Instant Automatic Emptying of Panoramic Indoor Scenes

Giovanni Pintore, Marco Agus, Eva Almansa, and Enrico Gobbetti

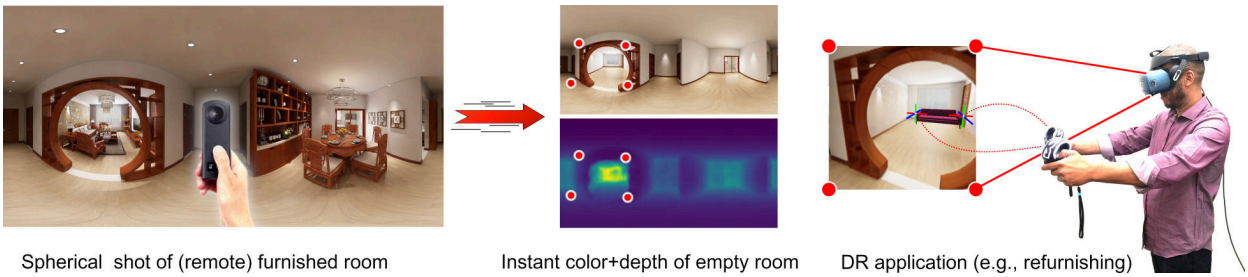


Fig. 1: Given a 360 panoramic photo of a cluttered indoor scene, our end-to-end approach automatically returns a photorealistic view and depth of same scene emptied of furniture and clutter. Both visual appearance and depth, estimated at interactive speed, are highly suitable for compelling and immersive XR applications, such as (re-)furnishing or planning of interior spaces.

Abstract—Nowadays 360° cameras, capable to capture full environments in a single shot, are increasingly being used in a variety of Extended Reality (XR) applications that require specific Diminished Reality (DR) techniques to conceal selected classes of objects. In this work, we present a new data-driven approach that, from an input 360° image of a furnished indoor space automatically returns, with very low latency, an omnidirectional photorealistic view and architecturally plausible depth of the same scene emptied of all clutter. Contrary to recent data-driven inpainting methods that remove single user-defined objects based on their semantics, our approach is holistically applied to the entire scene, and is capable to separate the clutter from the architectural structure in a single step. By exploiting peculiar geometric features of the indoor environment, we shift the major computational load on the training phase and having an extremely lightweight network at prediction time. Our end-to-end approach starts by calculating an attention mask of the clutter in the image based on the geometric difference between full and empty scene. This mask is then propagated through gated convolutions that drive the generation of the output image and its depth. Returning the depth of the resulting structure allows us to exploit, during supervised training, geometric losses of different orders, including robust pixel-wise geometric losses and high-order 3D constraints typical of indoor structures. The experimental results demonstrate that our method provides interactive performance and outperforms current state-of-the-art solutions in prediction accuracy on available commonly used indoor panoramic benchmarks. In addition, our method presents consistent quality results even for scenes captured in the wild and for data for which there is no ground truth to support supervised training.

Index Terms—Mediated and diminished reality, Omnidirectional, 360, Real-time performance issues, AR/MR/VR for architecture, Computer vision, Machine learning

1 Introduction

Current 360° cameras offering viable low-cost and energy-efficient solutions for full-context single-shot capture are increasingly popular in many application fields [11]. Since the captured 360° content, also known as *panoramic*, *spherical*, or *omnidirectional* imagery, covers the entire sphere around the viewer, even a single shot cannot be statically experienced at once, making it fundamentally different, more immersive and more dynamic, than traditional 2D imagery [5]. In particular, when consumed through Head-Mounted-Displays (HMDs), the user actively focuses on the desired content via natural head movements, just like humans do in real world, achieving a very high degree of immersion [52]. For this reason, omnidirectional imagery is becoming a fundamental component for creating immersive content from real-world scenes, and for supporting a variety of Virtual Reality (VR) applications [23]. Notably, virtual tours based on spherical images are extremely popular in the real estate domain, and have rapidly increased their appeal in the pandemic period [45]. A pure exploration of existing environments through the original spherical photos, is, however very limiting. Prominent examples of additional needs include the emptying of rooms before their presentation to virtual visitors (if only for privacy reasons), or the refurnishing or redecorating of interior spaces [59]. In this context, fast and effective Diminished Reality (DR) techniques, which conceal real-life parts from the view field, are paramount to remove the furniture and other clutter that masks the architectural structure. In particular, DR features are essential to allow users to immediately compare the furnished and unfurnished scene, and to support Augmented Reality (AR) applications in placing objects in the empty scene [42, 43]. Making these features available on novel environments with minimum latency, ideally in real-time, would, in addition, enable their usage in remote collaboration contexts, without the need for prior modeling [48].

While a variety of object erasing and image inpainting solutions have been presented in the literature (Sect. 2), DR for interior environments must generate images of empty indoor spaces that not only have a realistic appearance, but respect the context in stricter ways, in particular by inferring a plausible organization of the permanent architectural structure that bounds the room's interior [4]. Data-driven solutions, that learn hidden relations from examples, are emerging as viable approaches for this class of problems. However, state-of-the-art methods for image inpainting are mostly focused on photorealism [57, 62], and additional information about the scene is exploited only from the semantic point-of-view [4, 30, 64]. Current pipelines make limited use of the structure of the observed scene, and reconstruction accuracy is achieved at the price of high computational complexity or increased user intervention, using, for example, recursive networks [18], multi-branch architectures [62], and manual definition of specific parts of the

- *G. Pintore, Eva Almansa, and E. Gobbetti are with CRS4, Italy. E-mail: giovanni.pintore@crs4.it, eva.almansa@crs4.it, enrico.gobbetti@crs4.it*
- *M. Agus is with HBKU, Qatar. E-mail: MAgus@hbku.edu.qa*

original image to be removed [30].

In this work, we present a novel light-weight end-to-end deep network that, from an input 360° image of a furnished indoor space automatically returns, with very low latency, an omnidirectional photorealistic view and architecturally plausible depth of the same scene emptied of all clutter.

By harnessing the availability of large scale, photorealistic synthetic datasets, we train our network on pairs using a set of examples composed of registered equirectangular images of the cluttered environment color, the empty environment color, and their depth. The final end-to-end network is decomposed in two blocks, which are trained separately to reduce training costs. The first block learns an attention mask of the uncluttered parts of the input image, generating training examples from the cluttered input image and the depth pairs. The second block takes as input the attention mask and the cluttered image, and performs the synthesis of the uncluttered scene, using for training indoor-specific losses that embed our knowledge of expected indoor environments. Contrary to other object removal approaches, our approach is holistically applied to the entire scene, removing all clutter in a single step without user intervention. Rapidly emptying the room without manual intervention is the essential building block upon which the other features required for a DR application. For instance, removing a single object (or keeping only a single object) is achieved by compositing the empty room image from our network with the original image, while taking into account the computed object mask (see, e.g., the design of Gkiktas et al. [4]). Moreover, by inferring the room's geometry while removing clutter, we provide support various scene edits, including adding/positioning furniture while resting on floor or attached to a wall (see Fig. 1).

Our main contributions are summarized as follows:

- We propose a light-weight end-to-end deep-learning technique (Sect. 3), which provides, at interactive rate, a panoramic indoor scene emptied automatically without user intervention and suitable for use in XR applications. Our prediction network develops in a linear fashion, with no need to fuse features from parallel branches [4, 62], or to refine the result recursively [18]. In order to alleviate the burden of convolutional gating for generic user-assisted inpainting [57], we adopt instead a depth-separable gated convolution strategy, reducing the number of parameters and processing time while maintaining the effectiveness [53]. Furthermore, both visual and geometric constraints are applied only at training time, where the visual ones follow a strategy of transfer learning [6] and the geometric ones adopt robust and efficient losses that encode our prior knowledge on interior environments (Sect. 3.3).

- We predict a geometric representation paired with the output image, that is a dense depth estimation of the empty scene. This geometric representation can be directly used as a basis for further processing in XR application (e.g., to aid object positioning or to compute occlusions). It is obtained jointly with the visual representation and without the need of onerous parallel branches [4,62]. We also exploit it to define a robust and effective pixel-wise prior together with other 3D priors and losses (Sect. 3.3) The generation of a geometric clue as output reduces the need to add additional semantic analysis on the image or to use GAN strategies [8,25] to disambiguate the results obtained, as demonstrated by our results (Sect. 4). By contrast, current inpainting methods are mainly focused on the visual and perceptual output [57,62], where structure preservation is handled at image-feature level [17] or semantically [30, 64]. Other approaches are based, instead, on manual and simplified annotations of the underlying layout, which does not necessarily represent the true 3D geometry. This information is best interpreted as a 2D semantic prior rather than a geometric one [4, 64].

- We drive our training using a loss function that combines photorealistic and geometric terms. In particular, our geometric terms exploit both pixel-wise information from depth maps and the concept of virtual normals generated by triples of points at a large distance [54], to efficiently recover the salient characteristics of man-made indoor structures, in terms of flatness and smoothness, without falling into restrictive structures such as Manhattan World, Atlanta World or even vertical walls [36].

Our results show that our method outperforms current state-of-the-art approaches, using common benchmarks with a measurable ground truth, in terms of accuracy, quality and less computational complexity (Sect. 4.3). Moreover, our model is also able to produce compelling predictions even on images from common datasets where no ground truth is available for training, as well as on novel images captured by an user (Sect. 4.4).

2 Related Work

DR for indoor scenes builds on techniques for data-driven inpainting and image-to-image translation, and must extend them in order to produce a realistic and geometrically consistent environment, eventually estimating the depth of the uncluttered scene. In the following, we focus on the methods that are most closely related to ours.

Diminished reality for indoor spaces DR applications provide the illusion of concealing, eliminating, and seeing through objects while perceiving an environment. In contrast to AR and MR, which superimpose virtual objects to real-world representations, they require techniques to detect the unwanted objects and replace them with the hidden background in generated images. In most DR applications, the objects to be removed are already determined as targets of interest, and specific techniques are employed for their detection (e.g., pedestrians [13] or buildings [47]). In indoor spaces, the most basic operation is the removal of interior clutter (furniture and other non-permanent objects) [4, 40, 42, 43, 59], which is supported either through interactive mask definitions (e.g., [4]), or through semantic or instance segmentation (e.g., [64]). In this work, instead, we learn, from synthetic examples, a geometric definition of clutter, that includes anything with an appreciable geometric volume that is not part of the permanent architectural structure. A wide variety of approaches have been proposed in the literature for synthesizing the hidden background (see [26] for a comprehensive survey). A number of methods employ reprojections of actual background images, generated through a prior observation of the same scene [27, 38] or a concurrent observation from other points of view, e.g., by employing multiple cameras [24]. Since these approaches require considerable effort and/or specialized setups, much research has focused, instead, on generating plausible background rather than recovering actual ones. Early solutions recovered background textures from the same image, especially analyzing areas nearby removed objects (e.g., [12]). Since these methods are generally limited to small holes and fairly regular scenes, the focus has recently shifted towards data-driven solutions that learn from a large body of prior examples. We follow this trend by generating a plausible background of a novel scene using a single 360° observation, exploiting concepts from data-driven inpainting and image-to-image translation, recovering not only the color but also the geometry of the empty scene in the form of a depth map. Shape inference is very important for DR of indoor environments, since it improves texture reprojection [12, 28] and parallax effects [1], and offers a basis for the editing operations [19, 59]. However, prior DR solutions either expected a simplified geometry in the hidden area (e.g., a plane [12, 28]) or required particular capture setup (e.g., multi-view [24, 27, 38] or one or more RGB-D cameras [19, 40, 59]).

Data-driven inpainting The first data-driven inpainting approaches combined auto-encoders with an adversarial loss [32] or global and local discriminators [7] to produce photo-consistent images. They used regular or dilated convolutions [55] combining valid and masked parts of the image, thus leading to visual artifacts such as color discrepancy and blurriness. To overcome such limitations, Liu et al. [20] introduced *partial convolutions* to handle masking effects. Later, the partial convolution concept was revisited to incorporate structural information (edges) in the reconstructed feature map [17]. Recognizing the importance of edge preservation and generation, *EdgeConnect* [29] introduced an edge generator to hallucinate edges in the missing regions, to use them as structural guidance for the inpainting task. All

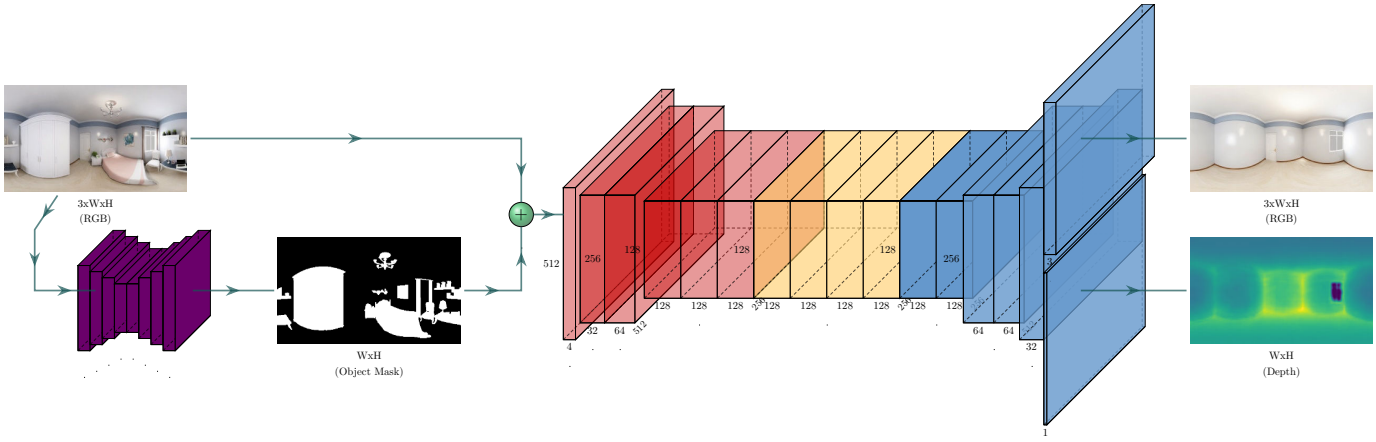


Fig. 2: **Model architecture.** We process the input equirectangular image to identify the cluttered area in the scene, exploiting a light-weight network (purple blocks - Sect. 3.1). The clutter mask and the input image are passed to the *empty scene synthesis network* (Sect. 3.2), including a gated encoder (red blocks), a dilation bottleneck (yellow blocks) and a gated decoder (blue blocks), whose last layer is split in 2 layers: one for the photorealistic equirectangular representation of the emptied scene and one for its depth. The scene synthesis network is trained end-to-end through the methods and losses described in Sect. 3.3.

the above methods assume that the mask is given and concentrate on the infilling part. Yu et al. [57] further extended the idea of partial convolutions by proposing gated convolutions to learn a mask automatically from a large number of examples. Combined with SN-PatchGAN [16, 25], this approach showed the ability of effectively supporting free-form user input as guidance. We also exploit gated convolutions, but learn to separate clutter from architectural structure using examples that exploit the availability of ground truth depths, without resorting to user input in any of our phases. In parallel developments, several authors have shown the importance of feature fusion at different scales, including the pyramid-context encoder approach [58] and the mutual encoder-decoder [21]. Conversely, Li et.al [18] propose a recurrent (i.e., iterative) method to inpaint missing regions from the outer regions of the hole towards the inner ones. Thanks to the data-driven design, the method is superior to previous techniques that assumed that gaps should be filled with similar content to that of the background, and can hallucinate new content for large holes. Zheng et al. [62] further extended the exploitation of global relations by designing a framework to generate multiple plausible results with reasonable content for each masked input, based on a probabilistic approach. To achieve that, they combine generative and variational synthesis approaches. None of the above methods, however, is applied to 360° imagery and exploits and generates geometric data.

Image-to-image translation DR can be also recast as an image-to-image translation problem [8], as it maps the input cluttered image to the output uncluttered image. Visual content and style preservation is very important in this context [3, 10]. Isola et al. [8] proposed conditional GANs as a general solution to various translation problems, semantic image synthesis being the most related to ours. The classic approach is to use semantic labeling, and reconstruct images from the semantic maps, preserving boundaries among classes [2]. Conserving semantic information fed to a deep layer built by stacking convolutional, normalization, and non-linear layers is, however, difficult, since normalization layers tend to blur semantic input. For this reason, Park et al. [30] introduced spatially adaptive normalization, in which the input map is exploited for modulating the activation in normalization layers through a spatially-adaptive, learned transformation. The approach has been recently extended by introducing per-region style encoding and allowing the user to select a different style input image for each semantic region [64]. Very recently, *PanoDR* [4] applied the above method to panoramic images of indoor scenes. In their approach, a pixel-wise semantic prior maps each pixel to the ceiling, wall, or floor class, while inpainting is performed exploiting a SEAN module [64]. As for classic inpainting methods, all these techniques are focused on the perceptual aspect and, in order to improve the realism of prediction, they exploit clues of image structure consistency (boundaries, edges), additional semantic information [30, 64] or user input [57]. Using such additional information mainly involves feature fusion from parallel branches [4, 62] or refining the result recursively [18], increasing the computational cost of the methods. In our approach, instead, we propose a linear pipeline avoiding feature fusion and recursion, leveraging the fact that the scene to be reconstructed has a specific geometry, and exploiting such an information only at training time. Furthermore, such geometric constraints reduce the need to use adversarial losses [8, 25] to disambiguate the results obtained, simplifying model structure and training approach. This leads to novel contributions in terms of network structure and loss functions.

Uncluttered depth estimation In order to integrate models into the empty scenes, or to perform geometric measures, DR requires shape information in addition to color. We do that by also inferring the depth map of the uncluttered image. Learning-based monocular depth estimation was introduced over a decade ago (e.g., Make3D [41]), and the emergence of deep learning, as well as the availability of large-scale 3D datasets, has contributed to significant performance improvements. Since the restricted field of view (FOV) of conventional perspective images inevitably results in a limited geometric context [61], much of the research on reconstruction of indoors from sparse imagery is now focused on inferring depth from a single omnidirectional interior image [33, 46, 50]. While these methods have been shown to cope with large amounts of clutter, they target the generation of the visible depth of the fully cluttered viewed room, rather than the estimation of the depth of the uncluttered one. For this reason, specific approaches for inferring the architectural layout are being actively researched. As noted by Zou et al. [65], most current data-driven layout reconstruction methods basically follow a pipeline that, based on specific indoor assumptions (e.g., Manhattan World), predicts layout elements in image space, followed by a post-processing for fitting a regularized 3D model to the predicted 2D elements. Recent solutions fully working in 3D [34] produce an approximate result in the form of a low-poly 3D mesh. By contrast, our approach, differently from all prior approaches strives to produce a per-pixel *uncluttered depth*, within a single-branch light-weight network that also produces the uncluttered color.

3 Methods

The overall architecture is illustrated in Fig. 2 and explained in the following sections. We first process the input image (i.e, equirectangular format) to identify the cluttered area in the scene, exploiting a separately-trained light-weight network (purple blocks in Fig. 2), described in Sect. 3.1. The returned clutter mask and the original input

image are then passed to the *empty scene synthesis network*, described in Sect. 3.2 (main network in Fig. 2), which returns the photorealistic equirectangular representation of the emptied scene and its registered depth. The scene synthesis network is trained end-to-end through the methods and losses described in Sect. 3.3.

3.1 Clutter mask prediction

The first stage of our method consists of an identification of the area containing the clutter that should be removed from the image ${I_f}^{h\times w}$ to generate the empty room color and depth panoramas. This identification consists of a binary mask $M^{h\times w}$, that contains 1 for pixels containing clutter, and 0 otherwise. Contrary to many current DR approaches, that are oriented to the removal of single objects, we do not expect that users define it interactively [4], eventually supported by object recognition and segmentation systems [64], but learn how to generate this mask directly from I_f through a lean segmentation network. We do that since we want our mask to identifies all the non-permanent structures that need to be removed at the same time, differentiating them from the architectural layout of the room.

For training, we exploit the large body of information provided by recent large-scale photorealistic synthetic datasets [63], which contain the registered representation of empty and non-empty rooms. Even though, in this paper, we only exploit the differentiation between clutter and layout, for maximum generality, we cast our problem as a classification problem, since many datasets contain, for each pixel also the type of object and/or the type of layout surface (ceiling, floor, wall). Such a classification might be of interest for reconstruction or when wanting to remove only particular kinds of objects.

For the present paper, we only consider a two-class situation (layout=0, clutter=1), that can be generated, in absence of annotations in the source datasets, by simply comparing the ground-truth depths of the empty (${D_e}^{h\times w}$) and non-empty (${D_f}^{h\times w}$) room representations, including in the clutter class the pixels for which $D_f < D_e$ and to the layout class all others.

With this approach, we define as clutter the portions of the environment that have an appreciable geometric volume in the room, but are not part of the bounding architectural structure of the room. Flat objects such as electric outlets or decorations (or mirror images) thus appear in the empty room by design. Such a definition is also commonly adopted for indoor structured reconstruction approaches [35, 37]. This choice avoids the need for semantic annotations, and lets the system learn a stable association between color and geometric shape using a completely automatic method using commonly available datasets. This approach does not exclude a combination with semantic information (e.g., [64]) to also remove flat objects.

We predict our full-empty mask from the image ${I_f}^{h\times w}$ as a dual channel probability map ${D_m}^{2\times h\times w}$ (i.e., full and empty channel), using a very lightweight encoder-decoder network based on the U-Net architecture, using just 256 channels as bottleneck (i.e., $4M$ parameters) and skip-connections [39].

The training of this network, the purple one in Fig. 2, is performed independently from the image synthesis network, as we experienced that training the clutter mask network simultaneously with the image synthesis network produces little or no advantages, but imposes an additional load on the entire training process (see Sect. 4).

For each of the two channels, training of the clutter mask is driven by binary cross-entropy loss:

$$-\frac{1}{n}\sum_{p\in {D_m}^c}(\hat{p}\log p+(1-\hat{p})\log(1-p)) \tag{1}$$

where ${D_m}^c$ is the slice c of D_m, p is the predicted probability of one pixel of being of class c, and $\hat{p}$ is the ground truth probability. The final predicted binary mask $M^{h\times w}$, that feeds the second stage of our complete network, is obtained by assigning each pixel to the class with maximum probability and setting the pixel value according to this classification.

3.2 Empty scene synthesis

To generate the empty scene image and depth, we adopt the architecture illustrated in Fig. 2. The overall encoder-decoder scheme follows a common design for image inpainting [7], exploiting dilated convolutions as bottleneck [55], and gated convolutions for encoding decoding [56]. Compared to the baseline [7, 56], our architecture is thinner, deeper, and with fewer parameters. Moreover, it has only a single branch and it includes several solutions (described below) to improve accuracy and reduce computational complexity. Furthermore, given the spherical nature of the image, we adopt circular padding along the horizon for convolutions, thus removing longitudinal boundary discontinuity, and reflection padding to alleviate the singularities at the poles [4].

The input of the network consists of a masked image of the cluttered room, with white in clutter regions, together with the binary mask indicating the hole regions (Sect. 3.1). The paired input is encoded through a sequence of light-weight gated convolutions having different strides (the 6 layers in red in Fig. 2), so that the original size is reduced by a factor four in each direction. Each encoding convolution is followed by instance normalization [49] and ReLU activation.

In our network, we adopt a specific form of gated convolution, that integrates a learnable gating technique when selecting features [56], since vanilla convolutions are ill-fitted for image inpainting [7, 56].

Considering a standard convolutional layer and a $C_{in}-channel$ input feature map, each pixel located at (y,x) in the $C_{out}-channel$ output map is computed as:

$$O_{y,x}=\sum_{i=-k'_h}^{k'_h}\sum_{i=-k'_w}^{k'_w} W_{k'_h+i,k'_w+j}\cdot I_{y+i,x+j} \tag{2}$$

where x, y represents the location along the x- and y-axis of the output map, k_h and k_w is the kernel size (e.g. 3×3), $k'_h=\frac{k_h-1}{2}$, $k'_w=\frac{k_w-1}{2}$, $W\in\mathbb{R}^{k_h\times k_w\times C_{in}\times C_{out}}$ are convolutional filters, and $I_{y+i,x+j}\in\mathbb{R}^{C_{in}}$ and $O_{y,x}\in\mathbb{R}^{C_{out}}$ are inputs and outputs. The application of the same filters at each spatial location (y,x) is not appropriate. This is because, for inpainting, the input will need to combine valid pixels/features coming from regions outside holes with invalid pixels/features (in shallow layers) or synthesized pixels/features (in deep layers) coming from masked regions [57]. Although simple partial convolutions [20] can be used to make the convolution dependent only on valid pixels, they are not suitable for our problem, since, essentially, they act as single-channel hard-gating.

Thus, we adopt a gated convolution (GC) approach [57], expressed as:

$$\begin{aligned} G &= conv(W_g, I)\\ F &= conv(W_f, I)\\ O &= \sigma(G)\bigodot\psi(F) \end{aligned} \tag{3}$$

where σ is the Sigmoid function, which outputs values in $[0,1]$, ψ is an activation function (ReLU in our case), and W_g and W_f are two different sets of convolutional filters, which are used to compute the gates and features respectively. GC enables the network to learn a dynamic feature selection mechanism. It should be noted that, according to Equation 2, W_g has $k_h\times k_w\times C_{in}\times C_{out}$ parameters, almost doubling the number of parameters and processing time in comparison to vanilla convolution. In order to simplify training and guarantee low latency at inference time, our network uses a modified version of GC called Light Weight Gated Convolutions (LWGC), which reduces the number of parameters and processing time while maintaining the effectiveness [53]. Specifically, we decompose G from Equation 3 into a depth-wise convolution [53] (i.e., 3×3) followed by a 1×1 convolution, having, as a result, the same gating step but with only $k_h\times k_w\times C_{in}+C_{in}\times C_{out}$ parameters.

Repeated dilations [55] are used for the bottleneck (Fig. 2, yellow blocks), thus increasing the area that each layer can use as input. It should be noted that this is done without increasing the number of learnable weights, but obtained by spreading the convolution kernel across the input map. The *dilated convolution operator* is then implemented as a gated convolution (i.e., Equation 3), but with some differences. It

is expressed as:

$$D_{y,x} = \sigma(b + \sum_{i=-k'_h}^{k'_h} \sum_{i=-k'_w}^{k'_w} W_{k'_h+i,k'_w+j} \cdot I_{y+\eta i,x+\eta j}) \quad (4)$$

where, assuming the same notation of Equation 2, η is a dilation factor, $\sigma()$ is a component-wise non-linear transfer function and $b \in \mathbb{R}^{C_{out}}$ is the layer bias vector. With $\eta = 1$, the equation becomes the standard convolution operation. In our model, we adopt, respectively, $\eta = 2, 4, 8, 16$ for the four bottleneck layers.

Using this strategy, we aggregate multi-scale contextual information without losing resolution, thus capturing the global context efficiently by expanding the receptive field, avoiding additional parameters and preventing information loss. This is important for the image completion task, as capturing sufficient context is critical for realism. By using dilated convolutions at lower resolutions, the model can effectively cover a larger area of the input image when computing each output pixel than with standard convolutional layers [7].

The network decoder (7 blue layers in Fig. 2) follows a scheme which is symmetrical with respect to the scheme of the encoder. Five layers, based on gated convolutions, restore the resolution of the output to the original input resolution, and a final double layer (two layers in parallel) is dedicated respectively to the synthesized *RGB image* and its depth (Fig. 2). These last two layers have two different activation functions, respectively *tanh* for the RGB output and *ELU* for the depth output.

3.3 Training and losses

During the training phase, we compute the parameters of the network (Sect. 3.2) using a supervised training approach. To this end, we currently exploit Structured3D [63]), a large-scale, synthetic database of indoor scenes. For each scene, a photorealistic, equirectangular rendering of the cluttered environment is matched with the rendering of the same empty scene and with its depth map. It should be noted that such a pixel-wise accurate matching between full and empty scenes and their depths is practically only possible with synthetic data. However, an important benefit of our method is the ability to perform transfer learning efficiently, so the model trained on the synthetic dataset also performs very well on real images, as demonstrated in our results (Sect. 4).

Our loss functions are designed to combine a visual term, that measures the photorealistic quality of the output, with a geometric term, that drives the solution towards a plausible reconstruction of an indoor environments.

The visual term is a combination of different domain losses to ensure the photorealistic quality of the predictions:

$$\mathscr{L}_{vis} = \lambda_{px}\mathscr{L}_{px} + \lambda_{perc}\mathscr{L}_{perc} + \lambda_{style}\mathscr{L}_{style} \quad (5)$$

The first term is a pixel-based $L1$ loss between the predicted RGB image I_{out} and the ground truth empty scene image I_{gt}. $\mathscr{L}_{perc}$ and $\mathscr{L}_{style}$ are the data-driven perceptual and style losses [3]. These enforce I_{out} and I_{gt} to have a similar representation in the feature space as computed by a CNN model ψ, which, as in many image synthesis approaches, is a pre-trained $VGG-19$ [44]. The perceptual loss is, thus, given by:

$$\mathscr{L}_{perc} = \sum_{n}^{N-1} \left\| \psi_n(I_{out}) - \psi_n(I_{gt}) \right\|_1 \quad (6)$$

and computes the $L1$ distance between the projection of I_{out} and I_{gt} into high-level features using the pre-trained network ψ, thus preserving *high-level* content of the image. In Equation 6, ψ_n is the activation map of the n-th selected layer. In our loss, we use *relu*11, *relu*21, *relu*31, *relu*41 and *relu*51 layers [10].

The style loss, calculated on the same layers of perceptual loss, is given by:

$$\mathscr{L}_{style} = \sum_{n}^{N-1} \left\| K_n(\psi_n(I_{out})^T \psi_n(I_{out})) - \psi_n(I_{gt})^T \psi_n(I_{gt}) \right\|_1 \quad (7)$$

which includes the *Gram matrix* function, where the high level feature $\psi(x)_n$ is of shape $(H_n W_n) \times C_n$, resulting in a $C_n \times C_n$ Gram matrix, and K_n is the normalization factor $1/C_n H_n K_n$ for the nth selected layer. Differently from the perceptual loss, this component gives more importance to local similarity (e.g., texture).

The geometric term is a combination of low- and high-order 3D constraints:

$$\mathscr{L}_{geom} = \lambda_d \mathscr{L}_d + \lambda_n \mathscr{L}_n \quad (8)$$

The low-order term $\mathscr{L}_d$ is a robust pixel-wise loss between the predicted depth D_{out} and the ground truth depth of the empty scene D_u (Sect. 3.1). Similarly to other recent state-of-the-art solutions (e.g., BiFuse [50] and SliceNet [33]), we adopt as objective function the *Adaptive Reverse Huber Loss (BerHu)* [15].

For the high-order term $\mathscr{L}_n$, we consider a geometric constraint from a global perspective to take long-range relations into account. This is achieved by exploiting the concept of *virtual normal* [54], i.e., the normal vector of a virtual plane formed by three randomly sampled non-collinear points in 3D space. By minimizing the direction divergence between a small set of ground-truth and predicted virtual normals, serving as a high-order 3D geometric constraint, we preserve the global shape of the model. Such an approach is very effective for indoor environments, typical composed of the union of a small set of smooth surfaces.

From the given depth map $D^{h \times w}$, a 3D point cloud is reconstructed by spherical projection, so that, for each pixel $p_i(u_i, v_i) \in D$, we obtain the location $P_i(x_i, y_i, z_i)$ in 3D coordinates with respect to the sphere center (i.e., camera point-of-view). N triples of points are randomly sampled from the point cloud. The three points $\{(P_a, P_b, P_c)\}$ in each triple are restricted to be non-collinear as defined by the following condition:

$$C = \{\alpha \geq \angle(\overrightarrow{P_a P_b}, \overrightarrow{P_a P_c} \leq \beta, \alpha \geq \angle(\overrightarrow{P_b P_c}, \overrightarrow{P_b P_a} \leq \beta\} \quad (9)$$

where $\alpha = 150^\circ$ and $\beta = 30^\circ$ in our experiments.

The normal vector n_i of the plane formed by the three points is computed by:

$$n_i = \frac{\overrightarrow{P_a P_b} \times \overrightarrow{P_a P_c}}{\left\| \overrightarrow{P_a P_b} \times \overrightarrow{P_a P_c} \right\|} \quad (10)$$

The high order loss $\mathscr{L}_n$ is computed by:

$$\mathscr{L}_n = \frac{1}{N} \sum_{i=1}^{N} \left\| n_i^{pred} - n_i^{gt} \right\| \quad (11)$$

A small number of triples is sufficient to produce effective results. As an example, for each predicted or ground truth depth, having 512×1024 size, we randomly sample 3600 triplets, from which we obtain a pair of 3600 virtual normals, i.e., less than 0.7% of the pixels. The contribution of geometric terms is highlighted in the ablation study at Sect. 4.5.

The relative importance of each loss term is determined by the values of the λ_x coefficients. In our experiments we use $\lambda_{px} = 4, \lambda_{perc} = 1, \lambda_{style} = 40, \lambda_d = 0.5, \lambda_n = 0.5$.

4 Results

Our approach was implemented using *PyTorch* [31] and has been tested on a large variety of indoor scenes. In this paper, we report on results on the benchmarks used by the majority of state-of-the-art works [22, 63]. In addition, we report on the applications to scenes captured by non-professional users.

Two examples of depth and color representations of empty rooms starting from a single-shot panoramic image of cluttered environments are presented in Fig. 3. The accompanying videos shows sequences taken from Diminished Reality applications in which users explore several panoramic scenes, going from the real cluttered view to the synthetic uncluttered view. The video also shows an example of refurnishing, where 3D models of furniture are placed within the virtually emptied room.

Fig. 3: Two examples of inference of color and depth of the empty room from a single-shot 360° panorama

4.1 Training and testing datasets

We use Structured3D [63] to train, validate and numerically compare our results to ground truth and other works. Structured3D [63] is a large-scale photo-realistic dataset containing $3.5K$ house designs created by professional designers with a variety of ground truth 3D structure annotations, including $21,000$ photo-realistic full-panoramic (i.e., equirectangular format) indoor scenes. These panoramic scenes are provided with or without furniture and objects. For all configurations, both RGB images and depth maps are provided, allowing us to immediately use them for training, validating, and testing without further configuration. The official splitting [63] is used, with no overlap among training and testing partitions.

It should be noted that, while our method makes no particular assumption on the architectural structure, Structured3D mostly includes Atlanta World structures [36], leading to a better performance on scenes also meeting these constraints, even though this constraint is not necessary for our network structure. As a further minor limitation of this dataset, we noted that the environment map of the outdoors seen through the windows is replicated in scenes that are part of the training and testing partition. We plan to generate higher variations both of room geometry and outdoor textures in our future works.

In addition to testing with Structured3D, we also adopt Matterport3D [22], a large-scale RGB-D dataset containing $10,800$ panoramic views from $194,400$ RGB-D images of 90 real building-scale scenes, to test our method on real-world scenes. We also exploit this dataset to demonstrate our transfer-learning capabilities. Furthermore, to demonstrate the robustness of the method towards real acquisitions of various types, we selected scenes from a variety of real-world datasets used in the field of automatic building reconstruction [36] or manually acquired by non-professional users using the widely available Ricoh Theta spherical cameras.

4.2 Setup and computational performance

We trained both the clutter mask prediction (Sect. 3.1) and the scene synthesis (Sect. 3.1) networks with the Adam optimizer [14], with $\beta_1 = 0.9$, $\beta_2 = 0.999$, on two NVIDIA RTX 2080Ti GPUs (11GB VRAM) with a batch size of 8 and a learning rate of 0.0001. When using the typical 512×1024 resolution, the average training time for the clutter mask prediction model is $32ms/image$, while it is $196ms/image$ for the scene synthesis model. We adopt for training both networks the Structured3D [63] official splitting. $VGG-19$ [44] pre-trained model is the one provided by TorchVision [31].

Table 1 presents our computational performance compared to state-of-the-art inpainting methods. Although our method is fully scalable in resolution (see Table 2), Table 1 shows the performance with a resolution compatible with the other baselines and adopted in previous comparisons [4], avoiding modifications of the other models (i.e., 256×512). Our method is clearly the most lightweight and has a lower computational complexity (e.g., GFLOPS) than the compared inpainting methods. Moreover, as our approach is designed to remove all the objects in the scene at the same time without user intervention, our presented statistics include the cost of both the clutter mask estimation and scene synthesis networks, while other methods, designed for general infilling, report results only for the synthesis part.

Method	Params↓	GFLOPS↓	ms/frame↓
RFR [18]	30.59 M	412.22	157
Deepfillv2 [57]	13.86 M	163.44	41
PanoDR [4]	20.88 M	122.53	270
Our	**6.06 M**	**41.03**	**17**

Table 1: **Computational performance.** We show our computational performance compared to other state-of-the-art works on a single NVIDIA RTX 2080Ti GPU.

Resolution	Params	GFLOPS	ms/frame
256×512	6.06 M	41.03	15
512×1024	6.06 M	164.11	41
1024×2048	6.06 M	656.45	174

Table 2: **Computational scalability.** We show our computational performance and latency time for different input resolution. Our results demonstrate how we diminish images with a very low latency even when resolution increase.

Table 2 shows, instead, how our approach scales to higher resolutions. As demonstrated in the results, we diminish images with a very low latency, even at the higher tested resolution (1024×2048). Applications can, thus, provide a quick feedback following a camera motion and/or environmental changes. While we currently exploit these advantages for interactively taken single-shot images, the achieved performance makes it possible to consider an extension to real-time room emptying during continuous capture. As a term of comparison, approaches such as PanoDR [4] take 1183 GFLOPS at the 512×1024 resolution (i.e., close to an order of magnitude larger than ours), making it hard to perform the inference on a single commodity graphics board.

4.3 Performance vs. ground truth and competitors

We compared our performance to the one achieved by several state of the art inpainting methods [4, 18, 57, 62], which are representative of the most related approaches discussed in Sect. 2. To provide a quantitative evaluation with respect to ground truth, we train all methods using Structured3D [63] and used the official implementation, minimally adapted to the equirectangular format, for the computational performance evaluation (Sect. 4.2). Table 3 presents results on the full Structured3D [63] test set, according to its official split.

(a) RGB input (b) ground truth (c) other [4] (d) our RGB (e) our depth

Fig. 4: We present qualitative performance and comparison vs. ground truth and other approaches on the Structured3D dataset [63]. We compare to panoDR [4], which has the best panoramic performances among the available methods. We additionally show our output depth paired with our visual results (Fig. 4d).

We adopt standard metrics: Mean Absolute Error (MAE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) [51] and the Learned Perceptual Image Patch Similarity (LPIPS) [60]. LPIPS is a metric that has been shown to better assess the perceptual similarity between two images. It measures the distance between the target and generated images using features extracted from a pre-trained VGG-16 model. Since other compared approaches assume that a mask of the parts of the image to be removed is provided by a user (i.e., the whole clutter), we have used as input the ground truth clutter mask (Sect. 3.1). Note that this aspects favors the other approaches, since for our method we use, instead, the mask estimated from the color input.

Even with this difference, our method outperforms the other approaches on all considered metrics. This can be explained by the fact that the currently available methods are designed to remove limited portions of the image or single objects, rather than the entire clutter preserving only the architectural layout, while our method is adapted to that situation. This fact clearly shows the advantage of designing a task-specific network.

Method	LPIPS↓	MAE↓	PSNR↑	SSIM↑
RFR [18]	0.418	0.201	10.885	0.745
PicNet [62]	0.472	0.204	10.922	0.733
Deepfillv2 [57]	0.354	0.188	11.235	0.729
PanoDR [4]	0.310	0.172	11.612	0.754
Our	**0.129**	**0.040**	**24.702**	**0.925**

Table 3: **Quantitative performance.** We show our quantitative performance compared to other state-of-the-art works.

Fig. 4 presents some examples of our qualitative performance (Fig. 4d), compared to ground truth (Fig. 4b) and to other methods (Fig. 4c). We choose to compare our approach with PanoDR [4], since it was specifically designed for diminished reality on panoramic images and, in our tests, it is the best performing among the other tested methods. Moreover, the method embeds several other state-of-the-art solutions for image inpainting [57, 64]. Our method performs well under different conditions, such as near and far objects, poorly or highly textured walls, more or less distorted foreground, as well as background occlusions.

Fig. 4e shows the depth produced by our method (Fig. 4e), which is not computed by the other inpainting solutions. To provide a term of comparison, we compared our method with state-of-the-art publicly available networks for panoramic depth prediction, i.e., SliceNet [33] and HoHoNet [46], trained on the Structured3D [63] dataset, and with the work of Jin et al. [9], which released the full-empty dataset adopted in this work. Since we target the estimation of the depth of the uncluttered scene, while competing methods do not differentiate clutter from architectural structure, the comparison is performed in the uncluttered areas for SliceNet [33] and HoHoNet [46], i.e., only for pixels not masked with the ground truth masks (Sect. 3.1). To compare ourselves with Jin et al. [9], we use instead the official data provided by the authors on the same data used by our method, since their original code is not available.

Table 4 provides depth results using the common metrics, i.e., mean absolute error (MAE), mean squared error (MSE), root mean square error of linear measures (RMSE) and relative accuracy δ_1, defined as the fraction of pixels where the relative error is within a threshold of 1.25. For MAE, MSE, and RMSE, smaller is better (unit is *meter*), while for δ_1 larger is better.

Method	MAE↓	MSE↓	RMSE↑	δ_1 ↑
HoHoNet [46]	0.101	0.076	0.206	0.932
Jin et al. [9]	-	0.071	0.642	0.958
SliceNet [33]	0.082	0.054	0.198	0.961
Our	0.091	0.073	0.197	0.954

Table 4: **Depth prediction performance.** We show our quantitative performance compared to other state-of-the-art works.

It should be noted that, despite the limited complexity of our network, and the fact that it also targets color estimation, the accuracy of our

(a) input [22] (b) prediction (c) input [22] (d) prediction

(e) user input (f) prediction (g) user input (h) prediction

Fig. 5: We present qualitative performance on data for which no ground truth or training set was available. Here, we show cases from the large scale real-world dataset Matterport3D [22] and from typical user-acquired scenes, where captured images are not perfectly aligned and the photographer is visible.

depth prediction appears comparable to the results of state-of-the-art specific methods for panoramic depth estimation [33, 46]. Our very good results with a much leaner network are due to the fact that, in this particular setting, we target reconstruction only of the fairly regular areas comprising the architectural layout of the room, while methods seeking to reconstruct the full depth [33, 46] must handle much more variable and discontinuous visible shape, due to the high presence of furniture and other objects that have to be measured.

4.4 Performance in-the-wild

Fig. 5 presents qualitative performance on data for which no ground truth or training set was available. This situation is the expected usage of our method.

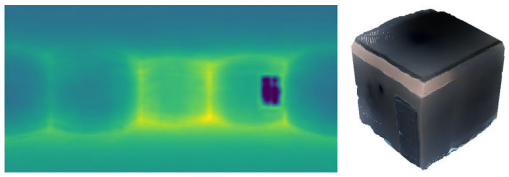
In the upper part of the figure, we show scenes from the large-scale real-world dataset Matterport3D [22]. In the bottom part of the figure, we show scenes acquired by non-professional users using commodity low-cost devices (i.e., Ricoh Theta V and Ricoh Theta Z), collected or acquired by us. In the case of Matterport data, the blur in the upper and bottom part of the scene is due to the fact that those areas are missing due to hardware limitation of the device, and have been approximated in the input scene with a color diffusion.

Although the training of our model was done on a synthetic dataset mainly including Atlanta World structures [36], our method makes no special assumption on the indoor scene kind, or about the precise alignment of the camera with respect to the ground [33, 46] (within the limits of rational, even manual, capture). Furthermore, the method automatically removes the photographer who takes a panoramic photo by holding the camera (i.e., that is considered as clutter). As an example, the last row of Fig. 5 shows our prediction when capture is not aligned to the ground and when the user is visible in the cluttered scene. In all cases, our method is able to predict compelling empty scenes on real data acquired with different devices, automatically removing various types of clutter and very heterogeneous furniture.

The biggest difference in the results, compared to standard synthetic testing sets, is on the lighting appearance of the resulting scene, which sometimes differs from the setup of the original cluttered scene (Fig. 5a). One of the evident consequences of this phenomenon is the different color tone of some scenes (Fig. 5b). This is not surprising, since our model does not, at the moment, make any assumption about lighting. This aspect could be object of future works (Sect. 5).

4.5 Discussion and ablation study

In this section, we discuss our major technical choices, supported by an ablation study, and several features and limitations of our method. As seen in the previous sections, our approach proves to be light-weight and scalable (see Table 2). It should also be noted that the 3D output allows for real-time 3D rendering applications, independent of image-based rendering applications only, for instance the usage of geometric features to help positioning virtual objects on the ground or aligned to walls (Fig. 1). As shown in Table 4, our depth estimation reaches state-of-the-art quality. Fig. 6 shows an example of the predicted point cloud, which represents a good approximation of the underlying layout of the scene.



(a) Output depth (b) Point cloud

Fig. 6: **Predicted depth and its point cloud.** Example of 3D point cloud generated from the predicted depth.

Throughout Table 5, we show the differences in performance, with the same setup as in previous numerical experiments (see Sect. 4.2), in exploiting the geometric features of the scene. In particular, *PWG* indicates the use or not of pixel-wise geometric loss and *HOG* of high-order geometric loss (Sect. 3.3). In the first case, we tested the model by passing the ground truth masks directly, and found no performance improvement over using masks learned from the model itself. It should be noted that this approach is much more efficient than using adversarial losses, as done in many infilling techniques (see Sect. 2). For the *GAN* option, we have tested a discriminator-based loss that is learned during training (i.e., PatchGAN [8]). We experiment that an adversarial loss gives a boost in performance without geometric hints (second row of Table 5), but does not give additional performance when already using geometric losses. Regarding the initial mask (Sect. 3.1), we

PWG	HOG	UI	GAN	LPIPS↓	SSIM↑	δ_1 D_{out} ↑
-	-	-	-	0.398	0.698	-
-	-	-	✓	0.302	0.748	-
✓	-	-	-	0.164	0.833	0.905
✓	✓	✓	-	0.136	0.914	0.954
✓	✓	-	✓	0.121	0.918	0.952
✓	✓	-	-	**0.129**	**0.925**	**0.954**

Table 5: **Ablation facts.** We show the effect of several key choices of our approach. In bold the adopted configuration. PWG: pixel-wise geometry loss; HOG: high-order geometric loss; UI: user intervention; GAN: adversarial-loss; LPIPS,SSIM and δ_1 metrics described in Sect. 4.3.

also experimented that an extreme accuracy of it is not required for our model to work (i.e., 97% IoU), as the gating mask is dynamically propagated and learned, and the purpose of the first network is more related to bootstrapping feature gating. We have also verified that training the clutter mask network in a separate stage is more efficient than training it simultaneously with the scene synthesis model, since it accelerates convergence and reduces the use of memory during training connected to loading all the data at the same time (i.e., both cluttered and uncluttered depth).

Our model proves versatile on different types of indoor scenes, even as the type of real or synthetic input data varies. However, there are cases, mainly in real-world scenes very different from training data, where our method did not produce plausible images. The first row in Fig. 7 shows one of these cases. The particular conditions of

(a) Input (b) Prediction

Fig. 7: **Limiting cases.** Due to the particular lighting condition our network returns a blurred model.

illumination and the presence of many reflections do not lead to a plausible reconstruction. The geometry, in this case, is not sufficient to model the scene, also for the presence of unconventional structures very distant from the domestic training set on which the network has been trained. This drop in performance under these particular lighting conditions is not surprising, as our method does not actually model the lighting of the scene. This aspect will be object of future work.

Moreover, while our method does not explicitly impose the typical restrictive priors of several competitors (e.g., Manhattan World, Atlanta World, vertical walls), and is, therefore, adaptable to more general architectures, the only currently available training dataset [63] is of the Atlanta World type. Thus, irrespective of the generality of our network architecture, performances clearly decay when moving away from this type of scenes. The second row in Fig. 7 shows a room characterized by a lot of clutter and a very sloping ceiling, where the lower perimeter walls are barely visible. Under these conditions, it is difficult to retrieve contextual and geometric information to reconstruct the missing parts, resulting in several artifacts. Since this limitation is related to lack of training examples, we expect major improvements when datasets containing this kind of architecture will become available.

5 Conclusions

We have presented a new data-driven approach that, from an input 360° image of a furnished and cluttered indoor space automatically returns, at interactive speed, a 360° photorealistic view and depth of the same scene emptied of all furniture and other clutter. Rather than casting the problem as a simple image infilling problem, we consider the correlation between color and geometry that occurs in indoor spaces. This allows us to exploit, beside perceptual and style objective functions, geometric losses of different orders, including robust geometric pixel-wise and high-order 3D losses targeted for indoor structures, simplifying the prediction model and its computational complexity. The experimental results demonstrate that our method provides interactive performance and outperforms current state-of-the-art solutions on commonly used indoor panoramic benchmarks and also for indoor scenes captured in the wild and for which there is no ground truth to support supervised training. While this article focused on a method to support low-latency emptying of single 360° shots, with subsequent exploration and editing of the produced static environment, the accuracy and speed achieved could make it possible to consider immersive dynamic scenarios with acquisition and modification of the scene, even in motion and in real-time. In the future, we plan to extend our work in this sense, also considering the lighting model and the spatial coherence of prediction.

Acknowledgments

The project received funding from the European Union's H2020 research and innovation program under grant 813170 (EVOCATION), and from Sardinian Regional Authorities under project VDIC.

REFERENCES

[1] T. Bertel, N. D. Campbell, and C. Richardt. MegaParallax: Casual 360° panoramas with motion parallax. *IEEE TVCG*, 25(5):1828–1835, 2019.

[2] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. In *Proc. ICCV*, pp. 1511–1520, 2017.

[3] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proc. CVPR*, pp. 2414–2423, 2016. doi: 10.1109/CVPR.2016.265

[4] V. Gkitsas, V. Sterzentsenko, N. Zioulis, G. Albanis, and D. Zarpalas. Panodr: Spherical panorama diminished reality for indoor scenes. In *Proc. CVPR Workshops*, pp. 3716–3726, 2021.

[5] Á. L. Guedes, G. d. A. Roberto, P. Frossard, S. Colcher, and S. D. J. Barbosa. Subjective evaluation of 360-degree sensory experiences. In *Proc. IEEE MMSP*, pp. 1–6, 2019.

[6] V. Gupta, R. Sadana, and S. Moudgil. Image style transfer using convolutional neural networks based on transfer learning. *International Journal of Computational Systems Engineering*, 5(1):53–60, 2019. doi: 10.1504/IJCSYSE.2019.098418

[7] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *ACM Trans. Graph.*, 36(4), jul 2017. doi: 10.1145/3072959.3073659

[8] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proc. CVPR*, pp. 1125–1134, 2017.

[9] L. Jin, Y. Xu, J. Zheng, J. Zhang, R. Tang, S. Xu, J. Yu, and S. Gao. Geometric structure based and regularized depth estimation from 360 indoor imagery. In *Proc. CVPR*, pp. 889–898, 2020.

[10] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proc. ECCV*, pp. 694–711, 2016.

[11] T. Jokela, J. Ojala, and K. Väänänen. How people use 360-degree cameras. In *Proc. MUM*, pp. 18–27, 2019.

[12] N. Kawai, T. Sato, and N. Yokoya. Diminished reality based on image inpainting considering background geometry. *IEEE TVCG*, 22(3):1236–1247, 2015.

[13] H. Kim, T. Kim, M. Lee, G. J. Kim, and J.-I. Hwang. Don't bother me: How to handle content-irrelevant objects in handheld augmented reality. In *Proc. VRST*, 2020. doi: 10.1145/3385956.3418948

[14] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *ArXiv e-print arXiv:1412.6980*, 2014.

[15] S. Lambert-Lacroix and L. Zwald. The adaptive BerHu penalty in robust regression. *Journal of Nonparametric Statistics*, 28:1–28, 2016.

[16] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European conference on computer vision*, pp. 702–716. Springer, 2016.

[17] J. Li, F. He, L. Zhang, B. Du, and D. Tao. Progressive reconstruction of visual structure for image inpainting. In *Proc. ICCV*, pp. 5961–5970, 2019.

[18] J. Li, N. Wang, L. Zhang, B. Du, and D. Tao. Recurrent feature reasoning for image inpainting. In *Proc. CVPR*, pp. 7760–7768, 2020.

[19] D. Lindlbauer and A. D. Wilson. Remixed reality: manipulating space and time in augmented reality. In *Proc. CHI*, pp. 129:1–128:13, 2018.

[20] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proc. ECCV*, 2018.

[21] H. Liu, B. Jiang, Y. Song, W. Huang, and C. Yang. Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In *Proc. ECCV*, pp. 725–741, 2020.

[22] Matterport. Matterport3D. `https://github.com/niessner/Matterport`, 2017. [Accessed: 2019-09-25].

[23] K. Matzen, M. F. Cohen, B. Evans, J. Kopf, and R. Szeliski. Low-cost 360 stereo photography and video capture. *ACM TOG*, 36(4):148:1–148:12, 2017.

[24] S. Meerits and H. Saito. Real-time diminished reality for dynamic scenes. In *Proc. ISMAR Workshops*, pp. 53–59, 2015.

[25] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. doi: arXiv:1802.05957

[26] S. Mori, S. Ikeda, and H. Saito. A survey of diminished reality: Techniques for visually concealing, eliminating, and seeing through real objects. *IPSJ Transactions on Computer Vision and Applications*, 9(1):1–14, 2017.

[27] S. Mori, F. Shibata, A. Kimura, and H. Tamura. Efficient use of textured 3D model for pre-observation-based diminished reality. In *Proc. ISMAR Workshops*, pp. 32–39, 2015. doi: 10.1109/ISMARW.2015.16

[28] Y. Namboku and H. Takahashi. Diminished reality in textureless scenes. In *International Workshop on Advanced Imaging Technology (IWAIT)*, vol. 11515, pp. 379 – 384. SPIE, 2020. doi: 10.1117/12.2566248

[29] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, and M. Ebrahimi. Edgeconnect: Structure guided image inpainting using edge prediction. In *Proc. ICCVW*, pp. 3265–3274, 2019. doi: 10.1109/ICCVW.2019.00408

[30] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proc. CVPR*, pp. 2337–2346, 2019.

[31] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *Proc. NIPS Workshop on Autodiff*, 2017.

[32] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proc. CVPR*, 2016.

[33] G. Pintore, M. Agus, E. Almansa, J. Schneider, and E. Gobbetti. SliceNet: deep dense depth estimation from a single indoor panorama using a slice-based representation. In *Proc. CVPR*, pp. 11536–11545, 2021.

[34] G. Pintore, E. Almansa, M. Agus, and E. Gobbetti. Deep3dlayout: 3d reconstruction of an indoor layout from a spherical panoramic image. *ACM Trans. Graph.*, 40(6):250:1–250:12, 2021.

[35] G. Pintore, F. Ganovelli, A. Jaspe Villanueva, and E. Gobbetti. Automatic modeling of cluttered floorplans from panoramic images. *Computer Graphics Forum*, 38(7):347–358, 2019.

[36] G. Pintore, C. Mura, F. Ganovelli, L. Fuentes-Perez, R. Pajarola, and E. Gobbetti. State-of-the-art in automatic 3D reconstruction of structured indoor environments. *Comput. Graph. Forum*, 39(2):667–699, 2020.

[37] G. Pintore, R. Pintus, F. Ganovelli, R. Scopigno, and E. Gobbetti. Recovering 3D existing-conditions of indoor structures from spherical images. *Computers & Graphics*, 77:16–29, 2018.

[38] G. Queguiner, M. Fradet, and M. Rouhani. Towards mobile diminished reality. In *Proc. ISMAR-Adjunct*, pp. 226–231, 2018. doi: 10.1109/ISMAR-Adjunct.2018.00073

[39] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proc. MICCAI*, pp. 234–241, 2015.

[40] H. Sasanuma, Y. Manabe, and N. Yata. Diminishing real objects and adding virtual objects using a RGB-D camera. In *Proc. ISMAR-Adjunct*, pp. 117–120, 2016. doi: 10.1109/ISMAR-Adjunct.2016.0055

[41] A. Saxena, M. Sun, and A. Y. Ng. Make3D: Learning 3D scene structure from a single still image. *IEEE TPAMI*, 31(5):824–840, 2009.

[42] S. Siltanen. Diminished reality for augmented reality interior design. *The Visual Computer*, 33(2):193–208, 2017.

[43] S. Siltanen, H. Saraspää, and J. Karvonen. [demo] a complete interior design solution with diminished reality. In *Proc. ISMAR*, pp. 371–372, 2014. doi: 10.1109/ISMAR.2014.6948494

[44] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[45] M. Z. Sulaiman, M. N. A. Aziz, M. H. A. Bakar, N. A. Halili, and M. A. Azuddin. Matterport: virtual tour as a new marketing approach in real estate business during pandemic COVID-19. In *Proc. IMDES*, pp. 221–226, 2020.

[46] C. Sun, M. Sun, and H.-T. Chen. HoHoNet: 360 indoor holistic understanding with latent horizontal features. In *Proc. CVPR*, pp. 2573–2582, 2021.

[47] Y. Takeuchi and K. Perlin. Clayvision: The (elastic) image of the city. In *Proc. CHI*, pp. 2411–2420, 2012. doi: 10.1145/2207676.2208404

[48] T. Teo, L. Lawrence, G. A. Lee, M. Billinghurst, and M. Adcock. Mixed reality remote collaboration combining 360 video and 3d reconstruction. In *Proc. CHI*, pp. 201:1—-201:14, 2019. doi: 10.1145/3290605.3300431

[49] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[50] F.-E. Wang, Y.-H. Yeh, M. Sun, W.-C. Chiu, and Y.-H. Tsai. BiFuse: Monocular 360 depth estimation via bi-projection fusion. In *Proc. CVPR*, pp. 462–471, 2020.

[51] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[52] M. Xu, C. Li, S. Zhang, and P. Le Callet. State-of-the-art in 360° video/image processing: Perception, assessment and compression. *IEEE Journal of Selected Topics in Signal Processing*, 14(1):5–26, 2020.

[53] Z. Yi, Q. Tang, S. Azizi, D. Jang, and Z. Xu. Contextual residual aggregation for ultra high-resolution image inpainting. In *Proc. CVPR*, pp.

7508–7517, 2020.

[54] W. Yin, Y. Liu, C. Shen, and Y. Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *Proc. ICCV*, pp. 5683–5692, 2019.

[55] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In Y. Bengio and Y. LeCun, eds., *Proc. ICLR*, 2016.

[56] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Generative image inpainting with contextual attention. In *Proc. CVPR*, pp. 5505–5514, 2018.

[57] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Free-form image inpainting with gated convolution. In *Proc. ICCV*, pp. 4471–4480, 2019.

[58] Y. Zeng, J. Fu, H. Chao, and B. Guo. Learning pyramid-context encoder network for high-quality image inpainting. In *Proc. CVPR*, pp. 1486–1494, 2019.

[59] E. Zhang, M. F. Cohen, and B. Curless. Emptying, refurnishing, and relighting indoor spaces. *ACM TOG*, 35(6):174:1–174:14, 2016.

[60] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. CVPR*, pp. 586–595, 2018.

[61] Y. Zhang, S. Song, P. Tan, and J. Xiao. PanoContext: A whole-room 3D context model for panoramic scene understanding. In *Proc. ECCV*, pp. 668–686, 2014.

[62] C. Zheng, T.-J. Cham, and J. Cai. Pluralistic image completion. In *Proc. CVPR*, pp. 1438–1447, 2019.

[63] J. Zheng, J. Zhang, J. Li, R. Tang, S. Gao, and Z. Zhou. Structured3D: A large photo-realistic dataset for structured 3D modeling. In *Proc. ECCV*, pp. 519–535, 2020.

[64] P. Zhu, R. Abdal, Y. Qin, and P. Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In *Proc. CVPR*, pp. 5104–5113, 2020.

[65] C. Zou, J. Su, C. Peng, A. Colburn, Q. Shan, P. Wonka, H. Chu, and D. Hoiem. Manhattan room layout reconstruction from a single 360 image: A comparative study of state-of-the-art methods. *International Journal of Computer Vision*, 129:1410–1431, 2021.