# Generating dense depth maps using LIDAR and stereo correspondences

Aniket Gupta
Northeastern University
Boston, MA
gupta.anik@northeastern.edu

Rajat Mehta
Northeastern University
Boston, MA
mehta.rajat@northeastern.edu

## Abstract

*For the safe operation of autonomous vehicles, a dense 3D map of the surroundings is required for operations like planning and navigation. Currently, to generate these dense 3D maps, point cloud data from LiDAR (sparse but long range) and Stereo cameras (Dense but short ranged) are fused together. However, both sensors vary significantly in their structures and modalities. Thus, a proper method for aligning and smoothening the point cloud data is critical for Stereo-LiDAR sensor fusion. Unlike previous approaches which use raw RGB images and Lidar data to generate dense depth maps, our approach utilizes the disparity generated from stereo images and combines it with the sparse data from Lidar sensor. Early testing of the models shows that it is able to produce high quality dense depth maps in accordance with some of the best approaches. The approach shows promise to be carried forward as a strong research project. The small footprint of the model also enables fast inference.*

## 1. Introduction

Estimation of Depth Maps has been one of the fundamental tasks in the computer vision field. Being able to acquire accurate depth maps is a first step for many vision tasks. Several tasks like 3D mapping and localization, 3D object detection are all highly dependent on successful estimation of correct depth maps. Such jobs underlie numerous applications including augmented reality, virtual reality, autonomous driving, and robotics.

Given a pair of rectified stereo images, the ultimate goal of depth estimation is to calculate the disparity d for each pixel in the reference image for a given pair of rectified stereo images. Disparity refers to the horizontal displacement between a pair of corresponding pixels on the left and right images. Camera's focal length f is useful in calculating depth d of a particular pixel.

Solving the Stereo matching problem has been a prominent method in constructing depth maps. The stereo matching pipeline involves finding the corresponding points based on matching cost and post-processing. Traditionally, this has been done by applying window-based methods [1] or global-optimization methods [6] to construct a disparity map. Recently, deep convolutional neural networks (DCNNs) have been used to solve the stereo matching problem [19], [11]. This is mainly because they have better performance than traditional approaches on a variety of vision tasks such as image classification and object detection. Generally, DCNNs are trained end-to-end with a large amount of ground-truth labels.

Recent research works [18]–[13] are using deep learning as an approach to exploit a synchronized RGB image for depth completion. These methods have surpassed conventional approaches [4] and are showing significant improvements. The research [13] train a network to estimate surface normal from both the RGB image and LiDAR data and then guide depth completion using the recovered surface normal. The aim of such models is to fuse the feature vectors from sparse depth and RGB image together directly for further processing. Hence, they adopt operations like concatenation or element-wise addition for such purposes.

In general, matching cost computation, cost support aggregation, cost volume regularization, and disparity refinement constitutes the 4-stage pipeline for a typical stereo matching algorithm [17]. In [22], a unique model for stereo matching network is achieved by proposing two techniques: Input Fusion to incorporate the geometric information from sparse LiDAR depth with the RGB images for learning joint feature representations, and Conditional Cost Volume Normalization to adaptively regularize cost volume optimization in dependence on LiDAR measurements. But this model has less flexibility when it comes to different network architectures.

At present, either light detection and ranging (LIDAR) or stereo matching algorithms are used to acquire such depth information. A high-resolution LIDAR is expensive and produces sparse depth maps at large ranges while the stereo matching algorithms are able to generate dense depth maps but are typically less accurate than LIDAR at long

ranges. Nowadays, an innovative model of combining these approaches together to generate high-quality dense depth maps is being used in several research works. The research in [5] follows a similar model which adopts a self-supervised training process and generates depth maps.

Hence, it is evident from the above discussion that accurate dense depth maps are a result of an adequate combination of stereo matching algorithms and LiDAR data. We implement a pyramid stereo matching network consisting of two main modules: spatial pyramid pooling and 3D CNN. The spatial pyramid pooling module takes advantage of the capacity of global context information by the aggregating context in different scales and locations to form a cost volume while the 3D CNN learns to regularize cost volume using stacked multiple hourglass networks in conjunction with intermediate supervision. Once we get good stereo correspondences from our model, they can be combined with the LiDAR data to get dense depth maps. Extensive experiments are conducted on the KITTI 360 Depth Completion Dataset to evaluate the effectiveness of our proposed method.

## 2. Background and Related work

In [17], dense depth maps are achieved using guided convolution between sparse images and guided RGB images. Both the inputs are passed through a set of convolution layers before generating the output through Guided convolution layers generating dense maps. The concept of guided image filtering is used for implementing guided convolution over images. The research work [21] applies Input Fusion and Conditional Cost Volume Normalization (CCVNorm) on the LiDAR information. The Stereo images are passed through a fusion layer of sparse LiDAR depths and the CCVNorm is used for the 2D Convolution. A few research works [22] focuses on scene completeness of sparse depth completion (SCADC). It uses depth maps with structured upper scene estimation using Stereo Cameras. Non-Local Spatial Propagation Network for Depth Completion [12] is one of the unique approaches to find depth completion by estimating non-local neighbors using RGB and Sparse depth images and then iterating over them to find refined dense depth completion using non-local spatial propagation procedure.

### 2.1. Stereo Matching

Stereo matching has been a fundamental problem in computer vision. In general, a typical stereo matching algorithm can be summarized into a four-stage pipeline, consisting of matching cost computation, cost support aggregation, cost volume regularization, and disparity refinement. This design paradigm is widely used even after the start of usage of deep learning for learning depth maps(which in turn gives very high performance). For instance, [5] propose to learn a feature representation for matching cost computation by using a deep Siamese network, and then adopt the classical semi-global matching (SGM) [12] to refine the disparity map. Disparity maps are typically derived from several layers of convolutional matching. For supervised learning techniques, the models are trained using ground-truth disparity maps. Several works are focusing on resorting to extra information to refine results [24], forming a correlation volume [10], and designing architectures to extract features [3]. In addition to the supervised approaches, unsupervised methods have also gained popularity [8]. These approaches rely on a warping error to provide a training loss. This error measures the difference between the input image from one side and the warped image of the stereo pair from the other side. The warping process is implemented using differential bilinear interpolation. Still, there is a noticeable gap between the supervised methods and unsupervised methods with respect to performance. Incorporating warping loss in supervised training has been shown to be beneficial over supervised training alone [15].

### 2.2. Depth estimation

Depth estimation is one of the fundamental tasks in computer vision.The depth completion task is a sub-problem of depth estimation. The depth completion task has strong priors on scene depth. There are several Sparsity-invariant operations for this task and they have proved to be more effective than regular convolutions [7]. With additional color images, the depth completion process can be guided by color information. Recent works show a performance boost using the color information contained in RGB data.

### 2.3. Guided Filtering

This type of filtering utilizes a reference or guidance image as prior and aims to transfer the structures from the reference image to the target image for color/depth image super-resolution, image restoration, etc. In [23], a guided filtering layer has been proposed by Wu et al. to perform joint upsampling. It reformulates the conventional guided filter and makes it differentiable as a neural network layer. Hence, the kernel weights are thus generated by the same close-form equation of a guided filter to filter the input image. This kind of operator is inapplicable to fill-in sparse LiDAR points, as commented

### 2.4. RGB Image and LiDAR Fusion

Fusion of RGB images and LiDAR data has obtained more attention because of its practicability and high performance for depth perception. There are mainly two different settings which are explored by several prior works: LiDAR fused with a monocular image or stereo ones. As the depth estimation from a single image is typically based on a regression from pixels, which is inherently unreliable and

ambiguous, most of the recent monocular-based works aim to achieve the completion on the sparse depth map obtained by LiDAR sensor with the help of rich information from RGB images or they refine the depth regression by having LiDAR data as a guidance. Current state-of-the-art studies focus on how to accurately compute the matching cost using CNNs and how to apply semi-global matching (SGM) to refine the disparity map. In Pyramid pooling, the empirical receptive field is much smaller than the theoretical receptive field. In ParseNet [9], global pooling with FCN enlarges the empirical receptive field to extract information at the whole image level and it actually improves semantic segmentation results. PSPNet [25] presents a pyramid pooling module to collect the effective multiscale contextual prior. Inspired by PSPNet, DeepLab v3 proposes a new ASPP module augmented with global pooling. Ideas of spatial pyramids have been used for optical flow. In SPyNet [14], image pyramids helps in estimating optical flow in a coarse-to-fine approach. PWC- Net [16] improves optical flow estimation by using feature pyramids.

# 3. Proposed Methods

## 3.1. Baseline Method

The initial plan of the project was to take forward the approach from [17] as our primary reference idea, which takes inspiration from learning kernel weights from a single guidance image. We had planned to study the effect on the model performance by introducing a dual branch guidance module consisting of stereo images. We faced major issues in compiling it's open-source code which required C++ compatibility on the Discovery server. Due to these issues, we decided to change the baseline model. Since most approaches for generating Dense depth maps utilize the same input (Single RGB and LiDar data), we decided to check the efficiency of our approach in comparison to [20]. The model uses a simple Encoder-Decoder architecture to encode the RGB and LiDAR data, concatenates them and passes them through the decoder architecture.

## 3.2. Proposed changes and Reasoning

We denote the sparse depth map as $D_s$ which is obtained from the LiDAR. The two stereo images obtained will be labeled as $I_l$ and $I_r$ in the report. The predicted dense depth map will be denoted as $D_t$. In our approach, we propose removing the RGB branch from the model and using a disparity branch instead.

For this disparity branch, we propose a dual branched model which takes the data from the two stereo images and generates a feature vector for both of them using a CNN encoder. This feature vector can then be passed through a Spatial pyramid pooling module(SPP) to increase the receptive field. The output of the SPP module can then be fed to the last convolution layer. The generate output feature vector from the two stereo images can thus be used to generate a cost volume which can be used to generate a dense depth map in conjunction with the LIDAR data.

### 3.2.1 Spatial Pyramid Pooling Module

Taking inspiration from [2], we employ a Spatial Pyramid Pooling Module (SPP) module in our implementation. SPP module helps in determining the contextual relationship between the different pixel intensities. It helps in identifying the relationship between a given object and its surroundings.

The SPP module is designed to remove the fixed-size constrain of CNN. In this project, the SPP is a four fixed layer average pooling block structure with filter sizes 64, 32, 16 and 8. It takes input the data from the last convolution block, runs average pooling with the given four filters and passes the outputs through four parallel convolution blocks. The output of these four blocks is then concatenated and upsampled to match with the output size of convolution block 3 and concatenated with it to produce the output feature map. This output feature map has high receptive field and an understanding of contextual information. Figure 2 represents the structure of the SPP module used in this project.

### 3.2.2 Encoder Network

We use three independent but identical encoder networks in our model. Two CNN encoders are used to encode the features from both of the stereo images while the third encoder network is used to generate a feature map from the LIDAR data. In general, a single encoder network contains 5 convolution layers with filter size of 3 for each of them. These convolution layers are coupled with a batch normalization and relu activation layer each. The first convolution block (convolution, batch normalization and relu) reduces the size of the input image by half, the next two convolution blocks again halve the size of the input and finally the last two layers bring down the size to 1/8 of the input image.

The encoder network contains the SPP module in itself and thus the output feature map generated has a larger receptive field coverage.

### 3.2.3 Cost Volume

The feature maps produced for each stereo image after the encoder CNN module and SPP module are 1/8 of the original input size and are processed using a pointwise coorelation layer to produce the cost volume for this branch. The coorelation layer correlates the features in the two images in a horizontal fashion. Since, we only consider a displace-

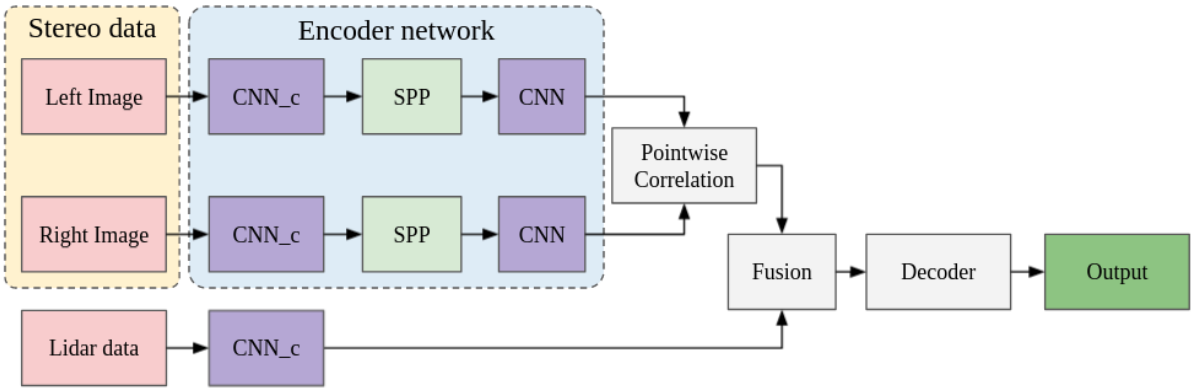


Figure 1. Model Pipeline

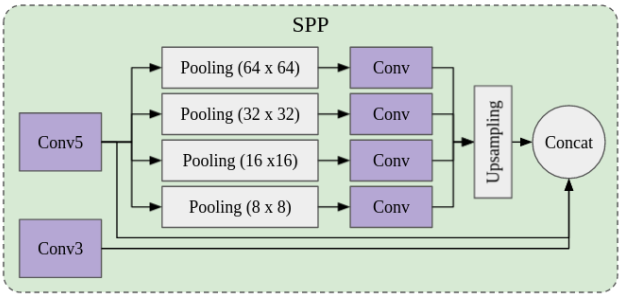


Figure 2. Spatial Pyramid Pooling Module

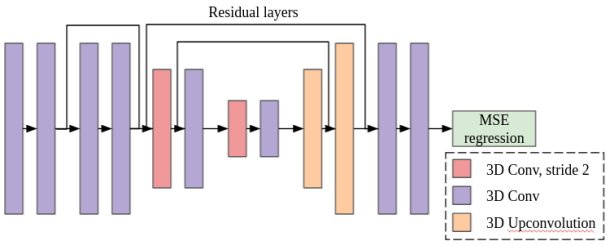


Figure 3. Stack hourglass decoder network

ment of 24 pixels in the two feature maps, the maximum disparity for the model is 192 pixels in the input image.

#### 3.2.4 Fusion

Unlike traditional approaches which sum the data from the two branches and work on it, in our fusion layer, we concatenate the cost volume produced by the disparity branch and the feature vector generated by the CNN for the LIDAR data. This concatenation preserves the information received from the two branches much better.

#### 3.2.5 Decoder Network

In order to learn more context information, we use a stacked hourglass (encoder-decoder) architecture, consisting of repeated top-down/bottom-up processing in conjunction with intermediate supervision.

This decoder network takes the output from the fusion unit and upsamples the output using 5 up-convolution layers. The output is then sent to a regression layer to produce the depth image and calculate the loss value.

#### 3.2.6 Loss function

The KITTI depth completion dataset does not contain the dense depth annotation. Only about 30% of the pixels in the depth image are annotated. The loss function therefore only uses the valid pixels in the reference ground truth depth map when computing the loss value. We use the Mean square error(MSE) to compute the error between the produced dense depth map and the ground truth for only the annotated pixels.

$$L_{depth} = \frac{1}{|V|} \sum_{i,j \epsilon V} ||D^g(i,j) - D^t(i,j)||_2^2 \qquad (1)$$

where $D^g(i,j)$ and $D^t(i,j)$ denote the ground truth and predicted depth values at $(i,j)$ pixel, respectively. $V$ denotes the set of valid depth points from semi-dense annotation. Along with this loss, we also use the second-order gradients of the ground truth depth map to produce smooth dense maps.

$$L_{smooth} = \frac{1}{N} \sum_{i,j} ||\delta_x^2 D_{ij}^t + \delta_y^2 D_{ij}^t||_2^2 \qquad (2)$$

The total loss therefore can be calculated as:

$$L = L_{depth} + L_{smooth} \qquad (3)$$

where $w_1$ is a constant to tune the smoothness parameter. High values of $w_1$ can cause the image to become too blurry and loose important details.

# 4. Experiments and Results

## 4.1. Datasets

The following datasets were used in this project:

### 4.1.1 KITTI depth selection dataset

This dataset provides sequential stereo images and LIDAR point clouds for different road environments. The total dataset consists of about 42,949 training samples and 1000 validation samples available publicly. The dataset is fairly complicated with upto 15 cars and upto 30 pedestrians visible per image on highways as well as in rural areas. The image size captured by the dataset is 1240x376. For our full model, we only use 5 full videos for training due to time constraint.

### 4.1.2 KITTI stereo dataset

The KITTI stereo dataset is a real-world dataset consisting of 194 training stereo image pairs and 195 testing image pairs. The ground truth data is obtained using a LIDAR sensor, which is a sparse depth image. The image size captured by the dataset is 1240x376, which is consistent with the depth selection dataset. For our training we divide the dataset into a 80:20 train and validation split.

## 4.2. Implementation details

To implement the model, we first trained a disparity model using the architecture in Figure 1. The first branch of the two stereo images was kept the same till the cost volume part and a decoder network similar to 3.2.5 was used in the model to train it to produce depth disparities. We utilized the Pretrained weights from [2] and divided the weight file as per our network architecture. We also used the CNN weights from the baseline model for training. The model was trained end-to-end with Adam optimizer. The maximum disparity(D) of the model was set to 192. We trained this model on the KITTI stereo dataset using a learning rate of 0.001 for 30 epochs.

The weight file of the first part of the disparity model was divided appropriately for the Encoder block and the Decoder (Stack-Hourglass model) blocks and used in the final model to speed up the training. For the final model, we again used the Adam optimizer with a learning rate of 0.001. We also utilized the weight file provided by the baseline model for the encoder blocks. The whole model was trained on 4 video blocks (Raw RGB images and LiDAR data) for 5 epochs. We could not train the model on the complete KITTI depth selection dataset as it is quite large and we had time constraints on the project.

| | RMSE | MAE | iRMSE | iMAE |
|---|---|---|---|---|
| Guidenet | 736.24 | 218.33 | 2.25 | 0.99 |
| Baseline | 792.80 | 225.81 | 2.42 | 0.99 |
| **Ours** | 1548.89 | 493.65 | 5.01 | 1.86 |

Table 1. Evaluation metrics

## 4.3. Evaluation Metrics

We used four standard metrics for evaluation of KITTI dataset: root mean squared error (RMSE), mean absolute error (MAE), root mean squared error of the inverse depth (iRMSE) and mean absolute error of the inverse depth (iMAE). Among them, RMSE and MAE directly measure depth accuracy, while RMSE is more sensitive. iRMSE and iMAE compute the mean error of inverse depth, which gives less weight for far-away points.

The comparison of our model performance is summarised in Table 1. As expected, the numerical results on the evaluation metrics are not great. This maybe due to multiple factors, one of the most prominent ones being lack of training on the complete KITTI dataset.

One of the noticeable things in the output image in Figure 2 is that the model still produces consistent depth images which seems to be in agreement with the outputs of the baseline model as well as the ground truth depth. There are some faulty depth measurements in the center of the image for faraway object which is left to be improved in future work.

The small footprint of the model enables fast inference times and is able to generate depth map from stereo images and Lidar data in about 0.4s on using a Nvidia GTX 1060 GPU and Intel i7-2.8Ghz CPU.

# 5. Conclusion and Future work

In this project, we proposed a model to produce dense depth maps from LIDAR data using stero information. LIDAR produces depth maps which are quite accurate but sparse and stereo disparities produce depth maps which are not very accurate but dense. The approach proposed in the project produces dense depth maps by utilizing the strong aspects from both the sensor data. Although, the results produced by the model are not great in comparison to the current state of the art models, the full potential of the proposed method is still to be explored. For example, better results can be obtained if the model is allowed to train longer on the full KITTI depth completion dataset. Results will also improve if we train the model end-to-end instead of training a disparity model first and utilizing it's weights.

For the future work, we would like to train the model

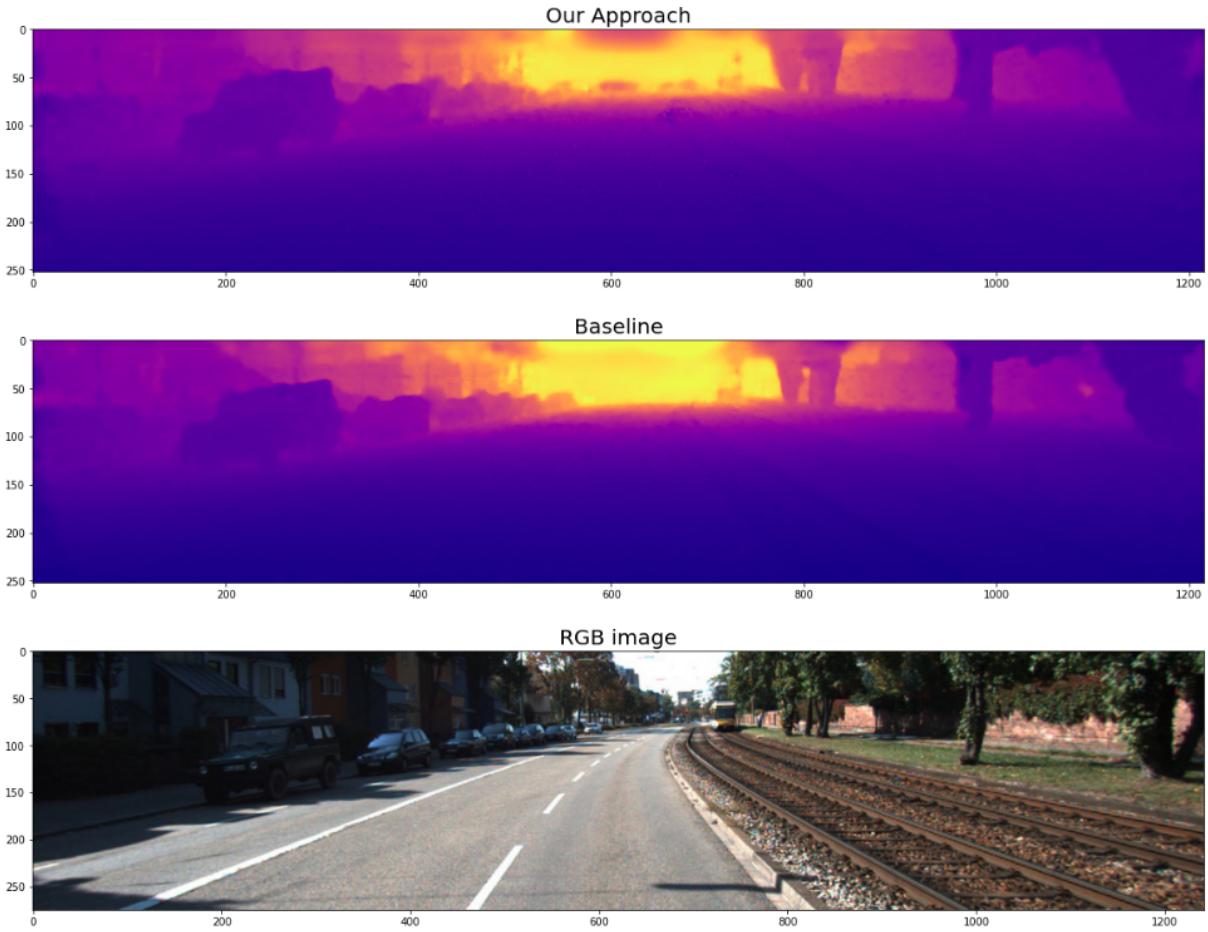


Figure 4. Comparison of results of depth map from our model vs baseline model

completely end-to-end on bigger datasets like SceneFlow to test the capabilities of the model. Moreover, we would also want to experiment with dilated convolutional layers in the model.

The research work to create dense depth maps is an interesting research area and can really help in cost reduction for autonomous driving and navigation. For example, high resolution LIDARs are comparatively much more expensive than low resolution ones and if a machine learning model can perform the task of dense depth estimation, it will be extremely useful for industrial applications.

## Acknowledgement

This work was done under the guidance of Prof. Huaizu Jiang, Assistant Professor, Khoury College of Computer Sciences, Northeastern University. Computation resources from the Northeastern's Discovery cluster were used for model training and evaluation.

## References

[1] Satyajit Adhyapak, Nasser Kehtarnavaz, and Mihai Nadin. Stereo matching via selective multiple windows. *J. Electronic Imaging*, 16:013012, 01 2007. 1

[2] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. pages 5410–5418, 06 2018. 3, 5

[3] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. 07 2018. 2

[4] Simon Hawe, Martin Kleinsteuber, and Klaus Diepold. Dense disparity maps from sparse disparity measurements. In *2011 International Conference on Computer Vision*, pages 2126–2133, 2011. 1

[5] Markus Herb. Computing the stereo matching cost with a convolutional neural network seminar recent trends in 3 d computer vision. 2015. 2

[6] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008. 1

[7] Zixuan Huang, Junming Fan, Shenggan Cheng, Shuai Yi, Xiaogang Wang, and Hongsheng Li. Hms-net: Hierarchical

multi-scale sparsity-invariant network for sparse depth completion. *IEEE Transactions on Image Processing*, PP:1–1, 12 2019. 2

[8] Zhang Junming, Katherine Skinner, Ram Vasudevan, and Matthew Johnson-Roberson. Dispsegnet: Leveraging semantics for end-to-end learning of disparity estimation from stereo imagery. *IEEE Robotics and Automation Letters*, PP:1–1, 01 2019. 2

[9] Wei Liu, Andrew Rabinovich, and Alexander Berg. Parsenet: Looking wider to see better. 06 2015. 3

[10] Wenjie Luo, Alexander G. Schwing, and Raquel Urtasun. Efficient deep learning for stereo matching. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5695–5703, 2016. 2

[11] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. pages 4040–4048, 06 2016. 1

[12] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and Soo-Ok Kweon. *Non-local Spatial Propagation Network for Depth Completion*, pages 120–136. 11 2020. 2

[13] 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. pages 3308–3317, 06 2019. 1

[14] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4161–4170, 2017. 3

[15] Nikolai Smolyanskiy, Alexey Kamenev, and Stan Birchfield. On the importance of stereo for accurate depth estimation: An efficient semi-supervised deep neural network approach. pages 1120–11208, 06 2018. 2

[16] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. pages 8934–8943, 06 2018. 3

[17] Jie Tang, Fei-Peng Tian, Wei Feng, Jian Li, and Ping Tan. Learning guided convolutional network for depth completion. *IEEE Transactions on Image Processing*, 30:1116–1129, 2021. 1, 2, 3

[18] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger. Sparsity invariant cnns. In *2017 International Conference on 3D Vision (3DV)*, pages 11–20, Los Alamitos, CA, USA, oct 2017. IEEE Computer Society. 1

[19] Jure Žbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *J. Mach. Learn. Res.*, 17(1):2287–2318, jan 2016. 1

[20] Qiang Wang, Shaohuai Shi, Shizhen Zheng, Kaiyong Zhao, and Xiaowen Chu. Fadnet: A fast and accurate network for disparity estimation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 101–107, 2020. 3

[21] Tsun-Hsuan Wang, Hou-Ning Hu, Chieh Lin, Yi-Hsuan Tsai, Wei-Chen Chiu, and Min Sun. 3d lidar and stereo fusion using stereo matching network with conditional cost volume normalization. 04 2019. 2

[22] Cho-Ying Wu and Ulrich Neumann. Scene completenesss-aware lidar depth completion for driving scenario. 03 2020. 1, 2

[23] Huikai Wu, Shuai Zheng, Junge Zhang, and Kaiqi Huang. Fast end-to-end trainable guided filter. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1838–1847, 2018. 2

[24] Guorun Yang, Hengshuang Zhao, Jianping Shi, Zhidong Deng, and Jiaya Jia. *SegStereo: Exploiting Semantic Information for Disparity Estimation: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part VII*, pages 660–676. 09 2018. 2

[25] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6230–6239, Los Alamitos, CA, USA, jul 2017. IEEE Computer Society. 3