Computational Visual Media
https://doi.org/10.1007/s41095-023-0358-0

Research Article

Deep panoramic depth prediction and completion for indoor scenes

Giovanni Pintore[1,*] (✉), **Eva Almansa**[1,*] (✉), **Armando Sanchez**[2], **Giorgio Vassena**[2,3], **and Enrico Gobbetti**[1] (✉)

Abstract We introduce a novel end-to-end deep-learning solution for rapidly estimating a dense spherical depth map of an indoor environment. Our input is a single equirectangular image registered with a sparse depth map, as provided by a variety of common capture setups. Depth is inferred by an efficient and lightweight single-branch network, which employs a dynamic gating system to process together dense visual data and sparse geometric data. We exploit the characteristics of typical man-made environments to efficiently compress multi-resolution features and find short- and long-range relations among scene parts. Furthermore, we introduce a new augmentation strategy to make the model robust to different types of sparsity, including those generated by various structured light sensors and LiDAR setups. The experimental results demonstrate that our method provides interactive performance and outperforms state-of-the-art solutions in computational efficiency, adaptivity to variable depth sparsity patterns, and prediction accuracy for challenging indoor data, even when trained solely on synthetic data without any fine tuning.

Keywords machine learning; image processing and computer vision; vision and scene understanding; 3D stereo scene analysis

1 Introduction

Integrated visual and depth capture of indoor environments is a key enabling component for a wide range of applications, including autonomous navigation, mobile augmented reality, indoor mapping, and 3D reconstruction. In most situations, synchronized high-resolution depth and color data for the widest possible coverage around the viewer should be fed with low latency to further processing and analysis modules [1, 2].

Depth estimation is a fundamental problem for which a variety of active and passive solutions have been proposed over the past decades. While classic approaches exploit the correlation among multiple views, acquired simultaneously (e.g., stereo) or over time (e.g., video), single-shot capture and depth estimation has also attracted a lot of attention, since it ensures the lowest latency, reduces system hardware and synchronization burden, and offers basic building blocks for multi-view methods [3, 4].

As current 360° cameras offer viable low-cost and energy-efficient solutions for omnidirectional single-shot indoor capture [5], many research efforts are currently being focused on generating 3D from panoramic images. However, even with the full context provided by 360° capture, depth generation from monocular input remains inherently ambiguous, and is particularly complex in indoor settings characterized by large texture-less surfaces, abundance of clutter, and severe occlusions [2]. Despite the very significant recent advances in this field, especially with emerging deep-learning solutions that exploit hidden relations discovered in large data collections [6–8], monocular depth estimation remains extremely challenging.

Depth can also be measured with depth-sensing devices. Current depth sensors exhibit, however, speed, cost, and resolution limitations that hamper their direct usability for full-frame dense 360° capture

* Giovanni Pintore and Eva Almansa contributed equally to this work.
1 Visual and Data-intensive Computing, CRS4, Cagliari 09134, Italy. E-mail: G. Pintore, giovanni.pintore@crs4.it (✉); E. Almansa, eval.m.almansa@gmail.com (✉); E. Gobbetti, enrico.gobbetti@crs4.it (✉).
2 Gexcel srl, Elmas (CA) 09097, Italy.
3 Department of Civil, Environment, Architectural Engineering, and Mathematics (DICATAM), Università degli Studi di Brescia (UNIBS), Brescia (BS) 25123, Italy.
Manuscript received: 2023-03-07; accepted: 2023-06-03

in interior scenes. In particular, stereo cameras require large baselines and tend to fail in texture-less indoor environments, and structured-light sensors are at lower resolution than comparable visual cameras, are very sensitive to illumination variations, and suffer from short ranging distance. Longer ranging LiDAR sensors are more robust and accurate, but can only provide extremely sparse measurements at real-time rates [10]. Figure 1 shows typical depth information provided by different low-latency techniques.

In view of these limitations, many research efforts have been devoted to exploit the coarse information coming from depth sensing to improve the performance of depth prediction from RGB [10]. Sparse depth input, in particular, has shown to be very useful to provide supervision at training time to pipelines that infer depth from visual data [11–14]. More and more often, it is used at inference time [15, 16] for guided and non-guided depth completion [10]. However, the sparse output from various kinds of sensors imposes fundamental challenges on machine learning methods, since data relevance is not uniform and further processing is required to either reconstruct or ignore missing regions [17].

Because of this imbalance, depth prediction from dense RGB input and depth completion from sparse depth input have often been treated separately, and solved with different methods [18–20]. The few state-of-the-art solutions that try to jointly tackle completion and prediction target outdoor planar [21] or small field-of-view (FOV) perspective [22] projections, are not efficiently applicable to 360° indoor capture (Sections 2 and 4).

In this work, we introduce an end-to-end deep-learning solution to jointly perform real-time dense depth prediction and completion from single-shot indoor 360° captures. This method, the first to work directly on equirectangular images of indoor environments, combines and extends state-of-the-art end-to-end deep-learning solutions, introducing several specific novelties. Our input is a single equirectangular image registered with a sparse depth map, as provided by a variety of common capture setups. We do not make assumptions on the sparsity structure of the input depth, which can range from the few dense stripes produced by LiDAR solutions to the regular and irregular point sampling produced by other active and passive vision-based approaches. We expect, however, the images to be approximately gravity-aligned, as in all common datasets available [23–28]. This condition is a de-facto standard for practically all indoor static and mobile acquisition setups, as they are equipped with automatic georeferencing and alignment systems [7, 8, 29–31]. It is worth noting that we can accommodate for large tolerances in gravity alignment. In our results (Section 4), we demonstrate how our system even works in the case of a backpacked LiDAR acquisition system with variable vertical tilt.

Assuming a rough gravity alignment allows us to optimize our network design. The network is constituted by a single-branch encoder–decoder, which jointly processes dense visual data and sparse geometric data in an efficient way. The initial residual encoder takes as input simultaneously 4 channels (i.e., RGB + sparse depth), and, through a gating system, returns fused visual and geometric features at different resolutions. Such features are efficiently compressed and flattened in an asymmetric way, by exploiting the intrinsic characteristics of gravity-aligned equirectangular projections of indoor scenes [8, 32]. In fact, since gravity plays an important role in the design and construction of interior environments, vertical and horizontal features

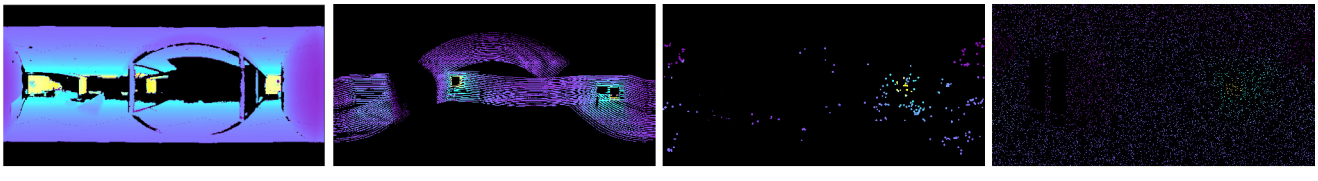


Fig. 1 Different kinds of sparse depth. First image (from the left): depth map captured by structured-light sensors (Matterport Pro 3D camera) has lots of missing areas when rooms are large, surfaces are shiny or thin, and strong lighting is abundant. Second image: a depth map captured by a LiDAR setup (two Velodyne VPN-16 shifted of the vertical direction with different direction) has lots of valid information but only for a few stripes. Third image: depth information may also come from triangulated features in purely image-based pipelines; indoor environments, however, have lots of flat textureless surfaces, and reliable features, here detected from SIFT, may be very sparse. Fourth image: a typical input from low-cost structured light sensors with sparse measurements only for a small subset of the captured camera pixels; for synthetic training, a typical approach is to use a Bernouilli distribution to sparsify inputs [9].

have different characteristics in most, if not all, man-made environments. Moreover, most 360° capture setups have a much more regular coverage along the horizontal than on the vertical direction because of masking effects [23]. As a result, we can exploit this anisotropy by compressing more on the vertical than on the horizontal direction. The resulting flattened features are refined through a lightweight self-attention module [33], which, acting as a bottleneck, exploits the wide context provided by omnidirectional capture in order to find the short- and long-range relations between parts of the scene which are typical of man-made environment. Decoding proceeds symmetrically to the encoder, but without need for gating, to reach the final output resolution.

Our contributions are summarized as follows:

- We introduce a novel residual encoder for the sparse-to-dense image-driven problem, which exploits lightweight gated convolutions [34] to process dense visual data and sparse geometric data together in a single branch at very little cost (Section 3.1). This design results in a much faster and more versatile network, with respect to the current approaches that process the data using multi-branch architectures and interconnections at various levels of the network [18, 26, 35–37]. Our encoder combines the advantages of a gating system, to handle different types of input in a single encoder, and of a residual architecture [38], allowing us to use deeper networks with respect to common inpainting solutions [36, 39], thanks to the efficient fusion and propagation of features at various resolutions and depth, without using skip connections that would increase the computational burden of the network [36]. As a result, the method meets real-time constraints even for the highest image and depth resolutions (Section 4.2).
- We introduce asymmetric feature compression and flattening for depth completion of gravity-aligned indoor panoramic imaging (Section 3.2), exploiting the intrinsic characteristics of equirectangular projections of indoor scenes [7, 8]. While gravity-aligned features have been employed earlier for depth estimation [8], they have not been used for designing depth completion networks. In this setting, this type of encoding remarkably maximizes the visual and geometric information gathered from a panoramic input, allowing, at the same time, the gathering of multi-resolution features and the use of a lightweight self-attention module (i.e., 1 layer, 4 heads) as bottleneck. Such an attention module allows the network to find the short- and long-range relations between parts of the scene, typical of man-made environment and panoramic images [32], relating features both spatially and at various levels of network depth (Section 3.1). Other state-of-the-art approaches, instead, employ dilated convolutions [36] as bottleneck, which are common in visual inpainting [39], renouncing to exploit deep-level features and, thus, losing part of the long-term information.
- We show how our approach is capable to handle a large variety of sparsity patterns and delivers excellent results when trained on synthetic data and applied to various real-world configurations with or without fine tuning (Section 4). In order to increase the robustness to various sampling patterns, we also complement approaches based on theoretical noise models for moderately dense and uniform RGB-D capture [10, 40] with a data augmentation module designed to model LiDAR behavior (Section 3.3.1). Such an augmentation is fundamental to increase the performance of our model in the LiDAR case, and increases also the performance of other methods, whose advertised accuracy was instead related to a specific capture pattern (Section 4.3).

We evaluated our approach on a variety of panoramic indoor scenes, ranging from commonly available panoramic indoor benchmarks [23, 26, 41] to novel real-world captures with mobile devices. Our results demonstrate that our approach outperforms current state-of-the-art solutions in terms of speed and accuracy (Section 4).

2 Related work

Depth estimation and completion from monocular input and indoor 3D reconstruction have a long history, and have recently attracted renewed interest with the emergence of deep-learning techniques. Here, we focus on the approaches most closely related to our work, referring the reader to recent surveys [2, 3, 10] for a general coverage.

2.1 Monocular depth estimation from RGB

Monocular depth estimation is a classic task in computer vision. While early solutions used various

combinations of feature detection, matching, and geometric reasoning, in recent years, a large body of deep-learning methods are being proposed for handling this traditional ill-posed problem under less restrictive constraints [4]. The *FCRN* architecture, proposed by Laina et al. [42], has become a common baseline. A variety of other solutions have been later proposed for improving inference for perspective images [12, 43–49]. However, it has been shown that, without specific adaptations, the direct application of these solutions to 360° depth estimation of indoor environments produces sub-optimal results [50]. For this reason, research has started focusing on methods to exploit the wide geometric context present in omnidirectional images. Several approaches convert equirectangular images to cubemaps and then make specific adaptation to methods designed for perspective images [51]. To make the network aware of the distortion, spherical convolution methods have been also proposed [50, 52–55]. Wang et al. [6] proposed instead a two-branch network, respectively for the equirectangular and the cubemap projection, based on a distortion-aware encoder [50] and the FCRN decoder [42]. Recent state-of-the-art data-driven solutions for panoramic depth estimation in indoor spaces [7, 8] have proposed to work directly on equirectangular images, as well as to leverage the concept of gravity-aligned features to reduce network size [8, 32]. While these concepts were applied to uniformly dense input for the depth estimation task, we extend them to efficiently handle both the image and sparse depth features, compressing them to gather visual and geometric information at various resolution and depth to the decoder, together with short- and long-range relations among parts of the scene (Section 3.1).

2.2 Guided monocular depth completion

Sparse-to-dense depth completion with the support of a guiding RGB image has been the focus of much research [10]. The majority of works focus, however, on small-FOV perspective poses [22] or planar projections for outdoor acquisitions [16, 56]. We discuss here only the approaches that can be directly applied or easily adapted to panoramic indoor environments.

In order to upsample and complete a sparse depth signal, generic scene methods that rely on registered RGB-based appearance as guidance either devise custom convolutions and propagate confidence to consecutive layers [57], or use content-dependent and spatially-variant guiding convolutions [58]. Alternative sources of information that are exploited for depth completion may also include confidence masks and object cues [59]. Cross-guidance between the RGB and depth encoders [60] has also been used. Moreover, to avoid the depth mixing typically induced by the standard MSE loss, a binned depth representation trained using a cross-entropy loss has been shown to be beneficial [15]. Recently, BIPS [61] proposes a bi-modal (RGB-D) panorama synthesis framework to jointly synthesize panoramic RGB and depth. Similar to our work, BIPS considers different kinds of sparsity patterns in depth input. However, the goal of BIPS is to provide realistic image inpainting and a complete 3D model for many applications (i.e., including layout), jointly synthesizing color and depth from partial input, rather than focusing on depth prediction and completion.

Even though deep learning has been widely used for inpainting of indoor scenes, extensions of those networks to color guided depth completion are still uncommon [16]. One of the main reasons is that large-scale training sets are not readily available for captured indoor RGB-D images paired with dense depth images. As a result, most methods for depth estimation have been classically trained and evaluated only for pixels that are captured by commodity RGB-D cameras [62]. From this data, they can, at best, learn to reproduce observed depths, but not complete depths that are unobserved, which in indoors have significantly different characteristics. To address this issue, Zhang and Funkhouser [26] introduced a new dataset based on the large-scale Matterport3D [23], which provides 105k RGB-D images aligned with dense depth images computed from multi-view reconstructions in 72 real-world environments, and proposed a hybrid solution to estimate surface normals and solve for indoor depth via a final global optimization. The method, however, has speed limitations and does not scale for different kinds of sparsity (see Section 1).

More recently, pure deep-learning solutions have been proposed for color guided depth completion. Cheng et al. [63] proposed an approach in which a low-FOV dense depth camera is registered with an omnidirectional camera, and the dense depth

from the limited FOV is extended to the rest of the recorded omnidirectional image through a convolutional network. We tackle, instead, the more general problem of omnidirectional sparse-to-dense depth estimation without any region in which a dense estimation is provided. This problem is tackled by several recent works. Huang et al. [36] exploited an inpainting self-attention network [64] to generate the dense depth map and a dedicated U-Net [65] to preserve depth boundary information. Skip connections [65] are also used in their method to adapt the generic inpainting network to the specific depth prediction task and to better recover fine-grained details. We handle more general sampling patterns, and propose a much faster solution. Park et al. [18] proposed an interactive Non-Local Spatial Propagation Network (NLSPN) that predicts non-local neighbors for each pixel and then aggregates relevant information using the spatially-varying affinities. To maximize the utility from the sparse source, Huang et al. [17] proposed a Sparse Signal Superdensity (S3) framework, tested for stereo sparse-guidance, for which expands the depth value from sparse cues while estimating the confidence of expanded region. Specifically targeted for guided monocular depth completion, Guizilini et al. [35] introduced Sparse Auxiliary Networks (SANs) to process the sparse signal separately from the dense RGB signal. Their pipeline consists of two parallel branches for the two signals, connected at encoder and decoder level by direct feature fusion. With a similar decoupled design, Liu et al. [66] advanced the pure depth prediction network RectNet [50] to support an SLAM-based reconstruction system where the scattered data are SLAM-SfM features. The method, however, costs 311 GFLOPs for a 512×256 image, while our solution takes 38.2 GFLOPs for a 1024×512 image.

These recent purely data-driven methods achieve state-of-the-art performance mainly on perspective views and at the cost of a significant computational cost (see Table 1). We propose, instead, a much leaner indoor solution for panoramic images, showing how our design can cope with a variety of dense sampling patterns and density and can achieve high accuracy even without any fine-tuning after a training on synthetic data.

3 Network architecture and training

Our network is designed to directly infer a panoramic depth map from a single equirectangular image registered with a sparse depth map. Figure 2 illustrates its structure for a 512×1024 input map. The architecture, is, however, fully scalable with respect to input resolution (Section 4).

The network input is given by the concatenation of the $3 \times 512 \times 1024$ RGB image with the $1 \times 512 \times 1024$ sparse depth map. On input, the RGB image is dense and contains a color value for each pixel. Valid pixels in the sparse depth map contain the distance from the camera in metric scale, while invalid pixels contain a zero.

The feature extraction is performed by 5 layers, each one having a residual architecture inside [38]. In order to process dense visual data and sparse geometric data together, each block is built around specific gated convolutions. The indoor panoramic format is also specifically handled through spherical padding and ELU activations. Encoding layers are described in Section 3.1. Similarly to other state-of-the-art solutions for 3D from RGB data [7, 8, 32, 67],

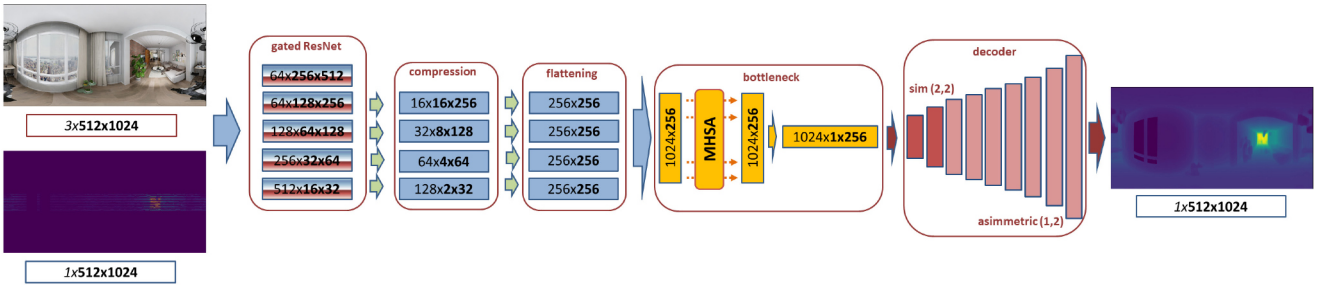


Fig. 2 Network architecture. Our network is constituted by a single-branch encoder–decoder, which processes together the dense visual and sparse geometric data. A residual-gated encoder takes as input 4 channels (RGB + sparse depth) returning fused features at different resolution. Multi-resolution features are compressed, flattened, and passed to an MHSA-single layer module (i.e., bottleneck). Decoding proceeds symmetrically to the encoder, but without using gating, to reach the final output resolution.

we start from the assumption that, in architectural indoor spaces, vertical and horizontal features have different characteristics along and across the gravity direction. We apply such concepts in our context by compressing the extracted features (i.e., 4 deeper feature maps) through an anisotropic contractive encoding that preserves the horizontal dimension and compresses the vertical one (Section 3.2). The resulting 4 feature maps, containing information at different spatial and depth levels, are flattened and concatenated in a single, sequential latent feature of feature dimension×sequence length. The encoding of the latent feature as a sequence allows the network to use a multi-head self-attention module (MHSA) [33] as bottleneck, leveraging complementary features in distant portions of the image and depth measurements rather than only local regions to support reconstruction. This makes it possible to cope with large changes due to occlusions and to take into account the short- and long-range relations between parts of the scene typical of man-made environment. As a result of these design choices, decoding proceeds very fast and without the need for skip connections, as it can just consist of a series of convolutions, upsampling, and activations until the output resolution is reached.

Our model is trained end-to-end supervised by sparse-dense depth map couples (Section 4.1), without specific assumptions on sparsity patterns, which are learned from training data. In addition to use variable depth density for RGB-D situation, we introduce a LiDAR-specific augmentation module that generates parametric LiDAR capture patterns at run-time during training (Section 3.3.1).

3.1 Feature extraction

The joined feature encoding of the mixed RGB+depth input is performed by a cascade of 5 blocks, i.e., 1 convolution-pooling block followed by 4 residual blocks. Given the spherical nature on the image, 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 [68].

Each residual block follows the *ResNet* scheme, involving two convolutions and one upsampling layer [38]. Here, for each convolution layer, we introduce a dynamic gating approach to efficiently process dense visual data and sparse geometric data together.

In a generic (vanilla) convolutional layer, for each pixel located at (y, x) in an input feature map F_n having n channels, the same filters are applied to produce the output for a generic convolutional filter.

However, the sparse depth channel does not contain all valid pixels, but for single channel tasks, like pure inpainting without RGB guidance, partial [69] convolutions can be adopted to make the convolution dependent only on valid pixels. Indeed, such solution is not very efficient for our problem, since, essentially, partial convolutions act as single-channel hard-gating, heuristically classifying each spatial location to be either valid or invalid, and setting to zeros or ones the mask in next layer no matter how many pixels are covered by the filter range in previous layer [39].

In our case, instead, we introduce a multi-channel *gated convolution* approach, where a multi-channel soft mask is automatically learned from data, taking decisions that jointly consider the sparse depth and the dense color channel. While gated convolutions are often adopted for pure image synthesis combined with dilated convolutions [39, 70, 71], here we use such a soft masking to model a kind of implicit confidence for multi-source features.

For each gated convolutional layer, gated features F'_m are

$$\begin{aligned} G_m &= \mathrm{conv}(W_{\mathrm{g}1}, \mathrm{conv}(W_{\mathrm{g}k}, F_n)) \\ F_m &= \mathrm{conv}(W_{\mathrm{f}}, F_n) \\ F'_m &= \sigma(G_m) \odot \psi(F_m) \end{aligned} \tag{1}$$

where σ is the Sigmoid function, whose output values are within $[0, 1]$, ψ is an activation function (in this paper we use ELU [72] to remove the need for batch normalization), $W_{\mathrm{g}1}$, $W_{\mathrm{g}k}$, and W_{f} are different sets of convolutional filters, used, respectively, to compute the gates ($W_{\mathrm{g}1}$, $W_{\mathrm{g}k}$) and features (W_{f}), and F_n is the input feature map.

In terms of computational complexity, the use of gated convolution should almost double the number of parameters in comparison to a standard, vanilla convolution [39]. To cope with this problem, we adopt here a lightweight solution, also called depth-separable convolution [34], which reduces the number of parameters and processing time while maintaining the effectiveness. Thus, we decompose a gated convolution soft mask G_m with $k_h \times k_w \times n \times m$ into a depth-wise convolution [34] (i.e., $k_h \times k_w$ kernel) followed by a 1×1 kernel convolution. Such solution has only $k_h \times k_w \times n + n \times m$ parameters, resulting

in a less overall computational cost for all the encoder without measurable loss in efficiency for our problem (Section 4.4).

Our encoder returns 4 feature maps having different depth and spatial size (Fig. 2), gathering fused information from both visual and geometric input. Beside data fusion, propagating these levels avoids using skip connections between encoders and decoders, such as those used by several other methods [18, 36, 73] to retrieve fine-grained details, drastically reducing the computational complexity (see Table 1). At the same time, propagating this information together in a deep architecture is not simple and requires an efficient compression system. To this end, we introduced a specific compression process described in Section 3.2.

3.2 Feature compression and decoding

In order to support an efficient gathering of information from the extracted features, taking into account the peculiar characteristics of indoor environments, we perform a specifically designed feature compression exploiting our knowledge of preferential directions. We start from the assumption that gravity plays an important role in the design and construction of interior environments, so world-space vertical and horizontal features have different characteristics in most, if not all, man-made environments. Moreover, the amount of information contained in the spherical equirectangular projection degrades significantly going towards the poles, and even disappears completely in the input depth due to the hardware limitations of the instrument.

According to these assumption, we perform an anisotropic contractive encoding that reduces the vertical direction while keeping the horizontal direction unchanged, so that separated vertical features can be better preserved. Specifically, we reduce the vertical dimension by a factor of 8 through an asymmetric convolution module with stride $(2, 1)$, applied 3 times, that contains a 2D convolution and an ELU module. We apply such a compression for each encoded feature map (i.e., 4 maps, Section 3.1). Finally, compressed features are reshaped to the same size and joined in a flattened latent feature, $L_s = (l_0 \dots l_s)$, as a sequence of s feature vectors of dimension l (i.e., s horizontal size of the less deep feature map—$s = 256$ and $l = 1024$ for a 512×1024 input).

Such a compressed representation contains a variety of information about the geometry of the scene, both local and non-local, which can be exploited to recover missing depth samples. In our case, we aim to leverage complementary features in distant portions of the image rather than only local regions, to support both depth completion and recovery. To do that, we adopt a single-layer multi-head self-attention (MHSA) scheme [33]. Our self-attention module takes the latent features $L \in \mathbb{R}^{s \times l}$ as input, and outputs a self-attention weight matrix $A \in \mathbb{R}^{s \times s}$:

$$A = \text{softmax}\left(\frac{(LW_q)(LW_k)^T}{\sqrt{l}}\right) \tag{2}$$

where $W_q, W_k \in \mathbb{R}^{l \times l}$ are learnable weights. The MHSA module has a particularly lightweight design with 4 heads and only 1 inner layer. We have verified experimentally that increasing the number of layers and heads does not affect performance.

Once passed to the MHSA module, the decoding of the latent feature ($1 \times 1 \times s$ in Fig. 2) is very fast, through convolutions, upsampling modules, and ELU activations, until we reach the target output resolution ($1 \times h \times w$ in Fig. 2).

3.3 Training strategy

During the training phase, we compute the weights of the network using a supervised training approach that exploits databases matching indoor equirectangular images with their correspondent sparse and dense depth maps (Section 4.2 for dataset details).

3.3.1 Coping with variable distributions of sparse depth samples

The distribution of the samples of the sparse maps can vary considerably depending on the acquisition methods. While sparse-dense datasets from structured-light sensors are available [26], it is not so for LiDAR data, even if these sensors are increasingly used also in indoor environments (Section 1). Generating those sampling patterns cannot be simply done by generic noise models (e.g., Refs. [10, 40]), but must take into account striping.

To this end, we adopt a sparsity simulation module to produce, under parametric control, different types of LiDAR patterns starting from a dense ground truth. Such a module can be used to generate specific, defined capture setup (e.g., 1 scan with fixed parameters), or to randomize sparsity at training time, thus augmenting the data to make the model

more robust to different inputs. Such a module extends existing generators [74–76] to provide run-time sparse samples extracted from ground truth dense depth maps.

Our sparsity simulator is driven by a limited number of parameters, that can be eventually randomized to augment data: the number of heads (sensors) and their position and orientation, and for each sensor, the horizontal aperture (i.e., 360 degrees), the vertical aperture, and the number of laser beams (e.g., 16 for a Velodyne16-like device), etc. Furthermore, a small 3D random noise is applied to simulate real-device noise. Head aperture and beam parameters are bounded to match to realistic setups (e.g., beams are multiple of 16).

It should be noted that even a *0-beam* case is contemplated during augmentation. This case allows the network to work even if there is no geometric input. In this case the prediction performance is aligned with that of recent state-of-the-art image-based methods [7, 8] (Section 4).

Using this augmentation module as a complement to those based on noise models, in addition to increase robustness, allows us to avoid locking the training to a specific device sampling pattern, since sparse data is generated from ground truth dense maps. In particular, as we will see in Section 4, differently from most previous work, we can train the model on purely synthetic datasets, and apply it to real-world data captured with a specific device even without any fine-tuning.

3.3.2 Loss function

Independently from the type of sparse depth distribution, learning is driven by a loss function combining two data terms:

$$\mathcal{L}_{\text{data}} = \mathcal{L}_{\text{d}} + \mathcal{L}_{\text{ss}} \tag{3}$$

where $\mathcal{L}_{\text{d}}$ is the robust *Adaptive Reverse Huber Loss* (*BerHu*) [77], which has proven to be effective in many recent works for panoramic depth estimation [6–8]. To further take into account structural information, we add the structural loss $\mathcal{L}_{\text{s}}$, based on the Structural Similarity Index Measure (SSIM) [78], which measures the preservation of highly structured signals with strong neighborhood dependencies. Since SSIM is higher if the two compared images are more structurally similar, we define $\mathcal{L}_{\text{ss}} = 1 - \text{SSIM}(D_{\text{gt}}, D_{\text{p}})$, where D_{gt} is the ground truth dense depth and D_{p} is the final inferred depth.

4 Results

Our approach is implemented with PyTorch 1.5.1 and has been tested on a large number of indoor scenes.

Source code and models will be available to the public at `https://github.com/crs4/PanoDPC`.

4.1 Benchmark datasets

Real-world capture of indoor environments is usually performed using a variety of settings, including panoramic cameras aligned with LiDAR-based setups (e.g., Velodyne) or stitching of structure-light-based sensors (e.g., Matterport). The limitations of these devices for indoor use [10] makes it difficult to find data corresponding to all the various use cases coupled with reliable full-frame ground truth data.

For training purposes, we employ in this paper the standard *Matterport3D-SD* (i.e., Matterport 3D sparse depth) [26] as well as a new dataset created on purpose that builds on Structured3D [25], dubbed *S3D-SD* (i.e., Structured 3D sparse depth).

4.1.1 Training and testing with Matterport3D-SD

Matterport3D was the first one to provide full-view indoor poses with paired sparse and dense depth maps, and for this reason, it has become a popular benchmark in recent papers and surveys [10, 18, 36]. For the sake of comparison with other results, and to show the behavior of our method on high-quality structured-light data, we thus include an analysis of our performance by training and testing our method on *Matterport3D-SD* compared to state-of-the-art works that use it. This dataset, however, is limited to a single kind of device operating in reasonably cooperative environments that ensure rather dense capture, so that even classical infilling or hybrid data-driven solutions may be adopted with some success [10]. Figure 3 shows representative examples. For this reason, we complement the dataset with much more challenging examples that cover other setups and less cooperative interiors.

4.1.2 Training and testing with S3D-SD

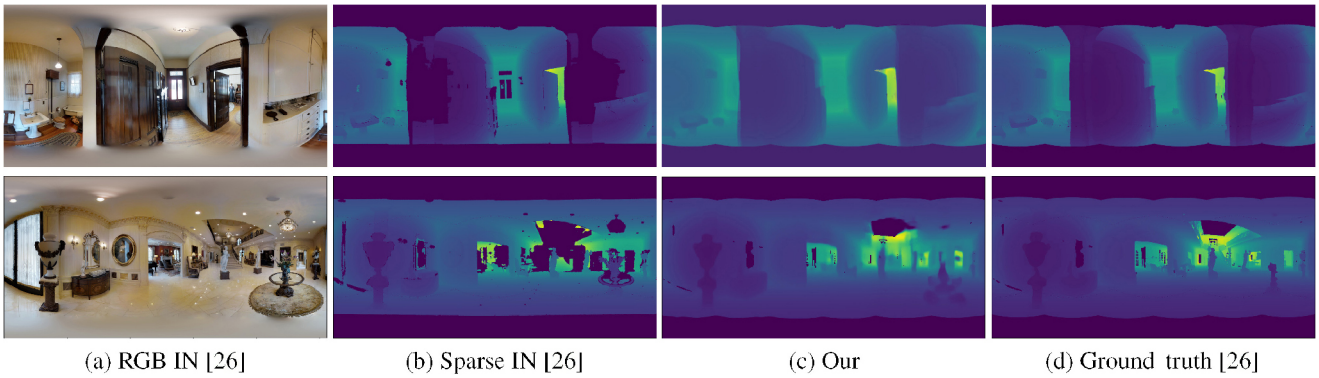
In order to cover a large variety of use cases, we created a novel dataset leveraging on synthetic data generated by sampling the large-scale Structured3D [25] photo-realistic synthetic dataset, containing 3.5k house designs created by professional designers with a variety of ground truth 3D structure annotations, including 21,000 photo-realistic full-panoramic (i.e.,

1024 × 512 equirectangular format) indoor scenes. The main advantage of such a synthetic dataset is that it provides a fully accurate dense ground truth for color and depth, which is not available with other common large-scale datasets, such as Matterport3D [23] or Stanford2D3DS [24], whose completeness, even if based on multi-view, is still limited by visibility and sensor limitations. For training purposes, we associate to each panoramic image and ground truth dense depth a sparse depth created through a sampling process that simulates a variety of setups. 50% of the depths simulate LiDaR setups, 25% RGB-D setups, and 25% data coming from SfM/stereo pipelines. The LiDaR setups emulate multi-beam mobile devices, selecting with equal probabilty 0, 16, 32, 48, 64, 80, and 96 beams on a rotating platforms. LiDaR simulation is performed by a parametric sampling process [75, 76, 79], using configurations mimicking *Velodyne* devices with 30°–40° vertical FOV. The 0-beam case is included to simulate pure visual capturte, while for the other multi-beam setups the depth coverage ranges from 16 beams (6% of pixels having depth values) to 96 beams (38%). As an extreme case, we also include a case where we have no depth input (i.e., data are purely visual, and depth maps have 0% valid pixels. Representative examples are included in Fig. 4. Moreover, to evaluate the method on different kinds of sparsity patterns, we simulate data coming from low-cost depth cameras using Bernouilli sampling [40] and input from SfM/stereo pipelines using an SIFT detector to place samples at feature locations. Training data are, thus, augmented with two parameterizations of Bernoulli samplings (24.68% and 6.17% of visible pixels having a depth), as well as with two different SIFT settings (with 0.91% and 2.99% valid depth pixels). Each of these 4 configurations comprise 12.5% of the training data. Representative examples are included in Fig. 5.

In order to validate the generalization capabilities of the model and the suitability of training on synthetic data, models trained on this dataset are tested both on S3D data and on completely novel data coming from other capture setups, including real-world ones.

4.1.3 Validating on novel real-world captured data

Furthermore, as another important point of our work, we tested our model with a real-world sparse and challenging capture campaign, not included in any of the training datasets, but supporting a dense capture as dense ground truth. Thus, we produce a novel dataset from a real LiDAR RGB-D acquisition (i.e., mobile device with 2 Velodyne VLP-16 and a registered Garmin spherical camera, Fig. 7) and a ground truth dense depth acquisition through a $FaroFocus3DX330TLS$. Each sparse scan takes about 300 ms and produces about 16% of pixels with valid depth. We have acquired, in a multi-floor and multi-room environment, about 150 scenes in equirectangular format aligned with dense ground truth and sparse depth maps. Note that the gravity alignment of the poses is directly the one provided by the tracking tools in the mobile device and has not been corrected through dense depth registration. This choice results in tilted sparse-dense pairs, which also provide us with a real-world benchmark to evaluate the robustness of our system to misalignment with respect to gravity direction (see Section 1). We use such a real-world benchmark for testing without any fine tuning, after training on *S3D-SD*, also demonstrating transfer-learning capabilities.



(a) RGB IN [26] (b) Sparse IN [26] (c) Our (d) Ground truth [26]

Fig. 3 Qualitative results on *Matterport3D-SD* dataset [26]. Masked samples in the results are missing samples in the ground truth.

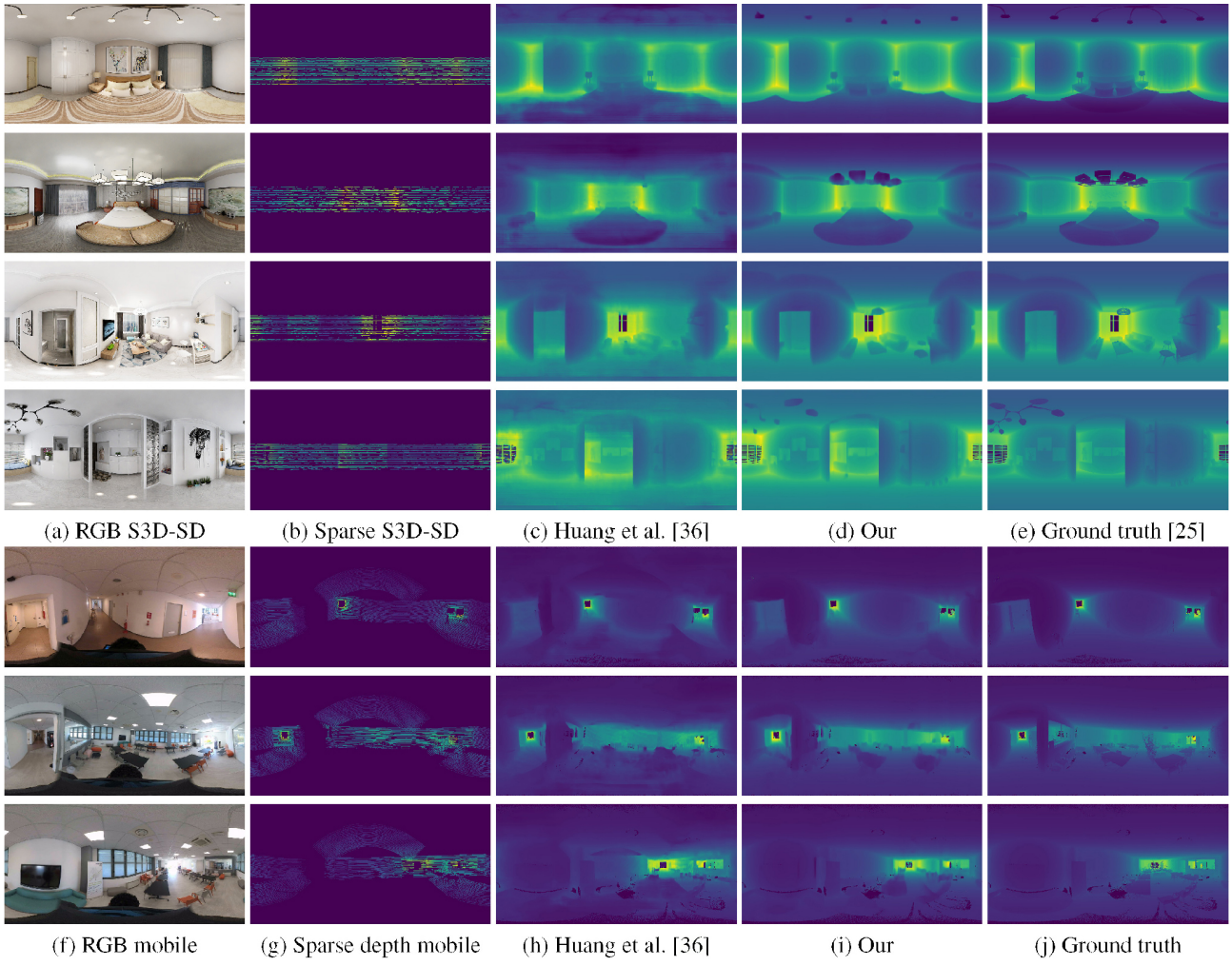


Fig. 4 Qualitative performance on S3D-SD with a LiDAR configuration with 32 beams and on real mobile LiDAR indoor capture. Qualitative results with the same setup of Table 2. Our results are compared to the Huang et al. [36] approach trained with the same equirectangular augmented S3D-SD dataset with varying sparsity patterns.

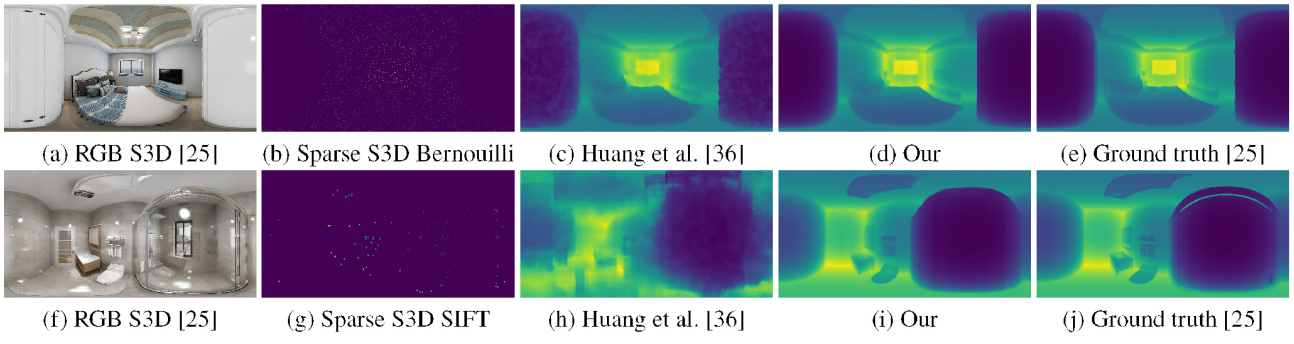


Fig. 5 Qualitative performance on S3D-SD with different input depth sparsity patterns. Qualitative results using simulated input from low-cost depth cameras using Bernouilli sampling and simulated input from SfM/stero pipelines, using an SIFT detector to place samples. Our results are compared to the Huang et al. [36] approach trained with the same equirectangular S3D-SD dataset.

4.2 Experimental setup and computational performance

We trained the network using the Adam optimizer [80] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, on four NVIDIA RTX 2080Ti GPUs (11 GB VRAM) with a batch size of 8 and a learning rate of 0.0001. For all benchmarks we adopt their original splits. Our new real-world dataset is not used for training, but for testing after training on synthetic data. With the given setup the

best valid epoch was around 170 epochs for *S3D-SD* and *Matterport3D-SD*. The average training speed on 4 GPUs is about 105 ms for each 512 × 1024 input image and depth pair.

Table 1 shows our computational complexity stats, compared with several state-of-the-art methods for the inference of a 512 × 1024 image and depth map. Our computational cost, in terms of GFLOPs, is significantly lower than for competing solution. Note that this increased performance is also with respect to networks with a lower number of parameters but with a far more complex structure. Moreover, our method produces depth maps directly from equirectangular inputs without pre- or post-processing steps and can thus be directly integrated in production systems without additional overhead.

As a result, the inference performance of our network guarantees a low-latency generation of dense depth, and we can therefore support full instantaneous frame-by-frame depth map generation directly at acquisition. In our case, starting from a 512 × 1024 image and depth map, we infer depth in under 16 ms on a single NVIDIA RTX 2080Ti, which is much faster than a single rotation of typical LiDARs covering a 360° view (e.g., 50–200 ms per rotation for a Velodyne VLP-16). The lean network structure also leads to a good scalability, as demonstrated by results with larger images included at the bottom of Table 1. We can, in particular, generate 2k×4k depth images from equally-sized inputs in less than 0.4 s.

Table 1 Computational cost and performance. Our method is compared to the best performing state-of-the-art competitors

Method	Size	Param	FLOPs↓	ms/frame↓
Ma et al. [81]	512 × 1024	26.10 M	765.1 G	137
GAENet [82]	512 × 1024	4.06 M	60.12 G	39
PENet [83]	512 × 1024	131.67 M	487.4 G	167
packNet+SAN [35]	512 × 1024	76.99 M	304.7 G	149
NLSPN [18]	512 × 1024	26.23 M	829.86 G	167
Huang et al. [36]	512 × 1024	13.10 M	1624.9 G	105
Our	512 × 1024	22.11 M	**38.2 G**	**16**
Our	1024 × 2048	44.14 M	**211.7 G**	**67**
Our	2048 × 4096	132.22 M	**1319.3 G**	**384**

4.3 Quantitative and qualitative evaluation

We evaluated our method with the same error metrics which are common to prior depth prediction and completion works and surveys [10, 26, 35, 36, 84]: mean absolute error (MAE), mean squared error (MSE), root mean square error of linear measures (RMSE), and three relative accuracy measures δ_n ($n = 1, 2, 3$), defined as the fraction of pixels where the relative error is within a threshold of 1.25^n. For MAE, MSE, and RMSE, smaller is better (i.e., unit is meter), while for δ_n larger is better.

We compare our results with state-of-the-art solutions for both indoor or generic scenes, for which the full code was available [18, 35, 36, 81–83] and an end-to-end training with equirectangular format was possible. The methods were adapted with minimal modifications to equirectangular images. We use 1024 × 512 for all tests.

Table 2 summarizes our performance and comparisons with related works using the augmented S3D-SD dataset to train every baseline compared (see Section 4.1), and LiDAR-specific examples for the inference. To select the training and the testing set, we adopt the official Structured3D split [25].

For synthetic tests, we considered all the simulated LiDAR configurations (i.e., 16–96 beams and various FOVs) discussed in Section 4.1. In Table 2, for clarity, we summarize only the results and comparisons for a

Table 2 Quantitative comparison on S3D-SD/LiDAR and real LiDAR capture. We show our performance evaluated on standard metrics and compared to the recent state-of-the-art approaches which are comparable with us. Here we present results simulating a 360° capture with 40° vertical FOV (−30 to 10 degrees) and 32 active beams in the synthetic dataset, and results using a real mobile device with 2 Velodyne VLP-16 and a registered Garmin spherical camera with ground truth obtained using a Faro Focus3D X 330 TLS (see Section 4.1)

Method	S3D-SD / LiDAR 32 beams							Mobile LiDAR 16+16 beams						
	MSE↓	MAE↓	RMSE↓	SSIM↑	δ_1↑	δ_2↑	δ_3 ↑	MSE↓	MAE↓	RMSE↓	SSIM↑	δ_1 ↑	δ_2 ↑	δ_3 ↑
GAENet [82]	0.086	0.394	0.160	0.149	0.466	0.753	0.889	0.041	0.472	0.105	0.202	0.230	0.555	0.748
packNet+SAN [35]	0.052	0.286	0.125	0.614	0.596	0.867	0.954	0.027	0.404	0.078	0.539	0.278	0.603	0.842
Ma et al. [81]	0.044	0.286	0.104	0.591	0.587	0.895	0.964	0.018	0.366	0.051	0.434	0.424	0.723	0.895
PENet [83]	0.028	0.210	0.090	0.595	0.671	0.930	0.976	0.010	0.252	0.035	0.512	0.578	0.835	0.969
NLSPN [18]	0.023	0.185	0.084	0.840	0.723	0.943	0.982	0.011	0.260	0.035	0.746	0.610	0.841	0.937
Huang et al. [36]	0.017	0.138	0.068	0.830	0.824	0.960	0.987	0.009	0.197	0.030	0.745	0.763	0.886	0.947
Our	**0.003**	**0.038**	**0.022**	**0.944**	**0.982**	**0.993**	**0.997**	**0.003**	**0.088**	**0.024**	**0.822**	**0.922**	**0.986**	**0.997**

40° vertical FOV and 32 active beams case, since other S3D-SD/LiDAR tests follow the same performance trend (see Fig. 8).

We also include results on the real-world scenes acquired with the mobile LiDAR system (i.e., here named *mobileLiDAR*16+16), compared to ground truth dense depth acquisition through a *FaroFocus*3*DX*330*TLS* (i.e., all models trained with *S*3*D LiDAR*).

Both the real-world benchmark and the synthetic data limited to LiDAR are used only as a testing set, without any fine-tuning, thus providing evidence of transfer learning capability.

Despite our lower computational complexity, already discussed in Section 4.1, our method outperforms competitors for every condition, showing that simply adapting general purpose pipelines to the specific panoramic indoor problem leads to unsatisfactory results.

Figure 4 presents qualitative results using the S3D LiDAR and mobile LiDAR test-sets adopted in Table 2. Here, we compare our method with the method of Huang et al. [36], which is the best performing among competitors in terms of quantitative results. In this case, with only a few stripes available from the scanner, our method benefits from its specific compression and information gathering features (Section 3.1) to recover more details in the final depth map.

Figure 6 shows additional experiments, where geometric visualization is obtained by unprojecting the depth map into 3D point clouds. Following the same setup of Table 2 and Fig. 4, we show, respectively: the RGB input (Fig. 6(a)); the sparse input depth as a point cloud (Fig. 6(b)); the point cloud predicted by the best competitor [36] (Fig. 6(c)); our prediction (Fig. 6(d)); and the ground truth point cloud (Fig. 6(e)). The illustrations complement the other qualitative and quantitative results with an easy-to-read illustration of the 3D reconstruction of the scene from a reference point of view. The performance improvement offered by the proposed approach is especially visible in regions where clear geometric structures (walls, ceilings or floor) are present.

Figure 7 shows instead examples of scenes acquired with the mobile backpacked device. Numerical data are presented in Table 2). As for the experiments Fig. 4, our method successfully completes the map, with better accuracy than competitors. Furthermore, it is also visually evident that the data acquired with the mobile backpacked device present a significant misalignment with respect to the direction of gravity, also variable along the user's trajectory, which results in a distortion of the equirectangular projection. The consistent results also in this case show that our method is robust with respect to such an inclination, tested in a real and mobile user-case. Note that, in practice, such inclinations can be reduced before entering the depth estimation pipeline, by using on-board IMUs as well as by aligning successive poses. We show here uncorrected results, to also demonstrate the possibility of using the pipeline we present for frame-by-frame inference, without any latency connected to the integration of multiple frames or the need for assistance from external sensors.

For completeness, we performed a further comparison of performance for different sparsity patterns. Table 3 summarizes the results obtained by emulating the pattern of low-cost structured

(a) RGB S3D [25] (b) Sparse S3D-SD PC (c) Huang et al. PC [36] (d) Our PC (e) Ground truth PC [25]

Fig. 6 Qualitative performance on S3D-SD by point cloud (PC). In these examples, 3D point clouds are obtained by unprojecting depth maps, using the same setting of Table 2, and visualizing them from a standard point of view. Note how the proposed approach improves reconstruction especially in regions where clear geometric structures from the architectural layout are present.

Table 3 Quantitative comparison on S3D-SD with Bernoulli and SIFT sparsity. We show our performance, compared to ground truth and other approaches, testing two different sparsity patterns: Bernoulli pattern with 1.97% of visible pixels, and SIFT detector pattern with 0.1 contrast, 5 edge threshold, and no more than 8k extracted features, thus resulting in 0.91% of visible pixels (see Section 4.1)

Method	S3D-SD/Bernoulli sparsity							S3D-SD / SIFT sparsity						
	MSE↓	MAE↓	RMSE↓	SSIM↑	δ_1 ↑	δ_2 ↑	δ_3 ↑	MSE↓	MAE↓	RMSE↓	SSIM↑	δ_1 ↑	δ_2 ↑	δ_3 ↑
GAENet [82]	0.093	0.410	0.161	0.149	0.465	0.748	0.885	0.093	0.410	0.161	0.149	0.465	0.748	0.885
packNet+SAN [35]	0.021	0.183	0.091	0.622	0.723	0.953	0.986	0.070	0.352	0.149	0.673	0.471	0.787	0.915
Ma et al. [81]	0.049	0.280	0.102	0.441	0.679	0.895	0.954	0.005	0.044	0.024	0.938	0.981	0.993	0.996
PENet [83]	0.036	0.248	0.109	0.416	0.629	0.894	0.969	0.040	0.259	0.118	0.499	0.557	0.859	0.960
NLSPN [18]	0.018	0.162	0.054	0.834	0.813	0.961	0.985	0.037	0.235	0.096	0.814	0.697	0.903	0.963
Huang et al. [36]	0.003	0.043	0.021	0.911	0.979	0.994	0.997	0.025	0.177	0.084	0.774	0.766	0.931	0.974
Our	**0.002**	**0.025**	**0.018**	**0.946**	**0.991**	**0.997**	**0.998**	**0.003**	**0.035**	**0.020**	**0.943**	**0.987**	**0.995**	**0.998**

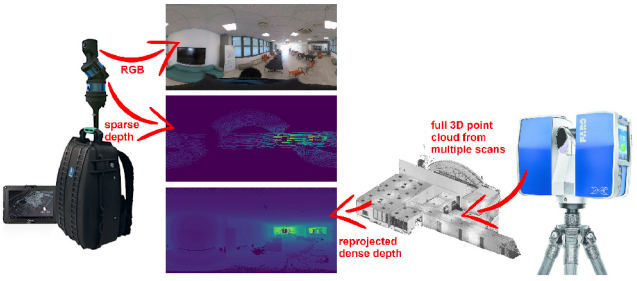


Fig. 7 Mobile RGB+LiDAR setup. To test our approach on a real-world panoramic RGB+LiDAR acquisition, we exploit a backpacked mobile scanner equipped with a full-view panoramic camera for the RGB capture and two LiDAR heads for sparse depth capture. Ground-truth dense depth for each pose is provided by reprojecting data coming from multiple poses of a static scanner.

light sensors (by a Bernoulli distribution [40]) and the pattern of an SIFT detector, emulating the typical sparse input that can be received from an SfM pipeline. Some qualitative examples with these patterns are illustrated in Fig. 5. Even in this situation our method demonstrates consistent performance, proving to be a versatile approach even when heterogeneous inputs vary.

Figure 8 summarizes the results of our experiments on the ability to cope with different levels of sparsity, tackling both purely visual input and seveal multi-beam LiDAR configurations. We illustrate our performance in comparison with the competitor method [36] that best performed in our experiments. We show the results on four different sparsity cases, ranging from no depth information to a full vertical FOV scan with 96 beams (38% pixel coverage, see Section 4.1 for details). For clarity, only the δ_1 metric is included in the graph, since the other metrics have, as shown in Table 2, a similar behavior.

The continuous lines illustrate the performance of the models when trained on the augmented S3D-SD dataset (i.e., the same setup of experiments in Table 2). The results indicate that our model, together with the proposed augmentation strategy, guarantees good performance for every type of sparsity. For the extreme case of a pure visual input, results are in-line with dedicated state-of-the-art [7, 8] approaches for panoramic depth estimation. On the other hand, the performance of the other approach [36] strongly depends on the number of available geometric samples. When training the model without data augmentation (dotted lines in the figure), but simply including in the training set the configuration used for testing, the performance of both models rapidly decays when moving away from the sampling used for training, even though our method remains superior at all sparsity levels. This experiment highlights how other methods can also benefit from our augmentation strategy, as it increases generalization without effects on use-case-specific performance.

For completeness, Table 4 summarizes our performance on *Matterport3D-SD* [26], compared to the results of other state-of-the-art approaches on the same benchmark [10, 26, 85, 86].

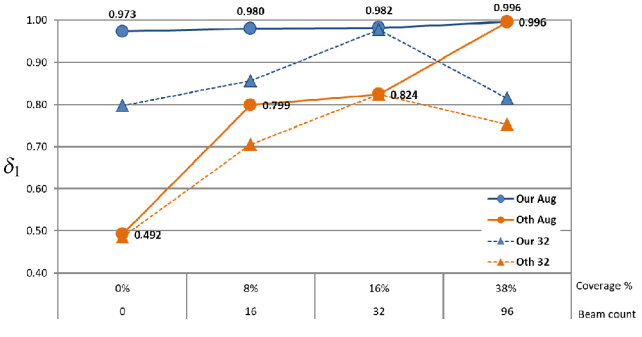


Fig. 8 Performance with variable sparsity level. The graph depicts the value of δ_1 as a function of input depth sparsity for our method and for the best competing method [36]. Continuous lines represent models trained with our augmentation strategy. Dotted lines show the same models but trained without augmentation (i.e., 40 degrees sparse coverage with 32 active beams).

Table 4 Quantitative comparison on *Matterport3D-SD*. We show our performance evaluated on standard metrics and compared to the recent state-of-the-art approaches on the indoor dataset provided by Zhang and Funkhouser [26]. We compare against the competitors best performance using their original perspective baselines, without considering additional error due to post-processing and stitching

Dataset	Method	MAE↓	RMSE↓	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$
M3D SD [26]	MRF [85]	0.618	1.675	0.651	0.780	0.856
	AD [86]	0.610	1.653	0.688	0.754	0.868
	Zhang and Funkhouser [26]	0.461	1.316	0.781	0.851	0.888
	Huang et al. [36]	0.342	1.092	0.850	0.911	0.936
	Xiong et al. [10]	0.462	0.866	0.863	0.930	0.942
	Our trained S3D-SD	0.464	0.803	0.834	0.908	0.942
	Our trained M3D SD [26]	**0.332**	**0.555**	**0.936**	**0.961**	**0.973**

As discussed in Section 4.1, such a benchmark presents a low-challenging sparsity distribution. The majority of the state-of-the-art solutions which adopted this benchmark are not end-to-end deep-learning networks, but hybrid pipelines [26], mainly focused on small-view perspective depth infilling [87]. Due to their hybrid nature, a direct computational complexity comparison is not feasible. It is also difficult, to create omnidirectional pipelines without major modifications to the code. In order to provide a uniform and fair evaluation in terms of prediction accuracy, we adopt here their official baselines and pre-trained models for perspective views, testing them with the original perspective viewports provided by Zhang and Funkhouser [26], and comparing the results for our code by extracting from the single equirectangular image we produce the perspective views required for testing. It should be noted that the exposed results for compared methods, thus, do not include the additional error due to the subsequent process of stitching the results necessary to obtain the final omnidirectional view, or other effects due to pipeline modifications in the case of adaptation to equirectangular projections.

We show our performance in the last two rows of Table 4. The bold row provides results obtained by training with *Matterport3D-SD* [26] training set, as for the compared methods, while, to also demonstrate our transfer learning capabilities, the other row summarizes the results obtained by inferring depth using the model trained with S3D-SD, with no fine-tuning. In both cases, our method provides consistent performance, well in line or outperforming other baselines that have been designed for this use-case. Although not directly comparable with the perspective results of the other pipelines (see Table 4), we show in Fig. 3 some qualitative results on the *Matterport3D-SD* dataset [26].

4.4 Ablation study

Our ablation experiments are presented in Table 5, with our configuration highlighted in bold. To test the key components of the approach, we use results obtained with S3D-SD, using for testing the LiDAR configuration with 3D beams (i.e., the same configuration of Table 2, 32 beams). The variations discussed in the ablation study are within the design space of our approach. For example, the use of gating in the encoder is essential for the model to work. Not using it leads to inconsistent results.

The first row of Table 5 presents a case without using some key-solutions of our model: multi-resolution features (MRF), asymmetric feature compression (AFC), multi-head self-attention feature refinement (MHSA), structural-similarity loss (SSIM), and data augmentation (AUG). Here we use the deeper layer of the residual feature encoder (see Section 3.1), and we perform a standard symmetric compression along the horizontal and vertical directions. This first case, which represents a common gated encoder–decoder scheme, demonstrates how this design is not sufficient to guarantee adequate performance without the subsequent contributions we have introduced. In the second row, we show the performance obtained by introducing multi-resolution features (MRF), which hallows gathering of information without using skip connections [35, 36]. Such a solution, without an efficient feature compression results in a significant increase of

Table 5 Ablation study performed on *S3D-SD*, using the LiDAR 32 beams confuguration for testing. MRF: multi-resolution features; AFC: asymmetric feature compression; MHSA: MHSA encoder; SSIM: SSIM loss; AUG: sparse data augmentation; LWGC: light-weight instead of standard gated convolution

MRF	AFC	MHSA	SSIM	AUG	LWGC	Param	GFLOPs	MAE	RMSE	δ_1
					✓	13.10	112.92	0.954	2.233	0.748
✓					✓	20.01	188.21	0.765	1.877	0.821
✓	✓				✓	20.01	43.15	0.312	1.384	0.877
✓	✓	✓			✓	22.11	38.16	0.121	0.084	0.951
✓	✓	✓	✓		✓	22.11	38.16	0.075	0.066	0.978
✓	✓	✓	✓	✓	✓	**22.11**	**38.16**	**0.038**	**0.022**	**0.982**
✓	✓	✓	✓	✓		31.86	61.62	0.035	0.021	0.985

computational complexity. The third row shows the benefits of asymmetric vertical compression (AFC), both in terms of lower computational complexity and in terms of accuracy. The fourth row shows instead the effects or using or not the MHSA module, without using specific losses or augmentation. It should be noted that MHSA feature refinement has a very low computational cost, but with a tangible increment of performance. The fifth and sixth rows show the increment in performance using augmentation, that limits overfitting.

At last, the seventh row shows that, in a setup using standard gated convolution instead of our light-weight choice (Section 3.1), performance is not improved despite the noticeable increment of computational cost.

4.5 Limitations and future works

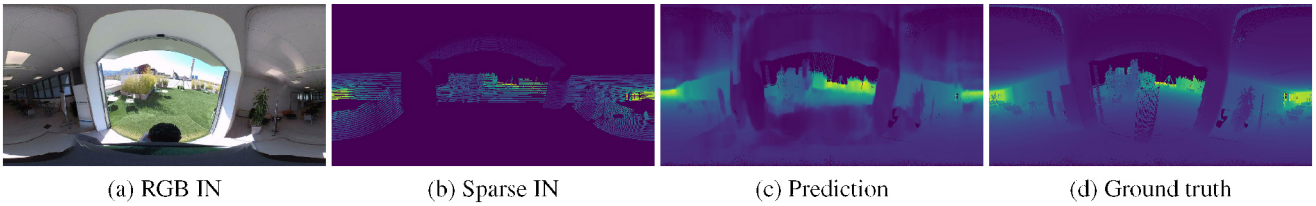
In our experiments, we experienced that the worst results are for datasets that do not closely match the assumptions of a closed indoor space, which are used in our design to construct an efficient network architecture (see Section 1). Figure 9 illustrates an example from a real-world capture. In this case, the sparse samples from the outdoor part, not properly masked, also negatively affect the reconstruction of the surrounding indoor parts.

It should be noted that the method has been specifically designed to exploit features in indoor structures. This behavior is mainly due to asymmetric feature compression and flattening of gravity-aligned indoor panoramic imaging (Section 3.2), which, in addition to providing efficient information gathering, allows the use of a transformer (MHSA) to retrieve the wide panoramic context. Without such indoor assumptions, compression, flattening, and self-attention are poorly effective. This design provides advantages in the prediction of depth for interior structures, as demonstrated by our results, while limiting the applicability of the method to scenes matching the assumptions.

Since such a domain-specific network design has shown to provide significant performance improvements with respect to more generic solutions, it is interesting to further extend this work by exploiting domain-specific constraints. One direction for future work would be to further exploit the indoor-specific design, e.g., by incorporating indoor-specific loss functions designed for architectural structures composed of large smooth surfaces, not necessarily planar, joining at possibly sharp edges [67]. Another direction would be, instead, to use the same concepts to design networks for other specific application contexts (e.g., outdoors, industrial plants), incorporating knowledge on plausible structures (e.g., presence of pipes) into the network representation and loss functions.

5 Conclusions

We have presented a novel end-to-end deep-learning solution for rapidly estimating a dense spherical depth map of an indoor environment starting from a single image and a sparse depth map. To realize a lightweight and efficient single-branch network, we combine and extend several technical solutions to offer a novel way to solve this specific problem. We adopted a residual encoder with a dynamic gating system to extract multi-resolution features from hybrid visual-geometric input. In order to efficiently gather such amount of information and to avoid onerous interconnections between encoder and decoder, we introduced a specific compression and feature flattening which exploits the characteristics of typical man-made environments and panoramic view. End-to-end training was instead carried out by introducing a data augmentation scheme capable of making it robust and versatile as the sparsity changes. As a result, our compact network outperforms in terms of speed and accuracy current solutions for color-guided sparse depth prediction and completion.



(a) RGB IN (b) Sparse IN (c) Prediction (d) Ground truth

Fig. 9 Bad case. Results on almost-outdoor environment. Sparse samples from outdoor part, not properly masked, negatively affect the whole reconstruction.

Availability of data and materials

The benchmarks presented are already based on publicly available datasets, that, where relevant, we have augmented with sparse depth information.

Funding

Giovanni Pintore and Enrico Gobbetti received funding from the Autonomous Region of Sardinia under project XDATA. Eva Almansa, Armando Sanchez, Giorgio Vassena, and Enrico Gobbetti received funding from the European Union's H2020 research and innovation programme under grant 813170 (EVOCATION).

Author contributions

Giovanni Pintore: Conceptualization, Methodology, Software, Validation, Writing - Original Draft, Writing - Review & Editing; Eva Almansa: Conceptualization, Methodology, Software, Validation, Data Curation, Writing - Original Draft; Armando Sanchez: Software, Validation, Writing - Original Draft; Giorgio Vassena: Supervision, Project administration, Funding acquisition, Writing - Original Draft; Enrico Gobbetti: Conceptualization, Methodology, Supervision, Project administration, Funding acquisition, Writing - Original Draft, Writing - Review & Editing.

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] Zollhöfer, M.; Stotko, P.; Görlitz, A.; Theobalt, C.; Nießner, M.; Klein, R.; Kolb, A. State of the art on 3D reconstruction with RGB-D cameras. *Computer Graphics Forum* Vol. 37, No. 2, 625–652, 2018.

[2] Pintore, G.; Mura, C.; Ganovelli, F.; Fuentes-Perez, L.; Pajarola, R.; Gobbetti, E. State-of-the-art in automatic 3D reconstruction of structured indoor environments. *Computer Graphics Forum* Vol. 39, No. 2, 667–699, 2020.

[3] Mertan, A.; Duff, D. J.; Unal, G. Single image depth estimation: An overview. *Digital Signal Processing* Vol. 123, 103441, 2022.

[4] Ming, Y.; Meng, X. Y.; Fan, C. X.; Yu, H. Deep learning for monocular depth estimation: A review. *Neurocomputing* Vol. 438, 14–33, 2021.

[5] Jokela, T.; Ojala, J.; Väänänen, K. How people use 360-degree cameras. In: Proceedings of the 18th International Conference on Mobile and Ubiquitous Multimedia, 1–10, 2019.

[6] Wang, F. E.; Yeh, Y. H.; Sun, M.; Chiu, W. C.; Tsai, Y. H. BiFuse: Monocular 360 depth estimation via bi-projection fusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 459–468, 2020.

[7] Sun, C.; Sun, M.; Chen, H. T. HoHoNet: 360 indoor holistic understanding with latent horizontal features. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2573–2582, 2021.

[8] Pintore, G.; Agus, M.; Almansa, E.; Schneider, J.; Gobbetti, E. SliceNet: Deep dense depth estimation from a single indoor panorama using a slice-based representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11531–11540, 2021.

[9] Lopez-Rodriguez, A.; Busam, B.; Mikolajczyk, K. Project to adapt: Domain adaptation for depth completion from noisy and sparse sensor data. In: *Computer Vision – ACCV 2020. Lecture Notes in Computer Science, Vol. 12622*. Ishikawa, H.; Liu, C. L.; Pajdla, T.; Shi, J. Eds. Springer Cham, 330–348, 2021.

[10] Xiong, X.; Xiong, H. P.; Xian, K.; Zhao, C.; Cao, Z. G.; Li, X. Sparse-to-dense depth completion revisited: Sampling strategy and graph construction. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12366*. Vedaldi, A.; Bischof, H.;Brox, T.; Frahm, J. M. Eds. Springer Cham, 682–699, 2020.

[11] Eigen, D.; Puhrsch, C.; Fergus, R. Depth map prediction from a single image using a multi-scale deep network. In: Proceedings of the 27th International Conference on Neural Information Processing Systems, Vol. 2, 2366–2374, 2014.

[12] Fu, H.; Gong, M.; Wang, C.; Batmanghelich, K.; Tao, D. Deep ordinal regression network for monocular depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2002–2011, 2018.

[13] Gan, Y. K.; Xu, X. Y.; Sun, W. X.; Lin, L. Monocular depth estimation with affinity, vertical pooling, and label enhancement. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11207*. Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 232–247, 2018.

[14] Yin, W.; Liu, Y. F.; Shen, C. H.; Yan, Y. L. Enforcing geometric constraints of virtual normal for

depth prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 5683–5692, 2019.

[15] Imran, S.; Long, Y. F.; Liu, X. M.; Morris, D. Depth coefficients for depth completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2438–2447, 2019.

[16] Qiu, J. X.; Cui, Z. P.; Zhang, Y. D.; Zhang, X. D.; Liu, S. C.; Zeng, B.; Pollefeys, M. DeepLiDAR: Deep surface normal guided depth prediction for outdoor scene from sparse LiDAR data and single color image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3308–3317, 2019.

[17] Huang, Y. K.; Liu, Y. C.; Wu, T. H.; Su, H. T.; Chang, Y. C.; Tsou, T. L.; Wang, Y.; Hsu, W. H. S^3: Learnable sparse signal superdensity for guided depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 16701–16711, 2021.

[18] Park, J.; Joo, K.; Hu, Z.; Liu, C. K.; Kweon, I. S. Non-local spatial propagation network for depth completion. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12358*. Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer Cham, 120–136, 2020.

[19] Eldesokey, A.; Felsberg, M.; Holmquist, K.; Persson, M. Uncertainty-aware CNNs for depth completion: Uncertainty from beginning to end. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 12011–12020, 2020.

[20] Ku, J.; Harakeh, A.; Waslander, S. L. In defense of classical image processing: Fast depth completion on the CPU. In: Proceedings of the 15th Conference on Computer and Robot Vision, 16–22, 2018.

[21] Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research* Vol. 32, No. 11, 1231–1237, 2013.

[22] New York University. NYU-Depth V2. 2012. Available at https://cs.nyu.edu/~sil berman/datasets/nyu depth v2.html

[23] Matterport. Matterport3D. 2017. Available at https://github.com/niessner/Matter port

[24] Stanford University. BuildingParser Dataset. 2017. Available at http://buildingpa rser.stanford.edu/ dataset.html

[25] Zheng, J.; Zhang, J. F.; Li, J.; Tang, R.; Gao, S. H.; Zhou, Z. H. Structured3D: A large photo-realistic dataset for structured 3D modeling. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12354*. Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer Cham, 519–535, 2020.

[26] Zhang, Y. D.; Funkhouser, T. Deep depth completion of a single RGB-D image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 175–185, 2018.

[27] Straub, J.; Whelan, T.; Ma, L. N.; Chen, Y. F.; Wijmans, E.; Green, S.; Engel, J. J.; Mur-Artal, R.; Ren, C.; Verma, S.; et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint* arXiv:1906.05797, 2019.

[28] Zioulis, N.; Karakottas, A.; Zarpalas, D.; Alvarez, F.; Daras, P. Spherical view synthesis for self-supervised 360° depth estimation. In: Proceedings of the International Conference on 3D Vision, 690–699, 2019.

[29] Xian, W. Q.; Li, Z. Q.; Snavely, N.; Fisher, M.; Eisenman, J.; Shechtman, E. UprightNet: Geometry-aware camera orientation estimation from single images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 9973–9982, 2019.

[30] Jung, R.; Lee, A. S. J.; Ashtari, A.; Bazin, J. C. Deep360Up: A deep learning-based approach for automatic VR image upright adjustment. In: Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces, 1–8, 2019.

[31] Davidson, B.; Alvi, M. S.; Henriques, J. F. 360° camera alignment via segmentation. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12373*. Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer Cham, 579–595, 2020.

[32] Sun, C.; Hsiao, C. W.; Sun, M.; Chen, H. T. HorizonNet: Learning room layout with 1D representation and pano stretch data augmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 1047–1056, 2019.

[33] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, 6000–6010, 2017.

[34] Yi, Z. L.; Tang, Q.; Azizi, S.; Jang, D.; Xu, Z. Contextual residual aggregation for ultra high-resolution image inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7505–7514, 2020.

[35] Guizilini, V.; Ambrus, R.; Burgard, W.; Gaidon, A. Sparse auxiliary networks for unified monocular depth prediction and completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11073–11083, 2021.

[36] Huang, Y. K.; Wu, T. H.; Liu, Y. C.; Hsu, W. H. Indoor depth completion with boundary consistency and self-attention. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop, 1070–1078, 2019.

[37] Yang, Y. C.; Wong, A.; Soatto, S. Dense depth posterior (DDP) from single image and sparse range. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3348–3357, 2019.

[38] He, K. M.; Zhang, X. Y.; Ren, S. Q.; Sun, J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778, 2016.

[39] Yu, J. H.; Lin, Z.; Yang, J. M.; Shen, X. H.; Lu, X.; Huang, T. Free-form image inpainting with gated convolution. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 4470–4479, 2019.

[40] Ma, F. C.; Karaman, S. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In: Proceedings of the IEEE International Conference on Robotics and Automation, 4796–4803, 2018.

[41] Kujiale.com. Structured3D Data. 2019.

[42] Laina, I.; Rupprecht, C.; Belagiannis, V.; Tombari, F.; Navab, N. Deeper depth prediction with fully convolutional residual networks. In: Proceedings of the 4th International Conference on 3D Vision, 239–248, 2016.

[43] Liu, F. Y.; Shen, C. H.; Lin, G. S. Deep convolutional neural fields for depth estimation from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5162–5170, 2015.

[44] Wang, P.; Shen, X. H.; Lin, Z.; Cohen, S.; Price, B.; Yuille, A. Towards unified depth and semantic prediction from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2800–2809, 2015.

[45] Cao, Y.; Wu, Z. F.; Shen, C. H. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for –Video Technology* Vol. 28, No. 11, 3174–3182, 2018.

[46] Xu, D.; Wang, W.; Tang, H.; Liu, H.; Sebe, N.; Ricci, E. Structured attention guided convolutional neural fields for monocular depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3917–3925, 2018.

[47] Godard, C.; Mac Aodha, O.; Brostow, G. J. Unsupervised monocular depth estimation with left-right consistency. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 6602–6611, 2017.

[48] Zhan, H. Y.; Garg, R.; Weerasekera, C. S.; Li, K. J.; Agarwal, H.; Reid, I. M. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 340–349, 2018.

[49] Ji, P.; Li, R. Z.; Bhanu, B.; Xu, Y. MonoIndoor: Towards good practice of self-supervised monocular depth estimation for indoor environments. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 12767–12776, 2021.

[50] Zioulis, N.; Karakottas, A.; Zarpalas, D.; Daras, P. OmniDepth: Dense depth estimation for indoors spherical panoramas. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11210.* Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 453–471, 2018.

[51] Cheng, H. T.; Chao, C. H.; Dong, J. D.; Wen, H. K.; Liu, T. L.; Sun, M. Cube padding for weakly-supervised saliency prediction in 360° videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 1420–1429, 2018.

[52] Su, Y. C.; Grauman, K. Learning spherical convolution for fast features from 360° imagery. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, 529–539, 2017.

[53] Tateno, K.; Navab, N.; Tombari, F. Distortion-aware convolutional filters for dense prediction in panoramic images. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11220.* Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 732–750, 2018.

[54] Payen de La Garanderie, G.; Atapour Abarghouei, A.; Breckon, T. P. Eliminating the blind spot: Adapting 3D object detection and monocular depth estimation to 360° panoramic imagery. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11217.* Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 812–830, 2018.

[55] Su, Y. C.; Grauman, K. Kernel transformer networks for compact spherical convolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 9434–9443, 2019.

[56] Liao, Y.; Xie, J.; Geiger, A. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 45, No. 3, 3292–3310, 2023.

[57] Eldesokey, A.; Felsberg, M.; Khan, F. S. Confidence propagation through CNNs for guided sparse depth regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 42, No. 10, 2423–2436, 2020.

[58] Tang, J.; Tian, F. P.; Feng, W.; Li, J.; Tan, P. Learning guided convolutional network for depth completion. *IEEE Transactions on Image Processing* Vol. 30, 1116–1129, 2021.

[59] Van Gansbeke, W.; Neven, D.; De Brabandere, B.; Van Gool, L. Sparse and noisy LiDAR completion with RGB guidance and uncertainty. In: Proceedings of the 16th International Conference on Machine Vision Applications, 1–6, 2019.

[60] Lee, S.; Lee, J.; Kim, D.; Kim, J. Deep architecture with cross guidance between single image and sparse LiDAR data for depth completion. *IEEE Access* Vol. 8, 79801–79810, 2020.

[61] Oh, C.; Cho, W.; Chae, Y.; Park, D.; Wang, L.; Yoon, K. J. BIPS: Bi-modal indoor panorama synthesis via residual depth-aided adversarial learning. In: *Computer Vision – ECCV 2022. Lecture Notes in Computer Science, Vol. 13676*. Avidan, S.; Brostow, G.; Cissé, M.; Farinella, G. M.; Hassner, T. Eds. Springer Cham, 352–371, 2022.

[62] Silberman, N.; Hoiem, D.; Kohli, P.; Fergus, R. Indoor segmentation and support inference from RGBD images. In: *Computer Vision – ECCV 2012. Lecture Notes in Computer Science, Vol. 7576*. Fitzgibbon, A.; Lazebnik, S.; Perona, P.; Sato, Y.; Schmid, C. Eds. Springer Berlin Heidelberg, 746–760, 2012.

[63] Cheng, X. J.; Wang, P.; Zhou, Y. Q.; Guan, C. Y.; Yang, R. G. Omnidirectional depth extension networks. In: Proceedings of the IEEE International Conference on Robotics and Automation, 589–595, 2020.

[64] Yu, J. H.; Lin, Z.; Yang, J. M.; Shen, X. H.; Lu, X.; Huang, T. S. Generative image inpainting with contextual attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5505–5514, 2018.

[65] Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Lecture Notes in Computer Science, Vol. 9351*. Navab, N.; Hornegger, J.; Wells, W.; Frangi, A. Eds. Springer Cham, 234–241, 2015.

[66] Liu, R. Y.; Zhang, G. D.; Wang, J. M.; Zhao, S. W. Cross-modal 360° depth completion and reconstruction for large-scale indoor environment. *IEEE Transactions on Intelligent Transportation Systems* Vol. 23, No. 12, 25180–25190, 2022.

[67] Pintore, G.; Almansa, E.; Agus, M.; Gobbetti, E. Deep3DLayout: 3D reconstruction of an indoor layout from a spherical panoramic image. *ACM Transactions on Graphics* Vol. 40, No. 6, Article No. 250, 2021.

[68] Gkitsas, V.; Sterzentsenko, V.; Zioulis, N.; Albanis, G.; Zarpalas, D. PanoDR: Spherical panorama diminished reality for indoor scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 3711–3721, 2021.

[69] Liu, G. L.; Reda, F. A.; Shih, K. J.; Wang, T. C.; Tao, A.; Catanzaro, B. Image inpainting for irregular holes using partial convolutions. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11215*. Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 89–105, 2018.

[70] Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. *arXiv preprint* arXiv:1511.07122, 2015.

[71] Zheng, C. X.; Cham, T. J.; Cai, J. F. Pluralistic image completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 1438–1447, 2019.

[72] Clevert, D. A.; Unterthiner, T.; Hochreiter, S. Fast and accurate deep network learning by exponential linear units (ELUs). *arXiv preprint* arXiv:1511.07289, 2015.

[73] Guizilini, V.; Li, J.; Ambrus, R.; Pillai, S.; Gaidon, A. Robust semi-supervised monocular depth estimation with reprojected distances. In: Proceedings of the Conference on Robot Learning, 503–512, 2020.

[74] Morales, J.; Plaza-Leiva, V.; Mandow, A.; Gomez-Ruiz, J. A.; Serón, J.; García-Cerezo, A. Analysis of 3D scan measurement distribution with application to a multi-beam lidar on a rotating platform. *Sensors* Vol. 18, No. 2, 395, 2018.

[75] Wu, T.; Fu, H.; Liu, B. K.; Xue, H. Z.; Ren, R. K.; Tu, Z. M. Detailed analysis on generating the range image for LiDAR point cloud processing. *Electronics* Vol. 10, No. 11, 1224, 2021.

[76] You, Y. R.; Wang, Y.; Chao, W. L.; Garg, D.; Pleiss, G.; Hariharan, B.; Campbell, M.; Weinberger, K. Q. Pseudo-LiDAR++: Accurate depth for 3D object detection in autonomous driving. *arXiv preprint* arXiv:1906.06310, 2019.

[77] Lambert-Lacroix, S.; Zwald, L. The adaptive BerHu penalty in robust regression. *Journal of Nonparametric Statistics* Vol. 28, No. 3, 487–514, 2016.

[78] Wang, Z.; Bovik, A. C.; Sheikh, H. R.; Simoncelli, E. P. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing* Vol. 13, No. 4, 600–612, 2004.

[79] Li, Y. W.; Dai, S. M.; Shi, Y.; Zhao, L. L.; Ding, M. H. Navigation simulation of a mecanum wheel mobile robot based on an improved A* algorithm in Unity3D. *Sensors* Vol. 19, No. 13, 2976, 2019.

[80] Kingma, D. P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint* arXiv:1412.6980, 2014.

[81] Ma, F. C.; Cavalheiro, G. V.; Karaman, S. Self-supervised sparse-to-dense: Self-supervised depth completion from LiDAR and monocular camera. In: Proceedings of the International Conference on Robotics and Automation, 3288–3295, 2019.

[82] Du, W. C.; Chen, H.; Yang, H. Y.; Zhang, Y. Depth completion using geometry-aware embedding. In: Proceedings of the International Conference on Robotics and Automation, 8680–8686, 2022.

[83] Hu, M.; Wang, S. L.; Li, B.; Ning, S. Y.; Fan, L.; Gong, X. J. PENet: Towards precise and efficient image guided depth completion. In: Proceedings of the IEEE International Conference on Robotics and Automation, 13656–13662, 2021.

[84] Eldesokey, A.; Felsberg, M.; Khan, F. S. Confidence propagation through CNNs for guided sparse depth regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 42, No. 10, 2423–2436, 2020.

[85] Harrison, A.; Newman, P. Image and sparse laser fusion for dense scene reconstruction. In: *Field and Service Robotics. Springer Tracts in Advanced Robotics, Vol. 62*. Howard, A.; Iagnemma, K.; Kelly, A. Eds. Springer Berlin Heidelberg, 219–228, 2010.

[86] Liu, J. Y.; Gong, X. J. Guided depth enhancement via anisotropic diffusion. In: *Advances in Multimedia Information Processing – PCM 2013. Lecture Notes in Computer Science, Vol. 8294*. Huet, B.; Ngo, C. W.; Tang, J.; Zhou, Z. H.; Hauptmann, A. G.; Yan, S. Eds. Springer Cham, 408–417, 2013.

[87] Xiong, X. H.; Huber, D. Using context to create semantic 3D models of indoor environments. In: Proceedings of the British Machine Vision Conference, 2010.

Giovanni Pintore is a senior researcher engineer in the Visual and Data-intensive Computing Group at the Center for Advanced Studies, Research, and Development in Sardinia (CRS4), Italy. He holds his Laurea (M.Sc.) degree (2002) in electronics engineering from the University of Cagliari, Italy. His research interests cover many areas of visual computing and are currently focusing on data-driven solutions to challenging graphics and vision problems.

Eva Almansa is an early stage researcher/Ph.D. student of the EVOCATION project, hosted in the Visual and Data-intensive Computing Group of the Center for Advanced Studies, Research, and Development in Sardinia (CRS4), Italy. She holds her bachelor of science in computer science and artificial intelligence (2015) and her master degree in data science (2017) from the University of Granada, Spain. Her research is focused on structured indoor reconstruction and exploration.

Armando Sanchez is an early stage researcher/Ph.D. student of the EVOCATION project, hosted at Gexcel, Italy. He holds his bachelor (2017) and master (2019) degrees in industrial engineering from Universitat Politècnica de València, Spain. His research is focused on developing tools for mobile 3D data capture.

Giorgio Vassena is an associate professor of geomatics and surveying at the Department of Civil Engineering, Architecture, Land and Environment and Mathematics of the University of Brescia (Italy). He holds his engineering and Ph.D. degrees in civil engineering from Politecnico di Milano. His main research topics are related to indoor mobile mapping technologies, GNSS and structures monitoring, geospatial applications.

Enrico Gobbetti is the director of Visual and Data-intensive Computing Group at the Center for Advanced Studies, Research, and Development in Sardinia (CRS4), Italy. He holds his engineering (1989) and Ph.D. (1993) degrees in computer science from the Swiss Federal Institute of Technology in Lausanne (EPFL). Prior to joining CRS4, he held research and teaching positions at EPFL, UMBC, and NASA. Enrico's research spans many areas of visual and data-intensive computing, with a focus on scalable solutions.