A Multi-Scale Guided Cascade Hourglass Network for Depth Completion

Ang Li[1] Zejian Yuan[1] Yonggen Ling[2] Wanchao Chi[2] Shenghao Zhang[2] Chong Zhang[2]
[1]Institute of Artificial Intelligence and Robotics, Xian Jiaotong University, China
[2]Tencent Robotics X, China

bennie.522@stu.xjtu.edu.cn yuan.ze.jian@xjtu.edu.cn
ylingaa@connect.ust.hk wanchaochi@tencent.com popshzhang@pku.edu.cn aerentzhang@gmail.com

Abstract

Depth completion, a task to estimate the dense depth map from sparse measurement under the guidance from the high-resolution image, is essential to many computer vision applications. Most previous methods building on fully convolutional networks can not handle diverse patterns in the depth map efficiently and effectively. We propose a multi-scale guided cascade hourglass network to tackle this problem. Structures at different levels are captured by specialized hourglasses in the cascade network with sparse inputs in various sizes. An encoder extracts multi-scale features from color image to provide deep guidance for all the hourglasses. A multi-scale training strategy further activates the effect of cascade stages. With the role of each sub-module divided explicitly, we can implement components with simple architectures. Extensive experiments show that our lightweight model achieves competitive results compared with state-of-the-art in KITTI depth completion benchmark, with low complexity in run-time.

1. Introduction

Accurate depth sensing is a critical basic component for many computer vision applications, including autonomous navigation, augmented realities, and unmanned aerial vehicles. A popular solution to these tasks is based on active depth sensors, given their precise measurements. Unfortunately, commodity-level sensors, such as LiDAR and Time-of-Flight sensor, barely produce sparse depth maps with considerable missing data. For instance, projecting 3D point cloud captured by Velodyne LiDAR HDL-64e into an image plane results in roughly 4% valid pixels on a depth map in KITTI dataset [21]. The scarcity of depth measurements makes it hard to meet the demands of high-level applications. As a consequence, a task of depth completion, i.e., to estimate the dense depth map from sparse input depth with the guidance from the aligned high-resolution camera image, catches growing attention in recent years.

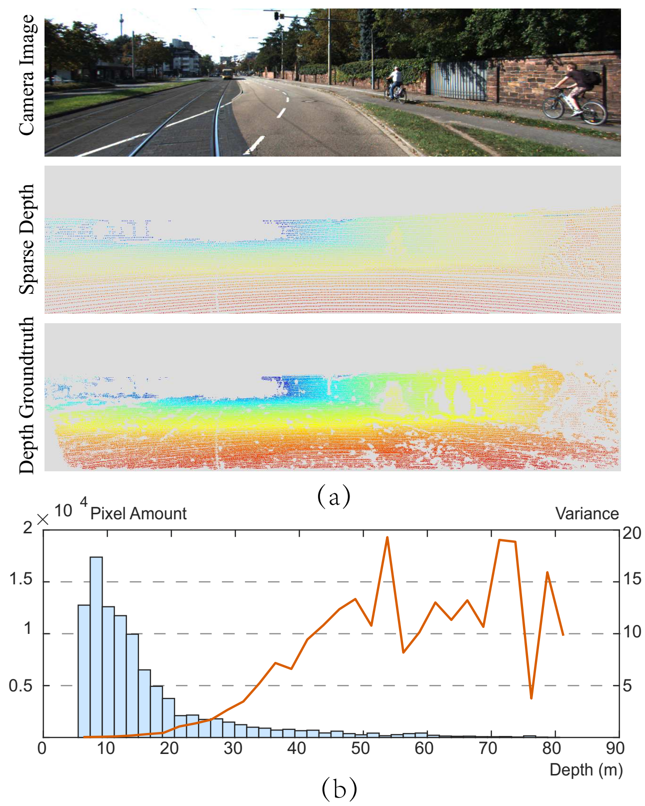

Figure 1. **Example Data for Depth Completion in KITTI dataset.** (a) Form top to bottom: the guidance color image, the input sparse depth which is dilated for visualization, and the groundtruth depth. In the camera image space, distant objects are shrunk, while near objects are enlarged due to perspective projection. (b) Far areas have larger depth variance (computed in a 9×9 window) but counts less proportion of pixels than the near areas do.

In addition to high accuracy requirement, a depth completion method always needs to have low demands for runtime and a small number of parameters in a practical system. Most recent methods [21, 23, 10] take advantage of convolutional neural networks (CNN) to tackle the problem. A common convolution kernel is shared spatially over the whole image so that the computation can be processed efficiently in parallel. One of the most popular backbones is the hourglass-shaped fully convolutional network (FCN) [10, 5, 16]. The contractive part in the hourglass traverses a large scope of field progressively by a series of down-sampling convolutions. The expanding part receives multi-scale features from the contractive part by skip connections to cope with diverse structures.

Although previous studies have achieved promising results, the plain hourglass architecture is actually not optimal for depth completion. This is mainly because of the extremely imbalanced distribution of scene structures across the camera image space, largely due to perspective projection. As shown in Fig. 1 (a), distant objects are shrunk and tend to appear as relatively fine structures compared to near parts. The deformation causes more severe variation in depth but much fewer supporting data points in far regions than those in the near, as evidenced by Fig 1 (b). In this situation, vanilla CNN based methods applying spatially shared filters can not fit heterogeneous patterns across the image. Hourglass networks collecting multi-scale features have the ability to cover diverse structures, while practical learning performance is not satisfactory. The imbalance in structures make layers for a certain scale may not be adequately trained to be effective. Consequently, a successful hourglass always depends on massive parameters trained with extensive data [13, 16, 10].

To alleviate the problem, we propose a **M**ulti-**S**cale **G**uided **C**ascade **H**ourglass **N**etwork (**MSG-CHN**). Instead of a single network bearing all the features, we employ cascade hourglass networks, which has been applied successfully to human pose estimation [14]. Different from these stacked hourglasses, we designate each sub-network to predict structures on a particular scale by feeding inputs at different resolutions. Multi-scale features from the high-resolution color image are extracted accordingly to provide guidance information for specific structures. By dividing sub-modules for refined tasks, the suppression effect among layers for different structures can be reduced. Furthermore, the redundant network can be replaced with a compact combination of simple architectures.

Specifically, we feed sparse inputs in the quarter, half, and full sizes to three stacked hourglasses, respectively. With the inputs down-sampled to different resolutions, we can implement component sub-networks with simple and similar architectures, whose roles are promoted by inputs. Even more valuable is that the run-time complexity can be considerably reduced.

As for the image guidance information, we apply a single deep encoder with layers of down-sampling convolutions on the full-resolution color image. Considering that the RGB image is densely informative and full of details, we do not perform manually-designed down-sampling as how we treat the sparse depth. To further reduce the computational complexity, the RGB encoder only operates once to extract all scales of features for three hourglasses. Guidance features are integrated in a "late" manner into every encoder in the cascade hourglass networks.

In order to activate the role of each sub-network, we apply a multi-scale training strategy. Each sub-module is trained to predict the depth at the corresponding resolution. With the intermediate supervision, cascade networks can be sufficiently trained into effect.

The major contributions and achievements of this paper are three-fold:

- We are, to the best of our knowledge, the first to apply cascade hourglass networks to handle the multi-scale structures for depth completion.
- We propose to extract multi-scale guidance information from image for the cascade networks. Ground-truth maps are down-sampled to the corresponding scales for auxiliary supervision.
- Our compact network achieves competitive accuracy with low run-time complexity compared to state-of-the-art methods on KITTI depth completion benchmark.

2. Related Work

Depth Completion

Most depth completion methods fall into two groups. One group focuses on how to dig out useful information from the sparse data. Uhrig *et al.* [21] propose a sparse convolution to explicitly consider the sparsity of input by only evaluating valid positions, which makes their model invariant to the level of sparsity. Huang *et al.* [8] extend the concept of sparsity-invariant convolution to other operations including summation, up-sampling, and concatenation, so that they can implement a multi-scale network HMS-Net. Eldesokey *et al.* [5] as well as Hua and Gong [7] replace the binary mask in the sparsity-invariant convolution with continuous confidence. Inspired by these works, we extend the average pooling operation to down-sample the sparse input.

The other group of methods investigates the proper way to integrate multi-modality data, i.e., the RGB image with depth measurements, to facilitate the completion. Compared to dealing with the sparse depth with RGB image together, previous works [8, 10, 16] agree on that combining them late at the feature level is a better choice. Some methods extract specific features from an image to assist in dense estimation. Qiu *et al.* [16] learn surface normals as the intermediate representation. Besides surface normals, Zhang and Funkhouser [23] estimate occlusion boundaries as well to better constrain the scene structures. MSG-Net [9] extracts multi-scale features to guide depth super-resolution

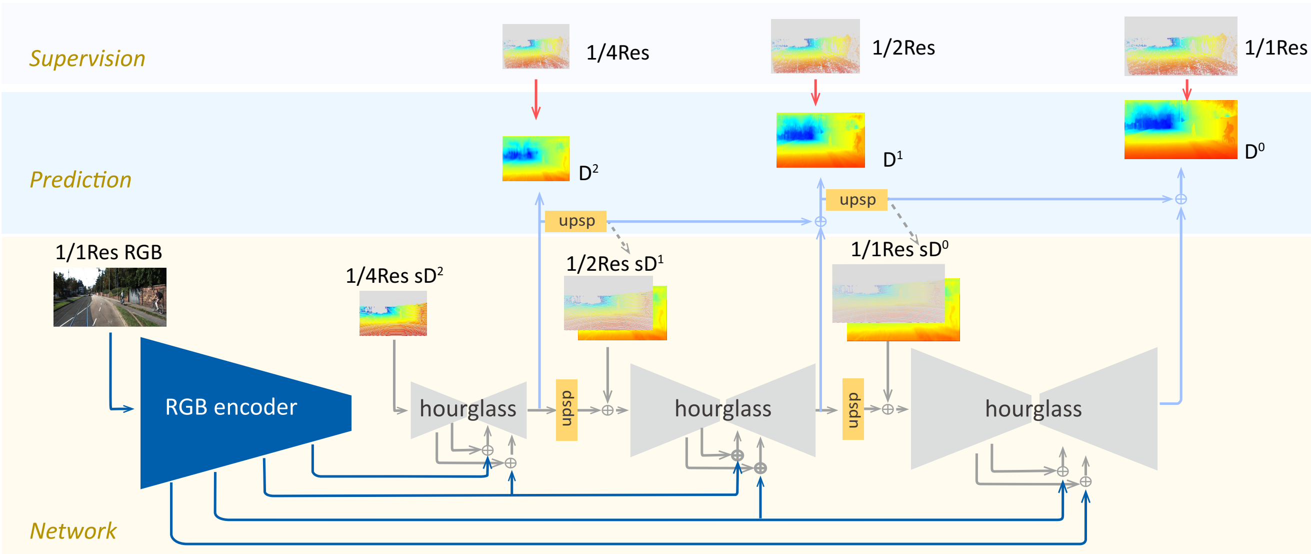

Figure 2. **Our Depth Completion Pipeline.** The sparse depth map is first resized to low- and medium- resolutions. Down-sampled maps together with the original one are fed into three cascade hourglass networks. A color image passes through an encoder to predict multi-scale guidance for each hourglass. Afterward, estimations from each hourglass are integrated by residual connections. Each network is trained with ground-truth depth down-sampled to the corresponding scale.

at different levels progressively. We follow them to employ layers of down-sampling convolutions on the image. Differently, we arrange the features in a principled way to guide all the sub-modules of different scales in the cascade network.

Multi-Scale Networks for Dense Estimation

Pixel-wise inference tasks, such as semantic segmentation and depth prediction, widely employ multi-scale networks. One kind of method extracts and integrates multi-scale features in a unified network. Encoder-decoder architecture with skip connections [19, 12] is one of the most popular backbones, which contains an encoder to extract multi-scale features sequentially and a decoder to collect features through skip connections. Some methods [25, 1, 20] utilize the spatial pyramid pooling (SPP) block to learn a coarse-to-fine representation via kernels in different spatial sizes. Nevertheless, training such networks sufficiently is challenging and always takes a long time.

The other type of methods resize the input to several resolutions and assign them to sub-networks. Each sub-network is in charge of a certain scale, and the final prediction is a combination of the multi-scale outputs. Chen *et al.* [2] deploy sub-networks in parallel, together with an attention model to merge multiple estimations. Eigen and Fergus [3] use stacked sub-networks to recover pixel-wise labels from the coarse-scale to the fine. Our architecture also contains stacked networks to fit diverse structures, while sub-networks are guided with multi-scale features from the color image. Furthermore, the function of our sub-networks is explicitly restricted during training by a multi-scale training strategy.

Real-time Pixel-Wise Prediction

Pixel-wise prediction methods achieve real-time performance in two ways. Some methods [15, 18, 17] speed up a model by pruning parameters. However, the high-speed brought by the lightweight model is at the cost of accuracy. Zhao *et al.* [24] argue that the computation complexity is related to the feature resolution. They propose an ICNet with cascade down-sampled inputs to achieve real-time inference without sacrificing much performance. Our network shares the same merit with ICNet. We properly take advantage of low-resolution sparse inputs for encoding coarse features. The design significantly reduces the demanding of run-time with little sacrifice on the capacity of the model.

3. Model

3.1. Overview

Given a sparse depth map sD and a guidance RGB image I which is aligned with sD, we are aiming at recovering a dense depth map. We proposed a multi-scale guided cascade hourglass network (MSG-CHN) to tackle the problem. The pipeline is illustrated in Fig. 2. The network treats the sparse depth and the image input in two different manners. Three cascade hourglasses take the quarter-sized sparse map sD^2, half-sized sD^1 and full-sized sD^0 as input respectively to capture structures in different scales. An RGB encoder with layers of down-sampling convolutions applies on image I for multi-scale guidance features. Image and depth features are coordinated at each encoder in the depth pathway. Residual connections integrate predictions from three hourglasses, to recover dense depth gradually from the coarsest D^2 to the finer D^1 and finally reaching the prediction D^0. We summarize the detailed layer-by-layer network configurations in Table. 1.

3.2. Cascade Hourglass Network

The depth pathway consists of three cascade hourglass modules, each takes a certain resolution of depth map input and gives predictions at the same resolution. At the beginning, 1/4-sized sparse depth is fed into the first hourglass to predict the coarsest scale of features, and an initial depth estimation at the same resolution. With the low-resolution input, the hourglass can capture large structures easily within only a couple of layers, which is sufficient to give an abstract of the scene. The initial prediction acting as a reference, together with the features from the first hourglass are up-sampled to the half-resolution and fed into the following module. 1/2 down-sampled sparse depth also participates in the second hourglass to supply for additional details. Outputs from the medium-level module are fused with the initial depth by residual connection to modify the estimation. The basic module repeats for the third time with the full-resolution inputs, to give the final dense map with fine-grained details.

Basic modules share the same network architecture. An hourglass sub-network starts with initial convolutional layers to turn the inputs into low-level feature maps. 2 down-sampling layers each with a stride of 2 contract the spatial size of feature maps and increase the perspective field progressively. Deconvolutional layers then up-sample the features to the input resolution for the pixel-wise prediction. To keep fine local information, we integrate features in different levels by skip connections. ReLU comes after each convolution except for the last layer, to better cope with the gradient vanishing problem.

When down-sampling the sparse input, it is important to keep as much information as possible. Inspired by the sparsity-invariant convolution proposed in [21], we adapt the standard average pooling to down-sample the sparse map. The data at position (x, y) on the down-sampled sparse depth map sD^k is an average over the valid neighbors of the pixel $(2^k x, 2^k y)$ on the original sparse map sD, in which 2^k is the down-sampling factor. This operation can be implemented simply as a division of the average pooling results of the original map sD and the indication map C, where $C(x, y) = 1$ indicates that pixel (x, y) in sD is observed, otherwise $C(x, y) = 0$. The implementation of the down-sampling operation $\phi^k_{x,y}(sD, C)$ can be formulated

Table 1. **Network Summary.** H and W are the height and width of the input color image. A stride of -2 in the decoder means a deconvolution layer with a stride of 2. k is the index of the hourglass. $+$ means pixel-wise addition of two features. Each layer is followed with ReLU, except for the last layer in every sub-module.

RGB Guidance Encoder					
Output	Kernel	Str.	Ch I/O	OutRes	Input
Initial layers					
F0_c	3×3	1	3/32	$H \times W$	I
	3×3	1	32/32		
Encoder					
F1_c	3×3	2	32/32	$\frac{1}{2}H \times \frac{1}{2}W$	F0_c
	3×3	1	32/32		
F2_c	3×3	2	32/32	$\frac{1}{4}H \times \frac{1}{4}W$	F1_c
	3×3	1	32/32		
F3_c	3×3	2	32/32	$\frac{1}{8}H \times \frac{1}{8}W$	F2_c
	3×3	1	32/32		
F4_c	3×3	2	32/32	$\frac{1}{16}H \times \frac{1}{16}W$	F3_c
	3×3	1	32/32		

Basic Depth Hourglass					
Output	Kernel	Str.	Ch I/O	OutRes	Input
Initial layers					
F0_d^k	3×3	1	1/32	$H/2^k \times W/2^k$	sD^k
	3×3	1	32/32		
Encoder					
F1_d^k	3×3	2	32/32	$\frac{1}{2}H/2^k \times \frac{1}{2}W/2^k$	F0_d^k+ F5_d^{k+1}
	3×3	1	32/32		
F2_d^k	3×3	2	32/32	$\frac{1}{4}H/2^k \times \frac{1}{4}W/2^k$	F1_d^k
	3×3	1	32/32		
Decoder					
F3_d^k	3×3	-2	32/32	$\frac{1}{2}H/2^k \times \frac{1}{2}W/2^k$	F2_d^k+ F(k+2)_c
	3×3	1	32/32		
F4_d^k	3×3	-2	32/32	$H/2^k \times W/2^k$	F3_d^k+ F(k+1)_c+F1_d^k
	3×3	1	32/32		
Predictor					
D^k	3×3	1	32/32	$H/2^k \times W/2^k$	F5_d^k+ F(k)_c+F0_d^k
	3×3	1	32/1		

as

$$\begin{aligned}\phi^k_{x,y}(sD, C) =& \frac{\sum_{i,j=0}^{2^k-1} sD_{2^k x+i, 2^k y+j}}{\sum_{i,j=0}^{2^k-1} C_{2^k x+i, 2^k y+j} + \epsilon} \\ =& \frac{\sum_{i,j=0}^{2^k-1} sD_{2^k x+i, 2^k y+j}/2^{2k}}{\sum_{i,j=0}^{2^k-1} C_{2^k x+i, 2^k y+j}/2^{2k} + \epsilon} \\ =& \frac{\psi^k_{x,y}(sD)}{\psi^k_{x,y}(C) + \epsilon},\end{aligned} \tag{1}$$

where $\psi^k_{x,y}(\cdot)$ is the average pooling, ϵ is a small number to avoid division by zero.

This kind of down-sampling essentially using adaptive weights to sample the data according to the validity. It can be easily extended to take the data positions into consideration. One can define a distance weight $\omega_{i,j}$ about the offsets i, j, and multiply it with $\omega_{i,j}$ to $sD_{2^k x+i, 2^k y+j}$ and $C_{2^k x+i, 2^k y+j}$. However, this extension introduces extra computations.

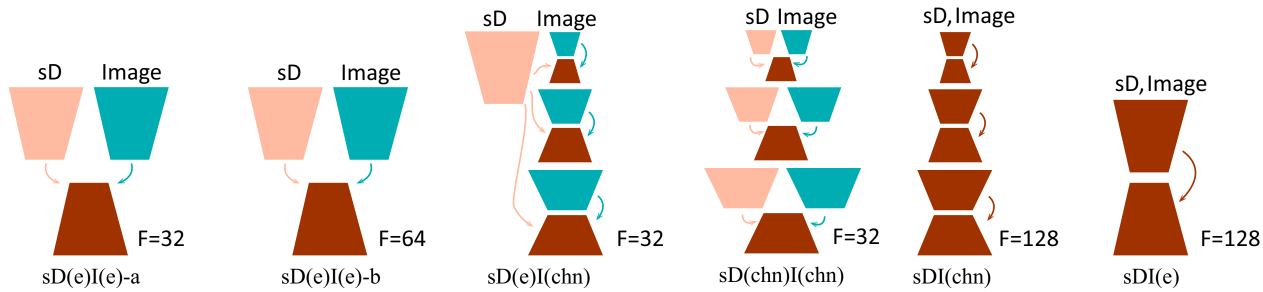

Figure 3. **Different Architectures.** F denotes the number of channels of the network.

3.3. Multi-Scale Guidance

Following the instructions from previous work [10, 20] that information of different modalities should be jointed with their high-level features, we design a specialized pathway to learn guidance features from the RGB signal.

The image pathway contains a single contracting network to encode the guidance information. Different from the sparse depth map, RGB image is densely informative. Thus we do not perform manual down-samplings on the image as the way in the depth branch. Instead, we begin with the full resolution image and use down-sampling convolution layers to extract multi-scale features. After 4 stacked layers of down-sampling convolutions each with a stride of 2, the deepest feature maps reach a 1/16 spatial resolution of the original input size, which is consistent with the minimal spatial feature size in the depth branch.

The extracted multi-scale image features are merged with depth features at the decoder phases in the depth pathway. Especially, our depth branch is composed of 3 prediction modules with different spatial resolutions. We combine RGB features and depth features that have the same resolution in each of the depth decoders. In this way, all the depth hourglasses can predict a dense map with the image guidance. With our design, decoding layers in the cascade network can always find guidance features that have the corresponding resolution. RGB and depth features are fused by the add operation, so as to compress the length of the feature channel in decoders.

4. Training with Multi-Scale Supervision

To further activate each functional part, we train the networks with intermediate supervision by using multi-scale ground-truth. We encourage the output of each sub-module to be close to the ground-truth at the corresponding resolution. The semi-dense ground-truth map is down-sampled by the strategy in Eq.(1). The total loss L is a summation of $\mathcal{L}_2$ losses for D^2, D^1 and D^0,

$$\begin{aligned} L =& \omega_2 \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_2(D_i^2, \hat{D}_i^2) + \omega_1 \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_2(D_i^1, \hat{D}_i^1) \\ &+ \omega_0 \frac{1}{N}\sum_{i=1}^{n}\mathcal{L}_2(D_i^0, \hat{D}_i^0), \end{aligned} \tag{2}$$

where i is the index of pixels, N is the total pixel number of the depth map. $\hat{D}_i^2$, $\hat{D}_i^1$, and $\hat{D}_i^0$ are the ground-truth maps at the quarter, half and full resolutions.

We adopt a multi-stage scheme during the training process. We set $\omega_2 = \omega_1 = \omega_0 = 1$ at first 10 epoches. Then ω_2 and ω_1 is reduced to 0.1 to emphasize the total performance. In the end, we set $\omega_2 = \omega_1 = 0$ after 20 epoches.

5. Experiment

In this section, we give an ablation study and a comparison with related work on the KITTI depth completion benchmark [21] to verify the design of our method. KITTI benchmark provides sparse depth maps captured by Velodyne LiDAR HDL-64e, aligned RGB images, and semi-dense ground-truth which is the coincident depth of the accumulated LiDAR and stereo estimation. The dataset contains 85895 training data, 1000 selected validation samples, and 1000 test data without ground-truth. Due to our limited computational ability, the networks in the ablation study is trained on only **10000** samples from the training set.

Our network is implemented in PyTorch. All models are optimized with Adam [11] ($\beta_1 = 0.9; \beta_2 = 0.999$). The learning rate begins at 0.01, and is reduced by a half every 5 epochs. We train for 31 epochs with a batch size of 4. Random cropping and left-right-flipping are performed as data augmentation. The training maps are cropped to a resolution of 1216×352. We initialize the networks with random parameters. A weight decay of 2×10^{-4} is applied for regularization.

For evaluation, we report four common metrics: the mean square error (MAE, mm), root mean square error (RMSE, mm), as well as their counterpart on the inverse depth iMAE (1/km) and iRMSE (1/km).

Table 2. **Quantitative Evaluation of Different Architectures.** The proposed MSG-CHN (sD(chn)I(e)) achieves the best performance with a small number of parameters and low run-time.

	MAE	RMSE	♯Params	Runtime
sD(e)I(e)-a	280.90	986.16	202K	0.007
sD(e)I(e)-b	279.52	973.45	806K	0.016
sD(e)I(chn)	263.64	972.97	421K	0.015
sD(chn)I(chn)	248.77	933.13	587K	0.017
sDI(chn)	266.16	1006.23	1117K	0.028
sDI(e)	291.91	1025.91	1887K	0.031
sD(chn)I(e) (ours)	245.28	910.37	364K	0.011

5.1. Ablation Study

a) Verification of Cascade Hourglass Architecture

To investigate our architecture design, we test to deal with the sparse depth and color image by several different network configurations. The diagrams of the network variants are sketched in Fig. 3, and comparison results are listed in Table 2. Here, we denominate our MSG-CHN architecture as *sD(chn)I(e)* , which means that the sparse depth pathway is the cascade hourglass architecture, and the image pathway is a single encoder. *sD(e)I(e)-a* is a variant in which both depth and image pathways are the single multi-scale encoders, and a decoder coordinates both features for prediction. *sD(e)I(e)-b* doubles the channel length of *sD(e)I(e)-a*. Sparse depth in *sD(e)I(chn)* passes a single encoder, while the image goes through stacked hourglasses. *sDI(chn)* performs data fusion at the input stage, and takes the RGBD input to a CHN architecture. *sDI(e)* contains a single hourglass to deal with the RGBD input. *sD(e)I(e)-a*, *sD(e)I(e)-b* and *sDI(e)* belong to the typical pipeline for depth completion.

Comparing the results of *sDI(chn)* with those of *sDI(e)*, we find that even with the data fused at the earliest stage, cascaded networks show superiority over single-stage networks. Moreover, replacing any of the current components with other configurations will cause degraded performance (*sD(e)I(e)-a/b*, *sD(chn)I(chn)*, and *sD(e)I(chn)*). It justifies the idea that sparse depth can be down-sampled at the beginning for multi-scale processing, while a single encoder with sub-sampling convolutions is suitable for the image data.

b) Verification of The Multi-Scale Designs

To study the effect of using multi-scale inputs, we compare with two variants. *CHN* takes the full resolution inputs to all the three hourglasses, and the encoder in the image path feeds the same scale guidance information for the three networks. *CHN-D* also takes the monotonous resolution inputs and image guidance, while dilated residual blocks [22] are added to the end of each depth encoder for capturing multi-scale structures. To keep the same receptive fields as ours, we add four dilated layers to the first hourglass, with the dilations set to 2, 2, 4, and 1. Two layers with dilations of 2 and 1 are added to the second hourglass.

Table 3. **Impact of the Multi-Scale Designs.** Our multi-scale network achieves high accuracy with low computational complexity.

	MAE	RMSE	♯Params	Runtime
CHN	262.63	946.39	364K	0.032
CHN-D	241.77	917.60	438K	0.040
MSG-CHN(ours)	245.28	910.37	364K	0.011

Results in Table 3 shows that *CHN* suffers from performance degradation both in the accuracy and efficiency. This is because the duplicated networks without multi-scale inputs and guidance lack the ability to describe the various structures. And the full-sized inputs introduce additional computational burdens. With the dilated blocks added to *CHN*, the variant *CHN-D* is enabled to cope with multi-scale features, and achieves similar accuracy as the proposed network. However, the computational complexity is increased even further. The comparison verifies that our networks with multi-scale inputs is efficient and effective to handle the diverse patterns and reduce the computational complexity at the same time.

c) Effect of the Sampling Strategy

Table 4. **Quantitative Evaluation of Different Sampling Strategies.** The model with sparse inputs sampled by our adapted average pooling performs the best.

	MAE	RMSE	iMAE	iRMSE
grid sampling	263.53	926.89	1.24	3.13
bilinear sampling	261.88	925.57	1.25	3.21
avg-pooling	255.20	922.06	1.17	2.98
max-pooling	246.93	920.94	1.09	2.82
ours	245.28	910.37	1.07	2.79

We sub-sample the sparse map to different resolutions for the network to learn multi-scale predictions, and meanwhile to accelerate the computation. Here, we analyze the influence of different down-sampling strategies. In table 4, we provide a comparison of the final results with inputs sampled by direct grid sampling, bi-linear sampling, avg-pooling, max-pooling, and the strategy introduced in Eq. (1). Sampling sparse data without considering the invalid positions causes a considerable reduction in accuracy. Max-Pooling controls the minus effect of invalid data to some extent, while it is still slightly inferior to the introduced strate-

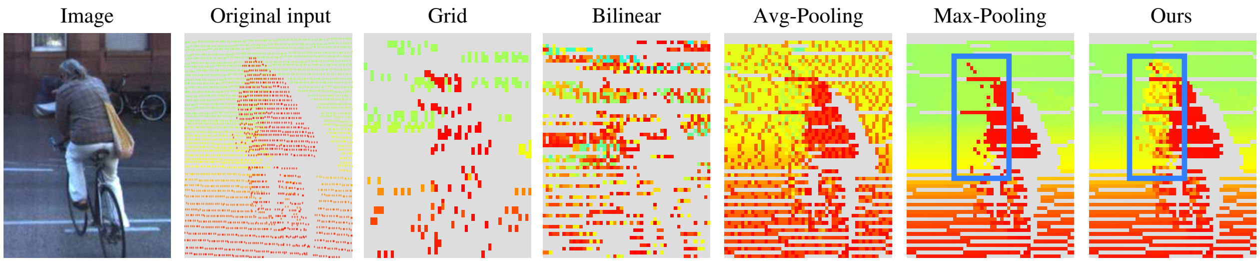

Figure 4. **Down-sampling Results by Different Strategies.** Direct grid sampling causes data missing. Bi-linear sampling and Avg-Pooling pollute the original information. Max-Pooling leads to the loss of fore-ground structures. Our strategy preserves original information as much as possible.

Table 5. **Quantitative Evaluation of Different Training Strategies.** Intermediate supervision can boost the performance.

	MAE	RMSE	iMAE	iRMSE
w/o intermediate	258.37	922.93	1.11	2.85
with intermediate (ours)	245.28	910.37	1.07	2.62

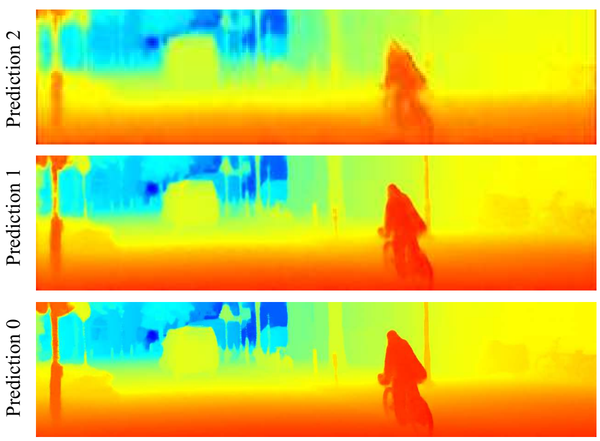

Figure 5. **Intermediate Results.** Form top to bottom are the predict maps from the first, the second, and the third hourglass.

gy. These straightforward down-sampling operations do not apply to the sparse data. As shown in Fig. 4, sampling directly at grids results in the vanishing of the surviving data. Bi-linear down-sampling and average pooling pollute the results with the invalid zero values. Moreover, max-pooling breaks up intrinsic structures.

d) Verification of The Multi-Scale Training

To verify the effectiveness of the multi-scale auxiliary training, we compare with a model trained end-to-end without intermediate supervision (*w/o intermediate*). Results in Table 5 imply that introducing intermediate supervision can boost performance. We visualize example intermediate predictions of each hourglass in Fig. 5. With our multi-scale training, the first hourglass learns to give a coarse abstract of the scene depth. Following stages gradually supply details at the corresponding scales.

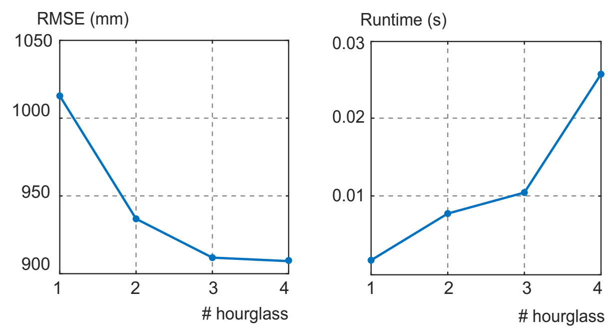

Figure 6. **The Influence of The Number of Hourglasses.** With the more hourglasses applied, the better performance a model will achieve, and the longer run-time it will cost.

e) The Number of Cascade Hourglasses

We study the right number of hourglasses in the depth pathway. With the number of hourglasses ranging from 1 to 4, the varying trends of RMSE error and run-time are visualized in Fig. 6. When reducing the number, we start removing from the lowest resolution, to ensure that the last finest hourglass is always kept. That is to say, the one-hourglass architecture only contains the last component in CHN, and the two -hourglass architecture contains the last two. When appending the architecture, another hourglass with full-resolution inputs follows our MSG-CHN.

Fig. 6 shows that as more hourglasses are introduced and the total number is no more than 3, errors reduce notably, and the run-time increases only in a limited amount. However, once the 4-th hourglass is added, the reduction in error becomes insignificant while the increment in run-time gets obvious. We can conclude that: (1) The effect of first three hourglasses is complementary. (2) Inference through a low-resolution network is fast, while a high-resolution network is slow. (3) Using three hourglasses is a decent trade-off between accuracy and speed.

5.2. Overall Performance

In this section, we compare with current publicly available state-of-the-art methods on the selected validation and

Table 6. **Comparison With State-of-The-Art on KITTI Benchmark.** The platform refers to Nvidia GPUs. Our method achieves competitive results compared with state-of-the-art, with low demand in run-time.

Methods	Selected Validation				Test				♯Params	Runtime(s)	Platform
	MAE	RMSE	iMAE	iRMSE	MAE	RMSE	iMAE	iRMSE			
DeepLiDAR [16]	215.38	687.00	1.10	2.51	226.50	758.38	1.15	2.56	144.0M	0.07	GTX 1080Ti
RGB_guide&certainty [6]	214	802	-	-	215.02	772.87	0.93	2.19	2.6M	0.02	Tesla V100
Sparse-to-Dense (gd) [13]	260.90	878.56	1.34	3.25	249.95	814.73	1.21	2.80	26.1M	0.08	Tesla V100
NConv-CNN-L2 (gd) [4]	233.25	870.82	1.03	2.75	233.26	829.98	1.03	2.60	355K	0.02	Tesla V100
Spade-RGBsD [10]	-	-	-	-	234.81	917.64	0.95	2.17	5.3M	0.07	-
MSG-CHN (ours)	215.14	750.52	0.95	2.32	220.41	762.19	0.98	2.30	1.2M	0.01	GTX 2080Ti

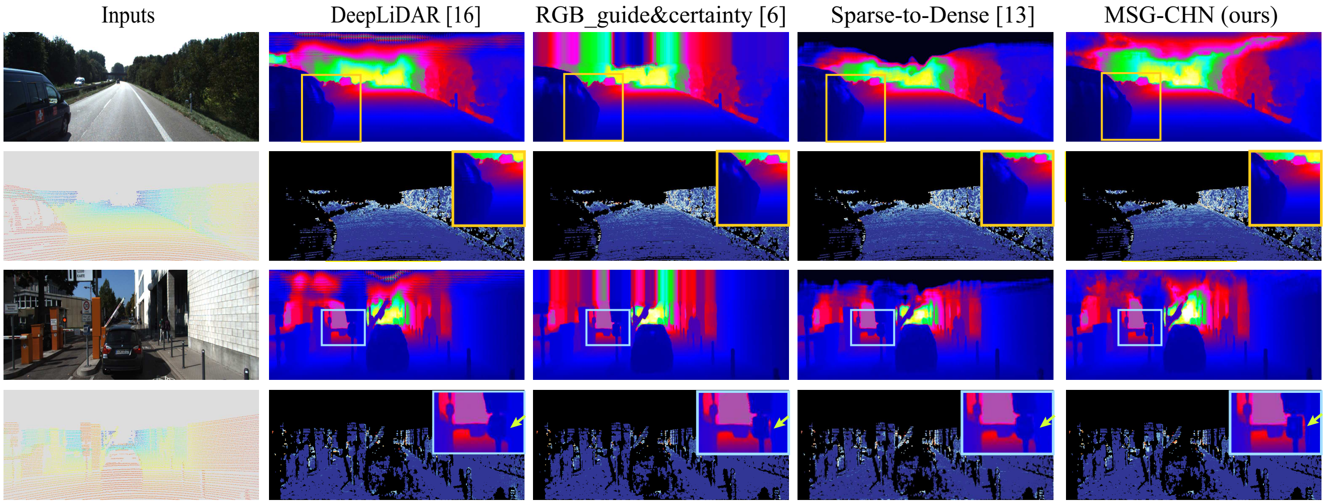

Figure 7. **Qualitative Results of Our Method and State-of-the-Art on KITTI test set.** From top to bottom, left to right: the input color image, the sparse depth input, the estimated dense depth maps, and the error maps (the warmer, the larger). We zoom-in the boxes of interest at the top-left corner on the error maps. Our method can handle both fine and coarse structures.

test sets of KITTI depth completion. The final model is trained with all training data provided by KITTI benchmark. We increase the model channels from 32 to 64 for final results. Table 6 reports the quantitative evaluation results, Fig. 7 shows several example maps.

Our method achieves competitive results compared with top methods on KITTI benchmark, with low run-time and model complexity [1]. The number of parameters of the proposed model is a hundredth of DeepLiDAR [16], while the sacrifice in accuracy is insignificant.

Qualitative results in Fig. 7 demonstrate that our method can adequately handle the large-scale structures as well as fine details. Specifically, we correctly recover the boundary of a large scale car (marked by the yellow boxes), as well as the fine-grained shape of the signs (pointed out by arrows in the blue boxes), where other methods fail.

[1]This is only a rough comparison since the run-time of each method is evaluated on different platforms. As a reference, Tesla V100 owns the most CUDA cores and highest computing performance, while GTX 1080Ti is with the lowest computational ability among the three platforms listed in Table 6.

6. Conclusion

In this work, we present a lightweight multi-scale guided cascade hourglass network for the task of depth completion. The cascade network consists of simple hourglasses with sparse depth inputs in multiple scales to specifically cope with diverse structures. Each hourglass receives the guidance at different levels from an RGB encoder. Subnetworks are encouraged to focus on particular patterns via a multi-scale learning strategy. By assigning modules with specialized functions, the network can be implemented with simple architectures. We performed comprehensive analyses on the KITTI benchmark and achieved competitive accuracy with low run-time and light weights.

Acknowledgement. This work was supported by the National Key R&D Program of China (No.2016YFB1001001), the National Natural Science Foundation of China (No.61573280, No.91648121, No.61976170), and Tencent Robotics X Lab Rhino-Bird Joint Research Program (No.201902, No.201903).

References

[1] J.-R. Chang and Y.-S. Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[2] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016.

[3] D. Eigen and R. 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 (ICCV)*, 2015.

[4] A. Eldesokey, M. Felsberg, and F. S. Khan. Confidence propagation through cnns for guided sparse depth regression. *CoRR*, abs/1811.01791, 2018.

[5] A. Eldesokey, M. Felsberg, and F. S. Khan. Propagating confidences through cnns for sparse data regression. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

[6] W. V. Gansbeke, D. Neven, B. D. Brabandere, and L. V. Gool. Sparse and noisy lidar completion with RGB guidance and uncertainty. In *International Conference on Machine Vision Applications (MVA)*, 2019.

[7] J. Hua and X. Gong. A normalized convolutional neural network for guided sparse depth upsampling. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

[8] Z. Huang, J. Fan, S. Yi, X. Wang, and H. Li. Hms-net: Hierarchical multi-scale sparsity-invariant network for sparse depth completion. *arXiv preprint arXiv:1808.08685*, 2018.

[9] T. Hui, C. C. Loy, and X. Tang. Depth map super-resolution by deep multi-scale guidance. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[10] M. Jaritz, R. de Charette, É. Wirbel, X. Perrotton, and F. Nashashibi. Sparse and dense data with cnns: Depth completion and semantic segmentation. In *Proceedings of the IEEE International Conference on 3D Vision (3DV)*, 2018.

[11] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[12] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[13] F. Ma, G. V. Cavalheiro, and S. Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *International Conference on Robotics and Automation ICRA*, 2019.

[14] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[15] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[16] J. Qiu, Z. Cui, Y. Zhang, X. Zhang, S. Liu, B. Zeng, and M. Pollefeys. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[17] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo. Efficient convnet for real-time semantic segmentation. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1789–1794, 2017.

[18] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems (ITS)*, 19(1):263–272, 2017.

[19] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[20] S. S. Shivakumar, T. Nguyen, S. W. Chen, and C. J. Taylor. Dfusenet: Deep fusion of RGB and sparse depth information for image guided dense depth completion. *CoRR*, abs/1902.00761, 2019.

[21] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger. Sparsity invariant cnns. In *Proceedings of the IEEE International Conference on 3D Vision (3DV)*, 2017.

[22] F. Yu, V. Koltun, and T. A. Funkhouser. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2017.

[23] Y. Zhang and T. A. Funkhouser. Deep depth completion of a single RGB-D image. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[24] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia. Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[25] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2017.