SliceNet: deep dense depth estimation from a single indoor panorama using a slice-based representation

Giovanni Pintore
Visual Computing, CRS4, Italy
giovanni.pintore@crs4.it

Marco Agus
CSE, HBKU, Doha, Qatar
magus@hbku.edu.qa

Eva Almansa
Visual Computing, CRS4, Italy
evaalmansa@crs4.it

Jens Schneider
CSE, HBKU, Doha, Qatar
jeschneider@hbku.edu.qa

Enrico Gobbetti
Visual Computing, CRS4, Italy
enrico.gobbetti@crs4.it

Abstract

We introduce a novel deep neural network to estimate a depth map from a single monocular indoor panorama. The network directly works on the equirectangular projection, exploiting the properties of indoor 360° images. Starting from the fact that gravity plays an important role in the design and construction of man-made indoor scenes, we propose a compact representation of the scene into vertical slices *of the sphere, and we exploit long- and short-term relationships among slices to recover the equirectangular depth map. Our design makes it possible to maintain high-resolution information in the extracted features even with a deep network. The experimental results demonstrate that our method outperforms current state-of-the-art solutions in prediction accuracy, particularly for real-world data.*

1. Introduction

Understanding the 3D layout of an indoor scene from images is a crucial task in many domains [45, 23, 24]. Fast depth estimation from single images is a fundamental sub-problem, as associating metric information to visual data is paramount for a variety of applications, including mobile Augmented Reality platforms, indoor mapping, autonomous navigation, 3D reconstruction, and scene understanding.

Since estimation of depth from single images is inherently ambiguous, all solutions must rely on prior information to guide reconstruction towards plausible architectural shapes that fit the input. In this context, we have recently seen an extraordinary development of data-driven methods that learn these priors from example data.

Early approaches were designed for a camera with a conventional limited field-of-view (FoV) (e.g., FCRN[14]). In recent years, however, 360° capture has emerged as a very appealing solution, since it provides the quickest and most complete single-image coverage and is supported by a wide variety of professional and consumer capture devices that make acquisition fast and cost-effective [37]. Since adapting monocular depth estimation models designed for traditional images to 360° depth estimation has been shown to produce sub-optimal results [44], specific 360° solutions have been recently introduced. In this context, many recent works [31, 44, 17] have adapted perspective depth estimation methods to omnidirectional imagery by proposing various types of distortion-aware convolution filters. However, few of them have explored the large-FoV nature provided by 360° images, which can provide, in one shot, the full-geometric context of an indoor scene [41].

In this work, we introduce a novel deep neural network solution, called *SliceNet*, which predicts the depth map of an indoor 360° image leveraging the characteristics of a gravity-aligned equirectangular projection of an interior scene. Since gravity plays an important role in the design and construction of interior environments, world-space vertical and horizontal features have different characteristics in most, if not all, man-made environments. Our network design starts from the assumption that capture of the scene through an equirectangular image is aligned to the gravity vector (i.e., camera is placed on an horizontal-ground plane), too, and, thus, it is rational to assume that gravity-aligned processing of images can directly exploit gravity-aligned world-space features [3]. In our network, an input equirectangular image is partitioned into vertical *slices* by performing a contractive encoding to reduce the input tensor only along the vertical direction, resulting in a compact and flattened sequence of slices made of a set of features. To preserve global information, we perform slicing over four different resolution levels, concatenating the result at the end (Sec. 3). This sequential representation enables the use of a convolutional long short-term memory (LSTM) network [26] to recover, with low computational

overhead, long- and short-term spatial relationships among slices. Decoding proceeds symmetrically with respect to encoding, thereby increasing only the vertical resolution of the feature map, until the target resolution is reached (Fig. 1(a)).

Our contributions are summarized as follows:

- We introduce a slice-based representation of an omnidirectional image that directly exploits the characteristics of the equirectangular projection of an indoor scene, without the need for distortion-aware convolution and transformation [44, 33], multi-branch architectures [33, 11] or additional information and priors [11]. Our representation based on vertical slices is very robust, as demonstrated by the important advantage in performance achieved in real-world cases (e.g., Stanford2D3D [27] and Matterport3D [19]), where a large area around the poles of the panorama is not acquired by the instrument (see Sec. 5.2 for details).
- We specialize and refine feature flattening, which has proven to be effective to regress one-dimensional tensors [30], for bi-dimensional depth encoding. In particular, we introduce an asymmetric contraction of the input tensor based on vertical slicing at different resolutions, so that the resulting feature map is flattened along a single direction (in our case, the sphere horizon), and we merge slices at different resolutions, so as to exploit deeper levels with larger receptive fields to capture global information, while at the same time exploiting higher resolution layers to preserve high-frequency details (Sec. 3). Our ablation study (Sec. 5.3) demonstrates the advantages of our approach.
- We introduce, for depth estimation from a single image, a LSTM multi-layer module to effectively recover long and short term spatial relationships between slices in the presence of a large number of features per slice due to the concatenation of multiscale representations. With this architectural choice, the decoder is simple and follows the same multi-layer scheme of the encoder with a vertical upsampling rather than a vertical reduction. We do not need, in particular, the chaining of up-projection blocks [10], making it easier to scale the method to different input resolutions. The ablation study (Sec. 5.3) confirms the benefits of the method by comparing different decoder configurations with or without LSTM and chaining up-projection blocks.

We tested our network on both synthetic and real datasets [27, 19, 44, 43, 42]. Our experimental results (Sec. 5) demonstrate that our method outperforms current state-of-the-art methods [14, 44, 33] in prediction accuracy, especially when working on real-world scenes. Exploiting gravity alignment leads to an efficient network structure, without significant limitations on the applicability of the approach. As mentioned, gravity-aligned capture is a very common setup, and, as determined by our tests, Sec. 5.3, all the public 3D indoor datasets commonly used for training and testing reconstruction solutions, both synthetic [43, 42] and real [27, 19], appear to have very small orientation deviations. Even in cases where this assumption is not verified at capture time, several orthogonal solutions exist to gravity-rectify images in a pre-processing step (e.g., [34, 12, 3]), simplifying the practical application of gravity-oriented methods. Moreover, as demonstrated by our ablation study (Sec. 5.3), our method is robust to small variations of the inclination.

2. Related work

Depth estimation from monocular input and 3D reconstruction of indoor environments are fundamental computer vision problem, which have recently attracted renewed interest with the emergence of deep learning techniques. A full review is beyond the scope of this paper. Here, we focus on the solutions most closely related to our work.

Depth from perspective images. Learning-based monocular depth estimation was introduced over a decade ago (e.g., Make3D [25]). The emergence of deep learning, as well as the availability of large-scale 3D datasets, has contributed to significant performance improvements. Eigen et al. [6] were the first to use CNNs for regressing dense depth maps from a single image in a two-scale architecture, where the first stage—based on the *AlexNet* feature encoder—produces a coarse output and the second stage refines the prediction. Their work was later extended to additionally predict normals and labels with a deeper and more discriminative model, based on *VGG* features encoder, and a three-scale architecture for further refinement [5]. Laina et al. [14], instead, combined *ResNet* [10] with an up-projection module for upsampling. They also proposed the reverse Huber [15] loss to improve depth estimation. This baseline, named *FCRN*, has become of common use even in the case of panoramic images. Lee et al. [16], instead, predicted depth from several cropped images combined in the Fourier domain. Conditional random fields (CRF) are also often exploited to refine prediction [18, 21, 1, 35]. Fu et al. [7] use dilated convolution to increase the receptive field and apply the ordinal regression loss to preserve the spatial relation among neighboring classes. Unsupervised training for depth estimation is instead performed using photometric loss [8, 40]. Directly adopting monocular depth estimation solutions for 360° depth estimation produces sub-optimal results [44], since several characteristics of panoramic images are not exploited, e.g., the fact that they capture global information which can improve reasoning.

Depth from a single omnidirectional image. One of the main limitation of single-image methods lies, in fact, in the restricted field of view (FOV) of conventional perspective images, which inevitably results in a limited geometric context [41]. With the emergence of consumer-level 360° cameras, a wide indoor context can now be captured with

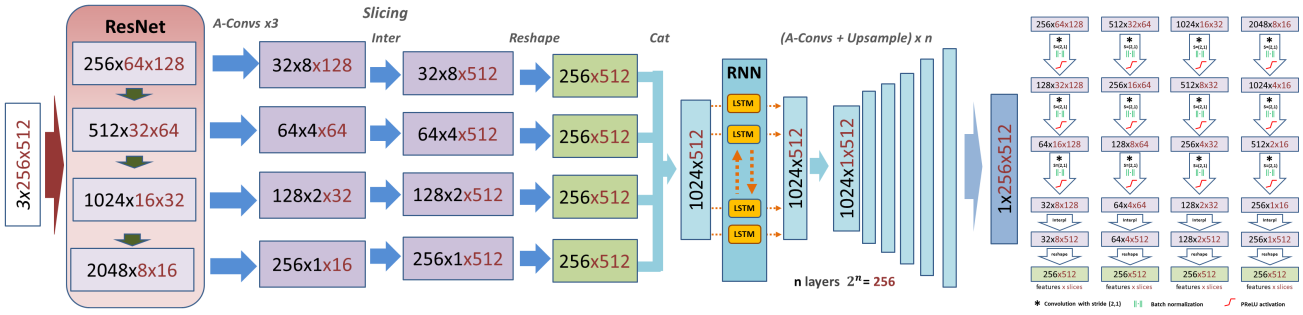


Figure 1. **Network architecture.** Our architecture is scalable with respect to the input resolution. In Fig. 1(a), to simplify comparison with other methods, we show an example with an input image having size $3 \times 256 \times 512$. A *ResNet50* encoder [10] extracts four layers at different resolutions. From each resolution layer we obtain a sliced feature map of 256×512 (purple blocks in Fig 1(a), details in Fig. 1(b)). By concatenating the resulting four layers we obtain a single bottleneck with 512 slices and 1024 features, which is refined using a RNN scheme (cyan blocks). The decoder proceeds symmetrically, producing a depth map at the same input image resolution.

one or at least few shots. As a result, much of the research on reconstruction of indoors from sparse imagery is now focused in this direction, even for directly recovering the room layout under specific conditions [46, 36, 30, 22]. In the specific case of depth estimation, a first approach is to convert an omnidirectional image into a cubemap [2], both to deal with the distortion of equirectangular projection and to take advantage of the consolidated monocular estimation techniques mentioned above. To make the network aware of the distortion, spherical convolution methods have been also proposed [29, 31, 20, 28]. Following this trend, Zioulis et al. [44] adopted the spherical layers of Su et al. [29] for depth estimation in the indoor environment, and proposed a large-scale synthetic dataset consisting of 22,096 re-rendered images from four existing datasets [43]. Wang et al. [33], which at the time of this writing provide the best results in terms of accuracy, proposed a two-branch network, respectively for the equirectangular and the cubemap projection, based on a distortion-aware encoder [44] and the FCRN decoder [14]. Recently, several orthogonal works [4, 38, 11, 39] exploit the correlation among depth, room layout, and semantics to improve prediction. Such promising solutions require much additional input for training (e.g., annotated room layout, normal maps and semantic segmentation), and exploit a depth estimation baseline based on one of the above-cited approaches. All the above methods bring back the spherical projection to a standard projection to apply encoding-decoding schemes designed for conventional images (e.g., FCRN [14]), while we introduce a scheme designed for equirectangular projections of indoor scenes.

3. Network architecture

Almost all CNNs for this task follow an encoder-decoder architecture [14]. Such a structure contains a contractive encoding part that progressively decreases the input image resolution through a series of convolutions and pooling operations, giving higher-level neurons large receptive fields, thus capturing more global information. As the target depth map is a high resolution image, the decoder regresses to the desired output by upscaling this representation. Our work introduces several important novelties in this structure.

Figure 1(a) illustrates the structure of our network for a 256×512 input. Note that our architecture is scalable with respect to the input resolution. In Sec.5 we provide results with the same input sizes adopted by recent state-of-the-art methods [14, 44, 33], including 512×1024 resolution.

The first part of our network is devoted to extracting relevant low/mid/high-level features from the input tensor. To do that, we exploit *ResNet-50*, a deep neural network that supports, through a residual learning framework, the training of very deep networks without degradation problems [10]. Differently from other approaches [14, 44, 33], we exploit not only the deepest layer of ResNet, but the last four layers, processing them in parallel, in order to build a multi-resolution spatial representation, discussed in detail below. Following our gravity-aligned model, we recover from these 4 layers (Fig. 1(a), red), 4 representative slice layers (Fig. 1(a), green), having all the same size of 256×512 (i.e., 256 features for 512 slices). Figure 1(b) illustrates how we produce the sliced representation from the ResNet layer. First, we reduce the vertical dimension by a factor of 8 through an asymmetric convolution module with stride $(2, 1)$ (*A-Conv*), applied 3 times, contains a 2D convolution, a batch normalization module and a Parametric Rectified Linear Unit [9] $\mathrm{PReLU}(x) := \max(0, x) + a * \min(0, x)$, where a is the coefficient of leakage learned during training. We selected PReLU instead of commonly adopted ReLU and Leaky-ReLU to minimize the vanishing gradient problems that are common in depth estimation. This kind of adaptive activation leads to convergence even on datasets with very different characteristics (e.g., real-world acquisition

with missing parts or synthetic rendering whih high levels of noise). Sliced encoding is then completed by horizontally interpolating each feature map to have the same number of slices (i.e., 512), and by vertically reshaping the features to the target size (i.e., 256).

Finally, the for layers are concatenated in a single sequence (i.e., 1024×512), obtaining 1024 features for each of the 512 vertical slices of the input sphere. In this way, we obtain a bottleneck representation that exploits deeper levels with larger receptive fields to capture global information, and higher resolution layers to preserve high-frequency details.

It should be noted that both indoor scenes and equirectangular projections have particular properties that we exploit in our design. For example, vertical lines are very common in the scene, and are practically not deformed in the projection while the horizontal ones are more so. Because of these characteristics, we expect each slice sequence along the dominant vertical direction be related to the others by both short-term and long-term spatial dependencies [32, 30, 22]. Thus, we start our decoder by feeding such a sequence to a RNN multilayer block [26]. In our case, we use a bi-directional LSTM (long-short term memory) having 512 hidden layers, which outputs a timestep of size 2×512 for each of the 512 slices, so that the final output is a feature map having the same size of the RNN block input, i.e., 1024×512. Once reshaped to $1024 \times 1 \times 512$, this flattened representation can be upsampled to the desidered output size (i.e., $1 \times 256 \times 512$) by following steps symmetrical to those used for encoding reduction. Actually, thanks to the flattened encoding and RNN features refinement, our network does not require the chaining of skipping up-projection blocks for upsampling, such as FCRN [14], also common in other recent works [33]. Our decoder, instead, consists of n layers, where for each layer we perform an upsampling of a factor of two of the height only, followed by a convolutional module *A-Conv* identical to that of the reduction phase (2D convolution and PReLU activation), but with stride (1,1). In the example of Fig. 1(a), the decoder consists of $n = 8$ layers, in order to achieve the targeted vertical resolution (i.e., $2^n = 256$), and the resulting map is a tensor of $1 \times 256 \times 512$ representing the depth prediction for each of the input pixels. We also tested different upsampling modules adapted to our data encoding, (e.g., FCRN [14]) but experiencing lower performance, given our particular slice-based model. Numerical details are exposed in the ablation study in Sec. 5.3.

4. Loss function and training strategy

Similarly to other recent state-of-the-art solutions (e.g., BiFuse [33]), we build our objective function on top of the robust *Adaptive Reverse Huber Loss (BerHu)* [15]:

$$B_c(e) := \begin{cases} |e| & |e| \leq c \\ \frac{e^2+c^2}{2c} & |e| > c \end{cases} \quad (1)$$

where e is the error term and the parameter c determines where to switch from L1 to L2. In order to set the c value adaptively, we follow the same approach of Laina et al. [14], so that c is set, in every gradient step, to 20% of the maximal error of the current batch. When applied to the depth maps, $e = D_{ij} - D^*_{ij}$ at each pixel (i, j), where D and D^* are, respectively, the predicted and the ground-truth depth maps. Since one of the typical problems encountered in predicting depths using convolutional networks is the loss of small details [14, 44], which is particularly noticeable when dealing with higher resolution images, we introduce an additional term by applying *BerHu* also to the gradient components obtained by convolving the maps with Sobel filters of width 3 to approximate the horizontal derivatives $\nabla_x D$ and $\nabla_x D^*$ and the vertical ones $\nabla_y D$ and $\nabla_y D^*$. Consequently, the full loss function L that guides our training is:

$$\begin{aligned} L_{c_1,c_2}(D, D^*) = & B_{c_1}(D - D^*) + \\ & B_{c_2}(\nabla_x D - \nabla_x D^*) + \\ & B_{c_2}(\nabla_y D - \nabla_y D^*) \end{aligned} \quad (2)$$

With a little abuse of notation, we intend the application of the function to the map as the sum of results on each individual pixel. The parameter c that determines the shape of each function B_c is computed at each batch independently for the depth term (c_1) and the two gradient terms (i.e., c_2 is independent from c_1 and shared for the x and y gradient terms). Moreover, in order to gracefully handle large areas with missing samples common in real-world data (e.g., the upper and lower parts of the hemisphere are not sampled by the instrument, as in Matterport [19]), we take the common approach [44] of ignoring errors on missing areas with a per-pixel binary mask.

In all experiments, we obtain the best performance when training with the loss in Eq. 2, even compared to other robust solutions [44], experiencing a noticeable difference when training and comparing with real-world datasets [27, 19], which contain noticeable amounts of noise. The gradient-based component improves image sharpening, as shown in the comparison presented in Sec. 5.3 and Fig. 5.

5. Implementation and results

Our approach is implemented using PyTorch 1.5.1 and has been tested on a large variety of indoor scenes. Source code and models will be made available to the public.

In this paper, we report results obtained on four publicly available datasets [27, 19, 43, 42] to facilitate comparison. These benchmarks were also adopted by the recent state-of-the-art works [14, 44, 33] comparable with our method. *Matterport3D* [19] and *Stanford2D-3D-S* [27] act as real-world examples. Similarly to Wang et al. [33], we used their official splitting and a resolution of 512×1024. *360D* [43] offers instead a synthetic benchmark. It contains 35,977 panora-

Table 1. **Quantitative performance on real and virtual world datasets.** We show our performance evaluated on standard metrics and compared to the recent state-of-the-art approaches. In all cases our approach outperforms the competition.

Dataset	Method	MRE	MAE	RMSE	RMSE log	δ_1	δ_2	δ_3
Stanford2D3D	FCRN [14]	0.1837	0.3428	0.5774	0.1100	0.7230	0.9207	0.9731
	OmniDepth [44]	0.1996	0.3743	0.6152	0.1212	0.6877	0.8891	0.9578
	BiFuse [33]	0.1209	0.2343	0.4142	0.0787	0.8660	0.9580	0.9860
	Our	**0.0744**	**0.1048**	**0.1214**	**0.0207**	**0.9031**	**0.9723**	**0.9894**
Matterport3D	FCRN [14]	0.2409	0.4008	0.6704	0.1244	0.7703	0.9174	0.9617
	OmniDepth [44]	0.2901	0.4838	0.7643	0.1450	0.6830	0.8794	0.9429
	BiFuse [33]	0.2048	0.3470	0.6259	0.1134	0.8452	0.9319	0.9632
	Our	**0.1764**	**0.3296**	**0.6133**	**0.1045**	**0.8716**	**0.9483**	**0.9716**
360D	FCRN [14]	0.0699	0.1381	0.2833	0.0473	0.9532	0.9905	0.9966
	OmniDepth [44]	0.0931	0.1706	0.3171	0.0725	0.9092	0.9702	0.9851
	BiFuse [33]	0.0615	0.1143	0.2440	0.0428	0.9699	0.9927	**0.9969**
	Our	**0.0467**	**0.1134**	**0.1323**	**0.0212**	**0.9788**	**0.9952**	**0.9969**

mas rendered by path-tracing scenes from two synthetic datasets (*SunCG* and *SceneNet*) and two realistic datasets (*Stanford2D3D* and *Matterport3D*). In this case, we adopted the splitting provided by Zioulis et al. [44] and a resolution of 256×512, which is a common baseline for many approaches [14, 44, 33]. At the time of this writing, the original *SunCG* data is no longer available for downloading due to legal reasons. Additionally, we present our performance on the recent *Structured3D* synthetic dataset [42] to support ablation and gravity-alignment robustness studies (Sec. 5.3).

5.1. Experimental setup and timing performance

We trained the network using the Adam optimizer [13] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, on four NVIDIA RTX 2080Ti GPUs (11GB VRAM) with a batch size of 8 and a learning rate of 0.0001 for real-world data and 0.0003 for synthetic data. We adopt the specific panoramic data augmentation proposed by Sun et al. [30]. With the given setup, starting from default weight initialization, the best valid epoch was around 60 for real-world data and 90 for synthetic data. The average training speed is about $55ms/img$ for a 256×512 input image and $117ms/img$ for a 512×1024 image. Single-GPU inference time is $74ms$ (13 fps) for a 1024×512 image and $44ms$ (23 fps) for a 512×256 input image, showing that our method can be integrated in interactive settings. It is important to note, in terms of computational complexity, that the best competing method, BiFuse [33], has 253M parameters and multi-branching, while our much simpler architecture has only 75M parameters, also leading to reduced inference time (e.g., $74ms$ vs. $616ms$ for a 1024×512 image). Additional details are provided in Sec. 5.3.

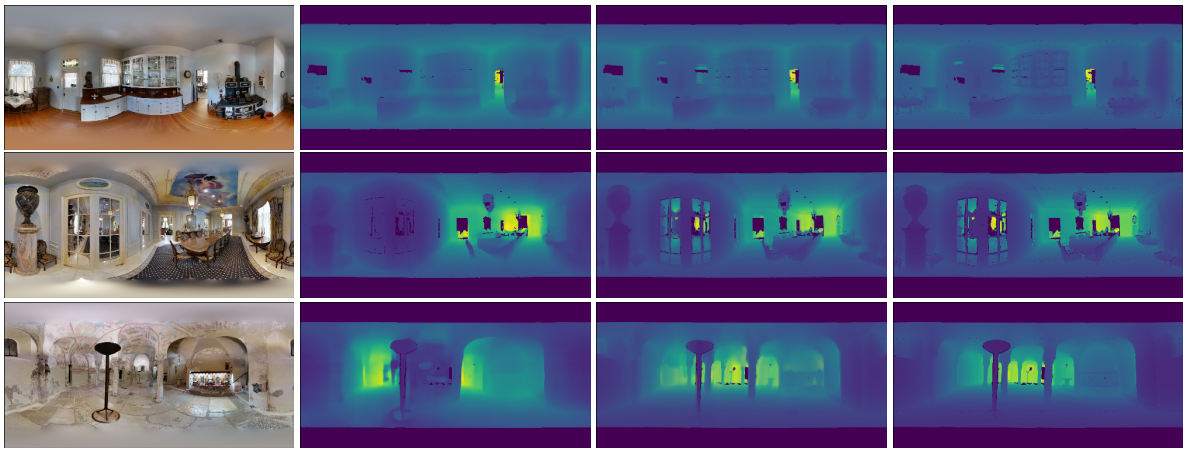
5.2. Quantitative and qualitative evaluation

We evaluated our method with the same error metrics used in prior depth estimation works [14, 44, 33]: mean absolute error (MAE), mean relative error (MRE), root mean square error of linear measures (RMSE), root mean square error of log measures (RMSE log scale invariant), and three relative accuracy measures δ_1, δ_2 and δ_3, defined, for an accuracy δ_n, as the fraction of pixels where the relative error is within a threshold of 1.25^n. Tab. 1 illustrates our quantitative results, in comparison with the most recent state-of-the-art works for which source code or numerical performance on the same data is available and using consistent training and testing setups. We compare with OmniDepth [44] (i.e., *RectNet*), BiFuse [33], as well as FCRN [14], which is the baseline of many current approaches (e.g., BiFuse [33]). Our method outperforms the others in terms of accuracy for all metrics, more markedly in cases of real data (Matterport3D and Stanford2D-3D-S in Tab. 1). In the case of synthetic data (360D in Tab. 1), our method also improves over other approaches, although here differences are closest, due to the fact that virtual renderings guarantee uniform 2D sampling and very few discontinuities [44] (except, for example, for occlusions), to the benefit of methods based on symmetrical 2D reduction and expansion.

Figures 2, 3, and 4 illustrate qualitative results on real and synthetic data. Figure 2 shows our prediction (Fig. 2(c)) on real-world RGB images (Fig. 2(a)) taken from Matterport3D[19], compared to ground truth (Fig. 2(d)) and BiFuse [33], for which the pre-trained model on Matterport3D was available. As we can see, our method finds a more accurate depth even in areas with smaller and repetitive structural details (first row of Fig. 2), in the case of large environments (second row of Fig. 2), and also for non-Manhattan-World but regular environments, as in the case of arched vaults (third row of Fig. 2). Figure 3 shows qualitative results on *360D* synthetic data [43], compared with the dataset creators' method [44]. The highlighted details illustrate qualitative differences. In particular, our method can infer a detailed reconstruction for typical man-made objects (Fig. 3, first row), even if they are far away (Fig. 3, second and third rows),

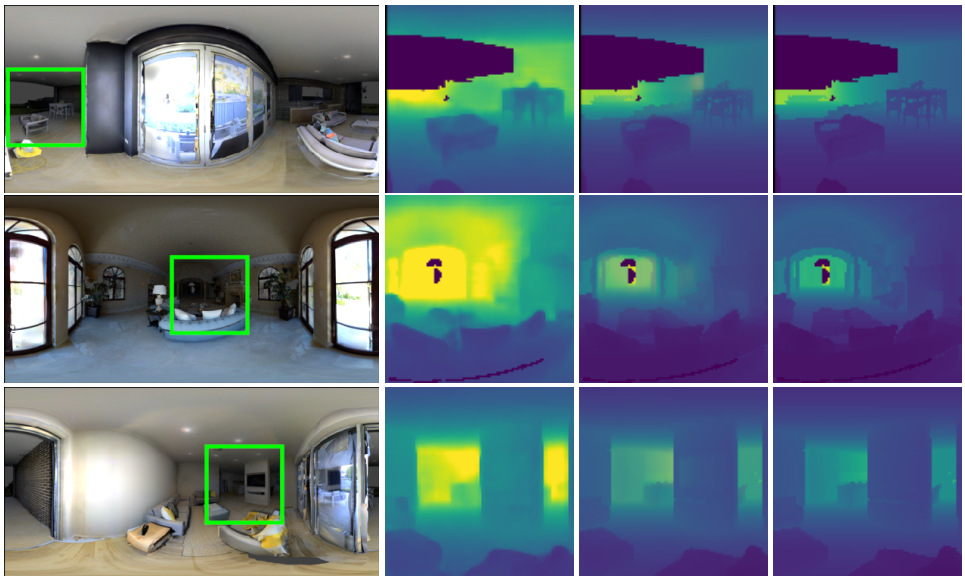
5.3. Ablation and Gravity Alignment Study

We present in this section the model ablation and computational costs (Tab. 2), and specific experiments showing the effectiveness of using the gravity-alignment prior (Tab. 3).



(a) *RGB* (b) *BiFuse* [33] (c) *Our* (d) *Ground truth*

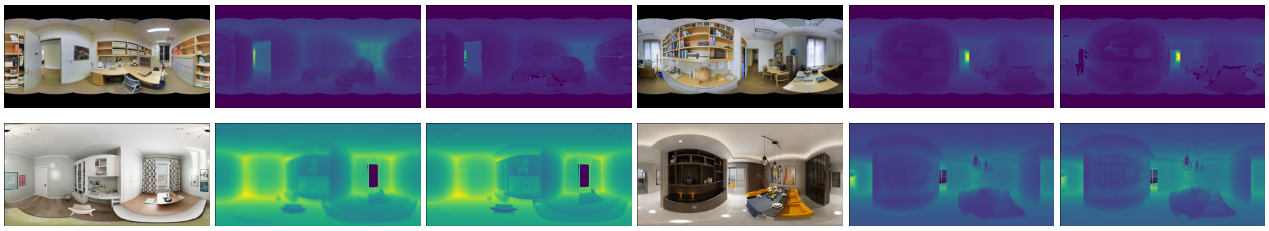
Figure 2. **Qualitative comparison on real-world datasets.** Depth maps are inferred from real-world captured RGB data (Matterport3D [19]). The first column is the input RGB image (Fig. 2(a)), the second one is the depth estimated by BiFuse [33] (Fig. 2(b)), the third one is the depth estimated by our method (Fig. 2(c)), and the fourth one is the ground-truth depth acquired by the instrument (Fig. 2(d)). Black pixels are missing samples in the ground-truth depth. All methods have been compared using the same original datasets and setting, without any further pre-process or alignment step.



(a) *RGB* (b) *OmniDepth* (c) *Our* (d) *Ground truth*

Figure 3. **Qualitative comparison on synthetic datasets.** Depth maps are inferred from synthetic data (360D [43]). We show in the first column the rendered RGB image (Fig. 2(a)), the estimated depth by OmniDepth [44] (Fig. 3(b)), by our method (Fig. 3(c)) and the rendered ground-truth depth (Fig. 2(d)). Black pixels are invalid pixels not rendered by the raytracer.

Ablation study and complexity. Our ablation experiments are presented in Tab. 2. To test the key components of the approach, we use results obtained with Structured3D [42], a synthetic dataset containing over 21,000 rendered rooms, that include, among other features, uniformly sampled color and very accurate depth panoramas. This very recent dataset has not yet been adopted by comparable works (Sec. 5.2), but provides an additional valuable benchmark for our method. The design variations discussed in the ablation study are those that consistently match decoder and encoder solution within our specific architecture and that better characterize our approach. Since our network has a simple single-branch structure, the computational cost of the model is directly related to the number of parameters of the model and its components. We thus illustrate the computational complexity of our method by presenting our network partitioned into macro blocks with their respective parameters: the *ResNet-50* features encoder block, the *Slicing* block (featuring slicing



(a) *RGB* (b) *Prediction* (c) *Ground truth* (d) *RGB* (e) *Prediction* (f) *Ground truth*

Figure 4. **Qualitative performance.** We present additional qualitative performance on Stanford2D3D [27] and Structured3D [42].

Table 2. **Ablation study.** The ablation study, performed on the *Structured3D* dataset[42], demonstrates how our proposed designs improve the accuracy of prediction. Results show only comparable-stable cases that actually increase it. We show in the last row the full architecture setup. PReLU activation provides identical benefits for each configuration in terms of convergence.

ResNet-50	Slicing	LSTM	Asym	Grad	Params	MRE	MAE	RMSE	RMSE log	δ_1	δ_2	δ_3
23.5M	24.8M (last 1)	-	6.3M	No	54.6M	0.4712	0.5520	0.1596	0.0341	0.6845	0.8684	0.8824
23.5M	33M (last 4)	-	6.3M	No	62.8M	0.2990	0.5014	0.0775	0.0154	0.7045	0.8784	0.9124
23.5M	24.8M (last 1)	12.5M	6.3M	No	67.1M	0.2988	0.4814	0.0750	0.0149	0.7702	0.8892	0.9222
23.5M	33M (last 4)	12.5M	6.3M	No	75.3M	0.0147	0.1223	0.0558	0.0102	0.8854	0.9376	0.9492
23.5M	**33M (last 4)**	**12.5M**	**6.3M**	**Yes**	**75.3M**	**0.0147**	**0.1180**	**0.0549**	**0.0109**	**0.9085**	**0.9451**	**0.9502**

and asymmetric dimensional reduction), the *LSTM* block and the *Asym* asymmetric upsample block. We also show the overall number of parameters for each setup (i.e., *Params*). For each block, the number of parameters needed is independent of the input image resolution, except for the LSTM block and the last upsampling, where the value indicated (i.e., 12.5M) is relative to the 256×512 resolution, which would be 16.8M for 512×1024. The results in Tab. 2 show the improvements obtained when using the last 4 ResNet layers, compared to using only the last one, in the *Slicing* block. Results at row 3 and 4, instead, show the benefits of adopting LSTM bottleneck-features refinement, which are appreciable already using only one ResNet output level, and become very consistent on the full pipeline. In addition, we present a comparison on whether or not to use the gradient component in the loss function, which mainly affects the sharpening of recovered depth details. Figure 5 shows a qualitative comparison between our model trained without or with the gradient loss. Many details typical of indoor environments (i.e., wall corners, objects with repetitive patterns), are lost without the contribution of the gradient component, even if from the point of view of the average numerical error the difference seems small. Since using the gradient, as for the PReLU activation (Sec. 3), provides identical benefits with every configuration, we expose the gradient contribution only for the last configuration. In particular, PReLU does not directly affect the best performance obtainable on single datasets but, instead, the ability to efficiently converge on both real and synthetic datasets. As an example, similar performances can be obtained using ELU without batch normalization on the synthetic OmniDepth dataset [43], but the same model would need batch normalization to work with Matterport3D [19], as also discussed in previous works [44, 33]. As shown in Tab. 2, each block adds a low and reasonable cost to the model, having as a counterpart a substantial increase in performance. In terms of computational cost, a standard decoder for equirectangular image based on FCRN [14], like the one adopted by BiFuse [33], needs about 38M of parameters, while the sum of our LSTM module (12.5M) and our actual decoder (6.3M) reaches 18.8M of parameters in total.

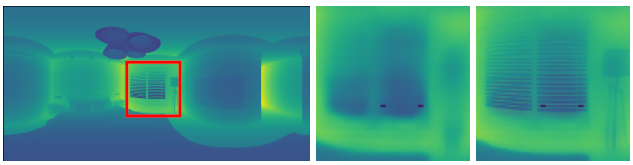
Table 3. **Gravity alignment study.** We test the robustness of our method to horizontal ground plane misalignment on Structured3D [42] and Matterport3D [19].

		MRE	MAE	RMSE	RMSE log	δ_1
Structured3D Our	0°	0.0147	0.1180	0.0549	0.1012	0.9085
	±2°	0.0217	0.1393	0.0658	0.1368	0.8776
	±5°	0.0263	0.1601	0.0714	0.1430	0.8527
Matterport3D Our	0°	0.1764	0.3296	0.6133	0.1045	0.8716
	±2°	0.2645	0.4205	0.7026	0.1334	0.7256
	±5°	0.3032	0.4806	0.7720	0.1482	0.6879
Matterport3D BiFuse [33]	0°	0.2048	0.3470	0.6259	0.1134	0.8452
	±2°	0.3888	0.5378	0.9805	0.1852	0.6144
	±5°	0.4905	0.6899	1.0225	0.2250	0.5440

Gravity evaluation of benchmark datasets. Our method assumes that the camera tripod is placed on a horizontal plane [3], which is common practice for capturing an indoor scene. We verified such feature on the four common publicly available datasets adopted above. All synthetic datasets [43, 42] are perfectly aligned by design. For real-world datasets [27, 19], we exploited the alignment pipeline of Zou et al. [46] to evaluate the misalignment with the ground plane. We found that the average misalignment with respect to the gravity vector of the Stanford2D3D [27] dataset is about 0.36 degrees, while the average misalignment of the Matterport3D [19] dataset is about 0.61 degrees

(full statistics in the supplementary material).

Robustness to gravity misalignment. Even if our method assumes to work with gravity-aligned scenes, we do not require perfect alignment, as demonstrated by our consistent results with the mentioned real-world datasets (Tab. 1). Moreover, we verified that the model, trained on the original aligned data, is robust to alignment errors, even larger than those appearing in practice. To test the behavior of our method in the presence of wider inclination errors, we exploit the Structured3D synthetic [42] dataset (such that the baseline is surely aligned to the ground plane) and Matterport3D [19] as real-world dataset. Starting from their initial baseline, we generate two new testing sets by randomly rotating the up vector of the camera, simulating a much wider misalignment to gravity — i.e., $\pm 2^\circ$ and $\pm 5^\circ$ maximum inclination error, as reported in Tab. 3. $\pm 2^\circ$ can be considered as a reliable error bound for a manual alignment without any correction, while $\pm 5^\circ$ is a deliberately wide range (additional tests are presented in the supplementary material). Results in Tab. 3 show that our method produces reliable predictions even with significant camera misalignment. Performance on the Structured3D dataset reaches good accuracy in all cases and low error values still competitive with state-of-art results. E.g., δ_1 is above 0.9 for the aligned case and degrades by only 0.03 for the moderate misalignment error of $\pm 2^\circ$ and 0.06 for the large misalignment error of $\pm 5^\circ$. The degradation obtained for Matterport3D are larger, but, by comparing the results with those in Tab. 1, we note that the results of our method on a dataset with $\pm 2^\circ$ error are still aligned with some of the state-of-the-art results obtained by other methods on perfectly aligned datasets. Moreover, we also present here the results obtained with BiFuse [33], for which the pre-trained model was available with the same training set, showing a much larger degradation in performance for non-gravity aligned data. This comparison shows how gravity alignment is also a fundamental assumption for other methods. It should be noted that these large errors can be avoided in practice by imposing capture constraints or performing a gravity-alignment pre-processing.

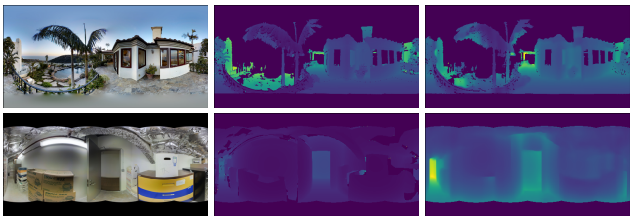


(a) *Ground truth* (b) *No grad.* (c) *With grad.*

Figure 5. **Loss function qualitative comparison.** Example of qualitative effects depending on gradient loss (Sec. 4).

5.4. Special cases and limits

In our experiments, we have verified that our model returns consistent results with all the man-made environments present in the tested datasets [27, 19, 43, 42], including scenes that can be defined as almost-outdoor (first row of Fig. 6). However, the quantitative and detailed performances depend on the ground truth data adopted, which in the case of depth often have masked parts due to lack of data from the sensor or unresolved ambiguities, such as reflections and fatal occlusions. In the second row of Fig. 6, we show one of these examples, that is one of the worst cases in our testes. Here the ground truth depth has numerous discontinuities and missing samples due to reflections, which are not easily predictable by our model. A large part of the structure is hidden by the insulating material.



(a) *Input* (b) *Ground truth* (c) *Output*

Figure 6. **Special cases.** First row: results on almost-outdoor environment. Second row: one of the worst cases in our tests.

6. Conclusions

We have introduced a novel deep neural network capable to rapidly estimate a depth map from a single monocular indoor panorama. Our design exploits gravity-aligned features, characterizing man-made interior environments through a compact representation of the scene into vertical spherical *slices*. We exploit long- and short-term relationships among slices to recover the equirectangular depth map, and maintain high-resolution information in the extracted features within a deep network. Our experimental results demonstrate that our method outperforms current state-of-the-art solutions in prediction accuracy, particularly in the case of real-world data with noise and missing data.

While the current method targets monocular reconstruction, we plan to extend it to multi-view in the context of structured 3D reconstruction of indoor environments. We are also looking at integrating it with interactive solutions, where we plan to use real-time depth estimation for volume and surface computation in AR settings. Moreover, while the approach was designed for indoor scenes, gravity alignment of features occurs also in other settings, especially man-made ones. We thus envision an extension of our approach to outdoor environments, in particular urban scenes.

Acknowledgments The project received funding from the European Union's H2020 research and innovation programme under grant 813170 (EVOCATION), and from Sardinian Regional Authorities under project VIGECLAB (POR FESR 2014-2020).

References

[1] Y. Cao, Z. Wu, and C. Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Trans. on Circuits and Systems for Video Technology*, 28(11):3174–3182, 2018.

[2] H. Cheng, C. Chao, J. Dong, H. Wen, T. Liu, and M. Sun. Cube padding for weakly-supervised saliency prediction in 360 videos. In *Proc. CVPR*, pages 1420–1429, 2018.

[3] Benjamin Davidson, Mohsan S. Alvi, and Joao F. Henriques Henriques. 360 camera alignment via segmentation. In *Proc. ECCV*, pages 579–595, 2020.

[4] M. Eder, P. Moulon, and L. Guan. Pano popups: Indoor 3D reconstruction with a plane-aware network. In *Proc. 3DV*, pages 76–84, 2019.

[5] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proc. ICCV*, pages 2650–2658, 2015.

[6] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2366–2374, 2014.

[7] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proc. CVPR*, June 2018.

[8] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proc. CVPR*, 2017.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. ICCV*, page 1026–1034, USA, 2015.

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

[11] Lei Jin, Yanyu Xu, Jia Zheng, Junfei Zhang, Rui Tang, Shugong Xu, Jingyi Yu, and Shenghua Gao. Geometric structure based and regularized depth estimation from 360 indoor imagery. In *Proc. CVPR*, June 2020.

[12] R. Jung, A. S. J. Lee, A. Ashtari, and J. Bazin. Deep360Up: A deep learning-based approach for automatic VR image upright adjustment. In *Proc. IEEE VR*, pages 1–8, 2019.

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ArXiv e-print arXiv:1412.6980*, 2014.

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

[15] Sophie Lambert-Lacroix and Laurent Zwald. The adaptive BerHu penalty in robust regression. *Journal of Nonparametric Statistics*, 28:1–28, 06 2016.

[16] J. Lee, M. Heo, K. Kim, and C. Kim. Single-image depth estimation based on fourier domain analysis. In *Proc. CVPR*, pages 330–339, 2018.

[17] Yeonkun Lee, Jaeseok Jeong, Jongseob Yun, Wonjune Cho, and Kuk-Jin Yoon. SpherePHD: Applying CNNs on a spherical polyhedron representation of 360° images. In *Proc. CVPR*, pages 9181–9189, 2019.

[18] F. Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proc. CVPR*, pages 5162–5170, 2015.

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

[20] Gregoire Payen de La Garanderie, Amir Atapour Abarghouei, and Toby P. Breckon. Eliminating the blind spot: Adapting 3D object detection and monocular depth estimation to 360 panoramic imagery. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Proc. ECCV*, pages 812–830, 2018.

[21] Peng Wang, Xiaohui Shen, Zhe Lin, S. Cohen, B. Price, and A. Yuille. Towards unified depth and semantic prediction from a single image. In *Proc. CVPR*, pages 2800–2809, 2015.

[22] Giovanni Pintore, Marco Agus, and Enrico Gobbetti. AtlantaNet: Inferring the 3D indoor layout from a single 360 image beyond the Manhattan World assumption. In *Proc. ECCV*, pages 432–448, 2020.

[23] Giovanni Pintore, Claudio Mura, Fabio Ganovelli, Lizeth Fuentes-Perez, Renato Pajarola, and Enrico Gobbetti. State-of-the-art in automatic 3D reconstruction of structured indoor environments. *Comput. Graph. Forum*, 39(2):667–699, 2019.

[24] Giovanni Pintore, Claudio Mura, Fabio Ganovelli, Lizeth Fuentes-Perez, Renato Pajarola, and Enrico Gobbetti. Automatic 3D reconstruction of structured indoor environments. In *SIGGRAPH 2020 Courses*, pages 10:1–10:218, August 2020.

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

[26] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Proc. NIPS*, page 802–810, 2015.

[27] Stanford University. BuildingParser Dataset. http://buildingparser.stanford.edu/dataset.html, 2017. [Accessed: 2019-09-25].

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

[29] Yu-Chuan Su and Kristen Grauman. Learning spherical convolution for fast features from 360 imagery. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 529–539, 2017.

[30] Cheng Sun, Chi-Wei Hsiao, Min Sun, and Hwann-Tzong Chen. HorizonNet: Learning room layout with 1D representation and pano stretch data augmentation. In *Proc. CVPR*, June 2019.

[31] Keisuke Tateno, Nassir Navab, and Federico Tombari. Distortion-aware convolutional filters for dense prediction in panoramic images. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Proc. ECCV*, pages 732–750, 2018.

[32] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *Proc. ICML*, pages 1747–1756, 2016.

[33] Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. Bifuse: Monocular 360 depth estimation via bi-projection fusion. In *Proc. CVPR*, June 2020.

[34] Wenqi Xian, Zhengqi Li, Matthew Fisher, Jonathan Eisenmann, Eli Shechtman, and Noah Snavely. UprightNet: geometry-aware camera orientation estimation from single images. In *Proc. ICCV*, pages 9974–9983, 2019.

[35] D. Xu, W. Wang, H. Tang, H. Liu, N. Sebe, and E. Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *Proc. CVPR*, pages 3917–3925, 2018.

[36] Shang-Ta Yang, Fu-En Wang, Chi-Han Peng, Peter Wonka, Min Sun, and Hung-Kuo Chu. DuLa-Net: A dual-projection network for estimating room layouts from a single RGB panorama. In *Proc. CVPR*, 2019.

[37] Yang Yang, Shi Jin, Ruiyang Liu, , and Jingyi Yu. Automatic 3D indoor scene modeling from single panorama. In *Proc. CVPR*, pages 3926–3934, 2018.

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

[39] Wei Zeng, Sezer Karaoglu, and Theo Gevers. Joint 3D layout and depth prediction from a single indoor panorama image. In *Proc. ECCV*, pages 666–682, 2020.

[40] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *Proc. CVPR*, June 2018.

[41] Yinda Zhang, Shuran Song, Ping Tan, and Jianxiong Xiao. PanoContext: A whole-room 3D context model for panoramic scene understanding. In *Proc. ECCV*, pages 668–686, 2014.

[42] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3D: A large photo-realistic dataset for structured 3D modeling. In *Proc. ECCV*, pages 519–535, 2020.

[43] Nikolaos Zioulis, Antonis Karakottas, Dimitris Zarpalas, Federic Alvarez, and Petros Daras. Spherical view synthesis for self-supervised 360° depth estimation. In *Proc. 3DV*, September 2019.

[44] Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, and Petros Daras. OmniDepth: Dense depth estimation for indoors spherical panoramas. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Proc. ECCV*, pages 453–471, 2018.

[45] Michael Zollhöfer, Patrick Stotko, Andreas Görlitz, Christian Theobalt, Matthias Nießner, Reinhard Klein, and Andreas Kolb. State of the art on 3D reconstruction with RGB-D cameras. *Comput. Graph. Forum*, 37(2):625–652, 2018.

[46] Chuhang Zou, Alex Colburn, Qi Shan, and Derek Hoiem. LayoutNet: Reconstructing the 3D room layout from a single RGB image. In *Proc. CVPR*, pages 2051–2059, 2018.