PanoFloor: reconstruction and immersive exploration of large multi-room scenes from a minimal set of registered panoramic images using denoised density maps

Giovanni Pintore*
CRS4, Italy
National Research Center in HPC, Big Data, and QC, Italy

Sara Jashari†
ICT Division
CSE, HBKU, Qatar

Marco Agus‡
ICT Division
CSE, HBKU, Qatar

Enrico Gobbetti§
CRS4, Italy
National Research Center in HPC, Big Data, and QC, Italy

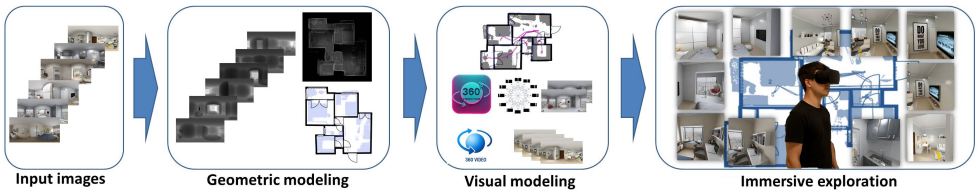


Figure 1: ***PanoFloor* overview.** Combining depth prediction, density map aggregation with diffusion-based refinement, structural segmentation, and view synthesis, we transform a small set of 360° panoramas into a 3D multi-room floor plan and a visual graph of stereoscopic views connected by optimized paths, enabling real-time exploration on standard VR headsets.

Abstract

We introduce a deep learning approach to automatically generate 3D floor plans and immersive multi-room virtual visit experiences from a small set of co-registered 360° panoramas – down to just one per room. We integrate novel neural networks that leverage panoramic image broad context and large annotated room datasets to build a geometric and visual graph. Nodes represent stereo-viewable multiple-center-of-projection (MCOP) 360° images at the capture locations, while arcs connect them with paths through doors, avoiding clutter and minimizing disocclusions to maximize visual quality. The process starts with depth prediction and floor-plan projection to create a comprehensive but noisy global density map, which is refined via a latent diffusion model. A segmentation network then extracts room layouts, openings, and clutter. This structured representation is lifted to a visual one by creating a 360° stereo-explorable MCOP representation at each node, produced using a view-synthesis network from the original image and its predicted depth map. Arc paths are then computed using an optimization process that considers structural constraints, including openings and obstacles, while minimizing visual discontinuities, occlusions, and disocclusions. Finally, 360° video transitions are synthesized using a specialized view-synthesis network to obtain a fully precomputed WebXR-ready explorable representation that can be efficiently experienced on Head-Mounted-Displays with limited graphics capabilities. The extracted floor plan not only aids in documenting the captured building but can also enhance immersive experiences by serving as a live map of the building. Our experiments show that the method achieves state-of-the-art reconstruction from sparse inputs and supports compelling immersive visits.

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

*e-mail: giovanni.pintore@crs4.it
†e-mail: sarajashari@infidev.mk
‡e-mail: magus@hbku.edu.qa
§e-mail: enrico.gobbetti@crs4.it

1 Introduction

Photorealistic virtual tours of built multi-room spaces like homes and offices provide interactive access to real environments [13], with applications ranging from remote property viewing in real estate [54] to emergency response planning [3] and beyond. Across domains, essential actions include establishing presence through visual exploration, recognizing room location within the building, and navigating between rooms [78]. Image-based techniques are the most practical way to generate realistic virtual visits, as manual 3D modeling is complex, costly to create and maintain, and often does not match the as-built or as-inhabited situation. 360∘ imaging has become standard, with consumer cameras enabling quick, full-scene capture [22]. Capturing just a few spherical images, ideally one per room, provides good coverage of he habitable and walkable space of indoor environments. This scenario is representative of common practices, particularly in the real estate industry, where millions of causal users create shareable virtual tours this way, using consumer cameras and overlapping views (e.g., open doors) for alignment [9, 18]. Moreover, viewing 360° images in a Head-Mounted Display (HMD) enables intuitive exploration via head movements, leading to easy-to-use Virtual/Augmented/Extended Reality (VR/AR/XR) applications [70, 32].

Creating virtual visits from a few connected spherical images is appealing, but current approaches are limited by reduced presence and high setup effort (Sec. 2). First of all, the locking of users to capture points and the lack of binocular stereo reduce presence in indoor spaces where close-range geometry is abundant [61]. Moreover, and crucially, immersive multi-room experiences need not only depth for parallax [13], but also detection of walls, doors, and clutter-free areas to map spaces and enable coherent room-to-room navigation [78].

To tackle these challenges, we introduce *PanoFloor*, a deep learning approach for creating floor plans and virtual visits from a minimal set of co-registered, gravity-aligned 360° shots (Fig. 1). Leveraging the broad context present in a panorama and patterns learned from large annotated datasets, we reconstruct a visual, geometric, and semantic model in the form of a graph using as few as a single image per room. Nodes represent multiple center-of-projection (MCOP) viewpoints at capture locations, and arcs define spherical videos following paths through doors and around clutter, minimizing disocclusions to enhance visual quality.

Our fully automatic pipeline progressively incorporates broader

global information (Sec. 4). We start from the common key assumption that indoor geometry and semantics can be largely inferred from the vertical geometry distribution relative to the ground [52]. From each input 360° image, the predicted depth is projected onto a common ground plane using the available relative poses to form a global density map. Since, differently from methods relying on dense and precise measurements coming from 3D devices or multi-view capture [8, 75], monocular visual inference at the individual room level may produce inconsistent global results, we refine the density map using a latent diffusion model to enhance geometry and semantics. We then segment the floor plan to extract room layouts, openings, and clutter maps. As a result, we obtain a consistent, plausible floor plan approximation from which to extract a full visual representation. First, we create nodes in the resulting graph using the panoramic depth to produce a stereoscopic MCOP image. Nodes are connected by arcs within or across rooms via doors. For each arc, we compute an optimal path that avoids clutter and minimizes visual discontinuities, generating a photorealistic 360° transition video using a view-synthesis network. The resulting precomputed representation in the form of a structured scene graph of stereoscopic views linked by panoramic videos enables fast, immersive, and coherent exploration across VR devices, including HMDs with minimal graphics capabilities. Moreover, the extracted floor plan provides valuable building documentation and can enhance visits by mapping the user's location within the space. Our approach provides the following key contributions.

- We introduce a latent diffusion method to refine noisy, incomplete density maps by enhancing their semantic content. Standard generative diffusion models [46], designed and trained for common imagery, are suboptimal for this task due to the sampling and detail requirements of occurrence maps. Our lightweight approach models noise as the difference between the diffusion input and ground truth density maps (Sec. 4.2).
- We introduce a lightweight autoencoder tailored for density maps to support latent diffusion processing (Sec. 4.2). Standard latent variational autoencoders [46] have higher computational costs and, even after fine-tuning, yield suboptimal results (Tab. 3). Our gated and dilated convolutions efficiently capture geometric and semantic indoor details (Sec. 7).
- We leverage geometric information to enhance view synthesis and navigation (Sec. 5) , computing optimal paths that balance geometric constraints (e.g., passing through doors and avoiding clutter) with visual quality (e.g., maximizing reconstruction quality by minimizing disocclusions). Additionally, we extend recent depth estimation techniques [41] to improve reconstruction and reduce visual artifacts in view synthesis.
- We propose an end-to-end pipeline to reconstruct an indoor multi-room model, including walls, openings, and clutter ma,p just starting from a set of registered images (Sec. 4), unlike competing techniques that require dense point clouds acquired with measurement instruments [52, 8, 75, 6].

Our experiments and performance analyses (Sec. 7) show that our methods improve over the state-of-the-art in different tasks, in terms of accuracy, quality, and computational complexity, producing image-based models ready to be interactively explored in HMD devices. We also show that our model, trained on synthetic data, can produce compelling predictions on user-captured images.

2 Related Work

Transforming sparse imagery of multi-room environments into image-based representations for VR/AR/XR exploration involves many research areas. In the following, we focus on the most relevant data-driven approaches for panoramic images.

Capture constraints Our method takes as input a small set of gravity-aligned, registered 360° panoramas. Most, if not all, indoor datasets preserve gravity alignment, as it is guaranteed by tripods or easily achievable in freehand capture by performing upright adjustment using IMU data, image-based vertical direction detection, or a combination of both [39, 55, 9]. Co-registration of images taken in different rooms can be done with printed markers [37], often used also for precise scale detection [9], but can also be achieved from raw capture data. Methods include exploiting the small overlaps that occur through open doors with photogrammetry [71, 72] or RGB-based learning [18], exploiting semantic cues like room types [48], or aligning the detected openings (i.e., doors) [25]. These problems are orthogonal to the ones treated in this paper, and our pipeline does not depend on the methods used for gravity alignment and co-registration.

Multi-room reconstruction Reconstructing a 3D floor plan from purely visual input in terms of rooms bounded by walls, ceilings, and floors, and connected through passages, provides the 3D geometry and the connections required to implement an effective 3D exploration. Since reconstruction from sparse visual input alone is an ill-formed problem, the regularities present in common environments, like homes, offices, and public buildings, are typically used to aid reconstruction from incomplete or noisy data [13, 19]. Early methods used image processing, architectural priors, and optimization to extract structure from images or depth data [12, 50, 5, 35, 19], but struggled in complex scenes due to reliance on feature detection (e.g., corners and edges) and strong assumptions (e.g., Manhattan World) to support geometric reasoning and optimization. More recent methods combine deep learning for extracting hidden relations with optimization for room reconstruction. Techniques include corner detection with neural networks, followed by integer programming for wall segment reconstruction [29], instance segmentation to identify room regions combined with path solvers for structured polygon generation [7, 36]. Some methods further refine room shapes using Monte Carlo Tree Search [52], while others use constrained diffusion and graph neural networks for multi-room layout synthesis [47, 14]. In contrast to hybrid approaches, end-to-end deep learning models aim to predict room structures directly from input density maps without complex post-processing. HEAT [8] introduces a bottom-up pipeline that integrates deformable transformers [81] to detect corners and classify edge connections, enabling a flexible representation of room boundaries. Similarly, RoomFormer [75] extends DETR-based architectures to predict structured floorplans from dense density maps, encoding indoor structures as variable-length polygon sequences. SLIBO-Net [53] builds on RoomFormer by incorporating additional geometric and semantic priors, improving accuracy through multi-view consistency and learned post-processing refinements. More recently, PolyDiffuse [6] has introduced a diffusion-based refinement technique that enhances polygonal reconstructions. By treating floorplan reconstruction as a conditional generative process, PolyDiffuse serves as a post-processing step that can refine outputs from various baseline methods, including RoomFormer. Our approach introduces key improvements to end-to-end learning solutions. First, extending our recent work on multi-room reconstruction [43], we predict dense depth maps and derive a single density map directly from images, without the need for 3D point clouds from multi-view capture or measuring instruments. Furthermore, we introduce a latent denoising approach to enhance geometric and semantic features in the predicted density map, and exploit reconstruction data to drive the construction of a visual and semantic graph for exploration. The proposed method is not constrained to Manhattan-World and can recover multi-room structures with slanted walls. Enforcing right angles, if needed for some application, can be eventually achieved by performing adaptive corrections as a post-process (i.e., rectifying walls meeting at $\approx 90°$ [21]).

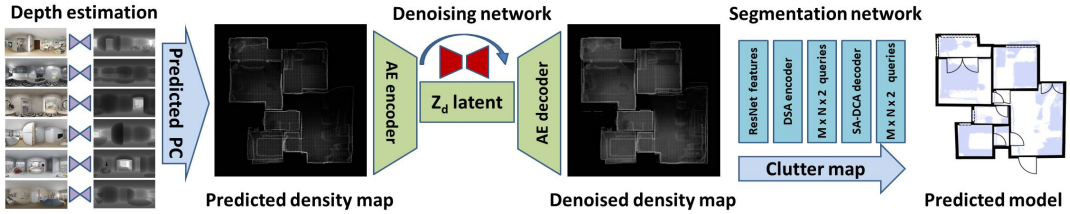


Figure 2: **Geometric modeling overview.** Depth maps predicted by a depth network guide both novel view synthesis and structural reconstruction. A denoising network and gated auto-encoder refine a unified density map, which is segmented to extract room layouts, openings, and clutter. The result is scaled to world dimensions for accurate floorplan modeling.

Immersive indoor exploration Structured representations capture multi-room geometry and topology, and must be enhanced for immersive visualization, enabling stereo cues, motion parallax, and navigation. Single-shot panoramas offer a convenient way to replicate real-world environments but are limited to head rotation with flat appearances, leading to artifacts [61, 32]. To address these constraints, methods targeting restricted viewer motion have been developed, supporting stereo or small head movements by generating compact representations for specific environments. Panoramic images combined with depth maps have enabled diverse view synthesis approaches, such as point cloud rendering [17], depth map-based meshes [57], and blended RGB-D data [30]. Despite advances in immersive deep learning-based solutions, computational challenges often limit their direct application to embedded devices, imposing remote rendering for Head-Mounted Displays (HMDs) [69, 41]. This has led to the emergence of specialized precomputed representations, such as Layered depth images [16] and multi-plane panoramas (MPI) [56, 28]. For stereo generation, techniques like omnidirectional stereo rendering [42] produce equirectangular images tailored to VR, although peripheral stereo accuracy remains a challenge. Gaze-adaptive rendering offers an alternative by dynamically adjusting depth images [31]. Larger displacements from the capture position require transforming panoramas into 3D models, often leveraging semantics-driven frameworks [68, 77]. However, such methods struggle with real-world environments filled with unrecognized objects. In that case, Neural Radiance Fields (NeRF) [33] and 3D Gaussian Splats (3DGS) [23] provide novel view synthesis, with recent single-view adaptations [20, 44] addressing panorama-based challenges through RGB-D inpainting and mesh refinement. Notably, iterative mesh optimization has shown promise for handling occlusions and underfitted geometries [44]. While methods like PERF [62] focus on generating 3D-consistent novel views and representations, trajectory constraints and blurring in occluded areas persist as limitations. More recent solutions iteratively refine meshes by leveraging inpainting techniques, incorporating occluded color and geometry to improve mesh quality [44]. Generation, however, is still costly and does not provide the quality and resolution required for real-time exploration on an HMD. In our case, we fully precompute a 3D representation made of stereo panoramas connected by transitions on optimized paths. For stereo exploration, we refine a deep-learning solution to generate omnidirectional stereo from a single 360° image [42], exploiting the geometric data available from multi-room reconstruction. For navigation, we introduce an approach to synthesize omnidirectional videos along paths that minimize view-synthesis artifacts in a reconstructed environment. The generation of video transitions from panoramas is a well-researched problem. The classic solution is to match feature points in the start and end panoramas and use them for constraining warping and cross-blending [79]. Since this class of methods suffers from ghosting and deformation artifacts caused by warping the non-matching geometry, later work has exploited depth estimation and view synthesis to achieve a more natural interpolation between wide-baseline panoramas (e.g., [27, 51]). We use the same machinery, as explained above, to synthesize views along transition paths, exploiting the same view synthesis network used for stereo. Indoors, however, straight line interpolation of viewpoints is not sufficient and must be complemented by curved path computation. Classic methods (e.g., [10]) generated a connecting path through collision avoidance and smoothness optimization, eventually taking into account the maximization of viewpoint selection quality (e.g., [11, 4]). These methods, however, rely on detailed geometry, typically from 3D models or dense multiview capture. Starting from purely visual data, several authors proposed optimizing neural aesthetic metrics [64] on synthesized views to select appealing viewpoints [2, 58] and paths [67], though they focus on single-room navigation. We adopt a similar strategy, but generate paths across rooms by linking door-connected segments, avoiding clutter and minimizing disocclusions.

3 Overview

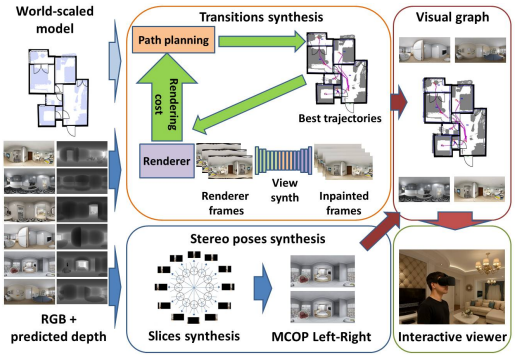


Figure 3: **Visual modeling overview.** Visual modeling aligns RGB-D poses to the floorplan, forming a navigable graph with immersive node views and smooth visual transitions. A view-dependent renderer and a view-synthesis network generate stereoscopic images for nodes and 360° videos for arcs, enabling real-time WebXR visualization.

Our method builds an explorable model by combining neural networks in an end-to-end pipeline for geometric (Fig. 2) and visual (Fig. 3) modeling. At runtime, it requires only omnidirectional images (at least one per room) and their relative poses. Training uses paired RGB+depth panoramas with pixel-accurate depth that includes clutter as ground truth, as provided by standard indoor datasets like Structured3D [80].

The geometric step (Sec. 4) starts with depth prediction by a *depth estimation network* (Sec. 4.1), used for both view synthesis and structural reconstruction via a unified density map (Sec. 4.2). A *denoising network* with a gated auto-encoder refines this map, which a *floorplan segmentation network* then converts into room layouts, openings, and clutter (Sec. 4.3). The model is scaled to

world dimensions using data gathered in density map computation (see *world-scaled floorplan* in Fig. 3).

In the visual modeling step, original images and predicted depth maps compose RGB-D poses that are aligned to the world-scaled floorplan to create a navigable graph (Fig. 3), with nodes as stereoscopic omnidirectional views and arcs as transitions. Imagery is generated using a *view-dependent renderer* and a *view-synthesis network*. For each node, we generate an MCOP omnidirectional stereoscopic image (Sec. 5.2), and for each arc we exploit the inferred floorplan to determine optimal trajectories, considering walls, openings, clutter, and a renderer-estimated visual disocclusion cost (Sec. 5.1). Selected trajectories are used to synthesize interactive 360° videos. The final navigable graph is WebXR-ready for real-time immersive viewing on all devices using fully precomputed data, with the floorplan also serving for a variety of purposes, including the display of a navigable map for VR/AR/XR that also shows where the user is currently located inside the building.

4 Geometric Modeling

Geometric modeling of the multi-room environment is performed through a sequential approach that progressively incorporates broader global information. First, we determine the shape of the visible portion of each room by associating a depth with each pixel of the captured 360° images (Sec. 4.1). Then, exploiting the original co-registration, we project the depths onto a common floor plan to create a density map of the entire multi-room environment. This comprehensive, but partial and noisy, model is then denoised and enhanced in terms of accuracy and semantic richness through a novel latent diffusion-based approach (Sec. 4.2). Finally, the global density map is segmented into a floor plan made of rooms bounded by walls, floor, and ceiling, and connected through openings, and a clutter map defining the walkable space (Sec. 4.3).

4.1 Depth maps prediction

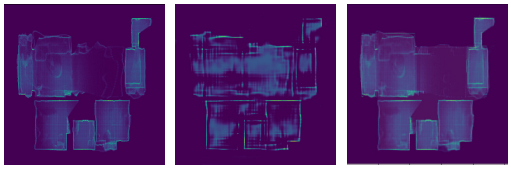
Monocular depth reconstruction from each 360° image is the first step in our pipeline and serves multiple purposes. First, the fusion of the inferred scene depth around the observer provides a first geometric approximation of the environment, which is used as a starting point for 3D floorplan generation. Second, depth is exploited for occlusion-aware reprojection and disocclusion handling in the construction of the visual model, both for supporting stereo and viewpoint translation and for creating transition paths. Many methods exist for estimating depth from equirectangular indoor images. We build upon the depth estimation branch of a state-of-the-art architecture, named ADM [41], originally designed for depth and layout estimation in view synthesis. The depth estimation network is designed around gravity-aligned features (GAFs), built with asymmetric convolutions, to take into account the fact that world-space vertical and horizontal features have different characteristics in man-made environments (see original work for details [41]). We further build on this concept by including an additional loss term that compares predicted and ground truth density maps obtained by counting the occurrences of the visible 3D points on the floor plan. The loss function $\mathscr{L}_d$ of the predicted depth compared to ground truth (see Sec. 7.1 for details) thus becomes:

$$\mathscr{L}_d = \mathscr{L}_e - 0.5\mathscr{L}_{ss} + \mathscr{L}_{om}. \tag{1}$$

where $\mathscr{L}_e$ and $\mathscr{L}_{om}$ are respectively, the *Adaptive Reverse Huber Loss* [26] for the predicted depth and for the density map and $\mathscr{L}_{ss}$ is the Structural Similarity Index Measure (SSIM). Explicitly including the density map term, as in DDD++ [40], allows us to drive the network to produce depths that generate good quality projections on the floor, which is important for our floor plan estimation steps. As a result, the adopted approach achieves both quantitative and computational state-of-the-art performances, as reported in Sec. 7.1 and Sec. 7.2.

4.2 Multi-room density map reconstruction

To fully take advantage of deep learning methods in floorplan reconstruction, it is a common practice to convert 3D point clouds into an intermediate 2D representation, often named density map [7, 8]. Such representation is typically a fixed-size map (e.g., 256×256 in common implementations and benchmarks), where the original point cloud is normalized and scaled to fit the image size. In our work, we first convert each depth map into a local point cloud through a spherical to Cartesian transformation, then we project all of them into a floorplan density map [8] using the co-registration information. However, such a predicted density map is noisy and



(a) Density map (b) $\mathscr{D}(E(\delta))$ (c) Denoised map

Figure 4: **Density denoising.** Comparison between predicted (Fig. 4c) and denoised (Fig. 4c) density map (images enhanced for illustration purposes). Although the two maps may look the same, the denoising process recovers some of the missing structural information in the latent space as noise $E(\delta)$. To better show, such latent difference has been decoded with the autoencoder in Fig. 4b.

incomplete, and must be processed to obtain a full, plausible representation of the entire multi-room environment. Our solution is to adopt a novel latent diffusion-based approach to enhance them.

Working in latent space requires the definition of an appropriate auto-encoder [46]. While very effective foundational variational autoencoders exist [45], they are designed and trained on standard imagery for generative tasks. Density maps, which count occurrences of points at every pixel, have, however, different characteristics. Thus, we designed a lightweight auto-encoder targeting only this data type. The selected lightweight architecture, dubbed *GAE*, is based on gated [74] and dilated [73] convolutions. While similar schemes have proven effective for image-to-image view synthesis, we introduce some differentiating features, see Fig. 5b.

In our network, gating functions as a *self-attention weight mask*, unlike in inpainting, where the mask is provided to specify missing pixels. Here, the network is designed to encode a latent representation z and decode it into a refined density map.

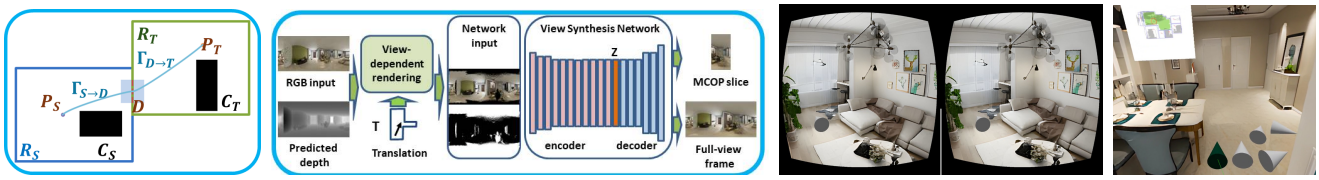
The encoder $\mathscr{E}$ includes 6 LWGC (light-weight gated-convolution [74]) layers, followed by 6 repeated dilations (as view-synth dilations in Sec. 5.2), thus increasing the area that each layer can use as input. As a decoder $\mathscr{D}$, 4 LWGC layers restore the original density map resolution, followed by a *Sigmoid* activation function, instead of *tanh* adopted for view-synthesis:

$$z_{\text{gt}} = \mathscr{E}(d_{\text{gt}}) \qquad \hat{d}_{\text{gt}} \quad = \mathscr{D}(z_{\text{gt}}) \tag{2}$$

Defining a ground truth density map as $d_{\text{gt}} \in \mathbb{R}^{1\times 256\times 256}$, ideally, $\hat{d}\text{gt} \approx d\text{gt}$. In our experiments, z_{gt} is $1 \times 32 \times 64 \times 64$, starting with 32 latent channels in the first convolutional layer (Sec. 5.2). The GAE autoencoder is trained on the ground truth density maps provided by Structured3D[80] with the following loss function:

$$\mathscr{L}_{\text{GAE}} = \mathscr{L}_{\text{d}}(\mathscr{D}(\mathscr{E}(d\text{gt})), d_{\text{gt}}) + \lambda \cdot \mathscr{L}_{\text{ss}}(\mathscr{D}(\mathscr{E}(d\text{gt})), d_{\text{gt}})) \tag{3}$$

where $\lambda = 0.05$ and, similarly to Eq. (1), $\mathscr{L}_d$ is the *Adaptive Reverse Huber Loss* [26] and $\mathscr{L}_{ss}$ is the Structural Similarity Index Measure (SSIM). As a result, the GAE network has much less computational and training complexity than foundation autoencoders [46], as shown in Tab. 3.



(a) Path computation (b) View-synthesis baseline (c) Cardboard interface (d) Oculus interface

Figure 5: **Visual modeling main tasks.** Visual modeling finalizes the geometric model by constructing a navigable graph with immersive visual content for nodes and transitions. Fig. 5a shows the estimation of optimal navigation trajectories between panoramic viewpoints. This process accounts for structural constraints, including openings and obstacles, while minimizing visual discontinuities and occlusions during transitions. Fig. 5b shows the baseline [41] adopted for view-synthesis, which, in our use, can output spherical slices to compose MCOP or full-view spherical images for transitions. Fig. 5c and Fig. 5d show HMD user interface.

Once the latent space is defined, we model the noise in the input as the difference $\delta = d_{\text{in}} - d_{\text{gt}}$, where d_{in} is the input, noisy density map, which implies that the noisy latent representation can be approximated as:

$$z_{\text{in}} = z_{\text{gt}} + E(\delta) \tag{4}$$

In other words, predicting the latent $z_d = E(\delta)$ means predicting the missing information in z_{in} to be more similar to z_{gt}. According to this, the forward diffusion process corrupts z_{gt} by adding Gaussian noise:

$$z_t = \sqrt{\bar{\alpha}_t}\, z_0 + \sqrt{1-\bar{\alpha}_t}\, \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0,I) \tag{5}$$

where t is the diffusion timestep and $\bar{\alpha}_t$ are predefined schedule coefficients. The latent diffusion model, parameterized by a UNet $\varepsilon_\theta(z_t,t)$, is trained to predict such a noise component. Inference, given z_{in}, applies denoising by inverting Eq. (5):

$$\hat{z}_d = \frac{z_t - \sqrt{1-\bar{\alpha}_t}\, \varepsilon_\theta(z_t,t)}{\sqrt{\bar{\alpha}_t}} \tag{6}$$

to estimate $\hat{z_d}$ from z_{in} as input. Finally, we reconstruct the denoised density map as the decoding $\hat{d}_d = \mathscr{D}(z_{in} - \hat{z}_d)$.

Assuming a frozen GAE autoencoder, we train the latent diffusion with the loss:

$$\mathscr{L}_{\text{LDM}} = |\hat{z_d} - z_d|_1 + 0.1|\mathscr{D}(z_{in} - \hat{z_d}) - \mathscr{D}(z_{\text{gt}})|_1 \tag{7}$$

As an example, Fig. 4 shows a comparison between the predicted (Fig. 4a) and the denoised (Fig. 4c) density map. While the two maps appear visually similar (with enhancements applied for better illustration), the denoising process restores missing structural details in the latent space as noise $E(\delta)$. To better illustrate this latent difference, Fig. 4b shows its decoding in the density map space.

4.3 Floorplan segmentation

Taking the denoised multi-room density map as input, we formulate floorplan segmentation as the prediction of a polygon set, where each polygon can represent a *room*, defined by an ordered sequence of 2D vertices, or an *opening*, encoded by a degenerate polygon with only two vertices. Rooms and openings are predicted as a set of queries, $Q \in \mathbb{R}^{M\times N\times 2}$, using a deformable transformer [81]. We define M as the maximum number of polygons (rooms and openings) and N as the maximum number of vertices per polygon ($M = 50, N = 40$ in our experiments). Each vertex is associated with a binary value indicating its validity. The *segmentation network* (Fig. 2) carries out the prediction by integrating multi-scale feature extraction, flattening, deformable self-attention (DSA) for transformer encoding, and a combination of self-attention and deformable cross-attention (SA-DSA) for decoding.

The first layers extract multi-scale features (four levels) from the density map using a *ResNet* scheme [15], flatten them to a feature sequence to serve as input to a six-layer transformer encoder. Each encoder layer consists of a multi-scale deformable *self-attention* module and a simple feed-forward network [81] (DSA in Fig. 2). Similarly to encoding, decoding is performed by six stacked layers; however, unlike the encoder, each layer consists of a self-attention module, a multi-scale deformable cross-attention module, and a feed-forward network (SA-DSA in Fig. 2). The decoder output is passed to a shared feed-forward network to predict binary class labels c for each query, indicating its validity as a corner. Finally, the decoder outputs a $M \times N \times 2$ polygons map and a $M \times N \times 1$ classification map, from which a set of valid 2D polygons is extracted [75]. While the ground truth contains an arbitrary number of polygons with an arbitrary number of vertices, the transformer decoder always outputs M polygons of N vertices, setting the invalid ones to 0. We train the network with the Structured3D [80] dataset, which provided fully annotated floorplans including geometries and semantic elements (see Sec. 7.1 for details), using the following recently introduced losses [75, 6]:

$$\mathscr{L}_{floor} = \sum_{m}^{M} (2\mathscr{L}^m_{cls} + 5\mathscr{L}^m_{coord} + \mathscr{L}^m_{ras}) \tag{8}$$

where $\mathscr{L}_{cls}$ is a standard binary cross-entropy (i.e., class loss), $\mathscr{L}_{coord}$ (i.e., coordinates loss) is the L_1 distance, $\mathscr{L}_{ras}$ (i.e., raster loss - only for rooms) is the Dice loss [34].

Quantitative and qualitative results show how our solution predicts rooms, doors, and windows with state-of-the-art accuracy while starting from purely visual input (Sec. 7.2, Sec. 7.3), ready to be exploited and integrated in the immersive visual model Sec. 5.

For our application, in addition, we complete the model with a clutter map, defining the walkable space in the environment used to generate our transitions. To this end, we calculate a partial density map from the fraction of point clouds below the observer's horizon, therefore removing ceilings and hangings, and consider as clutter the points on the density maps that have occurrences above a certain threshold (10% in our experiments), to remove from clutter the walkable floor (which counts as at least one occurrence) and eventual projection noise.

5 Visual Modeling

Visual modeling completes the geometric model by creating a navigable graph (Sec. 5.1), including the immersive visual content for nodes and transitions (Sec. 5.2). The final output is ready for interactive viewing with a WebXR-compatible device (Sec. 6).

5.1 Door-based graph topology

From our floorplan segmentation (Sec. 4.3), each door is identified by a small two-vertex segment. We first verify that the door connects exactly two distinct rooms. If so, we associate the door with the pair of *nodes* (i.e., cameras) that occupy those rooms. This yields a high-level graph structure: each door triggers a potential connection (arc) between two nodes. This representation is

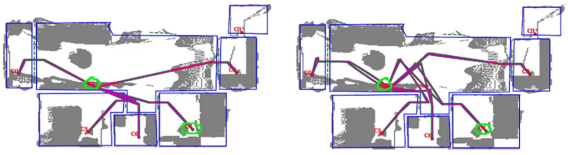


Figure 6: **Path optimization.** At initialization, paths connect source to target poses, passing through doors (left). After optimization, paths avoid walls and clutter and minimize disocclusions (right). The green mark identifies a very difficult case, in which the detected door is invisible in one of the views.

already sufficient to automatically support multi-room navigation through exploration at the original location and teleportation to connected rooms. To improve location and movement awareness, we refine each arc into a continuous path composed of two segments: (1) *Source-to-Door*: a parametric curve from the source viewpoint to the door location; (2) *Door-to-Target*: a parametric curve from the door location to the target viewpoint. Both sub-paths must be stitched into a single smooth trajectory at the door location. To ensure continuous derivatives (i.e., no abrupt orientation changes), we model each sub-path with a *Hermite spline* [59] (see Fig. 5, left).

Constructing a single multi-segment path from the source camera to the target camera via the door involves optimizing several control points and tangents. We use *dual simulated annealing* (DSA) to minimize a custom cost function [60, 66]. DSA randomly perturbs the free control points and tangent vectors over multiple iterations, gradually "cooling" to a stable solution that locally minimizes our *cost function*. Let the full path Γ be a concatenation of Hermite segments Γ_k, each defined by control points and tangents $(\mathbf{P}_k, \mathbf{P}_{k+1}, \mathbf{T}_k, \mathbf{T}_{k+1})$. The cost function combines four penalty components: *Path Length* C_ℓ favors shorter routes, reducing travel time and video frame count, computed using closed-form Hermite spline integration; *Disocclusion Penalty* C_d accounts for novel-view artifacts by sampling geometry along Γ and counting newly visible pixels in each frame; *Clutter Penalty* C_c discourages paths intersecting high-clutter areas, counting how many sampled path points intersect high-density clutter regions from the precomputed clutter map; *Wall Penalty* C_w ensures that Γ does not cross walls and remains within a safe distance from them, counting how many path points fall within a threshold distance from wall boundary polygons. The complete cost is:

$$C(\Gamma) = \omega_\ell C_\ell(\Gamma) + \omega_d C_d(\Gamma) + \omega_c C_c(\Gamma) + \omega_w C_w(\Gamma), \quad (9)$$

with weights $\omega_\ell = 0.01$, $\omega_d = 1.0$, $\omega_c = 100.0$, and $\omega_w = 100.0$. The weights were empirically determined with a few trials and kept fixed for all the experiments.

Once the optimized sub-path $\Gamma_{\text{src}\to\text{door}}$ is found, we repeat the procedure for $\Gamma_{\text{door}\to\text{tgt}}$. Merging them produces the complete door-based arc. For each segment, we sample a discrete set of points in parametric space (e.g., 10–100 points, depending on distance and resolution needs). These sampled points and their corresponding camera orientations form the keyframes used by our view-synthesis network to generate intermediate 360◦ frames.

By minimizing the *Disocclusion Penalty*, we ensure that the path remains in areas that are well sampled from the original viewpoint, reducing the need to generate novel data, thereby producing paths that give the impression of motion towards the door when leaving a room and away from the door when entering a new one. Good reconstruction in the transition area is only possible when a reasonably large mutually visible area exists, which may not be the case if one of the views does not see the door (e.g., it is around a corner - see Fig. 6, green highlight). If such a case is detected, we trim the portions of transitions closest to the occluded door to minimize visual artifacts and preserve the viewing experience.

5.2 Immersive view synthesis

To generate immersive visual content, it is necessary to synthesize novel panoramic images translated relative to the input poses. In our approach, we leverage a scheme already proven to be effective in VR [41], which integrates a *view-dependent renderer* and a *view synthesis* network. This architecture, illustrated in Fig. 5b, is very lightweight and suitable for an efficient synthesis of the many images required for visit precomputation. Compared to the inpainting baseline (see [41]), we introduce the minimization of the inpainting area through our path optimization method (Sec. 5.1), and incorporate structural constraints directly into the depth (Sec. 4.1). The view-dependent renderer exploits the predicted depth to convert source pixels to 3D points and translates and reprojects them to their new location in the target equirectangular image. The view synthesis network inpaints the missing disoccluded pixels.

At each node, we adopt a stereoscopic MCOP model [42] that uses these networks to generate panoramic slices, placing each slice along a circular path matching eye movement during head rotation. Each slice covers enough angular range for both eyes and contextual reconstruction (Fig. 5b). The final stereoscopic MCOP pair is efficiently composed by blending these precomputed slices in equirectangular format. For a typical human-sized configuration, assuming a head radius of 100mm and an interpupillary distance (IPD) of 65mm, we define a 45° portion of the image as sufficient to cover both eyes. To provide additional context for reconstructing missing areas and supporting eye convergence at finite distances, we expand this region to approximately 56°.

At each arc, instead, we render full-view, monoscopic equirectangular frames, following the optimal trajectories computed in Sec. 5.1. In our tests, we have experienced that monoscopic video is sufficient to ensure immersive exploration in motion, while also limiting the sense of discomfort from moving stereo or other artifacts. As a final step, we also perform upsampling [63] of all generated images to match the resolution of the display. This is because, currently, our synthesis is performed at a resolution smaller than the display size (i.e., a vertical slice resolution of 512 pixels vs 2048 for a typical headset). This limitation is not due to our scalable network architectures, but rather, to limitations in available ground-truth training sets (see Sec. 7.1 for details).

6 Interactive VR Exploration

As a proof-of-concept, we develop a virtual tour prototype built with WebXR and a minimal interface using *Three.js*. At runtime, the application imports the previously constructed graph data structure, wherein each *node* is an MCOP-stereo panoramic view of a given room, and each *arc* is a transition path, represented by a pre-rendered 360° video, which corresponds to the optimized trajectory computed in Sec. 5.1. Once a user puts on a VR headset (e.g., *Meta Quest* or *Cardboard*), they are placed in a node's stereoscopic panorama. The interface displays arrow-shaped placeholders indicating all possible door connections to adjacent rooms. Clicking or gazing at one of these placeholders triggers the 360° transition video associated with that arc, smoothly transporting the user to the corresponding target node. This design mimics intuitive room-to-room travel while avoiding abrupt scene jumps, since the underlying arc videos handle the in-between viewpoints and occlusions in a visually coherent manner. To satisfy common constraints on VR headsets, each transition video is encoded at a resolution of *5760×2880 pixels* with a frame rate of *30 fps*, meeting hardware performance requirements without compromising immersive quality. The system thus achieves fluid scene traversal across multiple rooms, letting users freely inspect each room's panoramic stereo view, then proceed to any connected room by activating the corresponding door arc. In practice, the entire experience runs within a standard WebXR-capable browser, making it readily deployable to standalone headsets or desktop devices with VR support.

7 RESULTS

Method	Params↓	FLOPs↓	Inf. time↓	MAE↓	RMSE↓	δ_1 ↑
Panoformer [49]	20 M	78 G	17 ms	0.254	0.793	0.747
EGFormer [76]	**15 M**	74 G	16 ms	0.220	0.684	0.797
Elite360D [1]	25 M	65 G	14 ms	0.148	0.496	0.874
ADM [41]	29 M	79 G	18 ms	0.080	0.124	0.968
Our depth e.	23 M	**38 G**	**7 ms**	**0.045**	**0.135**	**0.978**

Table 1: **Performance of depth inference.** We show our performance compared to other state-of-the-art works for a 512×1024 image. All methods were trained and tested using the same conditions. Other results are the same as reported in recent publications [1], while ADM [41] has been retrained by us.

Our approach was implemented using *PyTorch* [38] and WebXR and has been tested on a large variety of indoor scenes. The accompanying video shows its usage for exploration with HMDs.

7.1 Setup and computational performance

For training and benchmarking our solutions, we mainly exploit Structured3D [80]), a large-scale, synthetic database of fully annotated 3D floorplans with registered panoramic images. That dataset, in addition to being the larger dataset providing full annotations (including depthmaps and full semantic), provides the most accurate ground truth possible, which is essential, for example, for creating geometric-consistent density maps. We used Structured3D for training and testing the depth estimation network (Sec. 4.1), the autoencoder and diffusion networks (Sec. 4.2) and the floorplan segmentation network (Sec. 4.3). In all cases we follow the official splittings [80]. To train and test the view-synthesis network, instead, we exploit PNVS[69], which is a visual extension of Structured3D scenes providing, for each original panoramic image, three views translated by 0.2-0.3m along random directions, and three views translated by 1.0-2.0m. All models have been trained at the native Structured3D resolution, that is 1024×512, which have been upsampled to 4096×2048 for the final visual output, using *Real-ESRGAN* (with model *realesr-animevideov3*) [63] It is noteworthy that although the model is trained on synthetic data, it works successfully for reconstructing real-world multi-room scenes, as shown in Fig. 10. Here we exploit as benchmark ZinD [9], one of the largest real-world indoor datasets with 3D multi-room layout annotations, containing 71,474 panoramas from 1,524 real homes.

We train all the networks with the Adam optimizer [24], with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and an adaptive learning rate from 0.0001, on a NVIDIA RTX 4090 (24GB VRAM) with a batch size of 8. The average training time on the same machine is 10 $ms/image$ for the depth network with a batch size of 8 and 300 epochs, 22 $ms/image$ for the view-synthesis network with a batch size of 8 and 300 epochs, 402 $ms/image$ for the floorplan network with a batch size of 10 and 500 epochs, 4 ms/map for the gated autoencoder with a batch size of 32 and 1000 epochs, 140 ms/map for the diffusion network with a batch size of 64, 1000 epochs of 1000 steps. Both the autoencoder and the diffusion model are extremely lighter than common foundation models [46]. Below we collect a comparison based on our experiments in a Tab. 3. Here *VQ-VAE* is a standard implementation of a variational autoencoder without hierarchical latent [45] and *Fast latent diffusion* is a lightweight implementation of standard latent diffusion [65]. Tab. 1 summarizes computational complexity for depth inference, compared to other state-of-the-art works, on a NVIDIA RTX 4090 with 24Gb VRAM. Under the same setting, inferring a floorplan from a $1 \times 256 \times 256$ density map takes 4ms, and a 100-step denoising takes 142ms. Starting from the geometric model, visual modeling generation time depends on the number of stereoscopic poses and transition frames. The inference time for a full-view equirectangular frame of 1024×512 is 10ms, with 18ms additional time for super-resolution upsampling.

7.2 Quantitative performance analysis of geometric and visual modeling

Following the pipeline order, we first show results about depth estimation and view-synthesis, followed by performances specific to our major technical contributions related to the reconstruction of the entire floorplan for navigation.

Tab. 1 shows our performance in terms of depth estimation on standard metrics (mean absolute error (MAE), root mean square error of linear measures (RMSE), and relative accuracy δ_1 [1]), compared to the latest state-of-the-art approaches. Although our solution for depth estimation is an enhancement of ADM [41], it outperforms all the competitors in both accuracy and computational performance (Tab. 1) for the restricted task focused on structured data and scenes.

In addition, the higher efficiency in depth estimation also translates into better performance in view synthesis. As an example, we show in Tab. 4 the performance of our view-synthesis network, compared to the baseline GVS [41] and a baseline for the synthesis of MCOP slices [42], on PNVS [69] benchmark.

Tab. 2 summarizes a comprehensive quantitative evaluation of our approach on the Structured3D [80] dataset, focusing on the effectiveness of our denoised density maps in reconstructing indoor environments. To ensure a fair comparison, we generate competitor density maps using an alternative state-of-the-art (SoA) depth prediction method, ADM [41], which serves as a robust baseline for depth-based density estimation (see Tab. 1). We then compare our approach against two prominent approaches. The first is ADM combined with PolyDiffuse [6], which currently achieves the best segmentation performance for polygonal room layouts by leveraging diffusion-based polygonal generation. Since PolyDiffuse (as ours) builds upon the RoomFormer [75] architecture, an advanced transformer-based model for full floorplan segmentation (i.e., including room type and openings), we also compare our solution to the combination of ADM and RoomFormer.

Furthermore, to assess the contribution of our denoising step, we conducted an ablation study by comparing our full pipeline to a variant that directly utilizes raw density maps without denoising (Our plain density). This experiment helps isolate the impact of our denoising strategy on downstream tasks such as room segmentation and floorplan reconstruction. For each ground truth room, we find the best-matching reconstructed room among all predictions in terms of Precision, Recall, and F1 scores about Room (i.e., found or not), Corner (i.e, room corners), Angle (i.e., room angles) and openings (i.e, found or not). In all metrics, our proposed pipeline outperforms the other solutions.

7.3 Qualitative performance analysis of geometric and visual modeling

Fig. 7 shows several examples of model reconstruction from the Structured3D [80] dataset, for which a comparable ground truth is available. For each scene, we show the input denoised density map, our predicted model including openings, the ground truth model, and the predicted multi-room scene graph in world coordinates, including clutter map and computed trajectories (i.e., last column). Besides the reconstructed scene graph, we show representative equirectangular screenshots at the input poses (i.e., node pose) and synthesized screenshots along the computed trajectory. The third column shows, as comparison, the predicted model using ADM [41]+RoomFormer [75] (see Tab. 2), which provides an output comparable with ours. It should be noted that our approach can also distinguish between doors and windows, according to the relative queries (Sec. 4.3).

Method	Room			Corner			Angle			Openings		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
ADM [41]+PolyDiffuse [6]	0.880	0.851	0.871	0.697	0.660	0.675	0.662	0.629	0.612	-	-	-
ADM [41]+RoomFormer [75]	0.863	0.845	0.864	0.679	0.662	0.652	0.601	0.589	0.578	0.556	0.588	0.571
Our plain density	0.882	0.864	0.873	0.686	0.677	0.681	0.608	0.601	0.605	0.567	0.606	0.582
Our denoised density	**0.943**	**0.918**	**0.930**	**0.787**	**0.760**	**0.782**	**0.705**	**0.682**	**0.694**	**0.635**	**0.672**	**0.653**

Table 2: **Floorplan performance and ablation.** We show our quantitative performance on Structured3D (S3D) [80] (Our density denoised). For a fair comparison from predicted images, we adopt an alternative SoA depth prediction method to generate competitors density maps (ADM [41]). We compare then with ADM [41]+Polydiffuse [6], which is currently the method that performs best for segmentation of polygonal rooms, and RooomFormer [75], which is the same baseline of PolyDiffuse but providing full floorplan segmentation (ADM [41]+RoomFormer [75]). Finally, as ablation experiment, we compare to our pipeline but without denoising (Our plain density).

(a) Input density map (b) Our predicted (c) Ground truth (d) Oth predicted [75] (e) Our Predicted graph

Figure 7: **Examples of model reconstructions from Structured3D [80]**. For each scene, we show the denoised density map, our predicted model from denoised with openings, the ground truth, the prediction from original noisy density map, and the multi-room scene graph in world coordinates (several rotated for better fitting in the illustration), including clutter maps, computed trajectories and screenshots at primary and transition poses. Notably, our method distinguishes between doors and windows based on relative queries (Sec. 4.3).

	Simple VQ-VAE	GAE (our)	F. latent diffusion	Our denoise
Params↓	20 M	**0.11 M**	100 M	**7.6 M**
FLOPs↓	20 G	**0.68 G**	50 G	**31 G**

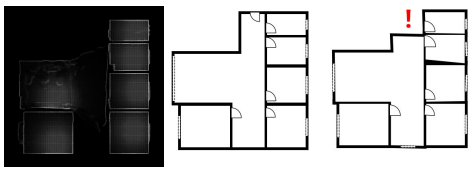
Table 3: **AE and diffusion computational performance.** We show our computational performance compared to a standard solution for the same density map size of 256×256.

Method	PSNR↑	SSIM↑	LPIPS↓
MCOP [42]	18.22	0.789	0.252
GVS [41]	22.97	0.817	0.178
Our	**23.02**	**0.824**	**0.187**

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

Additionally, Fig. 10 illustrates the capabilities of our approach on real-world captured scenes, leveraging ZinD [9] to obtain multi-room spherical captures. These experiments highlight our effectiveness in generalizing from synthetic to real-world scenarios, demonstrating the capability of domain transfer. Additionally, our results showcase the potential for zero-shot reconstruction, where the model successfully reconstructs previously unseen real-world environments without requiring additional fine-tuning or retraining on real data. We leverage geometric information to improve view synthesis and navigation (Sec. 5), computing optimal paths that account for both geometric constraints—like passing through doorways and avoiding obstacles—and visual quality by minimizing disocclusions. We demonstrate this in Fig. 8: the top row of each sequence illustrates transitions guided by our reconstructed model and optimized trajectories, while the bottom shows transitions from direct pose interpolation. Differences are especially evident in the final frames, even when the scene appears largely unchanged.

Figure 8: **Transition comparison.** Top: frames extracted from a transition obtained with our path optimizer and respecting disocclusion and clutter constraints. Bottom: frames extracted from a transition computed without path optimization.



(a) Density map (b) Ground truth (c) Prediction

Figure 9: **Failure example.** We show an example where the accuracy of the reconstruction affects the final result. Although the result appears to be a good reconstruction at first glance, with all doors present, the incorrect positioning of the wall at the top prevents access to the upper right room.

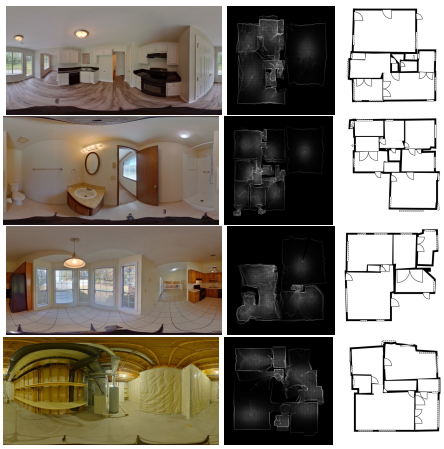
7.4 Failure cases

The process of reconstructing numerous structural elements, such as walls, doors, and windows, from a predicted density map presents many difficulties and possibilities for error. In many cases, (see Fig. 7 and Fig. 10), errors about the shape of rooms or the exact location of an opening do not prevent the construction of an immersive graph, bearing in mind precisely that the purpose of the reconstruction is to support exploration, and not to produce a detailed reconstruction. However, there are cases where the accuracy of reconstruction affects the final result, as in the case we selected in Fig. 9. Here, although the result may seem a good reconstruction and even all the doors are present, the wrong positioning of the wall at the top makes it impossible to reach the upper right room.

8 Conclusions

We presented a method for immersive modeling of complex indoor environments using a minimal set of registered 360° panoramas. Unlike prior single-room or post-constrained approaches, we reconstruct multi-room scenes by refining predicted depth with diffusion denoising and segmenting structure with transformers. By unifying geometry and semantics in a global density map, we recover a floor plan that documents the building and support path planning and immersive navigation in cluttered multi-room spaces. Since our final scene graph contains fully precomputed MCOP panoramic images and panoramic transition videos, scenes can be experienced immersively on commodity HMDs.

The discussed analyses and experiments show that the method achieves state-of-the-art reconstruction from sparse inputs and supports compelling immersive visits. Quantitative measures in multi-room reconstruction, in particular, demonstrate our ability to improve over competing solutions in geometric and structural reconstruction. From the user's point of view, we currently have only informal feedback on immersion and path planning, and a full user study is an important aspect of future work. In particular, we would like to formally evaluate user satisfaction, improvement in location awareness compared to just teleportation, and quantify any effects of cyber sickness that may arise from reconstruction artifacts.



(a) Image sample (b) D. map (c) Prediction

Figure 10: **Examples of reconstructions on real-world captures**, using our models trained on synthetic data [80] for prediction. Such experiments demonstrate the capabilities of domain transfer and zero-shot reconstruction.

The primary visual limitations stem from rendering parts that are very distant from the original viewpoint or, even more critically, those that are completely invisible from the original input views. In this regard, one direction for future work is to exploit conditional generative models of latent diffusion instead of deep-learning-based inpainting. We expect that this would allow synthesizing plausible, completely novel intermediate views and smooth motions through continuous trajectories. This would further close the gap between captured and fully navigable virtual environments.

Acknowledgments

This publication was supported by NPRP-S 14th Cycle grant 0403-210132 AIN2 from the Qatar National Research Fund (a member of Qatar Foundation). GP and EG also acknowledge the contribution of the Italian National Research Center in High-Performance Computing, Big Data, and Quantum Computing (Next Generation EU PNRR M4C2 Inv 1.4). The findings herein reflect the work and are solely the responsibility of the authors.

References

[1] H. Ai and L. Wang. Elite360d: Towards efficient 360 depth estimation via semantic-and distance-aware bi-projection fusion. In *Proc. CVPR*,

pp. 9926–9935, 2024. 7

[2] H. AlZayer, H. Lin, and K. Bala. Autophoto: Aesthetic photo capture using reinforcement learning. In *Proc. IROS*, pp. 944–951, 2021. 3

[3] P. Boguslawski, S. Zlatanova, D. Gotlib, M. Wyszomirski, M. Gnat, and P. Grzempowski. 3D building interior modelling for navigation in emergency response applications. *nt. J. Appl. Earth Obs. Geoinf*, 114:103066, 2022. 1

[4] S. Boorboor, Y. Kim, P. Hu, J. M. Moses, B. A. Colle, and A. E. Kaufman. Submerse: Visualizing storm surge flooding simulations in immersive display ecologies. *IEEE TVCG*, 30(9):6365–6377, 2023. 3

[5] R. Cabral and Y. Furukawa. Piecewise planar and compact floorplan reconstruction from images. In *Proc. CVPR*, pp. 628–635, 2014. 2

[6] J. Chen, R. Deng, and Y. Furukawa. Polydiffuse: Polygonal shape reconstruction via guided set diffusion models. *NeurIPS*, 36, 2024. 2, 5, 7, 8

[7] J. Chen, C. Liu, J. Wu, and Y. Furukawa. Floor-SP: Inverse cad for floorplans by sequential room-wise shortest path. In *Proc. CVPR*, pp. 2661–2670, 2019. 2, 4

[8] J. Chen, Y. Qian, and Y. Furukawa. HEAT: Holistic edge attention transformer for structured reconstruction. In *Proc. CVPR*, pp. 3866–3875, 2022. 2, 4

[9] S. Cruz, W. Hutchcroft, Y. Li, N. Khosravan, I. Boyadzhiev, and S. B. Kang. Zillow indoor dataset: Annotated floor plans with 360° panoramas and 3D room layouts. In *Proc. CVPR*, pp. 2133–2143, 2021. 1, 2, 7, 8

[10] M. Di Benedetto, F. Ganovelli, M. Balsa Rodriguez, A. Jaspe Villanueva, R. Scopigno, and E. Gobbetti. ExploreMaps: Efficient construction and ubiquitous exploration of panoramic view graphs of complex 3D environments. *Comput. Graph. Forum*, 33(2):459–468, 2014. 3

[11] M. Feixas, M. Sbert, and F. González. A unified information-theoretic framework for viewpoint selection and mesh saliency. *ACM Trans. Appl. Percept.*, 6(1):1:1–1:23, 2009. 3

[12] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Reconstructing building interiors from images. In *Proc. ICCV*, pp. 80–87, 2009. 2

[13] E. Gobbetti, G. Pintore, and M. Agus. Automatic 3D modeling and exploration of indoor structures from panoramic imagery. In *SIGGRAPH Asia Courses*, pp. 1:1–1:9, 2024. 1, 2

[14] A. Gueze, M. Ospici, D. Rohmer, and M.-P. Cani. Floor plan reconstruction from sparse views: Combining graph neural network with constrained diffusion. In *Proc. CVPR*, pp. 1583–1592, 2023. 2

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. CVPR*, pp. 770–778, 2016. 5

[16] P. Hedman and J. Kopf. Instant 3D photography. *ACM TOG*, 37(4):1–12, 2018. 3

[17] 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. 3

[18] W. Hutchcroft, Y. Li, I. Boyadzhiev, Z. Wan, H. Wang, and S. B. Kang. CoVisPose: Co-visibility pose transformer for wide-baseline relative pose estimation in 360° indoor panoramas. In *Proc. ECCV*, pp. 615–633, 2022. 1, 2

[19] S. Ikehata, H. Yang, and Y. Furukawa. Structured indoor modeling. In *Proc. ICCV*, pp. 1323–1331, 2015. 2

[20] A. Jain, M. Tancik, and P. Abbeel. Putting NeRF on a diet: Semantically consistent few-shot view synthesis. In *Proc. ICCV*, pp. 5885–5894, 2021. 3

[21] 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. 2

[22] T. Jokela, J. Ojala, and K. Väänänen. How people use 360-degree cameras. In *Proc. MUM*, pp. 18–27, 2019. 1

[23] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM TOG*, 42(4):139:1–139:14, 2023. 3

[24] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *ArXiv e-print arXiv:1412.6980*, 2014. 7

[25] J. Lambert, Y. Li, I. Boyadzhiev, L. Wixson, M. Narayana, W. Hutchcroft, J. Hays, F. Dellaert, and S. B. Kang. SALVe: Semantic alignment verification for floorplan reconstruction from sparse panoramas. In *Proc. ECCV*, pp. 647–664, 2022. 2

[26] S. Lambert-Lacroix and L. Zwald. The adaptive BerHu penalty in robust regression. *J. Nonparametric Stat.*, 28:1–28, 2016. 4

[27] D. Li, Y. Zhang, C. Häne, D. Tang, A. Varshney, and R. Du. OmniSyn: Synthesizing 360 videos with wide-baseline panoramas. In *Proc. VRW*, pp. 670–671, 2022. 3

[28] Q. Li and N. Khademi Kalantari. Synthesizing light field from a single image with variable MPI and two network fusion. *ACM TOG*, 39(6):229:1–229:10, 2020. 3

[29] C. Liu, J. Wu, and Y. Furukawa. FloorNet: A unified framework for floorplan reconstruction from 3D scans. In *Proc. ECCV*, pp. 203–219, 2018. 2

[30] B. Luo, F. Xu, C. Richardt, and J.-H. Yong. Parallax360: Stereoscopic 360 scene representation for head-motion parallax. *IEEE TVCG*, 24(4):1545–1553, 2018. Proc. IEEE VR. 3

[31] T. Marrinan and M. E. Papka. Real-time omnidirectional stereo rendering: generating 360° surround-view panoramic images for comfortable immersive viewing. *IEEE TVCG*, 27(5):2587–2596, 2021. 3

[32] 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, 3

[33] 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. *CACM*, 65(1):99–106, 2021. 3

[34] F. Milletari, N. Navab, and S.-A. Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *Proc. 3DV*, pp. 565–571, 2016. 5

[35] A. Monszpart, N. Mellado, G. J. Brostow, and N. J. Mitra. RAPter: Rebuilding man-made scenes with regular arrangements of planes. *ACM TOG*, 34(4):103:1–103:12, 2015. 2

[36] N. Nauata and Y. Furukawa. Vectorizing world buildings: Planar graph reconstruction by primitive detection and relationship inference. In *Proc. ECCV*, pp. 711–726, 2020. 2

[37] E. Olson. AprilTag: A robust and flexible visual fiducial system. In *Proc. ICRA*, pp. 3400–3407, 2011. 2

[38] 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. 7

[39] 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

[40] G. Pintore, M. Agus, A. Signoroni, and E. Gobbetti. Ddd++: Exploiting density map consistency for deep depth estimation in indoor environments. *Graphical Models*, 140:101281, August 2025. 4

[41] G. Pintore, F. Bettio, M. Agus, and E. Gobbetti. Deep scene synthesis of atlanta-world interiors from a single omnidirectional image. *IEEE TVCG*, 29, November 2023. 2, 3, 4, 5, 6, 7, 8

[42] G. Pintore, A. Jaspe-Villanueva, M. Hadwiger, J. Schneider, M. Agus, F. Marton, F. Bettio, and E. Gobbetti. Deep synthesis and exploration of omnidirectional stereoscopic environments from a single surround-view panoramic image. *Computers & Graphics*, 119:103907, March 2024. 3, 6, 7, 8

[43] G. Pintore, U. Shah, M. Agus, and E. Gobbetti. NadirFloorNet: reconstructing multi-room floorplans from a small set of registered panoramic images. In *Proc. CVPRW*, pp. 1986–1994, 2025. 2

[44] G. Pu, Y. Zhao, and Z. Lian. Pano2Room: Novel view synthesis from a single indoor panorama. In *Proc. SIGGRAPH Asia*, pp. 28:1–28:11, 2024. 3

[45] A. Razavi, A. Van den Oord, and O. Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019. 4, 7

[46] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proc. CVPR*, pp. 10684–10695, 2022. 2, 4, 7

[47] M. A. Shabani, S. Hosseini, and Y. Furukawa. HouseDiffusion: Vector floorplan generation via a diffusion model with discrete and continuous denoising. *arXiv preprint arXiv:2211.13287*, 2022. 2

[48] M. A. Shabani, W. Song, M. Odamaki, H. Fujiki, and Y. Furukawa.

Extreme structure from motion for indoor panoramas without visual overlaps. In *Proc. ICCV*, pp. 5683–5691, 2021. 2

[49] Z. Shen, C. Lin, K. Liao, L. Nie, Z. Zheng, and Y. Zhao. PanoFormer: Panorama transformer for indoor 360 depth estimation. In *Proc. ECCV*, pp. 195–211, 2022. 7

[50] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from RGBD images. In *Proc. ECCV*, 2012. 2

[51] K. Song and L. Zhang. Novel view synthesis with wide-baseline stereo pairs based on local–global information. *Computers & Graphics*, 126:104139, 2025. 3

[52] S. Stekovic, M. Rad, F. Fraundorfer, and V. Lepetit. Montefloor: Extending mcts for reconstructing accurate large-scale floor plans. In *Proc. CVPR*, pp. 16034–16043, 2021. 2

[53] J.-W. Su, K.-Y. Tung, C.-H. Peng, P. Wonka, and H.-K. Chu. SLIBO-Net: Floorplan reconstruction via slicing box representation with local geometry regularization. In *Proc. NeurIPS*, 2023. 2

[54] 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. 1

[55] 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

[56] R. Tucker and N. Snavely. Single-view view synthesis with multiplane images. In *Proc. CVPR*, pp. 548–557, 2020. 3

[57] 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, 2023. 3

[58] T. Uchida, Y. Kanamori, and Y. Endo. 3d view optimization for improving image aesthetics. In *Proc. ICASSP*, pp. 1–5, 2025. 3

[59] P. Wagner, J. Kotzian, J. Kordas, and V. Michna. Path planning and tracking for robots based on cubic hermite splines in real-time. In *Proc. IEEE ETFA*, pp. 1–8, 2010. 6

[60] B. W. Wah, Y. Chen, and T. Wang. Simulated annealing with asymptotic convergence for nonlinear constrained optimization. *Journal of Global Optimization*, 39:1–37, 2007. 6

[61] 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, 2022. 1, 3

[62] G. Wang, P. Wang, Z. Chen, W. Wang, C. C. Loy, and Z. Liu. PERF: Panoramic Neural Radiance Field from a single panorama. *IEEE TPAMI*, pp. 1–15, 2024. 3

[63] 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. 6, 7

[64] Z. Wei, J. Zhang, X. Shen, Z. Lin, R. Mech, M. Hoai, and D. Samaras. Good view hunting: Learning photo composition from dense view pairs. In *Proc. CVPR*, pp. 5437–5446, 2018. 3

[65] Z. Wu, P. Zhou, K. Kawaguchi, and H. Zhang. Fast diffusion model. *arXiv preprint arXiv:2306.06991*, 2023. 7

[66] Y. Xiang, S. Gubian, B. Suomela, and J. Hoeng. Generalized simulated annealing for global optimization: The gensa package. *The R Journal*, 5(1):13, 2013. 6

[67] D. Xie, P. Hu, X. Sun, S. Pirk, J. Zhang, R. Mech, and A. E. Kaufman. GAIT: Generating aesthetic indoor tours with deep reinforcement learning. In *Proc. ICCV*, pp. 7409–7419, 2023. 3

[68] J. Xu, B. Stenger, T. Kerola, and T. Tung. Pano2CAD: Room layout from a single panorama image. In *Proc. WACV*, pp. 354–362, 2017. 3

[69] 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. 3, 7

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

[71] Z. Yang, J. Z. Pan, L. Luo, X. Zhou, K. Grauman, and Q. Huang. Extreme relative pose estimation for RGB-D scans via scene completion. In *Proc. CVPR*, pp. 4531–4540, 2019. 2

[72] Z. Yang, S. Yan, and Q. Huang. Extreme relative pose network under hybrid representations. In *Proc. CVPR*, pp. 2455–2464, 2020. 2

[73] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In Y. Bengio and Y. LeCun, eds., *Proc. ICLR*, 2016. 4

[74] 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. 4

[75] Y. Yue, T. Kontogianni, K. Schindler, and F. Engelmann. Connecting the Dots: Floorplan Reconstruction Using Two-Level Queries. In *Proc. CVPR*, 2023. 2, 5, 7, 8

[76] I. Yun, C. Shin, H. Lee, H.-J. Lee, and C.-E. Rhee. EGformer: Equirectangular geometry-biased transformer for 360° depth estimation. In *Proc. ICCV*, pp. 6078–6089, 2023. 7

[77] C. Zhang, Z. Cui, C. Chen, S. Liu, B. Zeng, H. Bao, and Y. Zhang. DeepPanoContext: Panoramic 3D scene understanding with holistic scene context graph and relation-based optimization. In *Proc. ICCV*, pp. 12632–12641, 2021. 3

[78] J. Zhang, X. Xia, R. Liu, and N. Li. Enhancing human indoor cognitive map development and wayfinding performance with immersive augmented reality-based navigation systems. *Advanced Engineering Informatics*, 50:101432, 2021. 1

[79] Q. Zhao, L. Wan, W. Feng, J. Zhang, and T.-T. Wong. Cube2Video: Navigate between cubic panoramas in real-time. *IEEE Transactions on Multimedia*, 15(8):1745–1754, 2013. 3

[80] 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. 3, 4, 5, 7, 8, 9

[81] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *Proc. ICLR*, 2021. 2, 5