Monocular Depth Estimation Primed by Salient Point Detection and Normalized Hessian Loss

Lam Huynh[1] Matteo Pedone[1] Phong Nguyen[1]
Jiri Matas[2] Esa Rahtu[3] Janne Heikkilä[1]

[1]Center for Machine Vision and Signal Analysis, University of Oulu

[2] Center for Machine Perception, Czech Technical University, Czech Republic

[3]Computer Vision Group, Tampere University

Abstract

Deep neural networks have recently thrived on single image depth estimation. That being said, current developments on this topic highlight an apparent compromise between accuracy and network size. This work proposes an accurate and lightweight framework for monocular depth estimation based on a self-attention mechanism stemming from salient point detection. Specifically, we utilize a sparse set of keypoints to train a FuSaNet model that consists of two major components: Fusion-Net and Saliency-Net. In addition, we introduce a normalized Hessian loss term invariant to scaling and shear along the depth direction, which is shown to substantially improve the accuracy. The proposed method achieves state-of-the-art results on NYU-Depth-v2 and KITTI while using 3.1-38.4 times smaller model in terms of the number of parameters than baseline approaches. Experiments on the SUN-RGBD further demonstrate the generalizability of the proposed method.

1. Introduction

Acquiring accurate depth information from 2D images is crucial for computer vision, ranging from robotics, scene understanding, and augmented reality. Traditional multi-view setups [19, 53] obtain accurate results, but measurements are sparse and heavily depend on feature extractions, while active depth sensors are costly. On the other hand, recent learning-based monocular depth estimation approaches [3, 13, 21, 25, 34, 37, 38, 47, 48, 65] have achieved promising results making them potential alternatives to conventional multi-view methods.

Learning-based monocular depth estimation relies on the idea of training a model to predict dense depth values for every RGB pixel. However, training such networks typically requires substantial amount of data and massive network architectures. State-of-the-art methods tend to employ large encoders like ResNet [15, 33, 45, 47], ResNext-101 [66], SeNet-154 [3, 21], Transformer [48, 65], with sophisticated decoder strategies [3, 21, 33], and train with huge dataset such as PBRS [68], *MIX 6* [48] to achieve high accuracy. On the contrary, fast solutions [26, 58] suffer from low precision, manifesting an apparent compromise between accuracy and network size.

Depth completion is a related problem where the aim is to densify a sparse depth map by using machine learning techniques. The sparse depth measurements allow regularizing the nearby depth values, and as a consequence, high accuracy can be achieved with considerably smaller networks [23, 24, 42, 67]. However, the obvious disadvantage of depth completion is the requirement of additional data that is often obtained with active sensors such as LiDARs or ToF cameras.

In this paper, we propose an approach where known depth measurements are replaced with salient points to regularize the depth map. A major benefit compared to depth completion is that these salient points are determined from monocular RGB images while still providing similar advantages as the additional depth data. In this context, salient points are assumed to be image details where the local structure reveals the depth accurately even from a single view. Thus, it can be also considered to be a self-attention mechanism. For this purpose, we first train confidence predictors to highlight important keypoint positions from an RGB image as a confidence map. This map is then used to generate salient points where predicted depth values tend to be more accurate. These points are utilized to enhance the performance of our network similar to depth completion. Moreover, to further assist the network in learning local structures from a single image, we also introduce a normalized Hessian loss term invariant to scaling and shear along the depth direction. In summary, our work makes the following contributions:

- We propose a novel FuSaNet architecture for monocu-

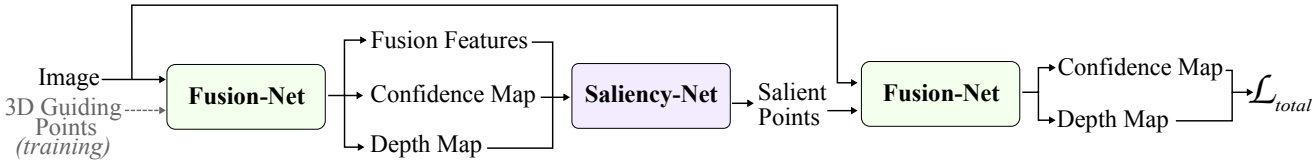

Figure 1. The overall structure of FuSaNet. At training time, the 3D guiding points are sampled from the ground truth depth map at keypoints locations. The Fusion-Net first takes the RGB image and 3D guiding points as inputs to produce a confidence map, a depth map, and a fusion features tensor. These outputs are used to detect the salient points that are fed to the Fusion-Net to generate the final confidence and depth map. At inference time, only the RGB image is required as input to the framework.

lar depth estimation that utilizes a self-attention mechanism based on salient point detection and 3D feature fusion to improve the depth estimation accuracy.

- We utilize a normalized Hessian loss term that is insensitive to the generalized bas-relief (GBR) ambiguity.
- We achieve state-of-the-art results with indoor (NYU-Depth-v2, SUN-RGBD) and outdoor (KITTI) scenes while using significantly fewer parameters than baseline approaches.

Implemention of FuSaNet will be made publicly available upon publication of the paper.

2. Related work

Monocular depth estimation: The interest in single image depth estimation has dramatically increased over the recent years since first proposed by Saxena et al. [50] and Eigen et al. [8, 9]. Pioneering studies obtained accurate depth maps mainly by using large network architectures [3, 21, 33]. Then, Jiao et al. [30] exploited semantic information while Qi et al. [45] utilized the duality between depth and surface normal to improve accuracy. Fu et al. [15] re-defined monocular depth estimation as a classification problem that later on inspired Ren et al. [49] to build their work as a mixture of both tasks. Reflecting the natural scale ambiguity, Lee et al. [35] suggested estimating relative instead of absolute depth values while Facil et al. [13] trained a network to recognize the camera calibration models. Recent depth estimation methods focused on learning the monocular priors such as occlusion [47], planarity both explicitly [37, 38, 66] and implicitly [25, 34]. Gonzalez and Kim [17] proposed to synthesize the right view from the left view for training from stereo images. Yang et al. [65] and Ranftl et al. [48] utilize transformer modules to estimate high-quality depth maps, while in contrast [26, 58] proposed fast depth estimation methods. However, there is a clear trade-off between accuracy and model size.

Depth completion: Depth completion methods produce a dense depth map starting from a set of incomplete depth measurements. Pioneering works from Diebel and Thrun [7] and Hawe et al. [20] proposed using Markov Random Fields or Wavelet analysis to tackle this problem. Then later, Uhrig et al. [55] utilized sparse convolution to build a network taking into account different input sparsities. Jaritz et al. [29] used semantic segmentation while Ma et al. [41] directly concatenated the sparse depth with the RGB image to improve accuracy. Imran et al. [27] introduced depth coefficients, and Xu et al. [64] suggested using surface normals as constraints. Eldesokey et al. [11] utilized confidence to enhance dense depth prediction. Qiu et al. [46] proposed fusing depth and surface normals using adaptive attention, while Chen et al. [4] and Huynh et al. [24] proposed merging appearance and geometry directly in the feature space. Cheng et al. [5, 6] and Park et al. [42] iteratively improved depth prediction using propagation architectures. Nevertheless, these methods are mainly developed to densify depth measurements from range sensors. Inspired by the recent depth completion method [24], we leverage its 3D point fusion architecture for the monocular depth estimation problem.

Attention mechanism: Xu et al. [63] was one of the first works utilizing attention for vision tasks. Later on, attention was implemented as spatial attention [1, 57], channel-wise attention [22, 54], and mix attention [56] to improve classification and detection accuracy. Recent monocular depth estimation methods [25, 32, 36, 39, 62] also applied the attention mechanism. However, these attention implementations require relatively heavy computational resources.

3. Proposed Method

The overall structure of our FuSaNet model is shown in Figure 1. It consists of two major components: Fusion-Net and Saliency-Net. At training time, the Fusion-Net first takes the RGB image and 3D guiding points as inputs to produce a confidence map, a depth map, and a fusion features volume. These intermediate outputs are utilized in Saliency-Net to detect a set of salient points. Both the RGB image and the salient points are fed to the network to produce the final depth prediction. However at testing time, only the RGB image is needed as input to the model.

3.1. Fusion-Net

Inspired by [24], we design the Fusion-Net as a fully convolutional framework as shown in Figure 2. Along with

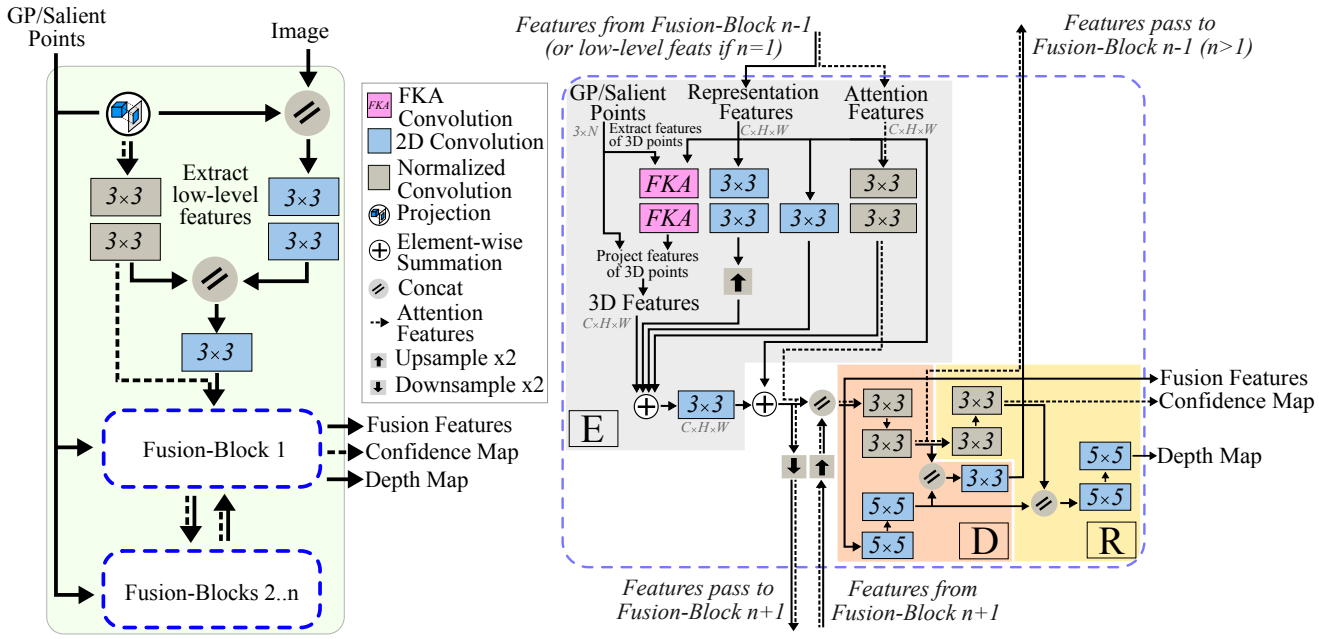

Figure 2. The Fusion-Net (left) consists of five Fusion-Blocks that extract and fuse 2D and 3D features at multiple-scale. The Fusion-Block (right) includes the feature fusion encoder [E], confidence predictor, decoder [D] and refinement [R] modules.

the RGB image, the input 3D points to the Fusion-Net can be the salient points or the 3D guiding points (GP), which the latter is only utilized during training. To generate the GP, we apply SIFT [40] to find keypoints locations from the input image and sample the GP from the ground truth depth map using these keypoint locations. However, one should notice that GP is only an initial guess of the salient points, and the network learns to detect the final points that are reliable in terms of monocular depth.

First, we obtain a sparse depth map and a sparse binary mask from the 2D locations of the input 3D points. The RGBD image is generated by stacking the RGB image with the sparse depth map. Next, two normalized convolutional layers (nConv) [10, 12] and two convolutional layers (Conv2D) are applied to the binary mask, sparse depth, and the RGBD image. Representation features of the two outputs are concatenated and passed to another Conv2D to create the low-level input features along with the attention features from the output of the previous nConvs. The Fusion-Net has five Fusion-Blocks that operate at multi-scale resolutions. Each Fusion-Block consists of a feature fusion encoder, a decoder, a refinement, and a confidence predictor module.

Feature Fusion Encoder. The feature fusion encoder is used to extract and fuse appearance, attention, and geometric features. This module takes a 3D attention tensor, a 3D representation tensor, and a set of salient points as inputs to produce the fusion features and attention volumes with the same shape as the input tensors.

Details of this module are presented in Figure 2 (right) that consists of two 2D branches, one normalized convolution branch, one 3D branch, and one convolutional layer for feature fusion. The 2D branches are convolved at two different resolutions to learn multi-scale representations from the input volume. The normalized convolution branch learns both appearance and attention features utilizing the nConv. The 3D branch aims to extract structural features from the salient points using the feature-kernel alignment convolution (FKAConv) [2]. The representation features from four branches have the same shape as the input tensors that are summed together before applying a Conv2D to output a 3D tensor. Finally, a residual connection is added to avoid vanishing gradient at training time. Meanwhile, the attention features are passed to the decoder module.

Confidence Predictor. The confidence predictor is a sequence of nConvs stretching from the encoder to the refinement module. Unlike the Conv2D, the nConv takes the representation features and the attention features as inputs to estimate a confidence map by propagating the attention volume weighting every pixel by its importance. In addition, output features from the confidence predictor are also used to guide the training of the encoder, decoder, and refinement blocks, as illustrated in Figure 2 (right).

Decoder and Refinement Modules. Many existing deep learning-based methods [3, 14, 21, 48, 59] for monocular depth estimation employ complicated decoders to obtain high levels of accuracy. However, it was shown in [24, 25, 44] that it is feasible to achieve competitive performance utilizing more lightweight and simplistic architectures. Following this line of work, we design our de-

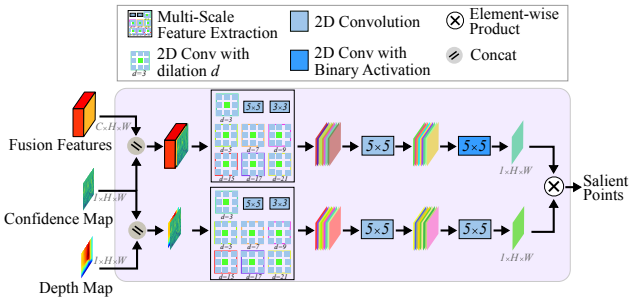

Figure 3. Structure of the Saliency-Net.

coder and refinement module using structure which consists of two branches: the upper branch uses several nConvs to decode learned features and predicts the confidence map, while the lower branch contains a series of Conv2Ds, that along with features from the upper one, is used to predict the depth map. Figure 2 (right) illustrates the structure of our decoder (D, the orange block) and refinement (R, the yellow block) modules.

3.2. Saliency-Net

Figure 3 shows the overall structure of the Saliency-Net. The 3D guiding points (used only at training) and RGB image are first passed through the Fusion-Net to produce a depth map, a confidence map, and a fusion features tensor. This confidence map is utilized for guidance, and it is separately concatenated with the fusion features and the depth map before passing them to the multi-scale feature extraction layer (MFE). The MFE contains two 2D convolutional layers (3x3, 5x5) and seven 3x3 2D atrous convolutions (d = 3, 5, 7, 9, 15, 16, 21) that is used to capture features at various spatial resolutions. Finally, extracted features are fed to convolutional layers and the element-wise product to generate the salient points that serve as an input for the Fusion-Net to predict the final depth map.

3.3. Training Loss Functions

Our loss function consist of normalized Hessian loss, sparse loss, and depth confidence loss terms.

Normalized Hessian loss: 3D reconstruction from a monocular view is inherently an ill-posed problem. There are many ambiguities present and one of them is the generalized bas-relief (GBR) transformation that exists when an unknown object with Lambertian reflectance is viewed orthographically. This is a well-known property that sculpturers have exploited since ancient times to make reliefs more shallow than they actually are. To deal with this ambiguity we observe that at each spatial location, the unit vector obtained by normalizing the three independent elements $\nabla^2 z = (z_{xx},\, z_{xy},\, z_{yy})$ of the Hessian of a depth map z, is invariant to scaling and shears along the z-axis i.e, the GBR transformation [43]. We thus define the quantity H_z based on the normalized Hessian of z as:

$$H_z = \frac{\nabla^2 z}{\|\nabla^2 z\| + \varepsilon} \tag{1}$$

where $\varepsilon = 10^{-20}$ is a small scalar quantity introduced to avoid divisions by zero. The quantities in $\nabla^2 z$ are in practice estimated using Gaussian second derivative filters. Given a predicted depth map $\widehat{d}$ and a ground-truth depth map d, we can measure their dissimilarity by calculating the root mean squared error of $H_{\widehat{d}}$ and H_d, and formulate the following loss function:

$$\mathcal{L}_H(\widehat{d}, d) = \sqrt{\frac{1}{N}\sum_{x,y} \left\| H_{\widehat{d}}(x,y) - H_d(x,y) \right\|^2} \tag{2}$$

where N denotes the total number of pixels of the depth map. Note that since $H_{\widehat{d}}$ and H_d are unit vectors, the term inside the summation in (2) represents the squared chordal distance between two points on the unit sphere. Since the operator H is invariant to linear transformations along the optical axis, it follows that whenever $\widehat{d}$ and d are related by a z-scaling or shear, their dissimilarity $\mathcal{L}_H(\widehat{d}, d)$ will be automatically 0, and consequently, the network will treat scaled versions of the same depth map as an entire equivalence class.

Sparse loss: The $\mathcal{L}_S$ is defined as the ratio between the root mean square error at ground truth sparse depth and all valid depth values. This term is used to minimize the error at sparse depth positions of the estimated and ground-truth depth maps, and is defined as follow:

$$\mathcal{L}_S = \sqrt{\frac{\sum_{j=0}^{M_s}(\hat{d}_j - d_j)^2}{M_s}} \Big/ \sqrt{\frac{\sum_{i=0}^{M}(\hat{d}_i - d_i)^2}{M}} \tag{3}$$

where M_s is the number of ground truth 3D point input and M is the number of valid depth values.

Depth Confidence loss: This loss term is defined as:

$$\mathcal{L}_{DC} = \mathcal{L}_{log} + \mu\mathcal{L}_{grad} + \theta\mathcal{L}_{norm} - \psi\frac{1}{p}\mathcal{L}_C \tag{4}$$

where $\mathcal{L}_{log}$ is a variation of the L_1 norm that minimizes error on the depth pixels, $\mathcal{L}_{grad}$ optimizes the error on edge structures, and $\mathcal{L}_{norm}$ penalizes angular error between the ground truth and predicted normal surfaces. These loss terms were introduced by Hu et al. [21] and widely adopted by state-of-art monocular depth estimation methods [3, 25]. We adopt the confidence loss proposed by [11] where p is the training epoch and $\mathcal{L}_C = C - \mathcal{L}_{log}C$ where C is the predicted confidence map. The full loss function that we utilize is

Table 1. Evaluation on the NYU-Depth-v2 dataset. Metrics with ↓ mean lower is better and ↑ mean higher is better. Methods with ‡ are trained using extra data while ⋆⋆ indicate the use of the whole training set.

Architecture		#params	REL↓	RMSE↓	δ_1↑	δ_2↑	δ_3↑
Eigen & Fergus	Eigen & Fergus'15 [8]⋆⋆	141.1M	0.158	0.641	0.769	0.950	0.988
FCRN	Laina'16 [33]⋆⋆	63.4M	0.127	0.573	0.811	0.953	0.988
PlaneNet	Liu'18 [38]‡	47.5M	0.142	0.514	0.812	0.957	0.989
Detail Preserving-depth	Hao'18 [18]	60.0M	0.127	0.555	0.841	0.966	0.991
Relative-depth	Lee'19 [35]	118.6M	0.131	0.538	0.837	0.971	0.994
DS-SIDENet	Ren'19 [49] ⋆⋆	49.8M	0.113	0.501	0.833	0.968	0.993
PAP-Net	Zhang'19 [69]	95.4M	0.121	0.497	0.846	0.968	0.994
DORN	Fu'19 [15]	110.0M	0.115	0.509	0.828	0.965	0.992
GeoNet	Qi'19 [45]	67.2M	0.128	0.569	0.834	0.960	0.990
SharpNet	Ramam.'19 ‡ [47]	80.4M	0.139	0.502	0.836	0.966	0.993
Revisited mono-depth	Hu'19 [21]	157.0M	0.115	0.530	0.866	0.975	0.993
SARPN	Chen'19 [3]	210.3M	0.111	0.514	0.878	0.977	0.994
VNL	Yin'19 [66]	114.2M	0.108	0.416	0.875	0.976	0.994
BTS	Lee'20 [34]	47.0M	0.110	0.392	0.885	0.978	0.994
DAV	Huynh'20 [25]	25.1M	0.108	0.412	0.882	0.980	0.996
AFDB-Net	Liu'21 [39]	139.2M	0.113	0.504	0.878	0.978	0.995
TransDepth	Yang'21 [65]	311.3M	0.106	0.365	0.900	0.983	0.996
DPT	Ranftl'21‡ [48]	123.9M	0.110	**0.357**	0.904	0.988	**0.998**
FuSaNet	Ours	**8.1M**	**0.104**	0.403	**0.915**	**0.989**	**0.998**

$$\mathcal{L}_{total} = \sum_{i=1}^{n=5} \gamma^i (\beta \mathcal{L}_{DC}^i + \phi \mathcal{L}_S^i + \lambda \mathcal{L}_H^i) \qquad (5)$$

where n is the number of resolution scales and $\gamma^i \in \mathcal{R}^+$ is the loss weight at scale i; $\beta, \phi, \lambda \in \mathcal{R}^+$ are weight loss coefficients. Subsection 4.2 describes in detail how the network is trained using these loss functions.

4. Experiments

In this section, we first describe the datasets and evaluation metrics that are used in our experiments followed by the implementation details of our model. The rest of this section presents a comparison with the state-of-the-art, ablation studies, and qualitative results.

4.1. Dataset and Metrics

We evaluate our proposed model on three datasets: NYU-Depth-v2 [51], KITTI [16], and SUN-RGBD [28, 52, 60]. NYU-Depth-v2 includes 120K RBG-D images captured from 464 indoor scenes, and we sample 50K images from the entire dataset to train and 654 test images of 215 scenes to test our model. To evaluate on KITTI outdoor driving dataset, we use the standard Eigen split [8, 9] for training (39K images) and testing (697 images). SUN-RGBD contains more than 10K images from a variety of indoor scenarios. We test our pre-trained model on 5050 images from its test set without any fine-tuning for this dataset. We follow the previous methods [8, 9] for evaluation by using the following metrics: mean relative error (REL), root mean square error (RMSE), thresholded accuracy(δ_i), scale-invariant mean square error (SI), mean absolute error (iMAE) and root mean square error (iRMSE) of the inverse depth values.

4.2. Implementation Details

The proposed model is trained for 150 epochs on a single TITAN RTX using batch size of 32, and the Adam optimizer [31] with $(\beta_1, \beta_2, \epsilon) = (0.9, 0.999, 10^{-8})$. We use all of the loss terms in Eq. 5 to train the NYU-Depth-v2 dataset while using the L_{log} and the L_S for training KITTI. The initial learning rate is $7 * 10^{-4}$, but from epoch 10 the learning is reduced by 5% per 5 epochs. We set the number of scales n in Eq. 5 to 5, weight loss coefficients μ, θ, β, ψ to 1.0; λ to 0.01; ϕ to 5.0 for NYU-Depth-v2 and 20.0 for KITTI. The scale weight losses $\gamma^1, \gamma^2, \gamma^3, \gamma^4, \gamma^5$ to $1.0, 0.75, 0.5, 0.25$ and 0.125 respectively. During training, we augment the input RGB and ground truth depth images using random rotation ([-5.0, +5.0] degrees), horizontal flip, rectangular window droppings, and colorization (RGB only). We train the network using 3D points sampled from ground truth depth maps at keypoint locations and for some random iteration

Table 2. Evaluation on the KITTI dataset. Metrics with ↓ mean lower is better and ↑ mean higher is better.

Method	#params	REL↓	RMSE↓	δ_1↑	δ_2↑	δ_3↑
DORN [15]	110.0M	0.072	2.626	0.932	0.984	0.994
AFDB-Net [39]	139.2M	0.071	2.848	0.933	0.983	0.995
VNL [66]	114.2M	0.072	3.258	0.938	0.990	0.998
BTS [34]	112.8M	**0.059**	2.756	0.956	0.993	0.998
PGA-Net [61]	168.3M	0.063	2.634	0.952	0.992	0.998
TransDepth [65]	311.3M	0.064	2.755	0.956	0.994	**0.999**
DPT [48]	123.9M	0.062	2.573	0.959	**0.995**	**0.999**
FuSaNet (Ours)	**8.1M**	**0.059**	**2.487**	**0.964**	**0.995**	**0.999**

Table 3. Cross-dataset evaluation with training on NYU-Depth-v2 and testing on SUN-RGBD.

Models	#params	REL	sqREL	SI	iMAE	iRMSE
Hu et al. [21]	157.0M	0.245	0.389	0.031	0.108	0.087
SARPN [3]	210.3M	0.243	0.393	0.031	**0.102**	0.069
DAV [25]	25.1M	0.238	0.387	0.030	0.104	0.075
DPT [48]	123.9M	0.230	**0.341**	**0.029**	0.103	**0.068**
FuSaNet (Ours)	**8.1M**	**0.225**	0.350	**0.029**	**0.102**	0.070

feeding zero points during training. At testing time, the model only takes the RGB image as input.

4.3. Performance Analysis

Tables 1, 2 and 3 present the quantitative comparison and the number of model parameters for our method and state-of-the-art approaches. The results show that, the proposed method, while being the smallest model, achieves comparable figures against the baselines.

In case of NYU-Depth-v2, the best approaches including TransDepth [65], SARPN [3], DPT [48], VNL [66], BTS [34], and DAV [25] use 38.4, 25.9, 15.3, 14.1, 5.8 and 3.1 times more parameters in contrast to ours, respectively. Compared to DPT [48], our depth maps have difficulties in tiny detailed structures such as leaves, legs of the chair. Nevertheless, our model produces high-quality depth maps compared to state-of-the-art approaches that employ large network [3, 21, 25, 34, 47, 48, 66] and train with a large amount of extra data [47, 48], as shown in Figure 7. Furthermore, the proposed method model performs well in uniform regions and large furniture like bookshelves, tables.

For KITTI, our approach performs on par with state-of-the-art methods while being (at least 13.5 times) more compact in terms of the number of parameters. As shown in Figure 4, our model yields high quality depth predictions, especially at object boundaries and contours.

Moreover, to assess the generalizability of our network, we perform a cross-dataset evaluation, where we train the model using NYU-Depth-v2 and test with SUN-RGBD without any fine-tuning. We also evaluate the methods from [3, 21, 25, 48] and present the results in Table 3 and Figure 5. As can be seen our model performs favourably compared to baseline approaches.

4.4. Ablation Studies

Effectiveness of the Saliency-Net: We implement models with and without the Saliency-Net to assess its effect to the performance. As presented in Table 4, applying the proposed module significantly improves the accuracy while increasing the runtime.

Table 4. Ablation studies of models without and with the Saliency-Net on the NYU-Depth-v2 dataset. Frame rate *(fps)* is measured using one GTX-1080 GPU.

Training	Frame rate↑	REL↓	RMSE↓	δ_1↑	δ_2↑
w/o Saliency-Net	**327**	0.122	0.494	0.857	0.971
w/ Saliency-Net	172	**0.104**	**0.403**	**0.915**	**0.989**

Table 5. Ablation studies of models using the 3D guiding points (GP) during training on NYU-Depth-v2. FuSaNet2 is trained using the RGB image and GP while FuSaNet1 uses only RGB information. FuSaNet3 is also trained with GP but for some random iteration no points are fed to the network at training time.

Training	REL↓	RMSE↓	δ_1↑	δ_2↑	δ_3↑
FuSaNet1	0.119	0.474	0.857	0.978	0.994
FuSaNet2	0.105	0.416	0.912	0.988	0.996
FuSaNet3	**0.104**	**0.403**	**0.915**	**0.989**	**0.998**

3D guiding points at training time: We experiment with three different schemes to study the impact of using the 3D guiding points at training time and report the results in Table 5. *FuSaNet1* is the model trained with only RGB images, while *FuSaNet2* also utilized the 3D guiding points as the input. The results show that *FuSaNet2* significantly outperforms *FuSaNet1* since the model trained with 3D guiding points is better at exploiting monocular priors from RGB images even when no points are fed to the network during testing. Moreover, we experiment with *FuSaNet3* by adopting a similar training procedure to *FuSaNet2*, except that we feed zero points to the network for some random iterations. This training procedure further robustifies our model to the absence of 3D points, as shown in Table 5.

Confidence predictor (CP): We conduct experiments with and without the confidence predictor to analyze how this module affects the performance by following the training scheme of *FuSaNet3* in Table 5. As shown in Table 6, the results clearly demonstrate the benefit of the CP block. Furthermore, we observe that the CP learns to highlight important locations from RGB images either with or without input 3D points at the inference time, as presented in the confidence maps (c_c, d_c) and salient points (c_s, d_s) in Figure 6. This, in turn, helps the proposed network to produce high-quality depth maps (d_d, Figure 6) with less distortion than the model without the CP module (e, Figure 6).

Training losses: We also study the impact of different loss terms by training our method with various settings and report the results in Table 7. Models that incorporated the normalized Hessian loss show clear improvements in all metrics.

Table 6. Ablation studies of models without and with the confidence predictor (CP) on NYU-Depth-v2.

Training	REL↓	RMSE↓	δ_1↑	δ_2↑	δ_3↑
w/o CP	0.126	0.530	0.843	0.967	0.992
w/ CP	**0.104**	**0.403**	**0.915**	**0.989**	**0.998**

Table 7. Ablation studies of different loss terms on the NYU-Depth-v2 dataset. Results are obtained from one iteration.

Training	REL↓	RMSE↓	δ_1↑	δ_2↑	δ_3↑
FuSaNet + L_{DC}	0.119	0.452	0.871	0.976	0.994
FuSaNet + L_{DC} + L_S	0.116	0.451	0.872	0.977	0.994
FuSaNet + L_{DC} + L_H	0.111	0.429	0.883	0.980	0.996
FuSaNet + L_{DC} + L_S + L_H	**0.104**	**0.403**	**0.915**	**0.989**	**0.998**

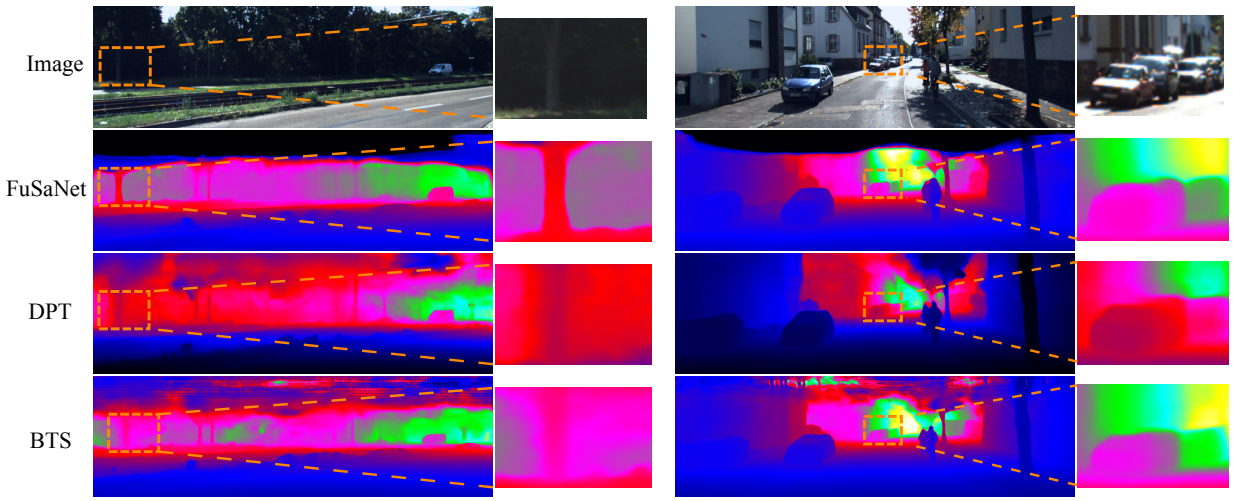

Figure 4. Comparison with BTS [34] and DPT [48] approaches on the KITTI dataset. Each example contains an image or a predicted depth map (left) with a zoom-in view (right) for visualization.

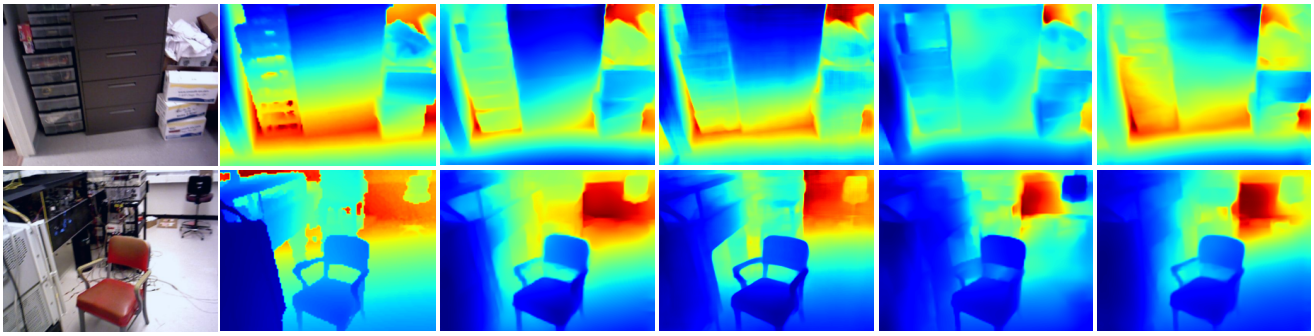

Figure 5. Cross-dataset evaluation on SUN RGB-D dataset. SARPN [3], DAV [25] and FuSaNet models were trained on NYU-Depth-v2 while DPT [48] was trained on *MIX 6* [48] before fine-tuning on NYU-Depth-v2.

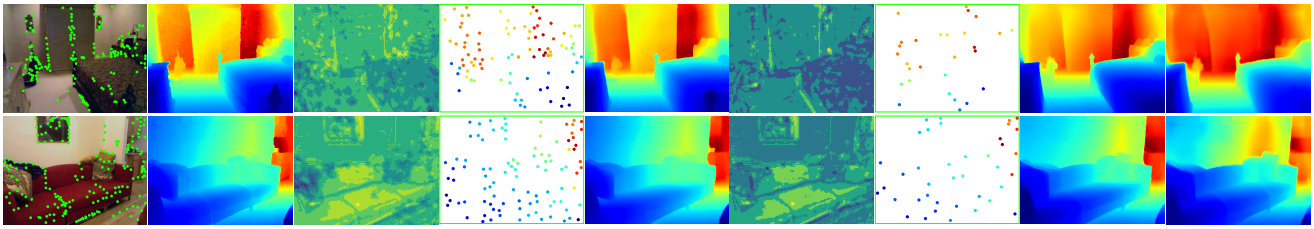

Figure 6. Examples from the NYU-Depth-v2 test set. From left to right: (a) RGB images and 3D guiding points inputs; (b) ground truth depth; (c_c) confidence map, (c_s) salient points, and (c_d) final predicted depth of model using the RGB image and 500 guiding points as inputs; (d_c), (d_s), (d_d) similar results of model using only the RGB image as input; and (e) predicted depth of model without the confidence predictor and using only the RGB image as input. (Points are enhanced for visualization)

5. Conclusion

This paper proposed a novel saliency-based self-attention mechanism and a normalized Hessian loss function to estimate high-quality depth prediction for indoor and outdoor environments. The proposed method achieves competitive performance while being at least three times more compact than state-of-the-art approaches. Our work provides a potential approach toward optimizing accuracy and network size for dense depth estimation without the need for using active depth sensors or multiple view geometry.

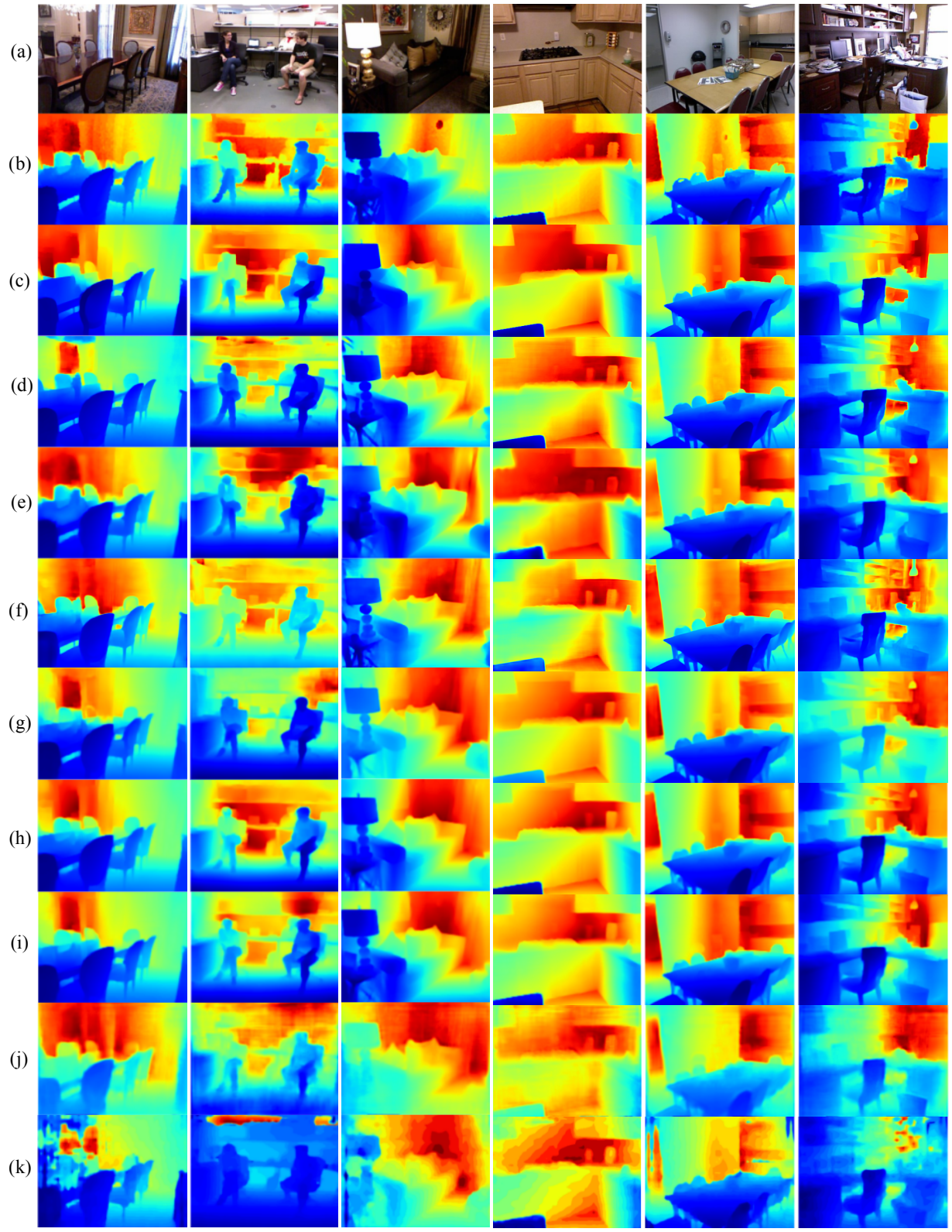

Figure 7. Examples from the NYU-Depth-v2 test set. (a) Input images, (b) ground truth depth. Results from (c) FuSaNet, (d) DPT [48], (e) DAV [25], (f) BTS [34], (g) VNL [66], (h) SARPN [3], (i) Hu et al. [21], (j) SharpNet [47], and (k) DORN [15].

References

[1] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3286–3295, 2019. 2

[2] Alexandre Boulch, Gilles Puy, and Renaud Marlet. FKAConv: Feature-Kernel Alignment for Point Cloud Convolution. In *15th Asian Conference on Computer Vision (ACCV 2020)*, 2020. 3

[3] Xiaotian Chen, Xuejin Chen, and Zheng-Jun Zha. Structure-aware residual pyramid network for monocular depth estimation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 694–700. AAAI Press, 2019. 1, 2, 3, 4, 5, 6, 7, 8

[4] Yun Chen, Bin Yang, Ming Liang, and Raquel Urtasun. Learning joint 2d-3d representations for depth completion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10023–10032, 2019. 2

[5] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–119, 2018. 2

[6] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning depth with convolutional spatial propagation network. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 2

[7] James Diebel and Sebastian Thrun. An application of markov random fields to range sensing. In *Advances in neural information processing systems*, pages 291–298, 2006. 2

[8] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2650–2658, 2015. 2, 5

[9] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014. 2, 5

[10] Abdelrahman Eldesokey, Michael Felsberg, Karl Holmquist, and Michael Persson. Uncertainty-aware cnns for depth completion: Uncertainty from beginning to end. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12014–12023, 2020. 3

[11] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Confidence propagation through cnns for guided sparse depth regression. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2423–2436, 2019. 2, 4

[12] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Propagating confidences through cnns for sparse data regression. In *The 29th British Machine Vision Conference (BMVC), Northumbria University, Newcastle upon Tyne, England, UK, 3-6 September, 2018*. BMVA Press, 2019. 3

[13] Jose M Facil, Benjamin Ummenhofer, Huizhong Zhou, Luis Montesano, Thomas Brox, and Javier Civera. Camconvs: camera-aware multi-scale convolutions for single-view depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11826–11835, 2019. 1, 2

[14] Zhicheng Fang, Xiaoran Chen, Yuhua Chen, and Luc Van Gool. Towards good practice for cnn-based monocular depth estimation. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1091–1100, 2020. 3

[15] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018. 1, 2, 5, 8

[16] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 5

[17] Juan Luis GonzalezBello and Munchurl Kim. Forget about the lidar: Self-supervised depth estimators with med probability volumes. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[18] Zhixiang Hao, Yu Li, Shaodi You, and Feng Lu. Detail preserving depth estimation from a single image using attention guided networks. In *2018 International Conference on 3D Vision (3DV)*, pages 304–313. IEEE, 2018. 5

[19] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1

[20] Simon Hawe, Martin Kleinsteuber, and Klaus Diepold. Dense disparity maps from sparse disparity measurements. In *2011 International Conference on Computer Vision*, pages 2126–2133. IEEE, 2011. 2

[21] Junjie Hu, Mete Ozay, Yan Zhang, and Takayuki Okatani. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In *IEEE Winter Conf. on Applications of Computer Vision (WACV)*, 2019. 1, 2, 3, 4, 5, 6, 8

[22] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018. 2

[23] Mu Hu, Shuling Wang, Bin Li, Shiyu Ning, Li Fan, and Xiaojin Gong. Towards precise and efficient image guided depth completion. *ICRA*, 2021. 1

[24] Lam Huynh, Phong Nguyen, Jiri Matas, Esa Rahtu, and Janne Heikkila. Boosting monocular depth estimation with lightweight 3d point fusion. *arXiv preprint arXiv:2012.10296*, 2020. 1, 2, 3

[25] Lam Huynh, Phong Nguyen-Ha, Jiri Matas, Esa Rahtu, and Janne Heikkilä. Guiding monocular depth estimation using depth-attention volume. In *European Conference on Computer Vision*, pages 581–597. Springer, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[26] Andrey Ignatov, Grigory Malivenko, David Plowman, Samarth Shukla, Radu Timofte, Ziyu Zhang, Yicheng Wang, Zilong Huang, Guozhong Luo, Gang Yu, et al. Fast and accurate single-image depth estimation on mobile devices, mobile ai 2021 challenge: Report. *arXiv preprint arXiv:2105.08630*, 2021. 1, 2

[27] Saif Imran, Yunfei Long, Xiaoming Liu, and Daniel Morris. Depth coefficients for depth completion. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12438–12447. IEEE, 2019. 2

[28] Allison Janoch, Sergey Karayev, Yangqing Jia, Jonathan T Barron, Mario Fritz, Kate Saenko, and Trevor Darrell. A category-level 3d object dataset: Putting the kinect to work. In *Consumer Depth Cameras for Computer Vision*, pages 141–165. Springer, 2013. 5

[29] Maximilian Jaritz, Raoul De Charette, Emilie Wirbel, Xavier Perrotton, and Fawzi Nashashibi. Sparse and dense data with cnns: Depth completion and semantic segmentation. In *2018 International Conference on 3D Vision (3DV)*, pages 52–60. IEEE, 2018. 2

[30] Jianbo Jiao, Ying Cao, Yibing Song, and Rynson Lau. Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 53–69, 2018. 2

[31] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[32] Shu Kong and Charless Fowlkes. Pixel-wise attentional gating for scene parsing. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1024–1033. IEEE, 2019. 2

[33] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 239–248. IEEE, 2016. 1, 2, 5

[34] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019. 1, 2, 5, 6, 7, 8

[35] Jae-Han Lee and Chang-Su Kim. Monocular depth estimation using relative depth maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2019. 2, 5

[36] Ruibo Li, Ke Xian, Chunhua Shen, Zhiguo Cao, Hao Lu, and Lingxiao Hang. Deep attention-based classification network for robust depth prediction. In *Asian Conference on Computer Vision*, pages 663–678. Springer, 2018. 2

[37] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. Planercnn: 3d plane detection and reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4450–4459, 2019. 1, 2

[38] Chen Liu, Jimei Yang, Duygu Ceylan, Ersin Yumer, and Yasutaka Furukawa. Planenet: Piece-wise planar reconstruction from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2579–2588, 2018. 1, 2, 5

[39] Peng Liu, Zonghua Zhang, Zhaozong Meng, and Nan Gao. Monocular depth estimation with joint attention feature distillation and wavelet-based loss function. *Sensors*, 21(1):54, 2021. 2, 5

[40] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 3

[41] Fangchang Mal and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018. 2

[42] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In *Proc. of European Conference on Computer Vision (ECCV)*, 2020. 1, 2

[43] Matteo Pedone, Abdelrahman Mostafa, and Janne Heikkilä. Learning non-rigid surface reconstruction from spatio-temporal image patches. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10134–10140. IEEE, 2021. 4

[44] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. Towards real-time unsupervised monocular depth estimation on cpu. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5848–5854. IEEE, 2018. 3

[45] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 283–291, 2018. 1, 2, 5

[46] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[47] Michael Ramamonjisoa and Vincent Lepetit. Sharpnet: Fast and accurate recovery of occluding contours in monocular depth estimation. *The IEEE International Conference on Computer Vision (ICCV) Workshops*, 2019. 1, 2, 5, 6, 8

[48] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ArXiv preprint*, 2021. 1, 2, 3, 5, 6, 7, 8

[49] Haoyu Ren, Mostafa El-khamy, and Jungwon Lee. Deep robust single image depth estimation neural network using scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 37–45, 2019. 2, 5

[50] Ashutosh Saxena, Sung H Chung, and Andrew Y Ng. Learning depth from single monocular images. In *Advances in neural information processing systems*, pages 1161–1168, 2006. 2

[51] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012. 5

[52] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015. 5

[53] Richard Szeliski. Structure from motion. In *Computer Vision*, pages 303–334. Springer, 2011. 1

[54] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2820–2828, 2019. 2

[55] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *2017 international conference on 3D Vision (3DV)*, pages 11–20. IEEE, 2017. 2

[56] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2017. 2

[57] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018. 2

[58] Diana Wofk, Fangchang Ma, Tien-Ju Yang, Sertac Karaman, and Vivienne Sze. Fastdepth: Fast monocular depth estimation on embedded systems. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6101–6108. IEEE, 2019. 1, 2

[59] Zbigniew Wojna, Vittorio Ferrari, Sergio Guadarrama, Nathan Silberman, Liang-Chieh Chen, Alireza Fathi, and Jasper Uijlings. The devil is in the decoder: Classification, regression and gans. *International Journal of Computer Vision*, 127(11-12):1694–1706, 2019. 3

[60] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1625–1632, 2013. 5

[61] Dan Xu, Xavier Alameda-Pineda, Wanli Ouyang, Elisa Ricci, Xiaogang Wang, and Nicu Sebe. Probabilistic graph attention network with conditional kernels for pixel-wise prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 5

[62] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3917–3925, 2018. 2

[63] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015. 2

[64] Yan Xu, Xinge Zhu, Jianping Shi, Guofeng Zhang, Hujun Bao, and Hongsheng Li. Depth completion from sparse lidar data with depth-normal constraints. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2811–2820, 2019. 2

[65] Guanglei Yang, Hao Tang, Mingli Ding, Nicu Sebe, and Elisa Ricci. Transformers solve the limited receptive field for monocular depth prediction. *arXiv preprint arXiv:2103.12091*, 2021. 1, 2, 5, 6

[66] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 1, 2, 5, 6, 8

[67] Yinda Zhang and Thomas Funkhouser. Deep depth completion of a single rgb-d image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 175–185, 2018. 1

[68] Yinda Zhang, Shuran Song, Ersin Yumer, Manolis Savva, Joon-Young Lee, Hailin Jin, and Thomas Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5287–5295, 2017. 1

[69] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4106–4115, 2019. 5