and Steven ZhiYing Zhou. Diminished reality using appearance and 3d geometry of internet photo collections. In *IEEE ISMAR*, pages 11–19. IEEE, 2013. 1
- [39] Zhen Li, Cheng-Ze Lu, Jianhua Qin, Chun-Le Guo, and Ming-Ming Cheng. Towards an end-to-end framework for flow-guided video inpainting. In *CVPR*, pages 17562–17571, 2022. 1, 2, 5, 6, 7, 3, 8, 9, 10
- [40] Liang Liao, Jing Xiao, Zheng Wang, Chia-Wen Lin, and Shin’ichi Satoh. Image inpainting guided by coherence priors of semantics and textures. In *CVPR*, pages 6539–6548, 2021. 1, 2
- [41] Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017. 4, 1
- [42] 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. Springer, 2018. 2, 1
- [43] Hongyu Liu, Bin Jiang, Yi Xiao, and Chao Yang. Coherent semantic attention for image inpainting. In *ICCV*, pages 4170–4179, 2019. 2, 3
- [44] 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. Springer, 2020. 2, 3
- [45] Rui Liu, Hanming Deng, Yangyi Huang, Xiaoyu Shi, Lewei Lu, Wenxiu Sun, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fuseformer: Fusing fine-grained information in transformers for video inpainting. In *ICCV*, pages 14040–14049, 2021. 1, 2, 5
- [46] Wei Liu, Xiaogang Chen, Jie Yang, and Qiang Wu. Robust color guided depth map restoration. *IEEE TIP*, 26(1):315–327, 2016. 2
- [47] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, pages 11461–11471, 2022. 2
- [48] John McCormac, Ankur Handa, Stefan Leutenegger, and Andrew J Davison. Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation? In *ICCV*, pages 2678–2687, 2017. 2
- [49] Siim Meerits and Hideo Saito. Real-time diminished reality for dynamic scenes. In *IEEE ISMAR Workshops*, pages 53–59. IEEE, 2015. 1
- [50] Shohei Mori, Fumihisa Shibata, Asako Kimura, and Hideyuki Tamura. Efficient use of textured 3d model for pre-observation-based diminished reality. In *IEEE ISMAR Workshops*, pages 32–39. IEEE, 2015. 1
- [51] Shohei Mori, Sei Ikeda, and Hideo Saito. A survey of diminished reality: Techniques for visually concealing, eliminating, and seeing through real objects. *IPSJ Trans. Comput. Vis. Appl.*, 9(1):1–14, 2017. 1
- [52] Shohei Mori, Okan Erat, Wolfgang Broll, Hideo Saito, Dieter Schmalstieg, and Denis Kalkofen. Inpaintfusion: Incremental rgb-d inpainting for 3d scenes. *IEEE TVCG*, 26(10):2994–3007, 2020. 1
- [53] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Structure guided image inpainting using edge prediction. In *ICCV Workshops*, 2019. 1, 2
- [54] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In *ECCV*, pages 120–136. Springer, 2020. 5, 6, 2, 4
- [55] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, pages 2337–2346, 2019. 2, 3
- [56] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035, 2019. 1
- [57] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, pages 2536–2544, 2016. 2
- [58] Giovanni Pintore, Marco Agus, Eva Almansa, and Enrico Gobbetti. Instant automatic emptying of panoramic indoor scenes. *IEEE TVCG*, 2022. 1, 2, 5
- [59] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE TPAMI*, 44(3), 2022. 2
- [60] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 2
- [61] Hajime Sasanuma, Yoshitsugu Manabe, and Noriko Yata. Diminishing real objects and adding virtual objects using a rgb-d camera. In *IEEE ISMAR Workshops*, pages 117–120, 2016. 1
- [62] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *GCPR*, pages 31–42. Springer, 2014. 2

- [63] Dmitry Senushkin, Mikhail Romanov, Ilia Belikov, Nikolay Patakin, and Anton Konushin. Decoder modulation for indoor depth completion. In *IEEE IROS*, pages 2181–2188. IEEE, 2021. 5, 2, 3, 4
- [64] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *NeurIPS*, 28, 2015. 4, 1
- [65] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 3
- [66] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760. Springer, 2012. 2
- [67] Sanni Siltanen, Henriikki Saraspää, and Jari Karvonen. A complete interior design solution with diminished reality. In *IEEE ISMAR*, pages 371–372, 2014. 1
- [68] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1
- [69] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, pages 1746–1754, 2017. 2
- [70] Yuhang Song, Chao Yang, Yeji Shen, Peng Wang, Qin Huang, and C-C Jay Kuo. Spg-net: Segmentation prediction and guidance network for image inpainting. In *BMVC*, 2018. 1, 2
- [71] Chuan Wang, Haibin Huang, Xiaoguang Han, and Jue Wang. Video inpainting by jointly learning temporal structure and spatial details. In *AAAI*, pages 5232–5239, 2019. 2
- [72] Haowen Wang, Mingyuan Wang, Zhengping Che, Zhiyuan Xu, Xiuquan Qiao, Mengshi Qi, Feifei Feng, and Jian Tang. Rgb-depth fusion gan for indoor depth completion. In *CVPR*, pages 6209–6218, 2022. 3
- [73] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Nikolai Yakovenko, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *NeurIPS*, 2018. 5
- [74] Alex Wong, Xiaohan Fei, Stephanie Tsuei, and Stefano Soatto. Unsupervised depth completion from visual inertial odometry. *IEEE Robot. Autom. Lett.*, 5(2):1899–1906, 2020. 2
- [75] Wei Xiong, Jiahui Yu, Zhe Lin, Jimei Yang, Xin Lu, Connelly Barnes, and Jiebo Luo. Foreground-aware image inpainting. In *CVPR*, pages 5840–5848, 2019. 2
- [76] Zhitong Xiong, Yuan Yuan, Nianhui Guo, and Qi Wang. Variational context-deformable convnets for indoor scene parsing. In *CVPR*, pages 3992–4002, 2020. 4
- [77] Rui Xu, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy. Deep flow-guided video inpainting. In *CVPR*, pages 3723–3732, 2019. 2
- [78] Hongyang Xue, Shengming Zhang, and Deng Cai. Depth image inpainting: Improving low rank matrix completion with low gradient regularization. *IEEE TIP*, 26(9):4311–4320, 2017. 2
- [79] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *CVPR*, pages 6721–6729, 2017. 2
- [80] Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, and Zhan Xu. Contextual residual aggregation for ultra high-resolution image inpainting. In *CVPR*, pages 7508–7517, 2020. 2, 3
- [81] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *CVPR*, pages 472–480, 2017. 3
- [82] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *CVPR*, pages 5505–5514, 2018. 2, 3
- [83] 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. 2, 5, 6, 7, 1, 3, 8, 9, 10
- [84] Yanhong Zeng, Jianlong Fu, Hongyang Chao, and Baining Guo. Learning pyramid-context encoder network for high-quality image inpainting. In *CVPR*, pages 1486–1494, 2019. 2
- [85] Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning joint spatial-temporal transformations for video inpainting. In *ECCV*, pages 528–543. Springer, 2020. 1, 2
- [86] Yu Zeng, Zhe Lin, Jimei Yang, Jianming Zhang, Eli Shechtman, and Huchuan Lu. High-resolution image inpainting with iterative confidence feedback and guided upsampling. In *ECCV*, pages 1–17. Springer, 2020. 1, 2
- [87] Edward Zhang, Michael F Cohen, and Brian Curless. Emptying, refurbishing, and relighting indoor spaces. *ACM TOG*, 35(6):1–14, 2016. 1
- [88] 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. 5
- [89] Yinda Zhang and Thomas Funkhouser. Deep depth completion of a single rgb-d image. In *CVPR*, 2018. 2
- [90] Yunfan Zhang, Tim Scargill, Ashutosh Vaishnav, Gopika Premankar, Mario Di Francesco, and Maria Gorlatova. In-depth: Real-time depth inpainting for mobile augmented reality. In *IMWUT*, pages 1–25. ACM, 2022. 2, 5, 6, 4
- [91] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017. 3, 4
- [92] Shengyu Zhao, Yilun Sheng, Yue Dong, Eric I-Chao Chang, and Yan Xu. Maskflownet: Asymmetric feature matching with learnable occlusion mask. In *CVPR*, 2020. 5, 2
- [93] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *ECCV*, pages 519–535. Springer, 2020. 2
- [94] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE TPAMI*, 40(6):1452–1464, 2017. 5

# DeepDR: Deep Structure-Aware RGB-D Inpainting for Diminished Reality

## Supplementary Material

### 6. Implementation details

#### 6.1. Architecture details

The input of our network is a 3-channel RGB image and a 1-channel depth map. RGB inputs are normalized to  $[-1, 1]$  and depth inputs are scaled to  $[0, 1]$ . Our generator uses a coarse-to-fine principle. First, we use two coarse networks identical to the ones of DeepFillV2 [83] to produce coarsely inpainted RGB images  $\tilde{I}_o^t$  and depth maps  $\tilde{D}_o^t$ . These outputs are used to coarsely fill the missing regions in the masked input, yielding  $\tilde{I}_m^t$  and  $\tilde{D}_m^t$ . These intermediate images are then concatenated along the channel dimension with the previous in- and outputs  $I^{t-1}$  and  $D^{t-1}$ , as well as  $I_o^{t-1}$  and  $D_o^{t-1}$ , and are fed into two separate, architecturally identical, image and depth fine encoders. In our final implementation, we set  $L = 3$ , thus, the fine encoders use three down-sampling layers. After separate encoding, image and depth features are fused and completed by a common completion bottleneck using dilation, followed by a ConvLSTM layer identical to the original implementation [64]. Afterward, they are fed into our structure-aware decoder with  $L = 3$  up blocks, whose architecture is described in Sec. 3.2. Finally, the architecture of our RGB and depth discriminators is identical and follows the dense, spectral-normalized patch discriminator (SN-PatchGAN) introduced in DeepFillV2 [83].

Hereafter, we denote kernel size, dilation, stride size and channel number as K, D, S, and C, respectively.

**Coarse generator:** K5S1C24 - K3S2C48 - K3S1C48 - K3S2C96 - K3S1C96 - K3D2S1C96 - K3D4S1C96 - K3D8S1C96 - K3D16S1C96 - K3S1C96 - K3S1C96 - up-sample(2) - K3S1C48 - K3S1C48 - up-sample(2) - K3S1C24 - K3S1C12 - K3S1C\*

**Refinement encoder:** K5S1C64 - K3S2C64 - K3S1C128 - K3S2C128 - K3S1C256 - K3S2C256 - K3S1C512

**Bottleneck:** concat - K3S1C512 - K3D2S1C512 - K3D4S1C512 - K3D8S1C512 - K3D16S1C512 - K3S1C512 - K3S1C512 - ConvLSTM

**Decoder:** up block C256 - up block C128 - up block C64 - K3S1C32 - K3S1C\*

In the output layers, the number of channels (C\*) is three for image outputs and one for depth outputs. We use gated convolutions [83], ReLU activation and instance normalization throughout convolution layers. Image output layers use the tanh activation function, while depth output layers clamp the output to  $[0, 1]$ .

#### 6.2. Details of training objectives

The SN-PatchGAN discriminators  $D_I$  and  $D_D$  are trained using Hinge loss [41], which is widely adopted for inpainting tasks. For an incomplete image and depth map  $I_m$  and  $D_m$ , and their ground truth counterparts  $I$  and  $D$ , the adversarial discriminator losses are

$$\mathcal{L}_{D_I} = -\mathbb{E}_{I \sim p_d} [\text{ReLu}(-1 + D_I(I))] - \mathbb{E}_{I_m \sim p_z} [\text{ReLu}(-1 - D_I(G_I(I_m)))], \quad (8)$$

$$\mathcal{L}_{D_D} = -\mathbb{E}_{D \sim p_d} [\text{ReLu}(-1 + D_D(D))] - \mathbb{E}_{D_m \sim p_z} [\text{ReLu}(-1 - D_D(G_D(D_m)))], \quad (9)$$

while the generator losses are defined as

$$\mathcal{L}_{adv,I}^G = -\mathbb{E}_{I_m \sim p_z} [D_I(G_I(I_m))], \quad (10)$$

$$\mathcal{L}_{adv,D}^G = -\mathbb{E}_{D_m \sim p_z} [D_D(G_D(D_m))]. \quad (11)$$

Here,  $p_d$  and  $p_z$  are the distributions of real data and the latent space, respectively, and  $\mathbb{E}_{I \sim p_d}$  denotes the expectation value of  $I$  with respect to distribution  $p_d$ .

The  $\ell_1$ -reconstruction loss on a pixel level is computed between ground truth images and depth  $I$  and  $D$  and the corresponding generator outputs  $I_o$  and  $D_o$  as

$$\mathcal{L}_{rec,I} = \|I - I_o\|_1, \quad (12)$$

$$\mathcal{L}_{rec,D} = \|D - D_o\|_1. \quad (13)$$

Perceptual and style loss operate in feature space, by computing the distance of  $i_{th}$  level features  $\phi_i$  of a pre-trained network, in our case, VGG-19 [68]. Specifically, perceptual loss is defined as

$$\mathcal{L}_{per} = \sum_i \frac{\|\phi_i(I) - \phi_i(I_o)\|_1}{N_i}, \quad (14)$$

where  $N_i$  is the number of elements in  $\phi_i$ , and style loss is given by

$$\mathcal{L}_{sty} = \|G_i^\phi(I) - G_i^\phi(I_o)\|_1, \quad (15)$$

with  $G_i^\phi$  being the Gram matrix constructed from activation  $\phi_i$ .

Our weight parameters are set empirically and according to literature [34, 42] as  $\lambda_{rec} = 10$ ,  $\lambda_{per} = 10$ ,  $\lambda_{sty} = 250$ ,  $\lambda_{grad} = 100$ ,  $\lambda_{seg} = 10$  and  $\lambda_t = 10$ .

#### 6.3. Hardware and training strategy

We implemented our model in PyTorch [56]. For training, we use an Nvidia Quadro RTX 8000 GPU, set the batch size to four and train for 1M iterations. We use an Nvidia GeForce GTX 1080 Ti for testing.



As training input, we select a series of  $T = 5$  consecutive RGB and depth frames with their semantic segmentation from our training datasets, together with randomly sampled object masks in the case of InteriorNet, and masks covering dynamic scene objects (pedestrians, vehicles) in case of DynaFill. For testing on InteriorNet and ScanNet, we set  $T = 100$  and use a fixed set of random object masks. Since DynaFill sequences have a shorter, varying number of frames, we set  $T$  according to the sequence length. We resize all inputs to a resolution of  $256 \times 256$  pixels during training and testing. Adam optimizer [30] with  $\beta_1 = 0.0$  and  $\beta_2 = 0.9$  is used for optimization, and we set the learning rate to  $2 * 10^{-4}$  for all modules. After 500k iterations, we reduce the learning rate to  $2 * 10^{-5}$ .

**Data augmentation.** Since InteriorNet already contains varying lighting conditions and different scene views, we do not apply additional data augmentation. Instead, to limit redundancy in the dataset, we sub-sample every fourth frame from each sequence during training. For DynaFill, we use the same data augmentation as in the original paper (brightness, contrast, saturation and hue modulation, as well as random horizontal flipping).

**Recurrent network training.** Recurrent network training incurs some additional computational costs, which we minimize through several strategies: Firstly, MaskFlowNet [92] for flow estimation is very lightweight with only 10.5 M params and 13.4 G MADs. Recent, more efficient flow estimation architectures like FastFlowNet [32] could further reduce the costs. Secondly, we perform sequence truncation by processing subsets of  $T = 5$ , while preserving the ConvLSTM hidden state for each sequence, which, according to our informal experiments, provides a good trade-off between capturing temporal dependencies and computational cost during training. In summary, recurrent training increases memory consumption by approximately 2.2 GB compared to single-image training, which we deem justified by the obtained quality gain (see Tab. 6). Importantly, the recurrent feedback loop has minimal impact on inference efficiency compared to other video inpainting methods (see Tab. 11).

**Obtaining semantic segmentations and depth.** We trained our models on synthetic datasets with accurate ground-truth segmentations and depth. Our experiments (Tab. 4, Fig. 10, Fig. 12) demonstrate their strong generalization to real-world data. As image segmentation techniques such as SAM [31] advance, obtaining high-quality segmentations from various image and video datasets will likely be feasible in the near future. Depth is usually available in our targeted mixed reality systems, as they require an understanding of their 3D surroundings. If not, recent

monocular depth estimation models [59] can be used to obtain depth.

## 7. Additional experiments

### 7.1. Analysis of RGB-D datasets for object removal

Datasets suitable for DR need to contain consecutive video frames of aligned RGB and depth. Our framework furthermore requires semantic segmentations: For supervising structural guidance during training, and for generating object masks for a convenient qualitative evaluation. The few works about fused RGB-D object removal [3, 11, 12, 58] use Structured3D [93], SceneNet RGBD [48] or DynaFill [3] (see Tab. 7). Structured3D does not fulfill the criteria of consecutive frames and we did not consider SceneNet RGBD due to its poor realism. Similarly, common depth completion benchmarks are not suitable, as shown in Tab. 8.

Table 7. Datasets used in related RGB-D object removal works.

	Segmentation	Consecutive frames	Photorealistic
Structured3D [93]	✓	✗	✓
SceneNet RGBD [48]	✓	✓	✗
DynaFill [3]	✓	✓	✓

Table 8. Common dense depth completion datasets and their suitability for DR.

	Segmentation	Consecutive frames	Available
NYU-depth V2 [66]	subset	subset	✓
Middlebury [20, 62]	✗	✗	✓
Matterport3D [6]	✓	✗	✓
VOID [74]	✗	✓	✓
DIODE [74]	✗	✗	✓
SUNCG [69]	✓	✓	✗

### 7.2. Comparison of different indoor depth completion methods

In total, we considered three methods designed for indoor and outdoor depth completion to fill missing depth regions in baselines that don’t handle depth: InDepth [90], NL-SPN [54] and DM-LRN [63]. These networks receive previously completed RGB and masked depth as input and are designed to leverage both RGB and depth features, with different methods to fuse them. Their goal is to fill missing depth based on *complete* RGB information, which is typically unavailable in DR. In Tab. 9, we compare their performance on our datasets, reporting root mean squared error (RMSE) for depth completion using inpainted color images from our baseline methods. Evidently, InDepth works

best for InteriorNet, while NLSPN performs best on ScanNet and DynaFill. We use these best-performing methods as baselines for the comparison of depth RMSE in Tab. 2, Tab. 3 and Tab. 4. Note that we did not consider works that complete depth from sparse measurements in different scenarios, *e.g.*, LiDAR-based depth completion in outdoor scenarios.

### 7.3. Influence of shadow mask

To fully diminish objects from a scene as if they were not there in the first place, it is also necessary to remove the shadow they cast. While automatic shadow segmentation remains a topic for our future work, we are interested in the performance of DeepDR in the case of a combined object and shadow mask. Thus, we have manually added shadow masks to the automatically derived object masks from InteriorNet and ScanNet. Fig. 9 provides visual results in order to demonstrate the performance of DeepDR for complete object and shadow removal. For comparison, we also provide results without a shadow mask. Apparently, DeepDR is capable of reliably inpainting shadow masks and moreover, results with masked shadows often look better than without. The reason for that is that our model does not need to hallucinate the very ambiguous shadow borders, leading to a more realistic color with fewer artifacts.

The same observation holds for the automatically derived object masks from ScanNet, which, due to inaccurate instance segmentation in the original dataset, sometimes do not cover the entire diminished object. In such cases, artifacts and flickering between consecutive frames can appear.

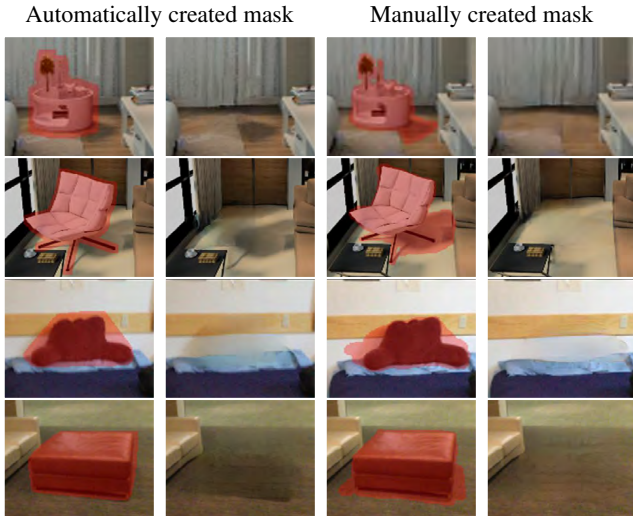


Figure 9. Comparison between automatically created mask using instance segmentations and manually created masks. DeepDR is also able to inpaint shadows, if they are appropriately masked.

### 7.4. Comparison with depth-from-RGB inpainting

As already mentioned in the main paper, depth inpainting literature mostly focuses on depth completion in regions that are visible in the corresponding RGB image. In-Depth, NLSPN and DM-LRN mentioned above are examples of such methods. While our task of inpainting hidden structures in diminished parts of the scene is fundamentally different from depth-from-RGB inpainting, it is compelling to draw comparisons between the two tasks. Therefore, we conducted an experiment applying DeepDR for depth-from-RGB inpainting on NYU-depth V2 [65] and ScanNet, which are common benchmarks in depth completion [2, 15, 55, 63, 72]. We directly compare to results reported in DM-LRN, which is the only work evaluating, but not trained on these datasets, allowing a fair comparison. We use the semi-dense sampling strategy reported in their paper to generate masks and feed complete RGB and masked depth to our framework. Results are shown in Tab. 10. The results on ScanNet show that DeepDR is effective in using visible RGB information to fill missing depth, resulting in an RMSE of 0.262 m compared to an RMSE of 0.484 m for the joint RGB-D inpainting task in Tab. 4. Still, DeepDR performs slightly worse on the depth-from-RGB inpainting task than the baseline method, particularly on NYU-depth V2, which is challenging for DeepDR due to its lack of consecutive frames.

## 8. Additional results

### 8.1. Computational complexity of ablation models

We analyze the computational complexity in terms of inference time, multiply-adds (MADs) and number of total parameters of our ablation models in Tab. 6. Evidently, RGB-D SPADE is the major driver of computational complexity, leading to an almost doubled inference time, as well as a significantly higher number of MADs and parameters. Our separate encoding strategy proves to be very efficient, improving the performance of our final model while decreasing the overall parameter count.

### 8.2. More qualitative results for the DR use case

Supplementary to the qualitative results in Fig. 5 and Fig. 6, we show more results of DeepDR in comparison to the baselines [3, 16, 39, 83] for DR object removal on InteriorNet in Fig. 10, DynaFill in Fig. 11 and on ScanNet in Fig. 12.

### 8.3. Qualitative results for 3D scene editing

While the importance of coherent image and geometry inpainting may not be immediately obvious, it becomes clear when looking at applications in 3D scene editing, such as interior re-design. In our indoor scene scenario, a typical use case is re-decorating rooms. To demonstrate this use case, we reconstruct a textured 3D mesh from the inpainted

Table 9. Root mean squared errors (RMSE) for different indoor depth completion methods based on color inpainting using our baseline methods. All measurements are given in meters.

	InteriorNet [37]			ScanNet [10]			DynaFill [3]		
Model	DeepFillV2	PanoDR	E2FGVI	DeepFillV2	PanoDR	E2FGVI	DeepFillV2	PanoDR	E2FGVI
NLSPN [54]	0.706	0.635	0.619	<b>0.508</b>	<b>0.536</b>	<b>0.512</b>	<b>7.92</b>	<b>8.12</b>	<b>7.83</b>
DM-LRN [63]	1.034	1.223	1.366	0.781	0.789	0.852	11.81	11.92	11.80
InDepth [90]	<b>0.572</b>	<b>0.564</b>	<b>0.563</b>	0.643	0.659	0.629	11.84	12.97	12.33

Table 10. RMSE in meters for D-from-RGB inpainting.

	NYU-d V2	ScanNet
DeepDR (ours)	0.281	0.262
DM-LRN	0.205	0.198

Table 11. Efficiency of DeepDR on a Nvidia GeForce GTX 1080 Ti GPU in comparison to the ablation models.

Model	Efficiency		
	Time ↓ (ms)	MADs ↓	Params ↓
no temporal	4.17	163.3 G	69.8 M
no RGB-D SPADE	<b>2.41</b>	<b>125.8 G</b>	<b>65.7 M</b>
joint encoder	4.42	174.6 G	71.1 M
DeepDR (Full model)	4.43	184.3 G	69.9 M

RGB-D pairs in 3D using pose and augment it with additional virtual light sources (Fig. 13a) and furniture or decoration objects (Fig. 13b).

As seen from these examples, incorrect depth inpainting leads to serious artifacts, such as ghost shadows or intersections and overlapping of the inpainted background with newly added objects. Since our method significantly outperforms related work in terms of depth inpainting, it does not cause such artifacts and is, therefore, best suited for 3D scene editing applications.

#### 8.4. Qualitative results using random object masks

As mentioned before, InteriorNet and ScanNet have no ground truth for the DR use case. Ideally, we would use training and testing pairs consisting of rooms before and after some items have been removed. Such data is very difficult to obtain in a real setting, but even synthetic data is costly to obtain, both in terms of computational and human resources as well as time. Therefore, for this datasets, we simulate the object removal task by overlaying random object masks over the scene and thus, the original image serves as ground truth. We use this strategy for both training and computing our quantitative results during testing. A qualitative comparison of inpainting using random object masks between our method and the baselines is given in Fig. 14 for

InteriorNet, and Fig. 15 for ScanNet.

Akin to the qualitative results for the DR use case, it is noticeable that DeepDR exceeds other methods in reconstructing sharp textures while preserving important structural properties of the scene. Furthermore, our method can reconstruct sharp depth edges, while the baselines fail to reconstruct the geometry of the scene, particularly for structures far away from the camera.

#### 8.5. Intermediate segmentation results

Our up blocks produce intermediate semantic segmentations of the scene at feature scale using a pyramid pooling module [91]. These maps are used to modulate the activations during decoding to ensure sharp and coherent boundaries in RGB and depth outputs. In Fig. 16, Fig. 17 and Fig. 18, we show these intermediate segmentations from each of the three up blocks in our final architecture. It is evident that the segmentation accuracy improves with higher feature dimensions. Notably, segmentation on DynaFill (Fig. 17) is more accurate, which we attribute to the lower variability and smaller number of semantic classes (12 vs. 40) in the dataset. Although our network does not produce perfect segmentations, it is able to accurately reconstruct clean object borders and plausible semantics, which leads to sharp edges and coherent textures in the resultant image and depth outputs. Still, in particular, on unseen, real data in ScanNet (Fig. 18), some regions are incorrectly classified, which might decrease the effectiveness of RGBD SPADE. We aim to overcome this limitation by fine-tuning our models on real data, reducing the number of semantic classes by merging similar classes, and by exploring RGB-D semantic segmentation strategies [4, 9, 76] to leverage depth information more effectively for intermediate semantic segmentation.



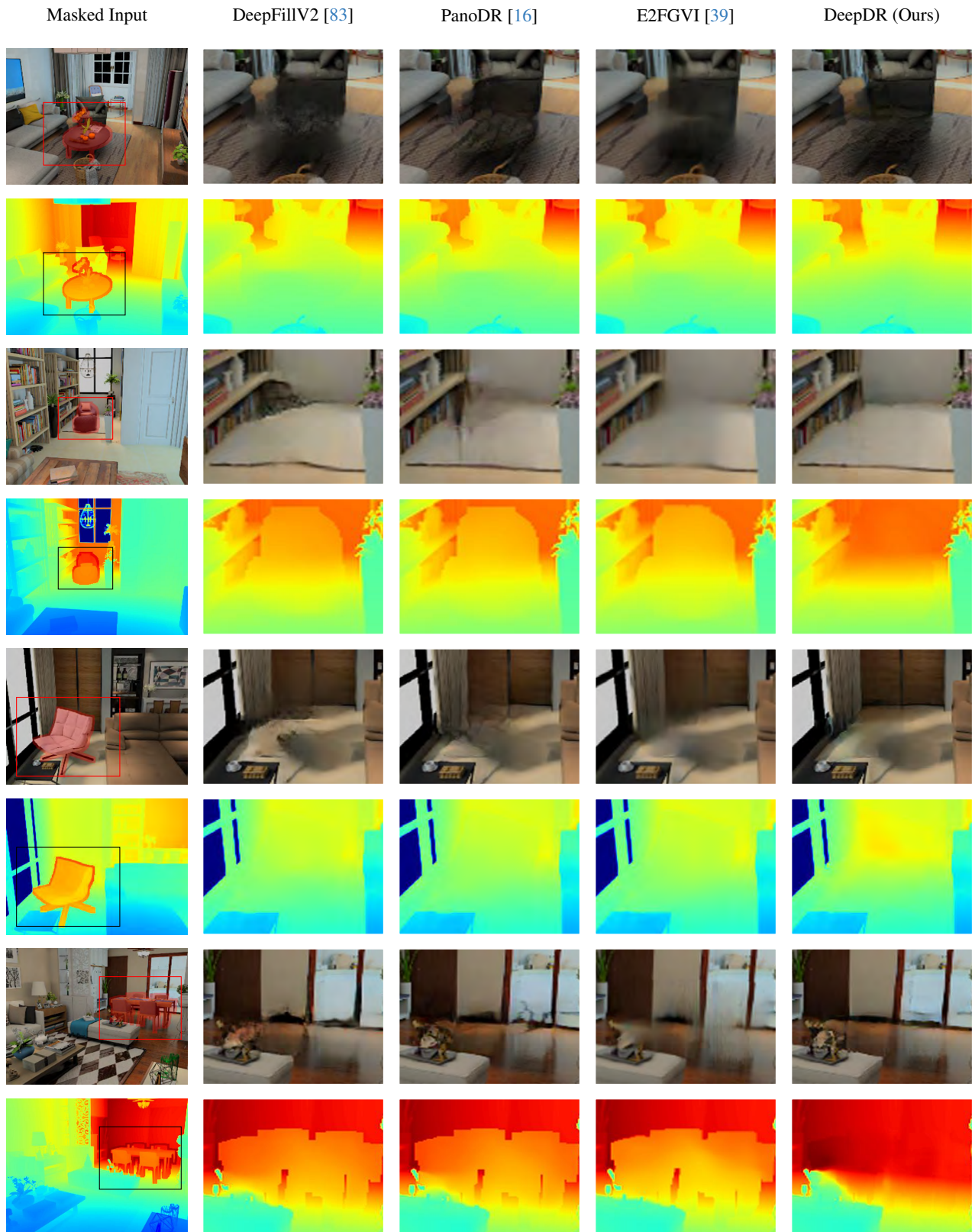


Figure 10. Qualitative comparison of color images and depth maps for diminishing objects from InteriorNet [37].



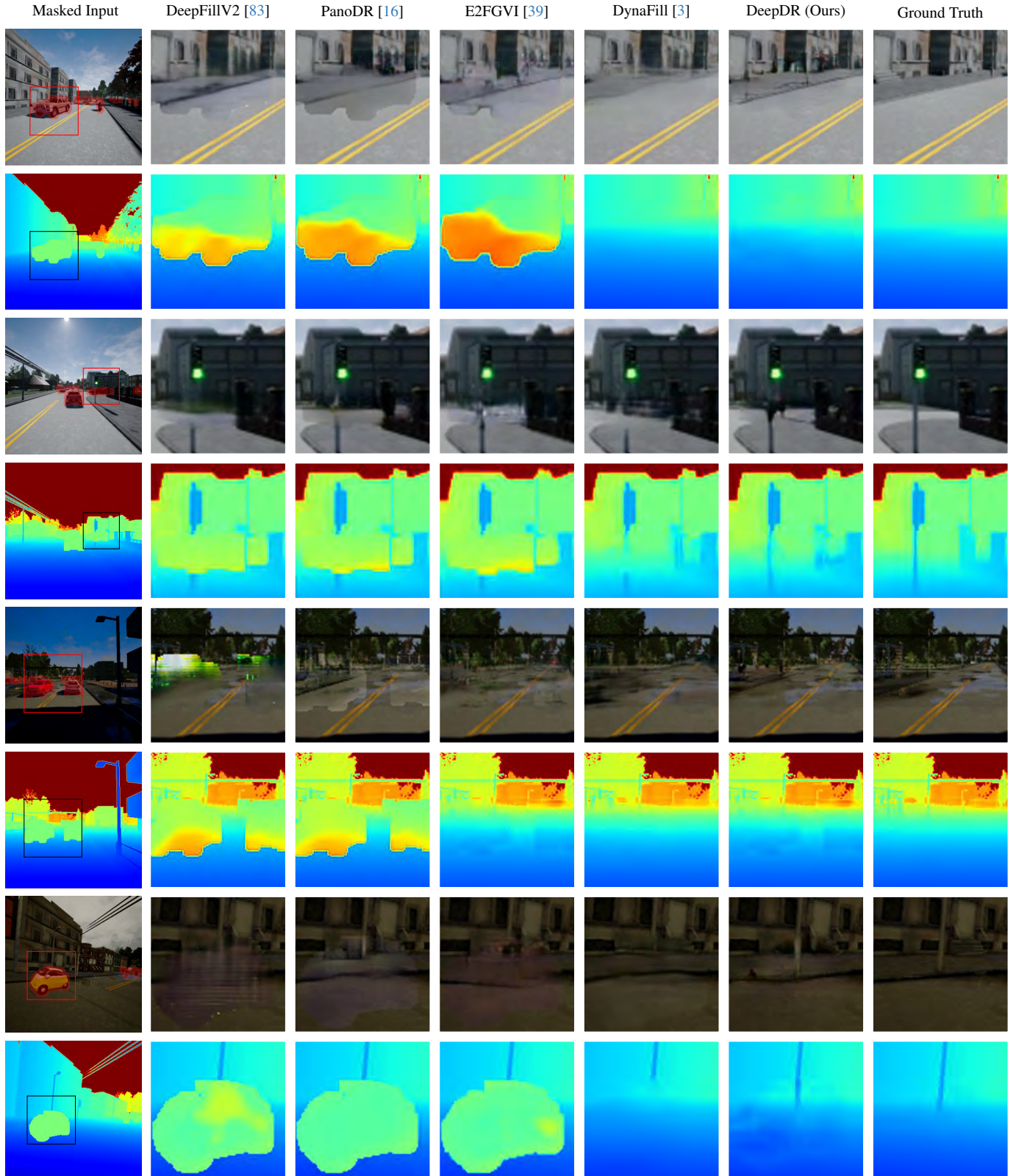


Figure 11. Qualitative comparison for diminishing objects from DynaFill [3].

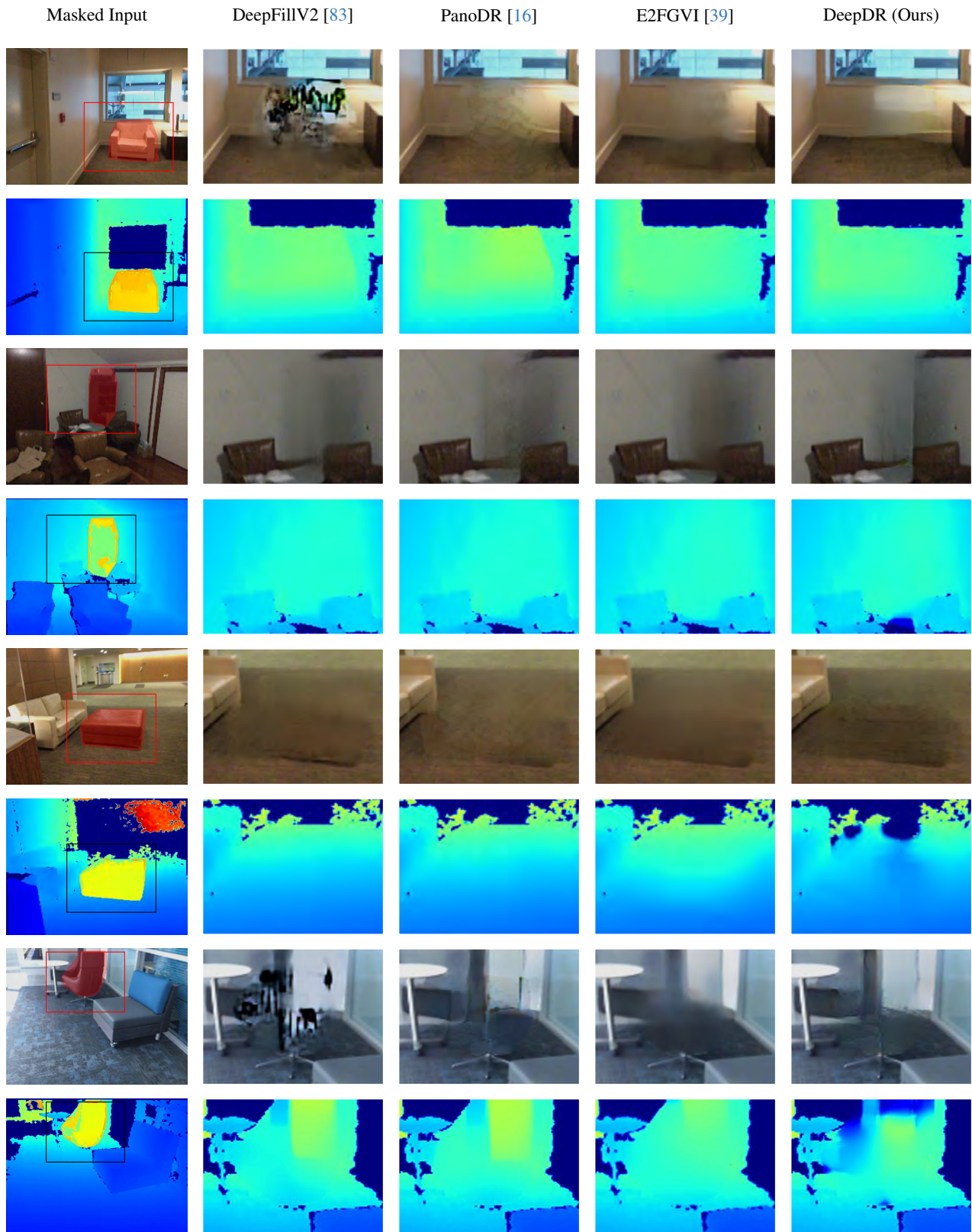


Figure 12. Qualitative comparison of color images and depth maps for diminishing objects from ScanNet [10].



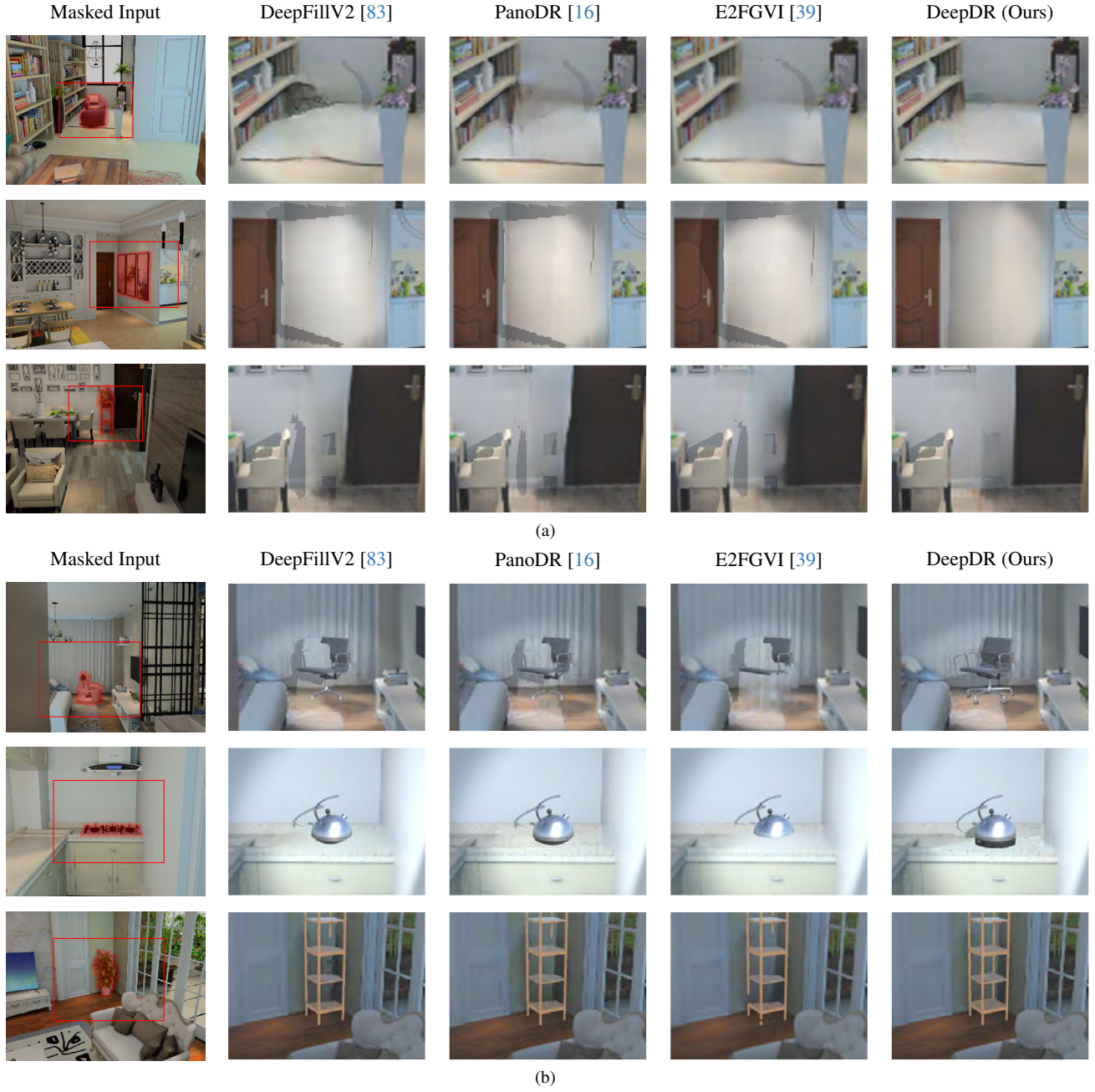


Figure 13. Qualitative comparison for 3D scene editing after diminishing objects via inpainting. The scene is reconstructed in 3D, light sources are added (a) and furniture or accessory items are replaced (b).

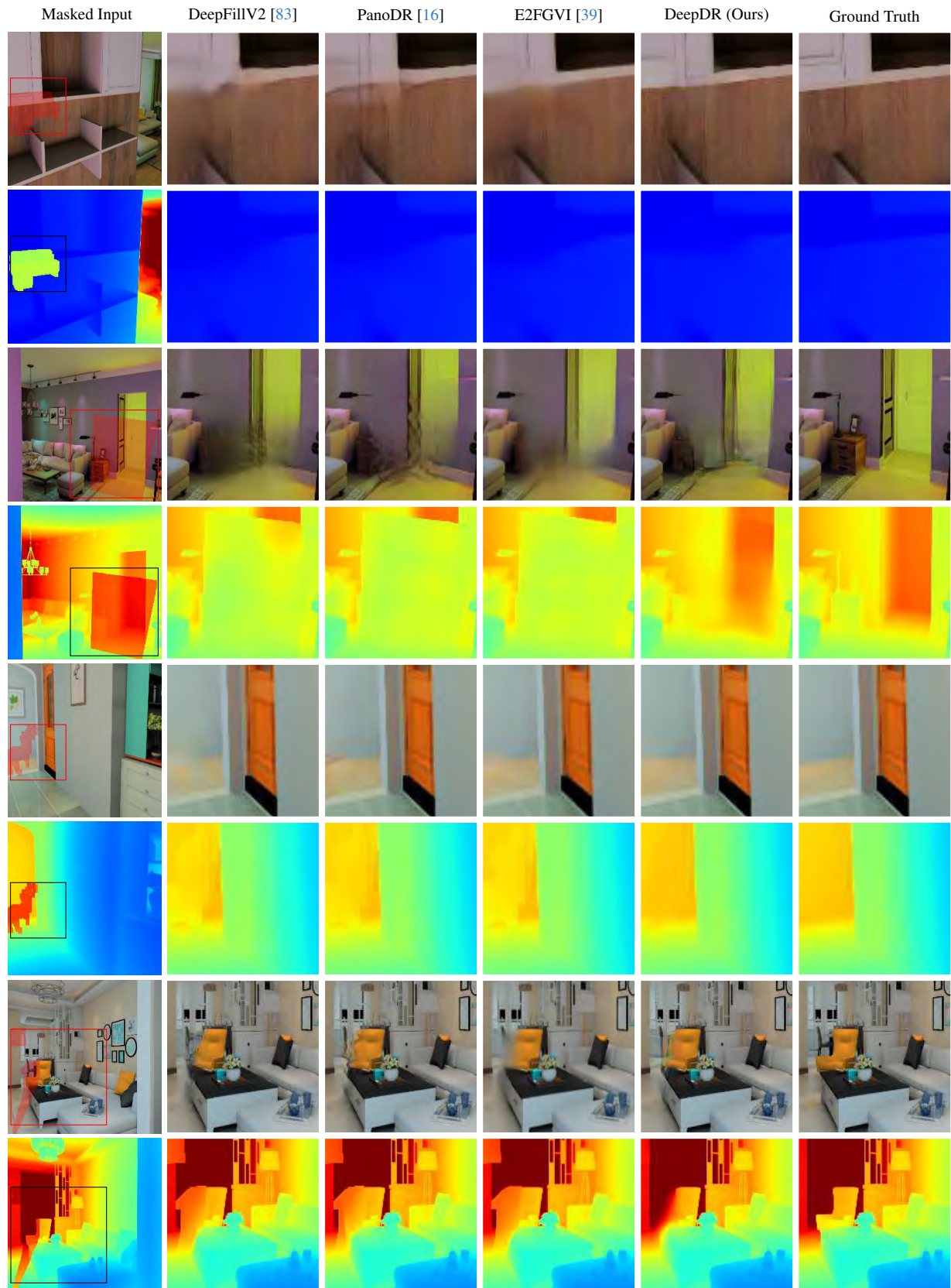


Figure 14. Qualitative comparison on InteriorNet [37] for inpainting random object masks.



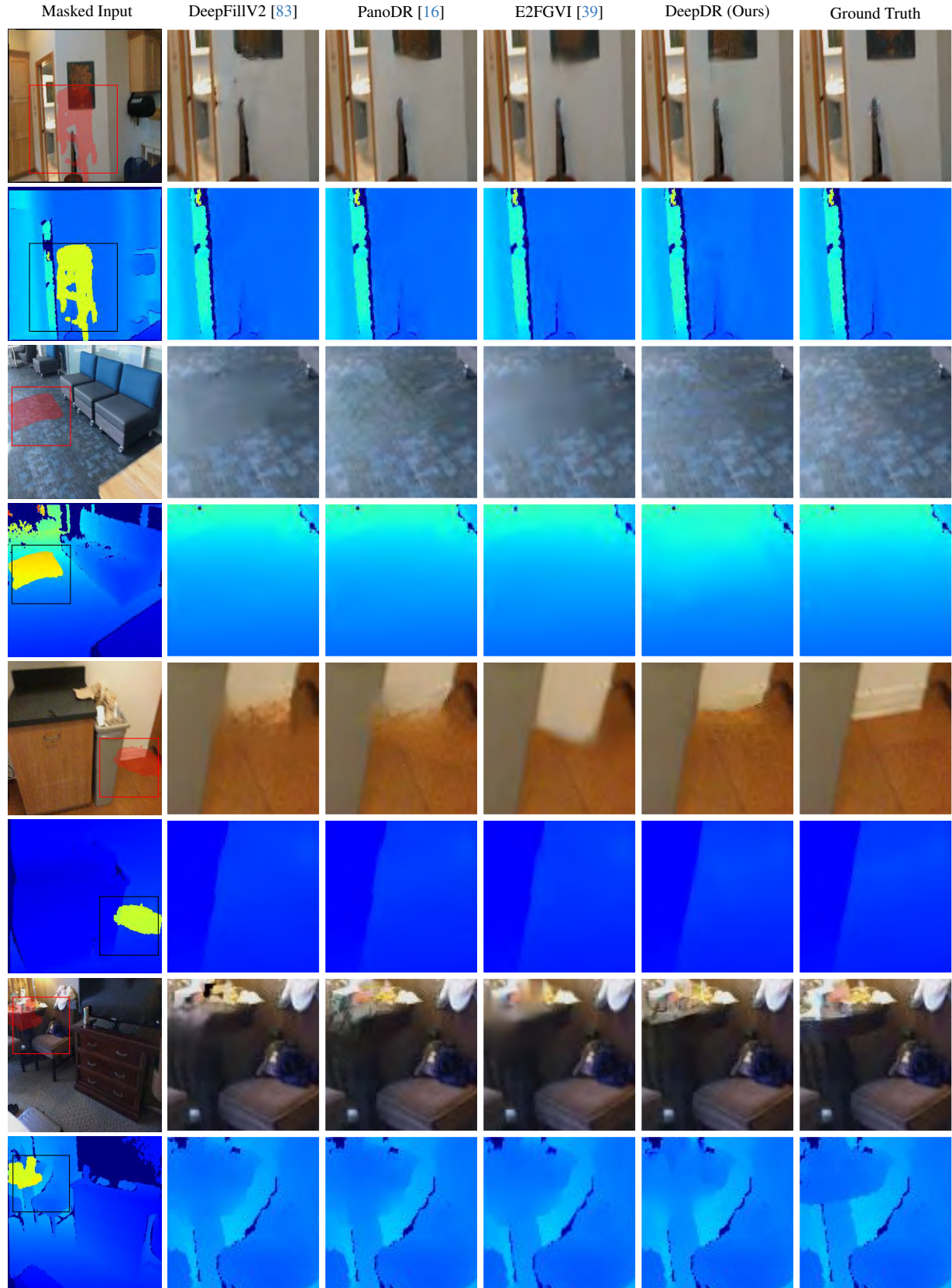


Figure 15. Qualitative comparison on ScanNet [10] for inpainting random object masks.



Figure 16. Analysis of the semantic segmentations produced within our up blocks on InteriorNet [37].



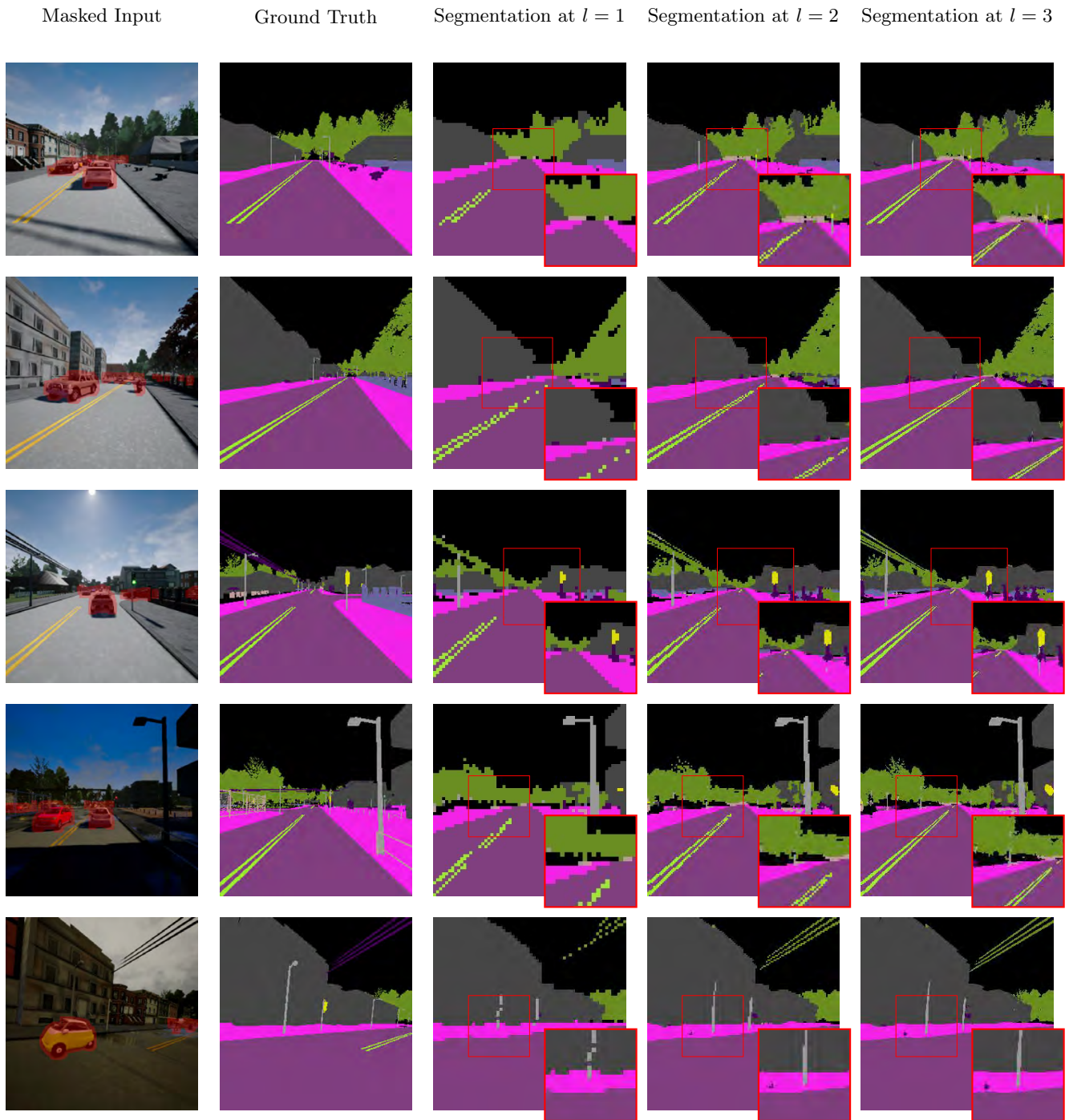


Figure 17. Analysis of the semantic segmentations produced within our up blocks on DynaFill [3].

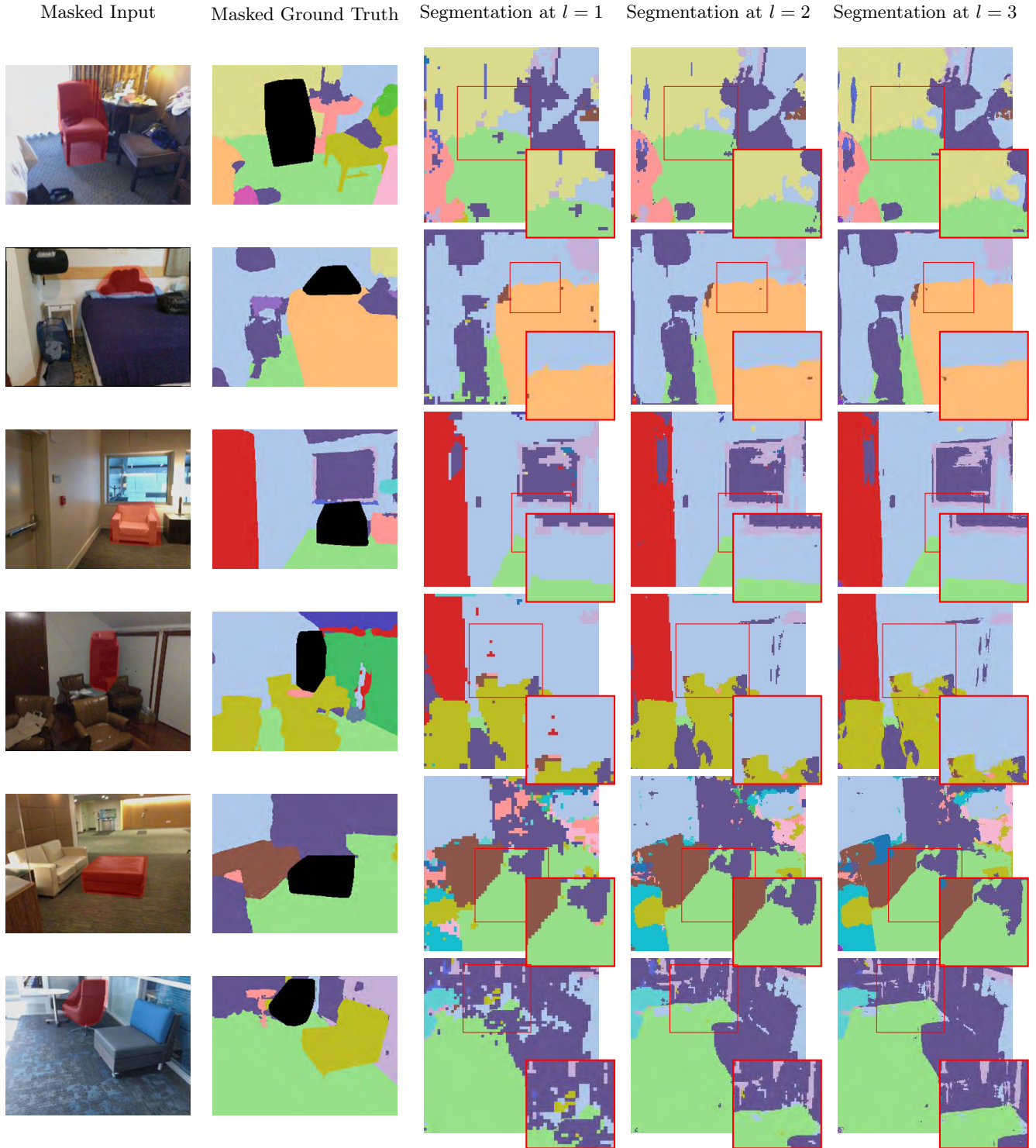


Figure 18. Analysis of the semantic segmentations produced within our up blocks on ScanNet [10].