OmniDepth: Dense Depth Estimation for Indoors Spherical Panoramas.

Anonymous ECCV submission

Paper ID **2775**

Abstract. Recent work on depth estimation up to now has only focused on projective images ignoring 360^o content which is now increasingly and more easily produced. However, we show that monocular depth estimation models trained on traditional images produce sub-optimal results on omnidirectional images, showcasing the need for training directly on 360^o datasets, which however, are hard to acquire. In this work, we circumvent the challenges associated to acquiring high quality 360^o datasets with ground truth depth annotations, by re-using recently released large scale 3D datasets and re-purposing them to 360^o via rendering. This dataset, which is considerably larger than similar projective datasets, is publicly offered to the community to enable future research in this direction. We use this dataset to learn in an end-to-end fashion the task of depth estimation from 360^o images. We show promising results in our synthesized data as well as in unseen realistic images.

Keywords: Omnidirectional Content, 360^o, Scene Understanding, Depth Estimation, Synthetic Dataset, Learning with Virtual Data

1 Introduction

One of the fundamental challenges in computer and 3D vision is the estimation of a scene's depth. Depth estimation leads to a three-dimensional understanding of the world which is very important to numerous applications. These vary from creating 3D maps [1] and allowing navigation in real-world environments [2], to enabling stereoscopic rendering [3], synthesizing novel views out of pre-captured content [4] and even compositing 3D objects into it [5]. Depth information can be utilized jointly with color information and has been shown to boost the effectiveness of many vision tasks related to scene understanding [6,7].

Similar to how babies start to perceive depth from two viewpoints and then by ego-motion and observation of objects' motions, researchers have tackled the problem of estimating depth via methods built on multi-view consistency [8,9] and structure-from-motion [10]. But humans are also driven by past experiences and contextual similarities and apply this collective knowledge when presented with new scenes. Likewise, with the advent of more effective machine learning techniques, recent research focuses on learning to predict depth using - mostly - CNNs [11,12,13,14,6,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29]. This has led to impressive results even with completely unsupervised learning approaches.

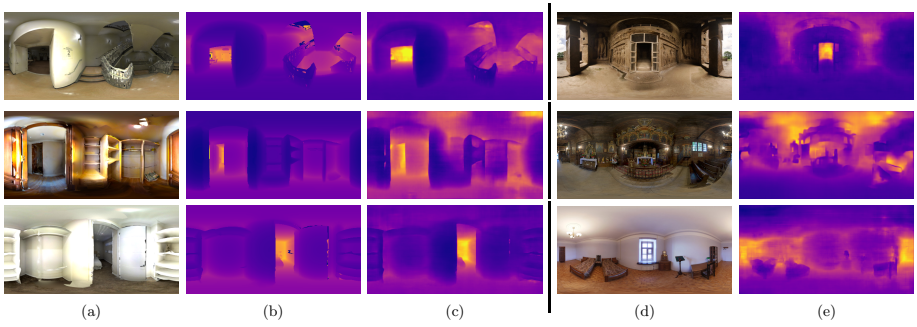

Fig. 1: We learn the task of predicting depth directly from omnidirectional indoor scene images. Example predictions by our RectNet model are presented. From left to right: (a) 360^o image samples from our test set, (b) ground truth depth, (c) predicted depth map, (d) 360^o unseen image samples from Sun360 dataset, (e) predicted depth map.

However, learning based approaches have only focused on traditional 2D content captured by typical pinhole projection model based cameras. With the emergence of efficient spherical cameras and rigs, omnidirectional (360^o) content is now more easily and consistently produced and is witnessing increased adoption in entertainment and marketing productions, robotics and vehicular applications as well as coverage of events and even journalism. Consumers can now experience 360^o content in mobile phones, desktops and, more importantly, the new arising medium – Virtual Reality (VR) – headsets.

Depth and/or geometry extraction from omnidirectional content has been approached similar to traditional 2D content via omnidirectional stereo [30,31,32,33] and structure-from-motion (SfM)[4] analytical solutions. There are inherent problems though to applying learning based methods to 360^o content as a result of its acquisition process that inhibits the creation of high quality datasets. Coupling them with 360^o LIDARs would produce low resolution depths and would also insert the depth sensor into the content itself, a drawback that also exists when aiming to acquire stereo datasets. One alternative would be to manually re-position the camera but that would be tedious and error prone as a consistent baseline would not be possible.

In this work, we train a CNN to learn to estimate a scene's depth given an omnidirectional (equirectangular) image as input[1]. To circumvent the lack of available training data we resort to re-using existing 3D datasets and re-purposing them for use within a 360^o context. This is accomplished by generating diverse 360^o views via rendering. We use this dataset for learning to infer depth from omnidirectional content. In summary, our contributions are the following:

[1] We use the terms omnidirectional image, 360^o image, spherical panorama and equirectangular image interchangeably in this document.

1. We present the first, to the authors' knowledge, learning based dense depth estimation method that was trained with and operates directly on omnidirectional content in the form of equirectangular images.
2. We offer a dataset consisting of 360^o color images paired with ground truth 360^o depth maps in equirectangular format. The dataset is available online[2].
3. We propose and validate, a CNN auto-encoder architecture specifically designed for estimating depth directly on equirectangular images.
4. We demonstrate how existing monocular depth estimation methods trained on traditional 2D images fall short or produce low quality results when applied to equirectangular inputs, highlighting the need for learning directly on the 360^o domain.

2 Related Work

Since this work aims to learn the task of dense depth estimation from omnidirectional input, and given that - to the authors' knowledge - no other similar work exists, we first review non-learning based methods for geometric scene understanding based on 360^o images. We then examine learning based approaches for spherical content and, finally, we present recent methods for monocular depth estimation.

2.1 Geometric understanding on 360^o images

Similar to typical pinhole projection model cameras, the same multi-view geometry [8] principles apply to 360^o images. By observing the scene from multiple viewpoints and establishing correspondences between them, the underlying geometrical structure can be estimated. For 360^o cameras the conventional binocular or multi-view stereo [9] problem is reformulated to binocular or multi-view spherical stereo [30] respectively, by taking into account the different projection model and after defining the disparity as angular displacements. By estimating the disparity (i.e. depth), complete scenes can be 3D reconstructed from multiple [34,33] or even just two [31,32] spherical viewpoints. However, all these approaches require multiple 360^o images to estimate the scene's geometry. Recently it was also shown that monocular 360^o videos acquired using a moving camera can also be used to 3D reconstruct a scene's geometry via SfM [4] and enable 6 degrees-of-freedom viewing in VR headsets.

There are also approaches that require only a single image to understand indoors scenes and estimate their layout. One such approach, PanoContext [35], generates a 3D room layout hypothesis given an indoor 360^o image in equirectangular format. With the estimations being bounding boxes, the inferred geometry is only a coarse approximation of the scene. Similar in spirit, the work of Yang et al. [36] generates complete room layouts from panoramic indoor images by combining superpixel information, vanishing points estimation and a geometric

[2] blindreview.com

context prior under a Manhattan world assumption. However, focusing on room layout estimation, it is unable to recover finer details and structures of the scene. Another similar approach [37] addresses the problem of geometric scene understanding from another perspective. Under a maximum a posteriori estimation it unifies semantic, pose and location cues to generate CAD models of the observed scenes. Finally, in [38] a spherical stereo pair is used to estimate both the room layout but also object and material attributes. After retrieving the scene's depth by stereo matching and subsequently calculating the normals, the equirectangular image is projected to the faces of a cube that are then fed to a CNN. The CNN predictions are projected back into the equirectangular image to finally reconstruct the 3D room layout.

2.2 Learning for 360^o images

One of the first approaches to estimate distances purely from omnidirectional input [39] under a machine learning setting utilized Gaussian processes. Instead of estimating the distance of each pixel, a range value per image column was predicted to drive robotic navigation. Nowadays, with the establishment of CNNs, there are two straightforward ways to apply current CNN processing pipelines to spherical input. Either directly on a projected (typically equirectangular) image, or by projecting the spherical content to the faces of a cube (cubemap) and then running the CNN predictions on the 2D projected images that are merged by back-projecting them to the spherical domain. The latter approach was selected by [40], an artistic style transfer work, where each face was re-styled separately and then the cubemap was re-mapped back to the equirectangular domain. Likewise, in SalNet360 [41], saliency predictions on the cube's faces are refined using their spherical coordinates and then merged back to 360^o. In contrast, following the former approach, and applying a CNN directly to the equirectangular image, was opted for in [42], that focuses on increasing the dynamic range of outdoor panoramas.

More recently, new techniques for applying CNNs to omnidirectional input were presented. Given the difficulty to model the projection's distortion directly in typical CNNs as well as achieve invariance to the viewpoint's rotation, the alternative pursued by [43] is based on graph-based deep learning. Specifically they model distortion directly into the graph's structure and apply it to a classification task. A novel approach taken in [44] is to learn appropriate convolution weights for equirectangular projected spherical images by transferring them from an existing network trained on traditional 2D images. This conversion from the 2D to the 360^o domain is accomplished by enforcing consistency between the predictions of the 2D projected views and those in the 360^o image. Moreover, recent work on convolutions [45,46] that in addition to learning their weights also learn their shape, are very well suited for learning the distortion model of spherical images, even though they have only been applied to fisheye lenses up to now [47]. Finally, very recently, Spherical CNNs were proposed in [48,49] that are based in a rotation-equivariant definition of spherical cross-correlation. However these were only demonstrated in classification and single variable regression

problems. In addition, they are also applied in the spectral domain while we formulate our network design for the spatial image domain.

2.3 Monocular depth estimation

Depth estimation from monocular input has attracted lots of interest lately. While there are some impressive non learning based approaches [50,51,52], they come with their limitations, namely reliance on optical flow and relevance of the training dataset. Still, most recent research has focused on machine learning to address the ill-posed depth estimation problem. Initially, the work of Eigen et al. [15] trained a CNN in a coarse-to-fine scheme using direct depth supervision from RGB-D images. In a subsequent continuation of their work [6], they trained a multi-task network that among predicting semantic labels and normals, also estimated a scene's depth. Their results showed that jointly learning the tasks achieved higher performance due to their complementarity. In a recent similar work [21], a multi-task network that among other modalities also estimated depth, was trained using synthetic data and a domain adaptation loss based on adversarial learning, to increase its robustness when running on real scenes. Laina et al. [12] designed a directly supervised fully convolutional residual network (FCRN) with novel up-projection blocks that achieved impressive results for indoor scenes and was also used in a SLAM pipeline [1].

Another body of work focused on applying Conditional Random Fields (CRFs) to the depth estimation problem. Initially, the output of a deep network was refined using a hierarchical CRF [28], with Liu et al. [29] further exploring the interplay between CNNs and CRFs for depth estimation in their work. Recently, multi-scale CRFs were used and trained in an end-to-end manner along with the CNN [20]. Dense depth estimation has also been addressed as a classification problem. Since perfect regression is usually impossible, dense probabilities were estimated in [23] and then optimized to estimate the final depth map. Similarly, in [17] and [27] depth values were discretized in bins and densely classified, to be afterwards refined either via a hierarchical fusion scheme or through the use of a CRF respectively. Taking a step further, a regression-classification cascaded network was proposed in [18] where a low spatial resolution depth map was regressed and then refined by a classification branch.

The concurrent works of Garg et al. [13] and Godard et al. [11] showed that unsupervised learning of the depth estimation task is possible. This is accomplished by an intermediate task, view synthesis, and allowed training by only using stereo pair input with known baselines. In a similar fashion, using view synthesis as the main supervisory signal, learning to estimate depth was also achieved by training with pure video sequences in a completely unsupervised manner [16,19,24,25,22,53]. Another novel unsupervised depth estimation method relies on aperture supervision [14] by simply acquiring training data in various focus levels. Finally, in [26] it was shown that a CNN can be trained to estimate depth from monocular input with only relative depth annotations.

3 Synthesizing Data

End-to-end training of deep networks requires a large amount of annotated ground truth data. While for typical pinhole camera datasets this was partly addressed by using depth sensors [54] or laser scanners [55] such an approach is impractical for spherical images due to a larger diversity in resolution for 360^o cameras and laser scanners, and because each 360^o sensor would be visible from the other one. As much as approaches like the one employed in [56] could be used to in-paint the sensor regions, these would still be the result of an algorithmic process and not the acquisition process itself, potentially introducing errors and artifacts that would reduce the quality of the data. This also applies to unsupervised stereo approaches that require the simultaneous capture of the scene from two distinct viewpoints. Although one could re-position the same sensor to acquire clean panoramas, a consistent baseline (both translational and rotational) would not be possible. More recently unsupervised approaches for inferring a scene's depth have emerged that are trained with video sequences. However, these work on the assumption of a moving camera as they rely on view synthesis as the supervisory signal which is not a typical setting for indoors 360^o captures, but for action camera like recordings.

360D Dataset: Instead, we rely on generating a dataset with ground truth depth by synthesizing both the color and the depth image via rendering. To accomplish that we leverage the latest efforts in creating publicly available textured 3D datasets of indoors scenes. Specifically, we use two computer generated (CG) datasets, SunCG [57] and SceneNet [58], and two realistic ones acquired by scanning indoor buildings, Stanford2D3D [59,60] and Matterport3D [61]. We use a path tracing renderer[3] to render our dataset by placing a spherical camera and a uniform point light at the same position $\mathbf{c} \in \mathbb{R}^3$ in the scene. We then acquire the rendered image $I(\mathbf{p}) \in \mathbb{R}, \mathbf{p} = (u, v) \in \mathbb{N}^2$, as well as the underlying z-buffer that was generated as a result of the graphics rendering process, that serves as the ground truth depth $D(\mathbf{p}) \in \mathbb{R}$. It should be noted that unlike pinhole camera model images, the z-buffer in this case does not contain the z coordinate value of the 3D point $\mathbf{v}(\mathbf{p}) \in \mathbb{R}^3$, corresponding to pixel $\mathbf{p}$, but instead the 3D point's radius $r = \|\mathbf{v} - \mathbf{c}\|$, in the camera's spherical coordinate system.

For the two CG datasets we place the camera and the light at the center of each house, while for the two scanned datasets we use the pose information available (estimated during the scanning process) and thus, for each building we generate multiple 360^o data samples. Given that the latter two datasets were scanned, their geometries contain holes or inaccurate / coarse estimations, and also have lighting information baked into the models. On the other hand, the CG datasets contain perfect per pixel depth but lack the realism of the scanned datasets, creating a complementary mix. However, as no scanning poses are available, the centered poses may sometimes be placed within or on top of objects and we also observed missing information in some scenes (e.g. walls/ceilings) that, given SunCG's size, are impractical to manually correct.

[3] https://www.cycles-renderer.org

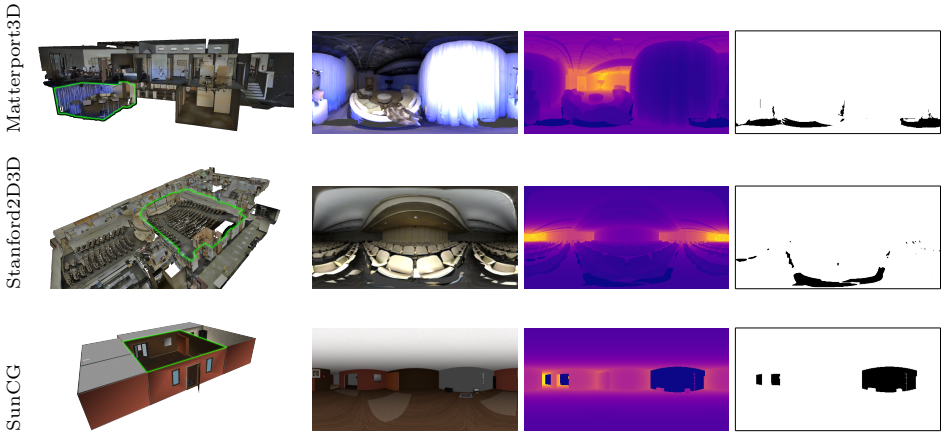

Fig. 2: Example renders from our dataset, from left to right: the 3D building with a green highlight denoting the rendered scene, color output, corresponding depth map, and the binary mask depicting the missing regions in black.

For each pose, we augment the dataset by rotating the camera in 90^o resulting in 4 distinct viewpoints per pose sample. Given the size of SunCG, we only utilize a subset of it and end up using **11118** houses, resulting in a 24.36% utilization. The remaining three datasets are completely rendered. This results in a total of **94098** renders and **23524** unique viewpoints. Our generated *360D* dataset contains a mix of synthetic and realistic 360^o color I and depth D image data in a variety of indoors contexts (houses, offices, educational spaces, different room layouts) and is publicly available at `blindreview.com` (more indicative samples in the supplement).

4 Omnidirectional Depth Estimation

The majority of recent CNN architectures for dense estimation follow the autoencoder structure, in which an encoder encodes the input, by progressively decreasing its spatial dimensions, to a representation of much smaller size, and a decoder regresses to the desired output by upscaling this representation.

We use two encoder-decoder network architectures that are structured differently. The first resembles those found in similar works in the literature [11,12], while the second is designed from scratch to be more suitable for learning with 360^o images. Both networks are fully convolutional [62] and predict an equirectangular depth map with the only input being an 360^o color image in equirectangular format. We use ELUs [63] as the activation function which also remove the need for batch normalization [64] and its added computational complexity.

UResNet: In this unbalanced ResNet, the encoding and decoding parts are not symmetrical, with the decoder being shallower. The encoder is built with skip connections [65], a technique that helps when training deeper architectures

by preventing gradient degradation, allowing for larger receptive fields. More detailed architectural information is presented in Fig. 3 where the network is decomposed into processing blocks.

RectNet: Omnidirectional images differ from traditional images in the sense that they capture global (full 360^o) visual information and, when in equirectangular format, suffer from high distortions along their y (i.e. latitude) axis. Therefore, the second architecture's design aims to exploit and address these properties of spherical panoramas while keeping some of the desirable properties of UResNet like skip connections. Capturing the 360^o image's global context is achieved by increasing the effective receptive field (RF) of each neuron by utilizing dilated convolutions [66]. Instead of progressive downscaling as in most depth estimation networks and similarly UResNet , we only drop the spatial dimensions by a factor of 4. Then, inspired by [67], we use progressively increasing dilations to capture a much larger RF, about half the input's spatial dimensions. In addition, within each dilation block we utilize 1×1 convolutions to reduce the spatial correlations of the feature maps.

The distortion factor of spherical panoramas increases towards the sphere's poles and is therefore different for every image row. This means that information is scattered horizontally, as we vertically approach the two poles. In order to account for this varying distortion we alter our input blocks, as their features are closer to natural image ones (e.g. edges). We use rectangle filters along with traditional square filters and vary the resolution of the rectangle ones to account for different distortion levels. However, this variation is done while also preserving

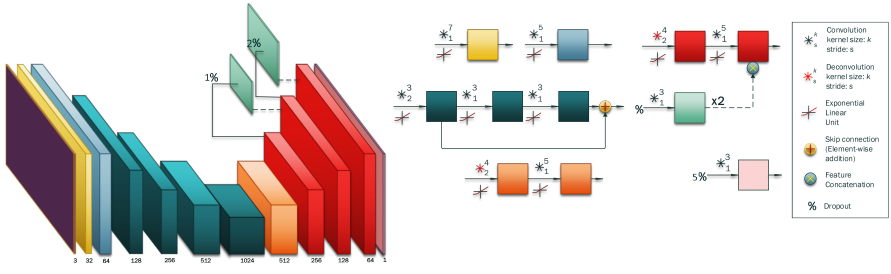

Fig. 3: **UResNet Architecture:** The encoder consists of two input preprocessing blocks, and four down-scaling blocks (dark green). The preprocessing blocks (yellow and blue) are single convolutional (conv) layers. The down-scaling blocks consist of a strided conv and two more regular convs. A skip / residual connection connects the strided conv to the block's last conv. The decoder contains one upscaling block (orange) and three up-prediction blocks (red), followed by the prediction layer (pink). The up-scaling block consists of a strided deconvolution followed by a conv layer. The up-prediction blocks are similar to the up-scaling block, but after the last conv, another conv layer branches out to estimate a depth prediction at the corresponding scale. This is concatenated with the features of the next block's last layer.

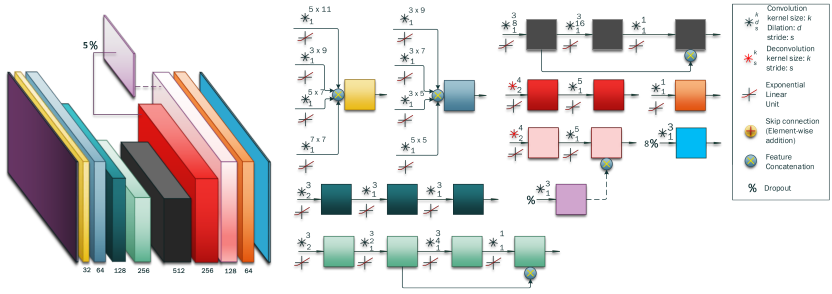

Fig. 4: **RectNet Architecture:** The encoder consists of two preprocessing blocks (yellow and blue) and a downscaling block (dark green), followed by two increasing dilation blocks (light green and black). The preprocessing blocks concatenate features produced by convolutions (convs) with different filter sizes, accounting for the equirectangular projection's varying distortion factor. The down-scaling block comprises a strided and two regular convs.

the area of the filter to be as close as possible to the original square filter's. The outputs of the rectangle and square filters are concatenated while preserving the overall output feature count. The detailed architecture is presented in Fig. 4.

Training Loss: Given that we synthesize perfect ground truth depth annotations D_{gt}, as presented in Section 3, we take a completely supervised approach. Even though most approaches using synthetic data fail to generalize to realistic input, our dataset contains an interesting mix of synthetic (CAD) renders as well as realistic ones. The scanned data are acquired from real environments and, as a result, their renders are very realistic. Following previous work, we predict depth D^s_{pred} against downscaled versions of the ground truth data D^s_{gt} at multiple scales (with s being the downscaling factor) and upsample these predictions using nearest neighbor interpolation to later concatenate them with the subsequent higher spatial dimension feature maps. We also use the dropout technique [68] in those layers used to produce each prediction. Further, we use L2 loss for regressing the dense depth output $E_{depth}(\mathbf{p}) = \|D_{gt}(\mathbf{p}) - D_{pred}(\mathbf{p})\|^2$ and additionally add a smoothness term $E_{smooth}(\mathbf{p}) = \|\nabla D(\mathbf{p})\|^2$ for the predicted depth map by minimizing its gradient.

Although our rendered depth maps are accurate in terms of depth, in practice there are missing regions in the rendered output. These are either because of missing information in the CAD models (e.g. walls/ceilings) or the imperfect process of large scale 3D scanning, with visual examples illustrated in Fig. 2. These missing regions/holes manifest as a specific color ("clear color"), selected during rendering, in the rendered image and as infinity ("far") values in the rendered depth map. As these outlier values will destabilize the training process, we ignore them during backpropagation by using a per pixel $\mathbf{p}$ binary mask $M(\mathbf{p})$ that is zero in these missing regions. This allows us to train the network even with incomplete or slightly inaccurate/erroneous 3D models. Thus, our final loss

Table 1: Quantitative results of our networks for 360^o dense depth estimation.

Network	Tested on	Abs Rel ↓	Sq Rel ↓	RMS ↓	RMSlog ↓	$\delta < 1.25 \uparrow$	$\delta < 1.2^2 \uparrow$	$\delta < 1.25^3 \uparrow$
UResNet	Test set	0.0835	0.0416	0.3374	0.1204	0.9319	0.9889	0.9968
RectNet	Test set	**0.0702**	**0.0297**	**0.2911**	**0.1017**	**0.9574**	**0.9933**	**0.9979**
UResNet	SceneNet	0.1218	0.0727	0.4066	0.1538	0.8598	0.9815	0.9962
RectNet	SceneNet	0.1077	0.699	0.3572	0.1386	0.8965	0.9879	0.9971
UResNet -S2R	Stanford	0.1226	0.0768	0.489	0.1667	0.8593	0.9756	0.9942
RectNet -S2R	Stanford	0.0824	0.0457	0.3998	0.1229	0.928	0.9879	0.9971
UResNet -S2R	SceneNet	0.1448	0.0991	0.517	0.1792	0.7898	0.9761	0.9935
RectNet -S2R	SceneNet	0.1079	0.0644	0.3778	0.1404	0.8966	0.9866	0.996

function is:

$$E_{loss}(\mathbf{p}) = \sum_s \alpha_s M(\mathbf{p}) E_{depth}(\mathbf{p}) + \sum_s \beta_s M(\mathbf{p}) E_{smooth}(\mathbf{p}) \,, \tag{1}$$

where α_s, β_s are the weights for each scale of the depth and smoothing term.

5 Results

We evaluate the performance of direct supervision with synthesized data using our two 360^o depth estimation networks in various ways. We first conduct an intra assessment of the two models and then we offer quantitative comparisons with other depth estimation methods. Finally, we also offer qualitative results of our models and other depth in completely unseen, realistic data of everyday scenes.

Training Details: Our networks are trained using Caffe [69] on a single NVidia Titan X. We use Xavier weight initialization [70] and ADAM [71] as the optimizer with its default parameters $[\beta_1, \beta_2, \epsilon] = [0.9, 0.999, 10^-8]$ and an initial learning rate of 0.0002. Our input dimensions are 512×256 and are given in equirectangular format, with our depth predictions being equal sized.

We split our dataset into corresponding train and tests sets as follows: (i) Initially we remove 1 complete area from Stanford2D3D, 3 complete buildings from Matterport3D and 3 CAD scenes from SunCG for our test set totaling 1,298 samples. (ii) We skip SceneNet entirely and use it as our validation set. (iii) Then, from the remaining SunCG, Stanford2D3D and Matterport3D samples we automatically remove scenes which contain regions with very large or small depth values ($> 5\%$ of total image area above $20m$ or under $0.5m$). Finally, we are left with a train-set that consists of 34,679[4] RGB 360^o images along with their corresponding ground truth depth map annotations. Our loss weights for UResNet are $[\alpha_1, \alpha_2, \alpha_4, \beta_1] = [0.445, 0.275, 0.13, 0.15]$, and for RectNet they are $[\alpha_1, \alpha_2, \beta_1, \beta_2] = [0.535, 0.272, 0.134, 0.068]$. For quantitative evaluation we use

[4] SunCG was not used in its entirety. Only a subset was used by prioritizing larger scenes as the rendering process is long. As data generation continued in parallel to the work, a larger subset will be publicly offered.

Table 2: Quantitative results against other monocular depth estimation models.

	Network	Abs Rel↓	Sq Rel ↓	RMS ↓	RMS(log) ↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
	UResNet	0.0835	0.0416	0.3374	0.1204	0.9319	0.9889	0.9968
	RectNet	**0.0702**	**0.0297**	**0.2911**	**0.1017**	**0.9574**	**0.9933**	**0.9979**
Equirect.	Godard et al. [11]	0.4747	2.3783	7.2097	0.82	0.297	0.79	0.751
Equirect.	Laina et al. [12]	0.3181	0.4469	0.941	0.376	0.4922	0.7792	0.915
Equirect.	Liu et al. [29]	0.4202	0.7597	1.1596	0.44	0.3889	0.7044	0.8774
Cubemap	Godard et al. [11]	0.2552	0.9864	4.4524	0.5087	0.3096	0.5506	0.7202
Cubemap	Laina et al. [12]	0.1423	0.2544	0.7751	02497	0.5198	0.8032	0.9175
Cubemap	Liu et al. [29]	0.1869	0.4076	0.9243	0.2961	0.424	0.7148	0.8705

the same error metrics as previous works [15,6,11,12,29] (arrows next to each metric in the tables denote the direction of better performance).

Model Performance: Table 1 presents the results of our two models in our test set, and in the unseen synthetic SceneNet generated data, after training for 10 epochs in all of our train set. We observe that RectNet – which was designed with 360^o input in mind – performs better than the standard UResNet even with far fewer parameters ($\sim 8.8M$ vs $\sim 51.2M$). In order to assess their efficacy and generalization capabilities we perform a leave-one-out evaluation. We train both networks initially only in the synthetic SunCG generated data for 10 epochs, and then finetune them in the realistic Matterport3D generated data for another 10 epochs. This train is suffixed with "-S2R". We then evaluate them in the entirety of the Stanford2D3D generated dataset, as well as in the SceneNet one. Comparable results to the previous train with all datasets are observed. Again, RectNet outperforms UResNet – albeit both perform slightly worse as expected due to being trained with less amount of data.

The increased performance of RectNet against UResNet in every error metric or accuracy, can be attributed to its larger RF, which for 360^o images is very important as it allows the network to capture the global context more efficiently. Despite the fact that UResNet is much deeper than RectNet and significantly drops the input's spatial dimensions, RectNet still achieves a larger receptive field. Specifically, UResNet has a 190×190 RF compared to that of RectNet which is 266×276. In addition, RectNet drops the input's spatial dimensions only by a factor of 4, maintaining denser information in the extracted features.

Comparison against other methods: Given that there are no other methods to perform dense depth estimation for 360^o images, we assess its performance against the state of the art in monocular depth estimation models. Since the predictions of these methods are defined in different scales, we scale the estimated depth maps by a scalar $\tilde{s}$, which matches their median with our ground truth like [16], i.e. $\tilde{s} = median(D_{gt})/median(D_{pred})$. Moreover, we evaluate the masked depth maps as mentioned in Section 3 in order to ignore the missing values. Table 2 presents the results of of state-of-the-art methods when applied directly on our test split in the equirectangular domain. We offer results for the model of Laina et al. [12], trained with direct depth supervision in indoor scenes, Godard et al. [11], trained in an unsupervised manner in outdoor driving scenes using

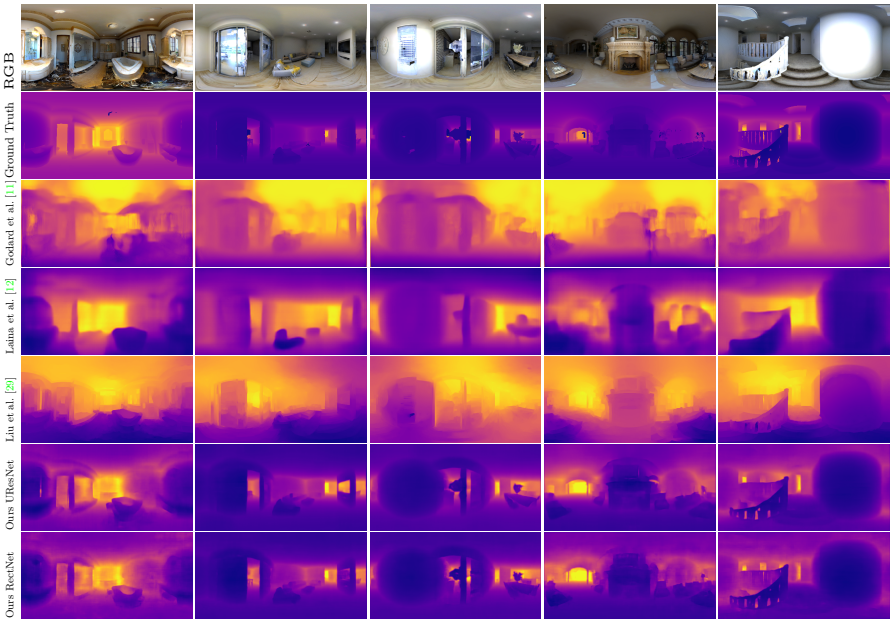

Fig. 5: Qualitative results on our test split.

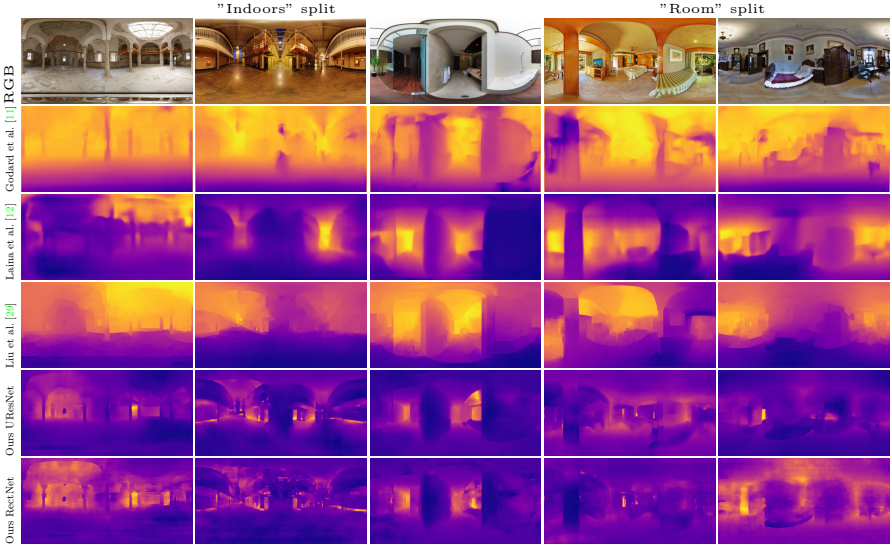

Fig. 6: Qualitative results on the "Room" and "Indoors" Sun360 splits.

calibrated stereo pairs, and the method of Liu et al. [29], which combines learning with CRFs and is trained in indoor scenes. As observed by the results, the performance of all the methods directly on equirectangular images is poor, and

Table 3: Per cube face quantitative results against other monocular models.

Network	AbsRel ↓	SqRel ↓	RMSE ↓	RMSElog ↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta N 1.25^3 \uparrow$
UResNet	0.0097	0.0062	0.1289	0.041	0.9245	0.9853	0.9955
RectNet	**0.008**	**0.0042**	**0.1113**	**0.03504**	**0.9497**	**0.9907**	**0.9969**
Godard et al. [11]	0.0453	0.1743	1.6559	0.1958	0.4524	0.7023	0.8315
Laina et al. [12]	0.03	0.0549	0.3152	0.1033	0.6353	0.8616	0.9412
Liu et al. [29]	0.0312	0.0532	0.3048	0.107	0.603	0.8412	0.9338

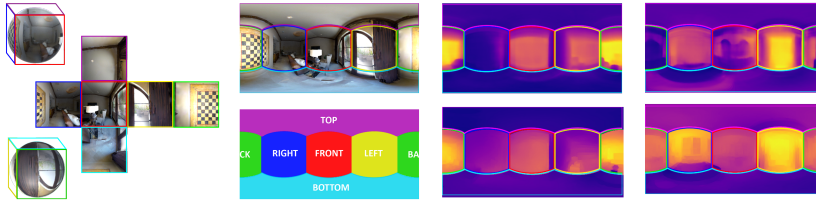

Fig. 7: Cubemap projection (left) and merged monocular predictions (right).

our main models outperform them. However, inferior performance is expected as these were not trained directly in the equirectangular domain but in perspective images. Nonetheless, Laina et al. [12] and Liu et al. [29] achieve much better results than Godard et al. [11]. This is also expected as the latter is trained in an outdoor setting, with very different statistics than our indoor dataset.

For a more fair comparison we use a cubemap projection (Fig. 7 (left)) of all spherical images and then run each model on the projected cube faces which are typical perspective images. After acquiring the predictions, we merge all cube faces' depth maps by projecting them back to the equirectangular domain to be evaluated. However, since the top and bottom cube face projections will be mostly planar, we ignore them during evaluation of all metrics. While monocular performance is improved compared to when applied directly to equirectangular images, their quantitative performance is still inferior to our models. Further, the runtime performance is also worse as multiple inferences need to run, one for each face, incurring a much higher computational cost. Moreover, another apparent issue is the lack of consistency between the predictions of each face. This is shown in Fig. 7 (right) where it is clear that the depth scale of each face are in different scales. This is in line with the observations in [40], but is more pronounced in the depth estimation case, than the style transfer one. Based on this observation, we evaluate each cube face separately against the ground truth values of that face alone which is also median scaled separately. The average values of the front, back, right and left faces for each monocular model against the obtained by our models on the same faces alone are presented in Table 3. Although the performance of the monocular models is further improved, our models still perform better. This can be attributed to various reasons besides training directly on equirectangular domain. One explanation is that 360^o images capture global information which can better help reasoning about relative

depth and overall increase inference performance. The other is that our generated dataset is considerably larger and more diverse than other indoor datasets. In addition, the cube faces are projected out of 512×256 images and are thus, of lower quality / resolution than typical images these models were trained in.

Qualitative Results: To determine how well our models generalize, we examine their performance on completely unseen data found in the Sun360 dataset [72], where no ground truth depth is available. The Sun360 dataset comprises realistic environment captures and has also been used in the work of Yang et al. [36] for room layout estimation. We offer some qualitative results on a data split from [36], referred to as "Room", as well as an additional split of indoor scenes that we select from the Sun360 dataset, referred to as "Indoors". These are presented in Fig. 6 for our two models as well as the monocular ones that were quantitatively evaluated. Our models are able to estimate the scenes' depth with the only monocular model to produce plausible results being the one of Laina et al. [12]. We also observe that UResNet offers smoother predictions than the better performing RectNet , unlike the results when applied on our test split. More qualitative results can be found in the supplementary material where comparison with the method of Yang et al. [36] is also offered.

6 Conclusions

In this work we have presented a learning framework to estimate a scene's depth from a single omnidirectional image. Our models were trained in a completely supervised manner with ground truth depth annotations. To accomplish this, we had to overcome the dataset unavailability for paired 360^o color and depth image pairs, which are difficult to acquire, especially in the scale required for deep learning. This was achieved by re-using large scale 3D datasets with both synthetic and real-world scanned scenes and synthesizing a 360^o dataset via rendering. 360^o depth information can be useful for a variety of tasks, like the composition of 3D elements within spherical content in an automatic, depth scale aware way, further boosting works where this was done manually [73].

Since our approach is the first work for dense 360^o depth estimation, there are many challenges that still need to be overcome. Our datasets only cover indoor cases and are generated with perfect camera vertical alignment with constant lighting and no stitching artifacts. This issue is further accentuated as the scanned datasets had lighting information baked into them during scanning. This can potentially hamper robustness when applied in real world conditions that also contain a much higher dynamic range of luminosity.

For future work, we want to explore unsupervised learning approaches that are based on view synthesis as the supervisory signal. Furthermore, robustness to real world scenes can be achieved, either by utilizing GANs as generators of realistic content, or by using a discriminator to identify plausible realistic images, or even by explicitly learning the depth estimation task in a lighting aware manner.

References

1. Tateno, K., Tombari, F., Laina, I., Navab, N.: Cnn-slam: Real-time dense monocular slam with learned depth prediction. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (July 2017) 6565–6574
2. Mo, K., Li, H., Lin, Z., Lee, J.Y.: The adobeindoornav dataset: Towards deep reinforcement learning based real-world indoor robot visual navigation. (2018)
3. Hedman, P., Alsisan, S., Szeliski, R., Kopf, J.: Casual 3d photography. ACM Transactions on Graphics (TOG) **36**(6) (2017) 234
4. Huang, J., Chen, Z., Ceylan, D., Jin, H.: 6-dof vr videos with a single 360-camera. In: Virtual Reality (VR), 2017 IEEE, IEEE (2017) 37–44
5. Karsch, K., Sunkavalli, K., Hadap, S., Carr, N., Jin, H., Fonte, R., Sittig, M., Forsyth, D.: Automatic scene inference for 3d object compositing. ACM Transactions on Graphics (TOG) **33**(3) (2014) 32
6. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 2650–2658
7. Ren, X., Bo, L., Fox, D.: Rgb-(d) scene labeling: Features and algorithms. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 2759–2766
8. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision second edition. Cambridge University Press (2000)
9. Furukawa, Y., Hernández, C., et al.: Multi-view stereo: A tutorial. Foundations and Trends® in Computer Graphics and Vision **9**(1-2) (2015) 1–148
10. Özyeşil, O., Voroninski, V., Basri, R., Singer, A.: A survey of structure from motion*. Acta Numerica **26** (2017) 305–364
11. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: CVPR. (2017)
12. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper depth prediction with fully convolutional residual networks. In: 3D Vision (3DV), 2016 Fourth International Conference on, IEEE (2016) 239–248
13. Garg, R., BG, V.K., Carneiro, G., Reid, I.: Unsupervised cnn for single view depth estimation: Geometry to the rescue. In: European Conference on Computer Vision, Springer (2016) 740–756
14. Srinivasan, P.P., Garg, R., Wadhwa, N., Ng, R., Barron, J.T.: Aperture supervision for monocular depth estimation. arXiv preprint arXiv:1711.07933 (2017)
15. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Advances in neural information processing systems. (2014) 2366–2374
16. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: CVPR. Volume 2. (2017) 7
17. Li, B., Dai, Y., He, M.: Monocular depth estimation with hierarchical fusion of dilated cnns and soft-weighted-sum inference. arXiv preprint arXiv:1708.02287 (2017)
18. Fu, H., Gong, M., Wang, C., Tao, D.: A compromise principle in deep monocular depth estimation. arXiv preprint arXiv:1708.08267 (2017)
19. Wang, C., Buenaposada, J.M., Zhu, R., Lucey, S.: Learning depth from monocular videos using direct methods. arXiv preprint arXiv:1712.00175 (2017)
20. Xu, D., Ricci, E., Ouyang, W., Wang, X., Sebe, N.: Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In: Proceedings of CVPR. (2017)

21. Ren, Z., Lee, Y.J.: Cross-domain self-supervised multi-task feature learning using synthetic imagery. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018)
22. Jiang, H., Learned-Miller, E., Larsson, G., Maire, M., Shakhnarovich, G.: Self-supervised depth learning for urban scene understanding. arXiv preprint arXiv:1712.04850 (2017)
23. Chakrabarti, A., Shao, J., Shakhnarovich, G.: Depth from a single image by harmonizing overcomplete local network predictions. In: Advances in Neural Information Processing Systems. (2016) 2658–2666
24. Mahjourian, R., Wicke, M., Angelova, A.: Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. arXiv preprint arXiv:1802.05522 (2018)
25. Yang, Z., Wang, P., Xu, W., Zhao, L., Nevatia, R.: Unsupervised learning of geometry with edge-aware depth-normal consistency. arXiv preprint arXiv:1711.03665 (2017)
26. Chen, W., Fu, Z., Yang, D., Deng, J.: Single-image depth perception in the wild. In: Advances in Neural Information Processing Systems. (2016) 730–738
27. Cao, Y., Wu, Z., Shen, C.: Estimating depth from monocular images as classification using deep fully convolutional residual networks. IEEE Transactions on Circuits and Systems for Video Technology (2017)
28. Li, B., Shen, C., Dai, Y., van den Hengel, A., He, M.: Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1119–1127
29. Liu, F., Shen, C., Lin, G., Reid, I.: Learning depth from single monocular images using deep convolutional neural fields. IEEE transactions on pattern analysis and machine intelligence **38**(10) (2016) 2024–2039
30. Li, S.: Binocular spherical stereo. IEEE Transactions on intelligent transportation systems **9**(4) (2008) 589–600
31. Ma, C., Shi, L., Huang, H., Yan, M.: 3d reconstruction from full-view fisheye camera. arXiv preprint arXiv:1506.06273 (2015)
32. Pathak, S., Moro, A., Yamashita, A., Asama, H.: Dense 3d reconstruction from two spherical images via optical flow-based equirectangular epipolar rectification. In: Imaging Systems and Techniques (IST), 2016 IEEE International Conference on, IEEE (2016) 140–145
33. Li, S., Fukumori, K.: Spherical stereo for the construction of immersive vr environment. In: Virtual Reality, 2005. Proceedings. VR 2005. IEEE, IEEE (2005) 217–222
34. Kim, H., Hilton, A.: 3d scene reconstruction from multiple spherical stereo pairs. International Journal of Computer Vision **104**(1) (Aug 2013) 94–116
35. Zhang, Y., Song, S., Tan, P., Xiao, J.: Panocontext: A whole-room 3d context model for panoramic scene understanding. In: European Conference on Computer Vision, Springer (2014) 668–686
36. Yang, H., Zhang, H.: Efficient 3d room shape recovery from a single panorama. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 5422–5430
37. Xu, J., Stenger, B., Kerola, T., Tung, T.: Pano2cad: Room layout from a single panorama image. In: Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on, IEEE (2017) 354–362

38. Kim, H., de Campos, T., Hilton, A.: Room layout estimation with object and material attributes information using a spherical camera. In: 3D Vision (3DV), 2016 Fourth International Conference on, IEEE (2016) 519–527
39. Plagemann, C., Stachniss, C., Hess, J., Endres, F., Franklin, N.: A nonparametric learning approach to range sensing from omnidirectional vision. Robotics and Autonomous Systems **58**(6) (2010) 762–772
40. Ruder, M., Dosovitskiy, A., Brox, T.: Artistic style transfer for videos and spherical images. arXiv preprint arXiv:1708.04538 (2017)
41. Monroy, R., Lutz, S., Chalasani, T., Smolic, A.: Salnet360: Saliency maps for omni-directional images with cnn. arXiv preprint arXiv:1709.06505 (2017)
42. Zhang, J., Lalonde, J.F.: Learning high dynamic range from outdoor panoramas. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 4519–4528
43. Frossard, P., Khasanova, R.: Graph-based classification of omnidirectional images. In: 2017 IEEE International Conference on Computer Vision Workshop (ICCVW), IEEE (2017) 860–869
44. Su, Y.C., Grauman, K.: Learning spherical convolution for fast features from 360 imagery. In: Advances in Neural Information Processing Systems. (2017) 529–539
45. Jeon, Y., Kim, J.: Active convolution: Learning the shape of convolution for image classification. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE (2017) 1846–1854
46. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 764–773
47. Deng, L., Yang, M., Li, H., Li, T., Hu, B., Wang, C.: Restricted deformable convolution based road scene semantic segmentation using surround view cameras. arXiv preprint arXiv:1801.00708 (2018)
48. Cohen, T., Geiger, M., Welling, M.: Convolutional networks for spherical signals. Principled Approaches to Deep Learning Workshop ICML 2017 (2017)
49. Cohen, T.S., Geiger, M., Köhler, J., Welling, M.: Spherical cnns. arXiv preprint arXiv:1801.10130 (2018)
50. Ranftl, R., Vineet, V., Chen, Q., Koltun, V.: Dense monocular depth estimation in complex dynamic scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 4058–4066
51. Liu, M., Salzmann, M., He, X.: Discrete-continuous depth estimation from a single image. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE (2014) 716–723
52. Karsch, K., Liu, C., Kang, S.B.: Depth transfer: Depth extraction from videos using nonparametric sampling. In: Dense Image Correspondences for Computer Vision. Springer (2016) 173–205
53. Yin, Z., Shi, J.: Geonet: Unsupervised learning of dense depth, optical flow and camera pose. arXiv preprint arXiv:1803.02276 (2018)
54. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: European Conference on Computer Vision, Springer (2012) 746–760
55. Saxena, A., Sun, M., Ng, A.Y.: Make3d: Learning 3d scene structure from a single still image. IEEE transactions on pattern analysis and machine intelligence **31**(5) (2009) 824–840
56. Matzen, K., Cohen, M.F., Evans, B., Kopf, J., Szeliski, R.: Low-cost 360 stereo photography and video capture. ACM Transactions on Graphics (TOG) **36**(4) (2017) 148

57. Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M., Funkhouser, T.: Semantic scene completion from a single depth image. IEEE Conference on Computer Vision and Pattern Recognition (2017)
58. Handa, A., Pătrăucean, V., Stent, S., Cipolla, R.: Scenenet: An annotated model generator for indoor scene understanding. In: Robotics and Automation (ICRA), 2016 IEEE International Conference on, IEEE (2016) 5737–5743
59. Armeni, I., Sax, S., Zamir, A.R., Savarese, S.: Joint 2d-3d-semantic data for indoor scene understanding. arXiv preprint arXiv:1702.01105 (2017)
60. Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S.: 3d semantic parsing of large-scale indoor spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 1534–1543
61. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3D: Learning from RGB-D data in indoor environments. International Conference on 3D Vision (3DV) (2017)
62. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 3431–3440
63. Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289 (2015)
64. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. (2015) 448–456
65. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
66. Yu, F., Koltun, V., Funkhouser, T.: Dilated residual networks. In: Computer Vision and Pattern Recognition. Volume 1. (2017)
67. van Noord, N., Postma, E.: Light-weight pixel context encoders for image inpainting. arXiv preprint arXiv:1801.05585 (2018)
68. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research **15**(1) (2014) 1929–1958
69. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22Nd ACM International Conference on Multimedia. MM '14, New York, NY, USA, ACM (2014) 675–678
70. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. Volume 9 of Proceedings of Machine Learning Research., PMLR (13–15 May 2010) 249–256
71. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
72. Xiao, J., Ehinger, K.A., Oliva, A., Torralba, A.: Recognizing scene viewpoint using panoramic place representation. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 2695–2702
73. Rhee, T., Petikam, L., Allen, B., Chalmers, A.: Mr360: Mixed reality rendering for 360 panoramic videos. IEEE transactions on visualization and computer graphics **23**(4) (2017) 1379–1388