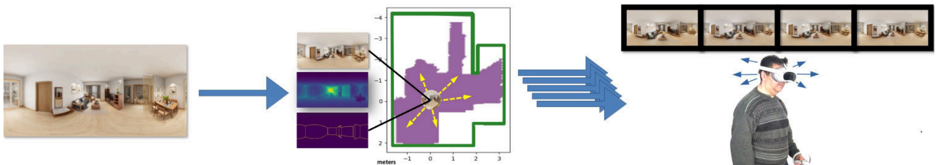
Deep scene synthesis of Atlanta-world interiors from a single omnidirectional image

Giovanni Pintore, Fabio Bettio, Marco Agus, and Enrico Gobbetti



Single-shot panorama
Visual, structural, and geometric representation
Novel views

Fig. 1: Given a single 360° panorama of an indoor scene, we compute an enriched geometric and structural representation, from which novel panoramas from other close-by viewpoints can be synthesized at interactive rates in response to user motion.

Abstract— We present a new data-driven approach for extracting geometric and structural information from a single spherical panorama of an interior scene, and for using this information to render the scene from novel points of view, enhancing 3D immersion in VR applications. The approach copes with the inherent ambiguities of single-image geometry estimation and novel view synthesis by focusing on the very common case of *Atlanta-world* interiors, bounded by horizontal floors and ceilings and vertical walls. Based on this prior, we introduce a novel end-to-end deep learning approach to jointly estimate the depth and the underlying room structure of the scene. The prior guides the design of the network and of novel domain-specific loss functions, shifting the major computational load on a training phase that exploits available large-scale synthetic panoramic imagery. An extremely lightweight network uses geometric and structural information to infer novel panoramic views from translated positions at interactive rates, from which perspective views matching head rotations are produced and upsampled to the display size. As a result, our method automatically produces new poses around the original camera at interactive rates, within a working area suitable for producing depth cues for VR applications, especially when using head-mounted displays connected to graphics servers. The extracted floor plan and 3D wall structure can also be used to support room exploration. The experimental results demonstrate that our method provides low-latency performance and improves over current state-of-the-art solutions in prediction accuracy on available commonly used indoor panoramic benchmarks.

Index Terms—Omnidirectional, 360, immersive view, AR/MR/VR for architecture, Computer vision, Machine learning.

1 Introduction

A single-shot 360° image, containing the entire scene around the viewer, is not consumed at once, but inherently requires a more dynamic exploration with respect to traditional 2D imagery. When it is presented through a Head-Mounted Display (HMD), the viewer is encouraged to actively focus on the desired content via natural head movements, leading to an intuitive VR interface [61]. For this reason, 360° image viewing is becoming one of the main exploration modes of real-world scenes in VR [29] and has widespread use in indoor navigation [1].

The reduction in degrees of freedom to just the rotation around the center of the panorama, leads, however, to constraints and artifacts [54], especially since only one or two shots per room are available in a typical virtual tour [60]. Moreover, binocular stereo and motion parallax, which are important aspects of immersion in VR, are totally missing. To fully support immersion, a system must thus also respond to viewpoint translation. While many solutions have been proposed for multiview capture setups (e.g., [2, 5]), performing view synthesis from single-shot panoramas is of primary importance, due to the convenience and diffusion of sparse capturing through monocular 360° cameras [54].

View synthesis requires the explicit or implicit estimation of the geometric shape of the imaged environment, in order to perform occlusion-aware reprojection and to synthesize the disoccluded content. Current state-of-the-art approaches (e.g., [12, 54]) focus on extending to single-shot panoramas the general data-driven view synthesis approaches designed for perspective views of objects and environments, such as Multi-planar images (MPI) [50] or Neural Radiance Fields (NeRF) [30] (Sec. 2). The mixing of large untextured surfaces, clutter, and non-cooperative materials in interior environments poses, however, important challenges to generic solutions [40]. In this context, it has been demonstrated that the knowledge of additional information, such as the position location of the room corners and edges, significantly improves the realism of the synthesis [60]. However, recovering the indoor layout directly from the input image is extremely challenging. Even the latest dedicated methods [16, 17, 69] still heavily rely on approximations and expensive heuristic post-processing [47], which significantly limit overall performance. As a result, their use for VR applications necessitating interactive-rate image generation is inhibited.

In our work, we propose a new end-to-end data-driven solution that, from a single 360° indoor panorama, assumed captured with approximate gravity alignment, produces with low latency a newly translated pose from which new perspective images can be extracted that respond to both position and orientation changes.

While some HMD solutions strive to fully run on the embedded platform, an alternative design is to compute images on high-performance servers. This approach, extensively employed for high-quality gaming, is made possible by the availability of low-latency tethered or wireless connections with sufficient bandwidth to feed the displays [19]. In our approach, a thin WebXR client directly handles head rotation, while relying on server-computed images to also respond to head translations. Our main novelty is in the indoor-specific deep-learning techniques that synthesize the views. Once per scene, we enrich the original panorama with geometric and structural information, and once per frame, we exploit pre-computed information to quickly perform view synthesis.

The approach copes with the inherent ambiguities of single-image geometry estimation and novel view synthesis in indoor environments by focusing on the very common case of interiors following the *Atlanta world* model (AWM) [40], in which the environment is expected to

- *G. Pintore F. Bettio and E. Gobbetti are with CRS4, Italy. E-mail: giovanni.pintore@crs4.it, fabio.bettio@crs4.it, enrico.gobbetti@crs4.it*
- *M. Agus is with HBKU, Qatar. E-mail: MAgus@hbku.edu.qa*

Author copy for Green Open Access. Digital Object Identifier: 10.1109/TVCG.2023.3320219

have horizontal floor and ceiling and vertical walls. Based on this prior, we introduce a novel end-to-end network to jointly estimate the depth and the underlying room structure of the scene, thus efficiently handling occlusions and disocclusions and enabling a plausible prediction even in the case of extensively occluded structures. The prior drives the network structure, which also exploits gravity-aligned features (GAFs) to take into account the fact that world-space vertical and horizontal features have different characteristics in man-made environments. In particular, AWM makes it possible to derive the 3D layout by extruding its 2D floor projection, while GAFs perform vertical compression, exploiting the fact that vertical lines, common in indoor scenes, are not deformed in equirectangular projections. Because of these characteristics, we expect scene GAFs to be inter-related by both short-term and long-term spatial dependencies, improving the quality of depth prediction and layout prediction [37], and, therefore, visual synthesis.

Starting from the enriched panoramic representation, a lightweight network infers at interactive rates novel panoramic images from translated positions in a working area around the original point of view suitable for VR applications. From these views, perspective images matching head rotations are produced and upsampled to display size. Moreover, the geometric and structural information recovered can also be used to support VR applications, e.g., to define walkable areas.

Our main novel contributions are the following:

- We present a novel approach, dubbed *Atlanta Depth Module - ADM*, to jointly estimate, starting from a single equirectangular image, the scene depth, a scene latent representation, the 3D room shape and a floor occupancy map (Sec. 3.1). ADM achieves state-of-the-art results on both geometric and structural reconstruction (Sec. 4), and provides many advantages for VR applications. First, it is much more lightweight than current solutions for depth or layout estimation commonly adopted in this context [37, 47, 75]. Second, the recovered AWM structure is segmented into ceiling, walls, and floor, and represented in metric units, including the prediction of ceiling-floor heights. This, besides improving view synthesis, supports the creation of a *floor occupancy map*, to generate consistent trajectories inside the room without collisions.
- We introduce novel objective functions to take into account the indoor structural consistency (Sec. 3.3) of the view synthesis. Such functions, based on GAF encoding [37, 39], support direct (i.e., target predicted depth loss) and latent-space losses. Latent-space losses guide a consistent structural reconstruction during training and are dual to visual losses, called *geometric perceptual* and *geometric style*, Such losses, combined with standard perceptual style transfer and adversarial losses, improve reconstructed scene quality (Sec. 4), shifting much of the computational load to the training phase, and making the inference phase much lighter.
- We introduce a fully data-driven, versatile, and lightweight approach to generate novel panoramic views from a single indoor panorama. Such a deep learning approach does not need dedicated processing for each scene [12, 54], but generalizes over indoor scenes that just follow AWM (Sec. 4). Once latent deep features and structural priors are applied at training time, novel pose synthesis is obtained through a network (GVS) without deep layers or complex pipelines. In fact, GVS consists of a limited number of layers, combining gated and dilated convolutions, focused on maximizing the level of detail (Sec. 3.2). As a result, we have a network with a limited and constant number of learnable parameters (Sec. 4.2), even as generated image resolution varies.

Our results (Sec. 4) improve over state-of-the-art approaches on common benchmarks with measurable ground truth, in terms of accuracy, quality and computational complexity. Moreover, compelling predictions are produced even on images where no ground truth is available for training, as well as on novel user-captured images.

2 Related Work

Effective view synthesis requires comprehensively understanding the 3D structure of a scene given an image [59]. Full coverage of this topic is outside the scope of this paper. In the following, we focus on the most closely related approaches, with a particular focus on data-driven solutions for panoramic images.

Depth estimation from panoramic images

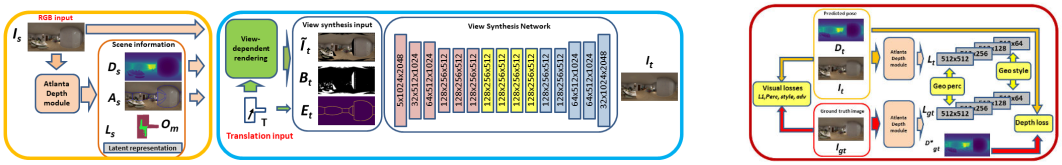
Monocular depth estimation is a classic task in computer vision. While early solutions used various combinations of feature detection, matching, and geometric reasoning, recent research is increasingly focusing on data-driven solutions that derive hidden relations from large amounts of examples [37]. Since it has been shown that directly applying perspective methods to 360° depth estimation in indoor environments produces suboptimal results [74], research has started to focus on explicitly exploiting the characteristics and wide geometric context present in omnidirectional images. A first breed of solutions concentrated on handling distortion through spherical convolution [35, 45, 46, 49, 74]. Wang et al. [55] proposed instead a two-branch network, respectively for the equirectangular and the cubemap projection, based on a distortion-aware encoder [74] and the FCRN decoder [22]. Recent solutions for panoramic depth estimation in indoor spaces [37, 48] have proposed to work directly on equirectangular images, as well as to leverage the concept of gravity-aligned features to reduce network size [37, 47]. A recent trend to mitigate panoramic distortion is to leverage perspective views sampled on panoramic images [25, 41] prior to combining depth maps using transformers. In this work, we leverage gravity-aligned features [37] to flatten image features and then process them with a lightweight network designed for interactive applications (Sec. 3.1). Compared to previous works, we achieve state-of-the-art performance at a much lower computational cost (Sec. 4).

Layout estimation from panoramic images

While depth estimation methods have shown impressive performances, they cannot produce seamless 3D boundary surfaces in case of self-occlusions, since they can only generate a single 3D position per view ray. For this reason, layout-specific approaches are being actively researched. Since man-made interiors often follow very strict rules, early pin-hole methods used geometric reasoning to match image features to simple constrained 3D models [40]. The effectiveness of geometric reasoning methods is, however, heavily dependent on the count and quality of extracted features (e.g., corners, edges, or flat patches). More and more research is thus now focusing on data-driven approaches [77]. Prominent examples are *LayoutNet* [75], which predicts the corner probability map and boundary map directly from a panorama, and *HorizonNet* [47], which simplifies the layout as three 1D vectors. The 2D layout is then obtained by fitting Manhattan World Model (MWM) segments on the estimated corner positions. To mitigate spherical distortion and maximize the efficiency of modern deep learning techniques such as transformers, many recent approaches project the equirectangular input image to planar surfaces [17, 38, 56, 62, 69]. These methods, however, require heavy pre-processing, such as detection of main Manhattan-world directions from vanishing lines analysis [24, 68, 77] and related image warping, or complex layout post-processing, such as Manhattan-world regularization of detected features [47, 62, 75]. LayoutNet [75], for instance, has been used to support view synthesis of individual panoramic images [60] by providing the location of corners in the image, but cannot run at interactive rates. Several methods have, thus, sought to relax the constraints of the Manhattan World model, while decreasing the computational load required by exploiting more general features of man-made structures [39, 40]. These methods, however, target the general reconstruction of the overall room shape but are not usable for the completion of photorealistic views, lacking well-defined parts and edges. In this work, we propose, instead, a new approach for fast estimation of a structured layout, where, unlike the mentioned methods, the estimation is done not from the *RGB* image but from its depth, appropriately transformed (Sec. 3.1). Moreover, we apply Atlanta World projection to depth values, and projection is not done on an arbitrary plane as in other transform-based approaches [40, 62].

Novel view synthesis

Our solution exploits recovered depth and layout for novel view synthesis from monocular input. Most view synthesis approaches exploit, instead, multi-view input, such as NeRF [30], the methods based on depth, proxy geometry, and flow [3, 4, 13, 27],



(a) Forward pipeline

(b) Training scheme

Fig. 2: **Approach overview.** At loading time, we process the input equirectangular image to recover depth D_s, Atlanta structure A_s, occupancy map O_m and latent scene representation L_s (ADM Sec. 3.1). When moving from the source position, the generation of the new translated views is done by a *soft* z-buffer and a gated neural network, dubbed *Gated View Synth network* - GVS (Sec. 3.2). Free viewpoint images can then be generated by extracting perspective views from the translated panoramas taking into account rotations. Supervised training of the GVS network combines visual and perceptual losses with novel indoor-specific losses (Sec. 3.3).

or, for 360° views, those using a layered image representation [10, 11], multi-depth panoramas [26], or layered mesh representations [5]. The method of Serrano et al. [43] extended the layered image representation approach to work with a single panoramic input image, but with only a few depth layers and extrapolation and in-painting to fill holes. Layered solutions have been also extended to be used as a target representation in an end-to-end learning pipeline. In multi-plane images (MPI) [71], each layer is a flat plane placed at a fixed depth from the capture point. The regularity of the representation makes it suitable to be the output of a convolutional neural network. Tucker and Snavely [50] introduced a method to infer an MPI from a single perspective image. *PanoSynthVR* [54] extended this approach by exploring the use of a multi-cylinder representation to approximate a 360 view. However, the method does not exploit information specific to indoors, produces blurry images at disocclusions and severely degrades quality when the viewpoint moves too far from the origin (Sec. 4). Xu et al. [60] recently extended to panoramic images the approach of SynSin [59], which uses a neural network to produce both a depth map and a feature map which can then be rendered to new perspective viewpoints. Moreover, similarly to us, they incorporate prior knowledge, in the form of screen corners, demonstrating the importance of using additional information that does not depend on the input viewpoint. Such an approach is capable of generating new, sparse views, but its performance depends on externally computed information and requires a considerable computational cost. An alternative solution is proposed by *OmniNeRF* [12], which proposes a self-supervised approach to generate novel views given a single panoramic image and its depth, with the goal to feed a NeRF [30] pipeline with multiple poses. Although it is one of the first works to adapt the NeRF concept to a panoramic image, the synthesized images feeding the training are a simple interpolated splatting of the original view, so that new views obtained at run time suffer from significant artifacts. In contrast, by introducing and exploiting important indoor priors at inference and training levels (Sec. 3.3), we generate new views with greater accuracy than the methods mentioned above and with a particularly lightweight end-to-end network. Our reconstruction is also not only visually but also spatially consistent, unlike other representations [2, 50, 54].

3 Methods

In our approach, a thin client explores an HMD synthesized panoramic images that are adapted to position changes through server-side computation.

The extraction, server side, of structural information from a room and the generation of a translated panorama are the most complex operations and are performed through a novel deep learning architecture, whose structure is depicted in Fig. 2.

Once per scene, we assign a viewer-independent geometry context to input source pixels, to create a room structure that can be used for various purposes and to propagate enriched input information to the new pose. Jointly with depth estimation, we infer a structured model of the underlying architectural structure through a deep network that exploits the Atlanta World prior. The network, dubbed Atlanta Depth Module (ADM) (orange module in Fig. 2a), feeds the second step of the pipeline, that, every time the viewer position changes, generates the translated panoramic image. While both modules could be trained as a whole, in our design, for performance reasons, we pre-trained ADM separately from view synthesis.

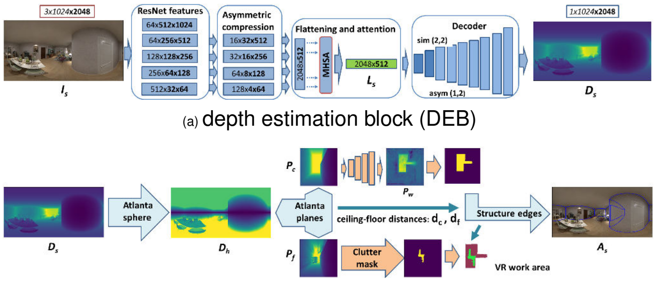
The pre-trained ADM returns the scene depth, a latent scene representation, the room 3D shape, and, additionally, a floor occupancy map (Sec. 3.1). Gravity-aligned features (GAFs) encode a panoramic image into a multi-resolution latent scene representation. GAFs are basically used in two ways. On one hand, their decoding produces a pixel-wise depth of the scene (Fig. 2a, orange); on the other hand, the GAF encoding supports specific geometric loss functions in latent space that guide the training of the view synthesis model (Fig. 2b). Moreover, downstream network layers process the recovered depth to predict a 3D model of the room. As this 3D model is viewer-independent, it is computed once at the time of image loading, albeit with low computational cost, and does not need to be re-executed at each view synthesis. Moreover, the 3D model can be used for a variety of purposes. For instance, it makes it possible to define a walkability map, as well as to limit the legal area for new viewpoints so that there are no walls crossed and we remain inside the room.

The panoramic view synthesis phase, depicted in the orange block of Fig. 2a, is performed at each viewer position change through a very light network that comprises a *soft* z-buffer block and a gated neural network, dubbed *Gated View Synth* (GVS) (Sec. 3.2). The GVS network is trained in a supervised way through a combination of specific losses (Sec. 3.3). In particular, in addition to standard metrics in view synthesis, we introduce novel geometric and structural metrics, that also exploit latent space GAFs.

The output of the view synthesis phase is the translated 360° view from which free-viewpoint images that respond to translation and rotation can be extracted. In our reference design, the view synthesis machinery is exploited in a client-server application, where a thin WebXR client runs directly on the HMD's embedded platform and communicates with the server that handles the compute-intensive tasks. Both client and server are initialized at each scene change with the initial view, which is used by the client for display and by the server to compute from visual data the augmented panoramic scene representation. The client is designed as a foveated panoramic image viewer, that maintains, in two textures, a low-res and a hi-res representation of portions of the scene's panorama in an equirectangular format centered approximately around the current lookat point. The low-res panorama typically comprises a 180x180 degree portion (full frontal view), while the high-res panorama covers at double resolution a 90x90 degree area (HMD FOV). A fragment shader combines the two textures at each frame to produce a seamless view. At each head position change, the head transformation is sent to the server. The position is communicated to the view synthesis network for producing the translated panorama. The rotation is used to determine the current look-at point. Since view synthesis, as for all current deep learning solutions, is performed at a resolution that is lower than current HMD capabilities, we perform image upsampling of the viewing region around the look-at point using state-of-the-art deep learning super-resolution methods [57]. The low-res image and hires image, together with the associated parameters are then sent to the client for display. The supplementary material provides detailed information on the structure of the client-server application.

The novel aspects of our work reside in the methods that are employed to generate the translated panorama, which, in addition to supporting view extraction, also generate auxiliary structural and geometric information that can be exploited for other needs. In the following we will describe the main components of this block, first focusing on depth and 3D layout prediction (Sec. 3.1), and then on novel view synthesis (Sec. 3.2) and training methods (Sec. 3.3).

3.1 Depth and 3D layout prediction



(a) depth estimation block (DEB)

(b) layout estimation block (LEB)

Fig. 3: **Atlanta depth module (ADM).** ADM is an end-to-end network that returns scene depth D_s, latent representation L_s, Atlanta-world 3D room shape, and floor occupancy map. Here we illustrate the two main cascading blocks: the depth estimation block (DEB) (a) and the layout estimation block (LEB) (b). DEB recovers from the input image the depth and its latent representation, while LEB recovers the layout from the predicted depth.

The Atlanta Depth Module (ADM) can be subdivided into two cascading blocks: depth estimation (Fig. 3a) and layout estimation (Fig. 3b). The first network block, dubbed depth estimation block (DEB), receives as input a single panoramic image ${I_s}^{h\times w}$ and returns as output a depth ${D_s}^{h\times w}$ in metric units (i.e., meters), coupled with a latent representation of the scene L_s, which is also used for the specific losses described in Sec. 3.3. The second block, dubbed layout estimation block (LEB), receives as input ${D_s}^{h\times w}$, and returns as output an Atlanta World representation A_s (Sec. 3.1.2).

The rest of this section summarizes the main aspects of DEB and LEB. We refer the reader to the supplementary material for more details.

3.1.1 Depth estimation block

From the input image, a cascade of five residual layers [9] creates four feature maps having different depth and spatial size (Fig. 3a). We consider, then, both the spherical and indoor nature of the scene to further process this representation. First, we adopt circular padding along the horizon for convolutions, to overcome the longitudinal boundary discontinuity, and reflection padding to alleviate the singularities at the poles [8]. Then, in order to support an efficient gathering of information from the extracted features, we perform a specifically indoor-designed feature compression exploiting our knowledge of preferential directions (i.e., gravity direction), assuming that world-space vertical and horizontal features (GAF, gravity aligned features) have different characteristics in most, if not all, man-made environments [37, 39].

Compressed latent features $L_s = (l_1 \dots l_4)$ contain a wealth of information on the geometry of the scene, both local and non-local, which can be exploited to recover depth and layout, as well as to provide a latent representation on the scene for further processing (Sec. 3.3).

For depth estimation, we aim to leverage complementary features in distant portions of the image rather than only local regions, to maximize the wide contextual information provided by omnidirectional images while keeping the computational cost low. To do that, we adopt a single-layer multi-head self-attention (MHSA) scheme [53] to process the latent feature (see Supplementary material). Once passed to the MHSA module, the decoding of the latent feature is very fast, through convolutions, upsampling modules, and ELU activations, until we reach the target output resolution ($1 \times h \times w$ in Fig. 3a).

3.1.2 Layout estimation block

According to the Atlanta World model [40], the indoor scene is expected to have a horizontal floor and ceiling and vertical walls, but without the restriction of walls meeting at right angles (supporting, e.g., curved walls). The indoor model can, thus, be fully represented as a 2D polygon around the observer, that defines the wall footprint, and two scalars d_c and d_f that respectively represent the distance of the observer from the ceiling and floor planes. Without loss of generality, we assume the observer is placed at the origin of a reference frame (Y front, Z up, and X right), and all measures are in metric units. In our network, we represent the layout as a probability map ${P_w}^{wp\times wp}$ tensor, along with the two scalars d_c and d_f. The probability map P represents the probability that a point in the floor plane is inside the room boundary. While the resolution of the depth map $h \times w$ only depends on the input image resolution, the probability map, by design, has a fixed size (i.e., $wp = 512$ in our experiments).

Since layout and depth are inter-related, as the layout must be consistent with the depth assigned to pixels that are part of the architectural structure, we estimate layout jointly with depth, with the added benefit for interactive applications of avoiding the use of separate branches that would increase computational burden (Tab. 1). Moreover, we assume that the prediction of the geometric layout can be extended to non-visible part by exploiting geometric features derived from the depth map, such as flatness, sharpness, and smoothness [39].

Since the recognition of planar and curved structures is not immediate in spherical space, we transform the equirectangular map D_s, representing the Euclidean distances of pixels from the camera center, according to the Atlanta World model. To do this, we apply a planar transformation to D_s, such that the distances are expressed not with respect to a point but with respect to the horizontal plane containing the camera center (Fig. 3b). This is accomplished by scaling each value in D_s, which corresponds to an azimuth angle θ (along w) and polar angle ϕ (along h), by $\|\sin\phi\|$ to obtain the map D_h. This specific transform imposes that horizontal structures, such as ceilings, floors, or even large tables or furniture, have a constant value so that structures identification is simpler, as shown in the Fig. 3b example.

This representation is in an equirectangular format that still has its natural spherical distortion. In order to further simplify structure recognition, we then apply the Atlanta World transform, which has been proven to be effective on equirectangular image analysis [40, 62]. Specifically, we project the equirectangular depth map ${D_h}^{h\times w}$ to two projective planes, ${P_c}^{wp\times wp}$ and ${P_f}^{wp\times wp}$ perpendicular to the Z-axis, so that ceiling projection P_c represents depth information belonging to the upper hemisphere of D_h, while the floor projection P_f to the bottom hemisphere. For every pixel in the perspective image at position (p_x, p_y), we recover a depth value from the corresponding pixel in the equirectangular map by defining a focal length:

$$f = \frac{wp}{2} \cot \frac{fov_p}{2}, \tag{1}$$

with $fov_p = 165°$ in our experiments. Then, we sample the equirectangular panorama at the coordinates

$$(u_s, v_s) = \left(\frac{\arctan(s_x/s_z)}{\pi}, \frac{\arcsin(s_y)}{\pi/2} \right), \tag{2}$$

where (s_x, s_y, s_z) is the direction vector from the origin to the point (p_x, p_y, f). Since the whole process is differentiable, it can be used in conjunction with back-propagation. Furthermore, both the depth transformation and perspective projection Eq. (2) are implemented with simple GPU operations (Sec. 4) with negligible computational cost.

Since the top view P_c is clearly more clutter-free, as it represents the ceiling, we use that view to derive the shape of the room, while we exploit the floor projection P_f to recover an occupancy map. Furthermore, the respective distances of the ceiling and floor planes from the center of the room are $d_c = max(P_c)$ and $d_f = max(P_f)$.

Simply extrapolating the layout of walls from P_c would return an incomplete reconstruction for eventually occluded parts (Fig. 3b, first transform). This is a common problem in almost all single view approaches (Sec. 2), which is commonly solved by a heuristic post-processing step [47]. However, such a solution, in addition to adding computational cost, works only for very simple cases.

In order to have a more versatile reconstruction we decide to introduce a specific data-driven solution. Starting from the projected depth P_c, we include in our ADM module a further multi-layer perceptron, named layout estimation network (LEN), which estimates a map P_w representing the probability of being inside the room footprint on the floorplan. As the visible shape of the room is already highlighted in the input P_c, the LEN exploits such contextual geometric information to efficiently complete the missing parts, as shown in the example of Fig. 3b. The LEN, integrated into the same ADM network, is very simple and lightweight (Sec. 4.2), as it is realized as a lightweight encoder-decoder network based on the U-Net architecture, using just 256 channels as a bottleneck ($4M$ parameters) and skip-connections [42]. We also tried different configurations for this task, experiencing no performance increment with deeper layers. Once a contour C_{xy} is obtained from P_w and scaled to metric units (see supplementary material for details), a full 3D layout A_s is obtained (i.e., d_c and d_f, respectively, z-up and z-down components).

3.1.3 Exploiting structural information

The information in the floor projection P_f can be used to recover a floor occupancy map, useful, for example, to define a collision-free VR work area (e.g., Meta Quest room-scale), or to generate a reliable trajectory for new poses, if only to ensure that new views do not cross wall boundaries and remain in the room interior. To efficiently perform those operations, we enrich the representation with a clutter binary mask ${O_m}^{h\times w}$ in an equirectangular format. To obtain that, many lightweight pre-trained networks are available [8, 36]. Using O_m on D_h, the floor projection will be automatically cleaned-up from clutter before transforming (Eq. (2)), so that P_f will return the valid work area of the floor (Fig. 3b). Some examples of this feature are presented in the supplementary material.

3.2 Novel view synthesis

While the previous operations are performed once per image, the *gated view synthesis network (GVS)*, illustrated in Fig. 2a (cyan block), is activated whenever a movement occurs. Its purpose is to compute a new plausible spherical image from a translated position (i.e., applying a translation T to the camera). Such spherical images can then be sampled with regular means by obtaining all possible rotated views.

The GVS network includes two cascading steps: a differential rendering step, which exploits depth D_s and translation T to move pixels information to the new position, and a panoramic view synthesis step, which transforms the reprojected information into a full output image. Such a network takes as input the translated pixels $\widetilde{I}_t$, the disocclusion mask B_t and the layout edges E_t, returning as output the novel view I_t.

3.2.1 Differential rendering

Reprojecting source pixels to their target position requires finding the mapping from the source-view pixels (u_s, v_s) to the target-view pixels (u_t, v_t), which is obtained by converting source pixels to a 3D point cloud PC_s, translating it by T and converting back the resulting point cloud PC_t to image space. As pixel coordinates $(u,v)^{h\times w}$ in an equirectangular image are associated to the view-directions (θ,ϕ), we apply the pixel-wise depth D_s at the same location to recover each point in PC_t as $(m_x, m_y, m_z) = \big(D_s\cos\theta\cos\phi - T_x, D_s\sin\theta\cos\phi - T_y, D_s\sin\phi - T_z\big)$. Then, for each triplet $(m_x, m_y, m_z) \in {PC_t}^{3\times h\times w}$ we obtain coordinates in the source image:

$$(u_s, v_s) = \left(\frac{w\arctan(m_x/m_y)}{2\pi} + \frac{w}{2}, \frac{h\arctan(m_z)}{\sqrt{{m_x}^2+{m_y}^2}} + \frac{h}{2}\right). \tag{3}$$

Since many source points may contribute to the same target image pixel, we want the closer ones to occlude the further ones. In traditional rendering, this can be achieved using a z-buffer, with only the closest point contributing to the rendering of a pixel. However, this process results in a discontinuous and non-differentiable rendering function that is unsuitable for a learning framework [52]. To this end, we adopt a *soft z-buffer* approach to assign a value to each pixel of $\widetilde{I}_t$:

$$\widetilde{I}_t(u_t, v_t) = \frac{\sum_{(u_s,v_s)} I_s(u_s,v_s)(\exp(-D_s(u_s,v_s)/\tau)}{\sum_{(u_s,v_s)} \exp(-D_s(u_s,v_s)/\tau + \varepsilon}. \tag{4}$$

The exponential factor, modulated by the temperature τ (i.e., $\tau = 20$ in our experiments), enforces higher precedence for points closer to the camera. A large value of τ results in *softer* z-buffering, whereas a small value yields a rendering process analogous to standard z-buffering [52]. ε is a small constant for numerical stability.

After forward splatting through soft z-buffering, as also shown in Fig. 2, the pixels visible in both the original and translated viewpoint get the expected content, but all disoccluded areas (i.e., the areas visible from the new viewpoint but invisible in the original one) remain empty.

3.2.2 Panoramic view synthesis

The goal of view synthesis is to produce a complete image from the partial information obtained after reprojection, exploiting all the auxiliary information we have generated in previous steps. To this end, several approaches splat source-view features, filling missing holes using feature interpolation approaches [59, 60]. This approach aims to convey more semantic content, but at the same time, it irretrievably loses details of the original image, thus requiring much deeper networks to arrive at the synthesis of the image [32] and much computational effort. To overcome these problems and better adapt novel view synthesis to the VR interactive context, in our approach we address the problem as an image completion and inpainting task that leverages the recovered indoor structure to ensure consistency at various levels.

We start from the rendered image $\widetilde{I}_t^{3\times h\times w}$. As in typical inpainting approaches, we define a binary inpainting mask ${B_t}^{1\times h\times w}$, identifying missing parts in the rendered image. In contrast to pure image-domain approaches [66, 72], we further enrich the input with the additional information provided by the indoor structure to guide image completion. Since the recovered layout L_s is represented in 3D space, we first project it to the target equirectangular pose. We experienced that the most effective format is through an *edgemap* with occlusions ${E_t}^{1\times h\times w}$, that is, a map storing the visible edges of the room boundary layout (Fig. 3b). E_t is then concatenated to $\widetilde{I}_t$ and with the mask B_t (i.e., along the batch dimension - 5 layers input). It should be noted that such representation acts as edge guiding [32], but with the main difference that E_t provides information even of parts inside the mask B_t, which are invisible in the source image I_s.

To process such input, we adopt the architecture illustrated in Fig. 2a. The overall encoder-decoder scheme follows a typical design for image inpainting [14], exploiting gated convolutions for encoding/decoding [65] and dilated convolutions as bottleneck [64]. Compared to common inpainting baselines [14, 65], our architecture is thinner, deeper, and with fewer parameters. Moreover, it has only a single branch and it includes several solutions to improve accuracy and reduce computational complexity.

To simplify training and guarantee low latency at inference time, our network uses a modified version of gated convolution called Light Weight Gated Convolutions (LWGC), which reduces the number of parameters and processing time while maintaining the effectiveness [63]. The input is then encoded through a sequence of lightweight gated convolutions having different strides (Fig. 2, red blocks). Repeated dilations [64] are instead used for the bottleneck (Fig. 2, yellow blocks). The *dilated convolution operator* is implemented as a modified gated convolution:

$$D_{y,x} = \sigma(b + \sum_{i=-k'_h}^{k'_h} \sum_{i=-k'_w}^{k'_w} W_{k'_h+i,k'_w+j} \cdot I_{y+\eta i, x+\eta j}), \tag{5}$$

where η is a dilation factor, $\sigma(\dot{})$ is a component-wise non-linear transfer function and $b \in \mathbb{R}^{C_{out}}$ is the layer bias vector. In our model, we adopt, respectively, $\eta = 2, 4, 8, 16$ for the four bottleneck layers. The network decoder follows a scheme that is symmetrical with respect to the scheme of the encoder until the input resolution is reached.

3.3 Training and losses

We train both the ADM (Sec. 3.1) and GVS (Sec. 3.2) networks using a supervised training approach (Fig. 2b) on synthetic data (see Sec. 4.1). ADM requires a dataset in which ground truth depth is also available, while GVS requires only original and translated views, as it can exploit a pre-trained ADM for geometric and structural information.

3.3.1 ADM loss functions

We train the ADM network by combining and extending the standard depth and layout losses:

$$\mathscr{L}_{adm} = \lambda_d \mathscr{L}_d - \lambda_{ss} \mathscr{L}_{ss} + \lambda_l \mathscr{L}_l + \lambda_h \mathscr{L}_h. \tag{6}$$

$\mathscr{L}_d$ is the robust *Adaptive Reverse Huber Loss (BerHu)* [23] for the predicted depth;$\mathscr{L}_{ss}$ is the Structural Similarity Index Measure (SSIM), which measures the preservation of highly structured signals with strong neighborhood dependencies; $\mathscr{L}_l$ is binary cross entropy wuth logits loss for the predicted probability map P_w Sec. 3.1; $\mathscr{L}_h$ is the $L1$ distance error for the predicted ceiling-floor distances D_c and D_f. The λ weights in our experiments are $\lambda_d = 1.0, \lambda_{ss} = 0.5, \lambda_l = 0.5, \lambda_h = 0.1$.

3.3.2 GVS loss functions

The novel view synthesis network is trained by combining visual terms and indoor-domain geometric terms: $\mathscr{L}_{vsn} = \mathscr{L}_{vis} + \mathscr{L}_{geo}$.

Visual terms include losses that measure the photorealistic quality of the output:

$$\mathscr{L}_{vis} = \lambda_{px} \mathscr{L}_{px} + \lambda_{perc} \mathscr{L}_{perc} + \lambda_{style} \mathscr{L}_{style} + \lambda_{adv} \mathscr{L}_{adv}. \tag{7}$$

Here the first term is a pixel-based $L1$ loss between the predicted RGB image I_t and the ground truth target image I_{gt}, $\mathscr{L}_{perc}$ and $\mathscr{L}_{style}$ are the data-driven perceptual and style losses [7], enforcing I_{out} and I_{gt} to have a similar representation in the feature space as computed by a pre-trained $VGG-19$ [44], while $\mathscr{L}_{adv}$ is a discriminator-based loss (i.e., PatchGAN [15]). λ weights are $\lambda_{px} = 1.0$, $\lambda_{style} = 100.0$, $\lambda_{perc} = 1.0$, $\lambda_{adv} = 0.2$. Such components are a common and effective solution for many single pose inpainting problems [65].

However, in our problem the scene to be reconstructed is from a different pose, thus standard inpainting techniques return many artifacts, especially for disoccluded structures, such as wall edges or hidden corners (Fig. 4). To this end, to better exploit the guiding of the structural information, we introduce in our method specific geometric and indoor-domain loss terms, exploiting the capabilities of ADM network to return indoor scene latent representations.

As described in Sec. 3.1, our latent, compressed scene representation is given by 4 layers $L = (l_1 \ldots l_4)$, which shapes (i.e., $l \times s$), for a 1024×2048 input, are: 512×512, 512×256, 512×128, 512×64.

Since the network is pre-trained to recover pixel-wise depth and the Atlanta World model of the room, including the parts occluded in the source view, we assume that L features contain important characterizing features of the indoor scene we want to reconstruct.

Analogous to the fundamental concepts of style-transfer [7], we expect that the $L1$ distance between the latent representation of I_t and I_{gt}, respectively the predicted and ground truth target images, preserves the *high-level* content of the scene, and thus global similarity:

$$\mathscr{L}_{geocont} = \sum_{n}^{4} \left\| L_n(I_t) - L_n(I_{gt}) \right\|_1. \tag{8}$$

According to the same concepts, we also define an objective function giving more importance to local similarity, acting as a kind of *geometric style* loss, based on the *Gram matrix* function of the same 4 layers:

$$\mathscr{L}_{geostyle} = \sum_{n}^{4} \left\| K_n(L_n(I_t)^T L_n(I_t)) - L_n(I_{gt})^T L_n(I_{gt}) \right\|_1, \tag{9}$$

where K_n is the Gram matrix normalization factor $1/s * l$ for the nth selected layer.

In addition, a direct depth loss term $\mathscr{L}_{tdepth}$ is included (i.e., (BerHu) [23]) to enforce geometric consistency. It should be noted that, since the available datasets [60] do not provide a ground truth depth for the target pose, the loss is calculated by assuming as ground truth the depth predicted by the ADM network, respectively on the target ground truth image I_{gt} (dubbed $D^*{}_{gt}$) and on the predicted image I_t (dubbed D_t) (Fig. 2b). For completeness, we also tried a self-supervised loss in order to estimate the target depth without ground truth [73], but with significantly less accurate results.

As a result, the full geometric term is:

$$\mathscr{L}_{geom} = \lambda_{gcont} \mathscr{L}_{gcont} + \lambda_{gstyle} \mathscr{L}_{gstyle} + \lambda_{tdepth} \mathscr{L}_{tdepth}. \tag{10}$$

Here λ weights are $\lambda_{gcont} = 1.0$, $\lambda_{gstyle} = 100.0$, $\lambda_{tdepth} = 0.01$.

4 Results

Our approach was implemented using *PyTorch* [34] and has been tested on a large variety of indoor scenes. The accompanying video shows its usage for the exploration of free viewpoint exploration of panoramic images with HMDs. In this section, we focus on analyzing the performance of our approach for depth and layout estimation and view synthesis.

4.1 Training and testing datasets

For training our solutions, we harness the availability of public panoramic scene datasets where ground truth is available. In particular, for training and testing ADM, we exploit Structured3D [70]), a large-scale synthetic database of indoor scenes comprising 21,000 photorealistic scenes, which provides ground truth depth and layout information for each panoramic image. To train and test GVS, instead, we exploit PNVS [60], a subset of Structured3D scenes providing, for each source panoramic image, three views translated by 0.2-0.3m along random directions, and three views translated by 1.0-2.0m. In contrast to the original PNVS setup, we included the zero-motion case for a fraction of the samples (i.e., 15%) to better adapt the dataset to a common VR use case where users may remain still for a portion of the time. It should be noted that in PNVS, the ground truth depth and layout are provided only for the source pose, while only the visual rendering is provided for the target views. The pre-trained ADM network is therefore used for providing the additional geometric and structural information, which is regarded as ground truth for GVS training. All these datasets provide data at a resolution of 1024x512 that we have used for all training.

In this paper, we also use Structured3D [70] and PNVS [60] as test datasets, using official splits that do not replicate data between training and testing sets, so as to make it possible to have a comparison with other solutions. Furthermore, to demonstrate transfer learning capabilities and versatility, we present results on real-world scenes captured by non-professional users.

We also considered other commonly used datasets, but none of them were fully suitable for our task. As an example, Matterport3D [28], provides incomplete depth maps not reliable for accurate rendering of points on novel views, while MatterportLayout [76] (annotated layouts for the Matterport3D dataset), only provides layout for a limited number of rectified scenes, with manual annotation not always coincident with the underlying image [76].

In our tests, we handle novel poses in a range of $50cm$, which is well above what is required for stereo (6-7cm) and assumed consistent with natural head movements to avoid full hallucination of image content [21, 59]. The accompanying video shows typical allowed motion.

4.2 Setup and computational performance

We trained both the ADM (Sec. 3.1) and the GVS (Sec. 3.2) networks with the Adam optimizer [20], with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and an adaptive learning rate from 0.0001, on an NVIDIA RTX A5000 (24GB VRAM) with a batch size of 8 for ADM and 2 for GVS. When using the Structured3D [70] 512×1024 native resolution for both

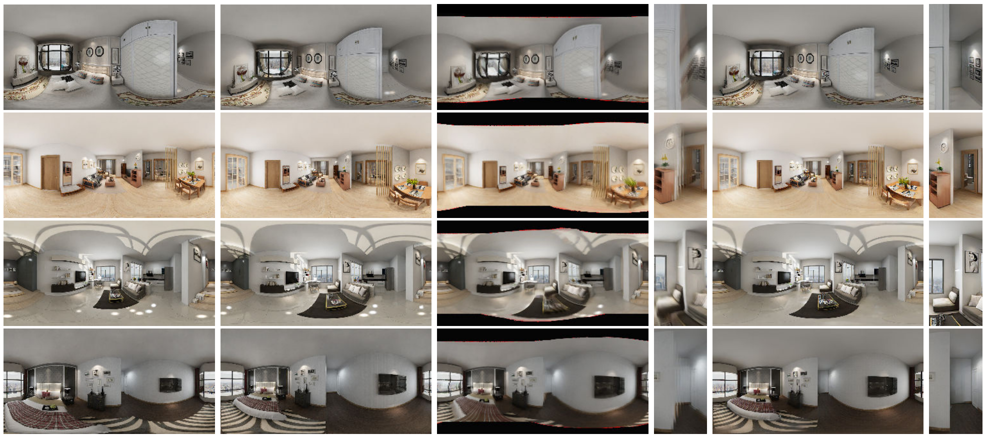


Fig. 4: We present qualitative performance and comparison vs. ground truth and PanoSynthVR [54] on the Structured3D dataset [70]. The average movement for each scene is about $50cm$ distributed on x, y, z axis.

training tasks [60, 70], the average training time for the ADM model is 76 ms/image and 428 ms/image for the GVS model. Inference time on the same NVIDIA RTX A5000 is 18 ms/image for ADM and 29 ms/image for GVS.

Method	Params↓	GFLOPS↓	Output type
Bifuse [55]	253 M	682	only depth
SliceNet [37]	79 M	101	only depth
AtlantaNet [38]	100 M	273	only layout
ADM (our)	**29 M**	**79**	**depth+layout**

Table 1: **Depth-layout estimation computational performance.** We show our computational performance compared to other specific state-of-the-art works for a 512×1024 image.

Tab. 1 presents the computational performance of our ADM compared to state-of-the-art depth and layout estimation solutions.

For depth estimation, we compare with SliceNet [37] and Bifuse [55], which are state-of-the-art methods commonly used as benchmarks in the latest panoramic works [25, 41]. Both works [37, 55] adopt as backbone a ResNet, which is often employed for depth estimation in stand-alone view synth pipelines [60], but do not use patch projection or transformers, that would add additional load to the pipeline making it less suitable for VR applications.

For layout estimation, we compare our method with AtlantaNet [38], a fast state-of-the-art solution that also handles the same scene types as ours. In particular, AtlantaNet does not use pre- and post-processing (i.e., usually done in CPU with considerable computational load), as done by pipelines based on LayoutNet [60, 75], or HorizonNet [47]. Our ADM approach is clearly the most lightweight and has lower computational complexity (GFLOPS) than the compared methods, even though it jointly performs both tasks.

Method	Params↓	GFLOPS↓	Output type
PNVS [60]	13.9 M	359	rgb
DeepFillv2 [66]	13.8 M	163	rgb
GVS (our)	**1.7 M**	**83**	**rgb-d**

Table 2: **View-synthesis computational performance.** We show our computational performance compared to other deep-learning approaches for view synthesis for a 512×1024 image.

Tab. 2 presents the computational performance for the view synthesis network (GVS). Here we compare pipelines that are, like ours, end-to-end deep learning networks. Thus, we compare to the PNVS [60] view synthesis branch, as well as, for completeness, with a state-of-the-art network for generic image inpainting (*DeepFill* [66]). Since the PNVS [60] source code is not available, we evaluate its computational cost from the information provided by the authors in the original paper, since the view synthesis network is an adaption of *EdgeConnect* [32] network.

Other types of approaches, not directly comparable in these terms, are also included for completeness. PanoSynthVR [54] exploits a pre-trained MPI network [50] to build, for each input panoramic scene, an MCI (multi-cylinder image) structure is about in $383ms$ (declared by the authors on an NVIDIA V100 GPU). Similar considerations apply to NeRF adaptations to equirectangular images. In this case, the training time is about $8h42m$ an NVIDIA RTX A5000 (24GB VRAM), with subsequent $14s$ inference time for each individual new scene view generated at 512×1024. In this case, much of the computational load is from generating new views around the main view.

4.3 Run-time performance

We ran the server connected to the display on a desktop machine equipped with an NVIDIA RTX 2080Ti. Predicting, once per scene, the enriched representation with ADM takes 32 ms, while performing per-frame view synthesis with GVS takes 39 ms. Cropping the image to 90x90° and upsampling it (2x) using *Real-ESRGAN* (with model *realesr-animevideov3*) [57] takes 39 ms. Transfering to CPU and image encoding, which in our prototype is done using TurboJPG takes an additional 5 ms/image. Server-side, thus, image computation can be performed at about 12fps, and is reduced to about 11fps including encoding and transmission. It should be noted that the machine is over 30% slower than the A5000 used for training. Moreover, encoding time could be reduced by integrating hardware JPEG encoding [33] or using alternative codecs, in particular for the ETC2 texturing format, which is widely supported on mobile platforms [31]. Client side, the WebXR application running on A PICO 4 VR headseat refreshes the display, in response to head rotations, at 72fps and updates the current panorama at the server speed. As a result, the proof-of-concept implementation supports about 70Hz refresh rate while updating panoramas at 11fps with a latency of $\approx 0.1s$, for a working volume of $\approx \pm 30cm$ around the original viewpoint. Latency was measured on the client by computing the difference from the time at which the position was sent to the server and the time the corresponding updated panorama is first displayed.

4.4 Performance vs. ground truth and competitors

As discussed in Sec. 3, the quality of the novel pose generation results strongly depends on the geometric information available, since the input image to be completed depends on the accuracy of reprojection, guided by the estimated depth, and the complementary information to guide inpanting depends on the indoor structure inferred (Sec. 3.2). For this reason, we present specific results for depth and layout estimation, followed by the results on the quality of the synthesized scene.

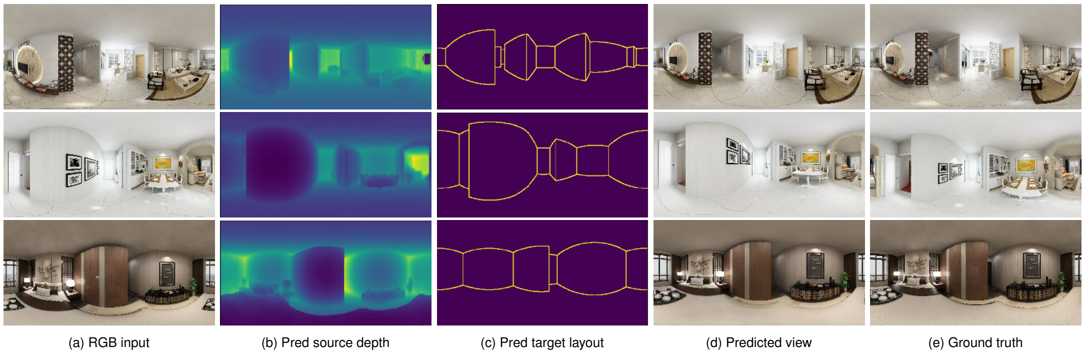


Fig. 5: We present our qualitative performance on scenes with structural occlusions. The average movement for each new pose is about 60 cm distributed on x, y, z axis.

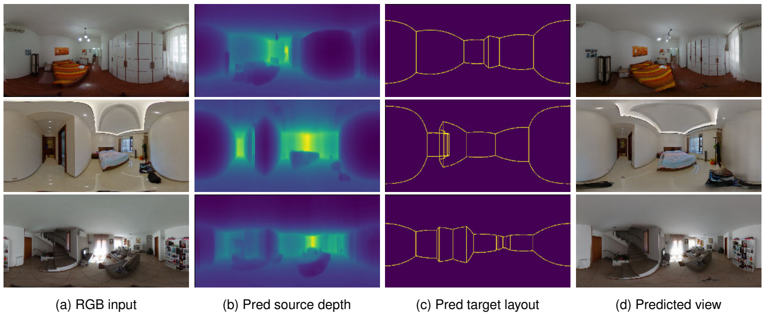


Fig. 6: We present our qualitative performance on scene acquired by non-professional users. Input resolution here is 6720×3360. The average movement for each scene is about 40 cm.

For depth estimation, in Tab. 3 we summarize our performance compared to SliceNet [37], and to the work of Jin et al. [18], which is a representative pipeline that jointly predicts depth and layout, where layout is predicted through LayoutNet [60, 75]. While the source code is available for SliceNet, for Jin et al. [18] we compare with their official results, where only depth estimation performance is available. For a fair comparison, we adopt the Structured3D [70] splitting of Jin et al. [18], adapting both SliceNet and our code to it. We adopt common metrics, i.e., mean squared error (MSE) and 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 layout estimation, we compare, instead, with AtlantaNet [38], an end-to-end solution that, like ours, does not require Manhattan World pre and post-processing to work [76]. Here, we adopt the common metrics IoU3D (volumetric intersection over union) and IoU2D (pixel-wise intersection over union). The results demonstrate how our method achieves state-of-the-art performance in both tasks, despite the lower computational burden compared to those baselines. To evaluate the gated view synthesis network (GVS), we compared our performance to the one achieved by state-of-the-art methods [12, 50, 54], which are representative and suitable for VR applications, as discussed in Sec. 2, and for which source code was available.

Method	mse↓	rmse↑	δ_1 ↑	iou3d↑	iou2d↑
Jin et al. [18]	0.103	0.666	0.91	-	-
SliceNet [37]	0.044	0.174	0.93	-	-
AtlantaNet [38]	-	-	-	82.45	85.78
AVN (Our)	**0.008**	**0.043**	**0.96**	**84.56**	**88.86**

Table 3: **Depth and layout performance.** We show our quantitative performance compared to other representative state-of-the-art works.

In Tab. 4, we present our quantitative results compared to the solutions already exploited for panoramic VR applications [54] that can be trained end-to-end and for which a comparison with respect to ground truth was possible [50, 54]. Specifically, MPI [50] is adopted by PanoSynthVR [54] to generate multi-cylinder-images (in their case with 32 layers) from a single panoramic view, as well as by MatryODShka [2] to generate low-resolution views form stereoscopic panoramic input. We also considered Synsin [59], but no official panoramic implementation was available, and only low-resolution results have been presented [60].

Method	PSNR↑	SSIM↑	LPIPS↓
MPI 32 [50, 54]	17.59	0.768	0.263
MPI 64 [50]	17.93	0.783	0.258
MPI 128 [50]	18.22	0.789	0.252
GVS (Our)	**22.97**	**0.817**	**0.178**

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

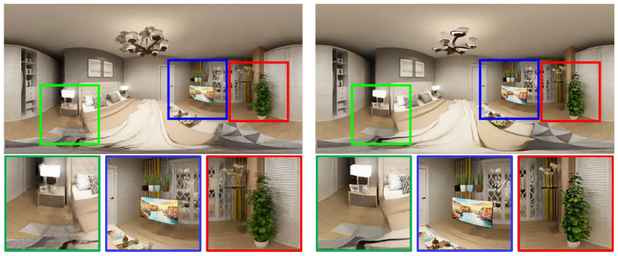
Since such multi-layer approaches have a limited working range, we adopt the PNVS benchmark called *easy set* [60], which limits motion to 0.1-0.2cm. Differently from the experiments proposed in the PNVS paper [60], we choose to test on Structured3D full resolution (i.e., 512×1024), since the benchmark proposed in the PNVS paper [60] was run at a resolution of 256×512, way too low for our applications.

Tab. 4 summarizes the results with standard metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) [58], and the Learned Perceptual Image Patch Similarity (LPIPS) [67]. These results show how our method outperforms other solutions in terms of accuracy in all metrics. It should be noted that the current approaches are adaptations of methods based on planar or semi-planar projections, while our method fully exploits depth and layout data.

Qualitative results on a variety of indoor scenes are presented in Fig. 4, Fig. 5, Fig. 6, and Fig. 7. Fig. 4 shows our performance compared with a recent approach for VR applications based on MPI [54], using the same target position and converting the cylindrical output of that system to an equirectangular map. As in Tab. 4, we adopt data for which ground truth is available [60]. In this regard, the same images provided by PanoSynthVR are not usable for direct comparison, since these are cylindrical crops and not full equirectangular scenes. The comparison shows how our method is able to predict occluded and disoccluded parts even in the presence of significant structural occlusions, such as corridors to particularly bulky furniture. Furthermore, besides our superior performance in terms of accuracy, it should be noted that the compared solution, although returning a perceptual plausible view, does not reconstruct a spatially reliable scene, probably due to approximation with a limited number of planes/cylinders.

In Fig. 5, we present instead qualitative examples of our performance, illustrating different tasks. Alongside the input source image, we show the predicted source depth, the predicted layout translated at the target position, our prediction at the target position, and the ground truth target. In this case, it is noticeable the correlation of depth and layout quality with the generated output view.

In addition to the results on synthetic scenes, we present in Fig. 6 qualitative performances on real-world, user-captured scenes. Here we exploit the training with Structured3D [70] to predict depth, layout, and novel views, from an input 6720 × 3360 images captured by a Ricoh Theta S. Also in this case our method returns visually realistic reconstructions. Finally, we present a qualitative comparison with a NeRF approach. As recent omnidirectional image-based methods attempt to build a NeRF structure from a single image (Sec. 2), in Fig. 7 we show a comparison to a NeRF-based approach for equirectangular images, OmniNeRF [12]. In this case, as no ground truth is available, we present a qualitative assessment of the data made available by the authors, considering their same spatial range (i.e., 30*cm*). As clearly highlighted in Fig. 7 details, our solution provides better accuracy in many parts of the scene. In this context, we expect that our work could be used to generate input views for further NeRF processing.



(a) OmniNeRF [12]

(b) Our

Fig. 7: **Comparison to NeRF.** we show a comparison to OmniNeRF [12], using data released by the authors. As the details clearly show, our solution provides better accuracy in many parts of the scene.

4.5 Discussion and ablation study

Tab. 5 illustrates the results of an ablation study made to analyze the major technical choices of our method. The first test (first and second row in Tab. 5) shows the importance of effective depth estimation. The accuracy of depth is critical to synthesize the new pose, as the correct displacement of elements visible from the new view depends on it [59]. In our ablation study, the first row presents results where depth is estimated with a domain-independent approach with a baseline that exploits only multi-resolution aggregation [60] but without compression according to a preferred direction. Using gravity-aligned features here results in a great increment in performance.

Row 3 and 4 in Tab. 5 show the contribution of layout knowledge to handle occlusion and disocclusion. In this case, we detail the performance difference between using a standard layout estimation with Manhattan World post-processing [47], vs. our approach (i.e., ATL - row 4). Finally, in row 5 we show our full-configuration performance, even using our novel geometric perceptual and style losses (see Sec. 3.3). It should be noted how this contribution mainly affects SSIM and LPIPS.

GAF	MW	ATL	GPS	PSNR↑	SSIM↑	LPIPS↓
-	-	-	-	16.87	0.693	0.325
✓	-	-	-	18.28	0.751	0.206
✓	✓	-	-	19.12	0.776	0.186
✓	-	✓	-	22.01	0.798	0.181
✓	-	✓	✓	**22.97**	**0.817**	**0.178**

Table 5: **Ablation stats.** We show the effect of several key choices of our approach. In bold is the adopted configuration. GAF: gravity aligned features-based depth estimation; MW: standard Manhattan World layout estimation; ATL: Atlanta transform structure estimation; GPS: geometric perceptual and style loss.

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. In such bad cases (i.e., illustrated and discussed in the supplementary material), the lack of performance depends on the quality of the depth associated with the initial view. This depth, in fact, depends on the projection of the visible pixels or any associated features, which is, in fact, the real input to the actual view synthesis network. In this sense, even the commonly used soft z-buffer method [52] may be subject to error and may be the subject of future work.

We also noted that, even if the structure and depth of the scene are predicted once and remain the same for all frames, visible instabilities may occur in the form of flickering in the application in which a new image is generated per frame. Examples are visible in the accompanying video. This is due, mainly, to small changes in the reprojection that trigger large changes in predicted images. We plan to mitigate this problem through the inclusion of regularization terms, as well as by augmenting the training sets.

Another important discussion point is the resolution of the generated images. Currently, the biggest limitation is the resolution of the available training datasets, which is still below the capabilities of modern VR viewers. Although new datasets will soon be available at higher resolutions, one practical solution is now the use of fast super-resolution methods [6, 57]. The accompanying video shows the configuration where a 2x upsampling of the presented view is combined with the prediction at the 1024x512 resolution.

5 Conclusions

We have presented a novel deep learning approach that extracts geometric and structural information from a single panorama in order to quickly synthesize plausible panoramic images from close-by viewpoints within a workspace suitable for VR applications.

Our end-to-end approach is particularly compact and lightweight, and introduces several innovations. In particular, our novel integrated network for estimating an environment's depth and permanent structure produces elements that are crucial requirements for ensuring reliable view synthesis. By incorporating novel domain-specific loss functions, we shift the major computational load on the training phase, and obtain an extremely lightweight network at prediction time. As a result, our method automatically produces compelling new poses ready for interactive use. Moreover, the extracted floor plan and 3D wall structure can also be used to support room exploration.

Our future work will concentrate on further improving the performance, especially on larger-size images, as well as the stability for its use in real-time exploration. We also plan to integrate this work with other solutions for the dynamic exploration of panoramic images, such as automatic room emptying and editing [36, 51].

Acknowledgments

This publication was made possible by NPRP-S grant NPRP14S-0403-210132 by Qatar National Research Fund (a member of Qatar Foundation) and by Sardinian Regional Authorities through project XDATA. The findings herein reflect the work, and are solely the responsibility of the authors.

References

[1] M. Aly and J.-Y. Bouguet. Street view goes indoors: Automatic pose estimation from uncalibrated unordered spherical panoramas. In *Proc. WACV*, pp. 1–8, 2012. 1

[2] B. Attal, S. Ling, A. Gokaslan, C. Richardt, and J. Tompkin. Matryodshka: Real-time 6dof video view synthesis using multi-sphere images. In *Proc. ECCV*, pp. 441–459. Springer, 2020. 1, 3, 8

[3] T. Bertel, N. D. Campbell, and C. Richardt. MegaParallax: Casual 360° panoramas with motion parallax. *IEEE TVCG*, 25(5):1828–1835, 2019. 2

[4] T. Bertel, M. Yuan, R. Lindroos, and C. Richardt. Omniphotos: casual 360 vr photography. *ACM Transactions on Graphics (TOG)*, 39(6):1–12, 2020. 2

[5] M. Broxton, J. Flynn, R. Overbeck, D. Erickson, P. Hedman, M. Duvall, J. Dourgarian, J. Busch, M. Whalen, and P. Debevec. Immersive light field video with a layered mesh representation. *ACM Transactions on Graphics (TOG)*, 39(4):86–1, 2020. 1, 3

[6] X. Deng, H. Wang, M. Xu, Y. Guo, Y. Song, and L. Yang. Lau-net: Latitude adaptive upscaling network for omnidirectional image super-resolution. In *Proc. CVPR*, pp. 9189–9198, June 2021. 9

[7] 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 6

[8] 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. 4, 5

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. CVPR*, pp. 770–778, 2016. 4

[10] P. Hedman, S. Alsisan, R. Szeliski, and J. Kopf. Casual 3d photography. *ACM Transactions on Graphics (TOG)*, 36(6):1–15, 2017. 3

[11] P. Hedman and J. Kopf. Instant 3d photography. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018. 3

[12] C.-Y. Hsu, C. Sun, and H.-T. Chen. Moving in a 360 world: Synthesizing panoramic parallaxes from a single panorama. *arXiv preprint arXiv:2106.10859*, 2021. 1, 2, 3, 8, 9

[13] J. Huang, Z. Chen, D. Ceylan, and H. Jin. 6-DOF VR videos with a single 360-camera. In *Proc. IEEE VR*, pp. 37–44, 2017. 2

[14] 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 5

[15] 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. 6

[16] H. Jia, H. Yi, H. Fujiki, H. Zhang, W. Wang, and M. Odamaki. 3d room layout recovery generalizing across manhattan and non-manhattan worlds. In *Proc. CVPR Workshops*, pp. 5188–5197, 2022. doi: 10.1109/CVPRW56347.2022.00567 1

[17] Z. Jiang, Z. Xiang, J. Xu, and M. Zhao. Lgt-net: Indoor panoramic room layout estimation with geometry-aware transformer network. In *Proc. CVPR*, pp. 1654–1663, 2022. 1, 2

[18] 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. 8

[19] V. Kelkkanen, M. Fiedler, and D. Lindero. Synchronous remote rendering for VR. *Int. Journal of Computer Games Technology*, 2021:1–16, 2021. 1

[20] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *ArXiv e-print arXiv:1412.6980*, 2014. 6

[21] P. K. Lai, S. Xie, J. Lang, and R. Laganière. Real-time panoramic depth maps from omni-directional stereo images for 6 dof videos in virtual reality. In *Proc. IEEE VR*, pp. 405–412. IEEE, 2019. 6

[22] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *Proc. 3DV*, pp. 239–248, 2016. 2

[23] S. Lambert-Lacroix and L. Zwald. The adaptive BerHu penalty in robust regression. *Journal of Nonparametric Statistics*, 28:1–28, 2016. 6

[24] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *Proc. CVPR*, pp. 2136–2143, 2009. 2

[25] Y. Li, Y. Guo, Z. Yan, X. Huang, Y. Duan, and L. Ren. Omnifusion: 360 monocular depth estimation via geometry-aware fusion. In *Proc. CVPR*, pp. 2801–2810, 2022. 2, 7

[26] K.-E. Lin, Z. Xu, B. Mildenhall, P. P. Srinivasan, Y. Hold-Geoffroy, S. DiVerdi, Q. Sun, K. Sunkavalli, and R. Ramamoorthi. Deep multi depth panoramas for view synthesis. In *Proc. ECCV*, pp. 328–344. Springer, 2020. 3

[27] B. Luo, F. Xu, C. Richardt, and J.-H. Yong. Parallax360: Stereoscopic 360 scene representation for head-motion parallax. *IEEE transactions on Visualization and Computer Graphics*, 24(4):1545–1553, 2018. Proc. IEEE VR. 2

[28] Matterport. Matterport3D. `https://github.com/niessner/Matterport`, 2017. [Accessed: 2022-09-25]. 6

[29] 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. 1

[30] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2, 3

[31] J.-H. Nah. QuickETC2: Fast ETC2 texture compression using luma differences. *ACM Trans. Graph.*, 39(6), nov 2020. 7

[32] 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 5, 7

[33] NVIDIA. nvJPEG Libraries: GPU-accelerated JPEG decoder, encoder and transcoder. `https://developer.nvidia.com/nvjpeg`, 2023. [Accessed: 2023-06-06]. 7

[34] 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. 6

[35] G. Payen de La Garanderie, A. Atapour Abarghouei, and T. P. Breckon. Eliminating the blind spot: Adapting 3D object detection and monocular depth estimation to 360 panoramic imagery. In *Proc. ECCV*, pp. 812–830, 2018. 2

[36] G. Pintore, M. Agus, E. Almansa, and E. Gobbetti. Instant automatic emptying of panoramic indoor scenes. *IEEE Transactions on Visualization and Computer Graphics*, 28(11):3629–3639, 2022. Proc. ISMAR. 5, 9

[37] 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. 2, 4, 7, 8

[38] G. Pintore, M. Agus, and E. Gobbetti. AtlantaNet: Inferring the 3D indoor layout from a single 360 image beyond the Manhattan World assumption. In *Proc. ECCV*, pp. 432–448, 2020. 2, 7, 8

[39] 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. 2, 4

[40] 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. 1, 2, 4

[41] M. Rey-Area, M. Yuan, and C. Richardt. 360monodepth: High-resolution 360deg monocular depth estimation. In *Proc. CVPR*, pp. 3762–3772, 2022. 2, 7

[42] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proc. MICCAI*, pp. 234–241, 2015. 5

[43] A. Serrano, I. Kim, Z. Chen, S. DiVerdi, D. Gutierrez, A. Hertzmann, and B. Masia. Motion parallax for 360 rgbd video. *IEEE Transactions on Visualization and Computer Graphics*, 25(5):1817–1827, 2019. Proc. IEEE VR. 3

[44] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6

[45] Y. Su and K. Grauman. Kernel transformer networks for compact spherical convolution. In *Proc. CVPR*, pp. 9434–9443, 2019. 2

[46] Y.-C. Su and K. Grauman. Learning spherical convolution for fast features from 360 imagery. In *Advances in Neural Information Processing Systems 30*, pp. 529–539, 2017. 2

[47] C. Sun, C.-W. Hsiao, M. Sun, and H.-T. Chen. HorizonNet: Learning room layout with 1D representation and pano stretch data augmentation. In *Proc. CVPR*, pp. 1047–1056, 2019. 1, 2, 5, 7, 9

[48] C. Sun, M. Sun, and H.-T. Chen. HoHoNet: 360 indoor holistic understanding with latent horizontal features. In *Proc. CVPR*, pp. 2573–2582, 2021. 2

[49] K. Tateno, N. Navab, and F. Tombari. Distortion-aware convolutional filters for dense prediction in panoramic images. In *Proc. ECCV*, pp. 732–750, 2018. 2

[50] R. Tucker and N. Snavely. Single-view view synthesis with multiplane images. In *Proc. CVPR*, pp. 551–560, 2020. 1, 3, 7, 8

[51] M. Tukur, G. Pintore, E. Gobbetti, J. Schneider, and M. Agus. SPIDER: A framework for processing, editing and presenting immersive high-resolution spherical indoor scenes. *Graphical Models*, 128:101182:1–101182:11, July 2023. 9

[52] S. Tulsiani, R. Tucker, and N. Snavely. Layer-structured 3d scene inference via view synthesis. In *Proc. ECCV*, pp. 302–317, 2018. 5, 9

[53] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, vol. 30, 2017. 4

[54] J. Waidhofer, R. Gadgil, A. Dickson, S. Zollmann, and J. Ventura. PanoSynthVR: Toward light-weight 360-degree view synthesis from a single panoramic input. In *Proc. ISMAR*, pp. 584–592. IEEE, 2022. 1, 2, 3, 7, 8

[55] 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. 2, 7

[56] F.-E. Wang, Y.-H. Yeh, M. Sun, W.-C. Chiu, and Y.-H. Tsai. LED2-Net: Monocular 360 layout estimation via differentiable depth rendering. In *Proc. CVPR*, pp. 12956–12965, 2021. 2

[57] X. Wang, L. Xie, C. Dong, and Y. Shan. Real-ESRGAN: Training real-world blind super-resolution with pure synthetic data. In *Proc. ICCVW*, 2021. 3, 7, 9

[58] 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. 8

[59] O. Wiles, G. Gkioxari, R. Szeliski, and J. Johnson. Synsin: End-to-end view synthesis from a single image. In *Proc. CVPR*, pp. 7467–7477, 2020. 2, 3, 5, 6, 8, 9

[60] J. Xu, J. Zheng, Y. Xu, R. Tang, and S. Gao. Layout-guided novel view synthesis from a single indoor panorama. In *Proc. CVPR*, pp. 16438–16447, 2021. 1, 2, 3, 5, 6, 7, 8, 9

[61] 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. 1

[62] S.-T. Yang, F.-E. Wang, C.-H. Peng, P. Wonka, M. Sun, and H.-K. Chu. DuLa-Net: A dual-projection network for estimating room layouts from a single RGB panorama. In *Proc. CVPR*, pp. 3363–3372, 2019. 2, 4

[63] 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. 5

[64] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In Y. Bengio and Y. LeCun, eds., *Proc. ICLR*, 2016. 5

[65] 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. 5, 6

[66] 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. 5, 7

[67] 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. 8

[68] 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. 2

[69] Y. Zhao, C. Wen, Z. Xue, and Y. Gao. 3d room layout estimation from a cubemap of panorama image via deep manhattan hough transform. In *Proc. ECCV*, pp. 637–654. Springer, 2022. 1, 2

[70] 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. 6, 7, 8, 9

[71] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018. 3

[72] P. Zhu, R. Abdal, Y. Qin, and P. Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In *Proc. CVPR*, pp. 5104–5113, 2020. 5

[73] N. Zioulis, A. Karakottas, D. Zarpalas, F. Alvarez, and P. Daras. Spherical view synthesis for self-supervised 360° depth estimation. In *Proc. 3DV*, pp. 690–699, 2019. 6

[74] N. Zioulis, A. Karakottas, D. Zarpalas, and P. Daras. OmniDepth: Dense depth estimation for indoors spherical panoramas. In *Proc. ECCV*, pp. 453–471, 2018. 2

[75] C. Zou, A. Colburn, Q. Shan, and D. Hoiem. LayoutNet: Reconstructing the 3D room layout from a single RGB image. In *Proc. CVPR*, pp. 2051–2059, 2018. 2, 7, 8

[76] 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. 6, 8

[77] C. Zou, J.-W. Su, C.-H. Peng, A. Colburn, Q. Shan, P. Wonka, H.-K. Chu, and D. Hoiem. 3d manhattan room layout reconstruction from a single 360 image. *ArXiv e-print arXiv:1910.04099*, 2019. 2