Sparsity Invariant CNNs

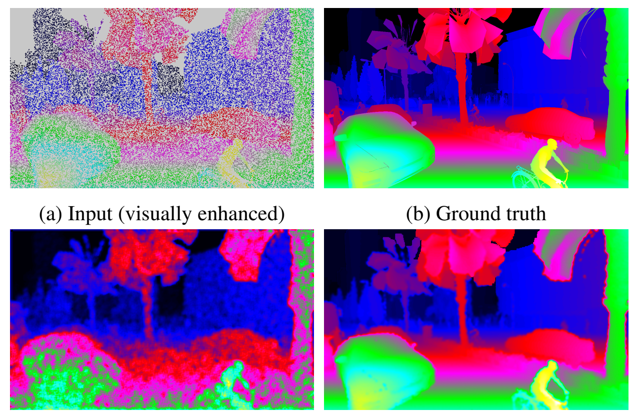
Jonas Uhrig[⋆,1,2] Nick Schneider[⋆,1,3] Lukas Schneider[1,4]
Uwe Franke[1] Thomas Brox[2] Andreas Geiger[4,5]

[1]Daimler R&D Sindelfingen [2]University of Freiburg
[3]KIT Karlsruhe [4]ETH Zürich [5]MPI Tübingen

{jonas.uhrig,nick.schneider}@daimler.com

Abstract

In this paper, we consider convolutional neural networks operating on sparse inputs with an application to depth completion from sparse laser scan data. First, we show that traditional convolutional networks perform poorly when applied to sparse data even when the location of missing data is provided to the network. To overcome this problem, we propose a simple yet effective sparse convolution layer which explicitly considers the location of missing data during the convolution operation. We demonstrate the benefits of the proposed network architecture in synthetic and real experiments with respect to various baseline approaches. Compared to dense baselines, the proposed sparse convolution network generalizes well to novel datasets and is invariant to the level of sparsity in the data. For our evaluation, we derive a novel dataset from the KITTI benchmark, comprising over 94k depth annotated RGB images. Our dataset allows for training and evaluating depth completion and depth prediction techniques in challenging real-world settings and is available online at: ***www.cvlibs.net/datasets/kitti***.



(a) Input (visually enhanced) (b) Ground truth

(c) Standard ConvNet (d) Our Method

Figure 1: **Depth Map Completion.** Using sparse, irregular depth measurements (a) as inputs leads to noisy results when processed with standard CNNs (c). In contrast, our method (d) predicts smooth and accurate depth maps by explicitly considering sparsity during convolution.

1. Introduction

Over the last few years, convolutional neural networks (CNNs) have impacted nearly all areas of computer vision. In most cases, the input to the CNN is an image or video, represented by a densely populated matrix or tensor. By combining convolutional layers with non-linearites and pooling layers, CNNs are able to learn distributed representations, extracting low-level features in the first layers, followed by successively higher-level features in subsequent layers. However, when the input to the network is sparse and irregular (e.g., when only 10% of the pixels carry information), it becomes less clear how the convolution operation should be defined as for each filter location the number and placement of the inputs varies.

The naïve approach to this problem is to assign a default value to all non-informative sites [3, 32]. Unfortunately, this approach leads to suboptimal results as the learned filters must be invariant to all possible patterns of activation whose number grows exponentially with the filter size. In this paper, we investigate a simple yet effective solution to this problem which outperforms the naïve approach and several other baselines. In particular, we introduce a novel sparse convolutional layer which weighs the elements of the convolution kernel according to the validity of the input pixels. Additionally, a second stream stream carries information about the validity of pixels to subsequent layers of the network. This enables our approach to handle large levels

⋆The first two authors contributed equally to this work.

of sparsity without significantly compromising accuracy.

Importantly, our representation is invariant to the level of sparsity in the input. As evidenced by our experiments, training our network at a sparsity level different from the sparsity level at test time does not significantly deteriorate the results. This has important applications, e.g., in the context of robotics where algorithms must be robust to changes in sensor configuration.

One important area of application for the proposed technique is enhancement of 3D laser scan data, see Fig. 1 for an illustration. While laser scanners provide valuable information about depth and reflectance, the resulting point clouds are typically very sparse, in particular when considering mobile scanners like the Velodyne HDL-64e[1] used in autonomous driving [13].

Learning models which are able to increase the density of such scans is thus highly desirable. Unfortunately, processing high-resolution data directly in 3D is challenging without compromising accuracy [44].

An alternative, which we follow in this paper, is to project the laser scan onto a virtual or real 2D image plane resulting in a 2.5D representation. Besides modeling depth prediction as a 2D regression problem, this representation has the advantage that additional dense information (e.g., RGB values from a color camera) can be easily integrated. However, projected laser scans are typically very sparse and not guaranteed to align with a regular pixel grid, hence leading to poor results when processed with standard CNNs. In contrast, the proposed method produces compelling results even when the input is sparse and irregularly distributed.

We evaluate our method in ablation studies and against several state-of-the-art baselines. For our evaluation, we leverage the synthetic Synthia dataset [45] as well as a newly proposed real-world dataset with 93k depth annotated images derived from the KITTI raw dataset [12]. Our dataset is the first to provide a significant number of high-quality depth annotations for this scenario. Besides attaining higher accuracy in terms of depth and semantics we demonstrate our method's ability to generalize across varying datasets and levels of sparsity. Our code and dataset will be released upon publication.

2. Related Work

In this section, we discuss methods which operate on sparse *inputs* followed by techniques that consider sparsity *within* the CNN. We briefly discuss the state-of-the-art in invariant representation learning and conclude with a review on related depth completion techniques.

CNNs with Sparse Inputs: The naïve approach to handling sparse inputs is to either zero the invalid values or to create an additional input channel for the network which encodes the validity of each pixel. For detecting objects in laser scans, Chen et al. [3] and Li et al. [32] project the 3D point clouds from the laser scanner onto a low resolution image, zero the missing values and run a standard CNN on this input. For optical flow interpolation and inpainting, Zweig et al. [59] and Koehler et al. [28] pass an additional binary validity mask to the network. As evidenced by our experiments, both strategies are suboptimal compared to explicitly considering sparsity inside the convolution layers.

Jampani et al. [25] use bilateral filters as layers inside a CNN and learn the parameters of the corresponding permutohedral convolution kernel. While their layer handles sparse irregular inputs, it requires guidance information to construct an effective permutohedral representation and is computationally expensive for large grids. Compared to their approach our sparse convolutional networks yield significantly better results for depth completion while being as efficient as regular CNNs.

Graham [15, 16] and Riegler et al. [44] consider sparse 3D inputs. In contrast to our work, their focus is on improving computational efficiency and memory demands by partitioning the space according to the input. However, regular convolution layers are employed which suffer from the same drawbacks as the naïve approach described above.

Sparsity in CNNs: A number of works [17, 33, 41, 54, 10] also consider sparsity *within* convolutional neural networks. Liu et al. [33] show how to reduce the redundancy in the parameters using a sparse decomposition. Their approach eliminates more than 90% of parameters, with a drop of accuracy of less than 1% on ILSVRC2012. Wen et al. [54] propose to regularize the structures (i.e., filters, channels and layer depth) of deep neural networks to obtain a hardware friendly representation. They report speed-up factors of 3 to 5 with respect to regular CNNs. While these works focus on improving efficiency of neural networks by exploiting sparsity *within* the network, we consider the problem of sparse *inputs* and do not tackle efficiency. A combination of the two lines of work will be an interesting direction for future research.

Invariant Representations: Learning models robust to variations of the input is a long standing goal of computer vision. The most commonly used solution to ensure robustness is data augmentation [50, 30, 31]. More recently, *geometric* invariances (e.g., rotation, perspective transformation) have been incorporated directly into the filters of CNNs [4, 55, 58, 24, 20]. In this paper, we consider the problem of learning representations invariant to the *level of sparsity* in the input. As evidenced by our experiments, our model performs well even when the sparsity level differs significantly between the training and the test set. This has important implications as it allows for replacing the sensor (e.g., laser scanner) without retraining the network.

[1] http://velodynelidar.com/hdl-64e.html

Depth Completion: We evaluate the effectiveness of our approach for the task of depth completion, which is an active area of research with applications in, e.g., stereo vision, optical flow and 3D reconstruction from laser scan data. While some methods operate directly on the depth input, others require guidance, e.g., from a high resolution image.

Methods for *non-guided depth upsampling* are closely related to those for single image superresolution. Early approaches have leveraged repetitive structures to identify similar patches across different scales in 2D [14, 36] and 3D [22]. More recently, deep learning based methods for depth [43] and image superresolution [56, 7, 8, 27] have surpassed traditional upsampling techniques in terms of accuracy and efficiency. However, all aforementioned methods assume that the data is located on a regular grid and therefore cannot be applied for completing sparse and irregularly distributed 3D laser scan data as considered in this paper.

Image guided depth completion, on the other hand, uses the underlying assumption that the target domain shares commonalities with a high-resolution guidance image, e.g., that image edges align with depth discontinuities. A popular choice for guided depth completion is bilateral filtering [2, 6, 29, 57, 34]. More advanced approaches are based on global energy minimization [5, 40, 9, 1, 42], compressive sensing [18], or incorporate semantics for improved performance [49]. While some of the aforementioned techniques are able to handle sparse inputs, they heavily rely on the guidance signal. In contrast, here we propose a learning based solution to the problem, yielding compelling results even without image guidance. Several approaches also exploit end-to-end models for guided depth upsampling of regular data [23, 51]. Unlike existing CNN-based approaches, the proposed convolution layer handles sparse irregular inputs which occur, e.g., in 3D laser scan data.

3. Method

Let f denote a mapping from input domain $\mathcal{X}$ (e.g., intensity, depth) to output domain $\mathcal{Y}$ (e.g., depth, semantics), implemented via a convolutional neural network. In this paper, we consider the case, where the inputs $\boldsymbol{x} = \{x_{u,v}\} \in \mathcal{X}$ are only partially observed. Let $\boldsymbol{o} = \{o_{u,v}\}$ denote corresponding binary variables indicating if an input is observed ($o_{u,v} = 1$) or not ($o_{u,v} = 0$). The output of a standard convolutional layer in a CNN is computed via

$$f_{u,v}(\boldsymbol{x}) = \sum_{i,j=-k}^{k} x_{u+i,v+j}\, w_{i,j} + b \tag{1}$$

with kernel size $2k+1$, weight w and bias b. If the input comprises multiple features, $x_{u,v}$ and $w_{i,j}$ represent vectors whose length depends on the number of input channels.

3.1. Naïve Approach

There are two naïve ways to deal with unobserved inputs. First, invalid inputs $x_{u,v}$ can be encoded using a default value, e.g., zero. The problem with this approach is that the network must learn to distinguish between observed inputs and those being invalid. This is a difficult task as the number of possible binary patterns grows exponentially with the kernel size. Alternatively, $\boldsymbol{o}$ can be used as an additional input to the network in the hope that the network learns the correspondence between the observation mask and the inputs. Unfortunately, both variants struggle to learn robust representations from sparse inputs (see Section 5).

3.2. Sparse Convolutions

To tackle this problem, we propose a convolution operation which explicitly considers sparsity by evaluating only observed pixels and normalizing the output appropriately:

$$f_{u,v}(\boldsymbol{x}, \boldsymbol{o}) = \frac{\sum_{i,j=-k}^{k} o_{u+i,v+j}\, x_{u+i,v+j}\, w_{i,j}}{\sum_{i,j=-k}^{k} o_{u+i,v+j} + \epsilon} + b \tag{2}$$

Here, a small ϵ is added to the denominator to avoid division by zero at filter locations where none of the input pixels $x_{u+i,v+j}$ are observed. Note that Equation 2 evaluates to a (scaled) standard convolution when the input is dense.

The primary motivation behind the proposed sparse convolution operation is to render the filter output invariant to the actual number of observed inputs which varies significantly between filter locations due to the sparse and irregular input. Note that in contrast to other techniques [43, 9] which artificially upsample the input (e.g., via interpolation), our approach operates directly on the input and doesn't introduce additional distractors.

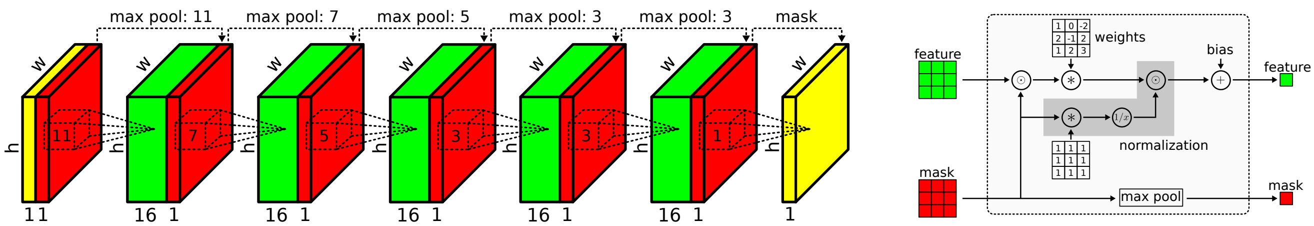
When propagating information to subsequent layers, it is important to keep track of the visibility state and make it available to the next layers of the network. In particular, we like to mark output locations as "unobserved" when none of the filter's inputs has been observed. We thus determine subsequent observation masks in the network $f^o_{u,v}(\boldsymbol{o})$ via the max pooling operation

$$f^o_{u,v}(\boldsymbol{o}) = \max_{i,j=-k,..,k} o_{u+i,v+j} \tag{3}$$

which evaluates to 1 if at least one observed variable is visible to the filter and 0 otherwise. In combination with the output of the convolution this serves as input for the next sparse convolution layer. The complete architecture of our network is illustrated in Fig. 2.

3.2.1 Skip Connections

So far we have only considered the convolution operation. However, state-of-the-art CNNs comprise many different



(a) **Network Architecture**

(b) **Sparse Convolution**

Figure 2: **Sparse Convolutional Network.** (a) The input to our network is a sparse depth map (yellow) and a binary observation mask (red). It passes through several sparse convolution layers (dashed) with decreasing kernel sizes from 11×11 to 3×3. (b) Schematic of our sparse convolution operation. Here, $\odot$ denotes elementwise multiplication, $*$ convolution, $1/x$ inversion and "max pool" the max pooling operation. The input feature can be single channel or multi-channel.

types of layers implementing different mathematical operations. Many of those can be easily generalized to consider observation masks. Layers that take the outputs of multiple preceding layers and combine them to a single output, e.g., by summation, are used frequently in many different network architectures, e.g., summation in inception modules [52] or skip connections in ResNets [19] as well as fully convolutional networks [35]. With additional observation indicators, the summation of input layers for each channel c and location (u, v) can be redefined as a normalized sum over the observed inputs x^l

$$f^+(\boldsymbol{x}, \boldsymbol{o}) = \frac{\sum_{l=1}^{n} o^l x^l}{\sum_{l=1}^{n} o^l} \quad (4)$$

where n denotes the number of input layers. If all pixels are observed, this expression simplifies to the standard operation $\sum_{l=1}^{n} x^l$.

4. Large-Scale Dataset

Training and evaluating the proposed depth completion approach requires access to a large annotated dataset. While evaluation on synthetic datasets [45, 11, 39] is possible, it remains an open question if the level of realism attained by such datasets is sufficient to conclude about an algorithm's performance in challenging real-world situations.

Unfortunately, all existing real-world datasets with sanitized depth ground truth are small in scale. The Middlebury benchmark [48, 47] provides depth estimates only for a dozen images and only in controlled laboratory conditions. While the Make3D dataset [46] considers more realistic scenarios, only 500 images of small resolution are provided. Besides, KITTI [13, 37] provides 400 images of street scenes with associated depth ground truth. However, none of these datasets is large enough for end-to-end training of high-capacity deep neural networks.

For our evaluation, we therefore created a new large-scale dataset based on the KITTI raw datasets [12] which comprises over 94k frames with semi-dense depth ground truth. While the KITTI raw datasets provide depth information in the form of raw Velodyne scans, significant manual effort is typically required to remove noise in the laser scans, artifacts due to occlusions (e.g., due to the different centers of projection of the laser scanner and the camera) or reflecting/transparent surfaces in the scene [13]. It is therefore highly desirable to automate this task.

In this paper, we propose to remove outliers in the laser scans by comparing the scanned depth to results from a stereo reconstruction approach using semi-global matching (SGM) [21]. While stereo reconstructions typically lead to depth bleeding artifacts at object boundaries, LiDaR sensors create streaking artifacts along their direction of motion. To remove both types of outliers, we enforce consistency between laser scans and stereo reconstruction and remove all LiDaR points exhibiting large relative errors. For comparing both measurements, we transform the SGM disparity maps to depth values using KITTI's provided calibration files. We further follow [13] and accumulate 11 laser scans to increase the density of the generated depth maps. While the environment is mostly static, some of the KITTI sequences comprise dynamic objects, where a laser scan accumulation causes many outliers on dynamic objects. Therefore, we use the SGM depth maps only once to clean the accumulated laser scan projection (instead of cleaning each laser scan separately) in order to remove all outliers in one step: Occlusions, dynamic motion and measurement artifacts. We also observed that most errors due to reflecting and transparent surfaces can be removed with this simple technique as SGM and LiDaR rarely agree in those regions.

4.1. Dataset Evaluation

Before using the proposed dataset for evaluation in Section 5, we verify its quality. Towards this goal, we exploit the manually cleaned training set of the KITTI 2015 stereo benchmark as reference data. We compute several error measures for our generated depth maps using the provided depth ground truth and compare ourself to the raw and accumulated LiDaR scans as well as the SGM depth

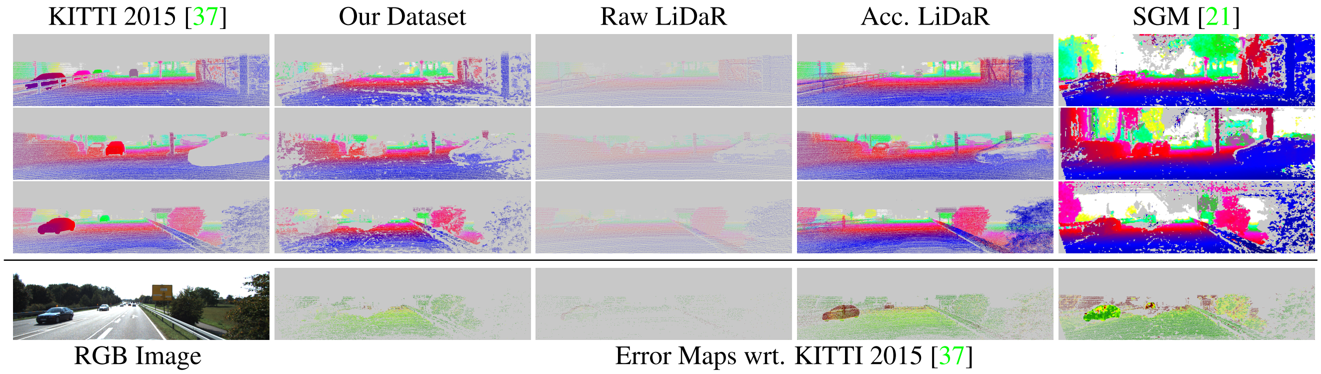


Figure 3: **Large-scale Dataset.** Qualitative results of our depth annotated dataset. From left to right we compare: depth maps of the manually curated KITTI 2015 dataset, our automatically generated data, raw and accumulated LiDaR scans, and SGM [21] results. Differences to the KITTI 2015 depth maps are shown in the last row from 0 (green) to 2 (red) meters.

maps in Table 1. The SGM reconstruction is very dense but also rather inaccurate compared to the raw laser scans. In terms of mean absolute error (MAE) our dataset reaches approximately the same accuracy level as the raw LiDaR scans. However, for the metrics "root mean squared error (RMSE)", "KITTI outliers" (disparity error $\geq$ 3px and $\geq 5\%$), as well as the δ inlier ratios (maximal mean relative error of $\delta_i = 1.25^i$ for $i \in \{1, 2, 3\}$), our dataset outperforms all baseline results. At the same time, we achieve four times denser depth maps than raw LiDaR scans. A qualitative comparison is presented in Fig. 3.

After manually separating the foreground and background regions on the benchmark depth maps, we evaluate the errors present on dynamic objects and background in Table 2. The result indicates that our proposed accumulation and clean-up pipeline is able to remove outliers in the raw LiDaR scans and at the same time significantly increases the density of the data. Qualitatively, we find only little errors in our dataset. Most of the remaining errors are located on dynamic objects or at high distances, cf. Fig. 3 (bottom). In comparison, SGM results are inaccurate at large distances and LiDaR scans result in occlusion errors due to the different placement of the LiDaR sensor and the virtual camera used for projection (we use the image plane of the KITTI reference camera for all our experiments). Note that dynamic objects (e.g., car on the left) lead to significant errors in the accumulated LiDaR scans which are largely reduced with our technique.

For the experimental evaluation in this work, we split our dataset into 86k images for training, 3k images for testing and 4k images for validation. For all splits we ensure a similar distribution over KITTI scene categories (city, road, residential and campus) while keeping the sequence IDs unique for each split to avoid overfitting to nearby frames. To bring forward the tasks of learned depth completion and single-image depth prediction, we will create a benchmark based on our data and an additional set of 1.5k held out test images. The benchmark paired with an online evaluation server will be published on the KITTI Vision homepage.

Table 1: Evaluation of reference depth maps using the manually curated ground truth depth maps of the KITTI 2015 training set [37]. Note that our dataset is generated fully automatically and achieves highest accuracy while providing high density. All metrics are computed in disparity space.

	Density	MAE [px]	RMSE [px]	KITTI outliers	δ_i inlier rates δ_1	δ_2	δ_3
SGM	82.4%	1.07	2.80	4.52	97.00	98.67	99.19
Raw LiDaR	4.0%	**0.35**	2.62	1.62	98.64	99.00	99.27
Acc. LiDaR	30.2%	1.66	5.80	9.07	93.16	95.88	97.41
Our Dataset	16.1%	**0.35**	**0.84**	**0.31**	**99.79**	**99.92**	**99.95**

Table 2: Evaluation of Table 1 split according to foreground (car) / background (non-car) regions.

Depth Map	MAE [px]	RMSE [px]	KITTI outliers	δ_i inlier rates δ_1	δ_2	δ_3
SGM	1.2/1.1	3.0/2.8	5.9/4.4	97.6 /96.9	98.2 /98.7	98.5/99.3
Raw LiDaR	3.7/**0.2**	10.0/1.9	17.4/0.9	84.3 /99.3	86.1 /99.6	88.6/99.7
Acc. LiDaR	7.7/1.1	12.0/4.8	59.7/4.3	55.7 /96.7	73.7 /98.0	83.0/98.8
Our Dataset	**0.9**/0.3	**2.2/0.8**	**3.0/0.2**	**98.6/99.8**	**99.0/99.9**	**99.3/99.9**

5. Experiments

5.1. Depth Upsampling

We investigate the task of depth map completion to evaluate the effect of sparse input data for our *Sparse Convolution Modules*. For this task, a sparse, irregularly populated depth map from a projected laser scan is completed to full image resolution without any RGB guidance.

We first evaluate the performance of our method with varying degrees of sparsity in the input. Towards this goal,

	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
5%	3.6	6.1	9.3	10.7	11.5	12.1	12.4	12.7	13.0	13.2	13.4
10%	7.1	2.4	5.4	7.2	8.3	8.9	9.3	9.5	9.5	9.4	9.1
20%	18.0	7.4	2.6	3.8	4.9	5.8	6.2	6.5	6.6	6.4	6.2
30%	23.6	11.9	4.1	1.9	2.6	3.4	3.9	4.0	4.1	3.7	3.7
40%	30.1	16.3	7.1	3.4	2.2	2.7	3.7	4.3	4.5	4.7	5.1
50%	46.5	24.9	26.7	5.8	2.8	1.8	4.3	3.2	3.8	4.3	4.9
60%	41.6	24.3	12.2	6.9	4.1	2.4	1.8	2.3	2.9	3.4	4.1
70%	64.9	38.4	19.5	11.8	7.1	4.2	2.5	1.7	2.2	2.9	3.8
80%	61.2	38.3	20.9	13.1	8.7	5.7	3.6	2.2	1.5	2.1	2.8
90%	159.6	84.4	41.1	25.2	16.5	11.0	7.2	4.4	2.4	1.4	2.4
100%	1288.5	886.9	506.6	318.9	202.9	128.0	76.6	37.7	14.8	6.1	1.2

(a) ConvNet

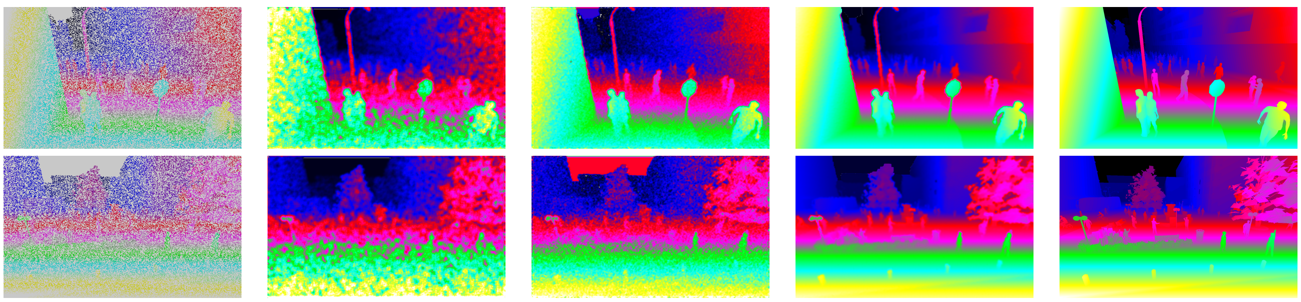
	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
5%	10.0	5.2	6.7	7.3	7.4	8.0	8.6	10.2	12.4	11.5	12.6
10%	16.6	13.3	5.7	8.7	10.3	11.2	13.0	12.0	12.5	12.5	14.2
20%	7.7	16.6	8.4	4.5	5.1	7.0	7.5	8.5	9.6	10.4	11.1
30%	8.5	6.4	5.8	6.3	4.3	7.1	14.2	33.4	46.4	35.1	23.6
40%	9.9	7.2	5.3	4.7	4.6	3.7	3.8	4.6	5.2	6.3	8.0
50%	12.1	8.3	5.5	3.9	4.8	5.1	4.5	5.1	6.6	17.4	166.5
60%	10.4	8.9	7.1	5.7	4.4	3.6	3.6	3.4	3.3	5.1	6.6
70%	14.7	9.9	7.8	6.9	5.8	4.4	3.3	3.5	3.9	6.1	8.0
80%	15.8	10.6	8.8	7.9	7.1	6.0	4.7	4.1	3.4	3.2	4.9
90%	13.4	10.0	7.9	6.7	5.7	5.3	4.7	3.7	3.0	6.4	17.4
100%	16.6	16.0	17.4	18.4	18.6	17.0	14.2	12.5	10.0	6.6	2.9

(b) ConvNet + mask

	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
5%	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
10%	0.98	0.98	0.97	0.98	0.97	0.97	0.97	0.97	0.97	0.97	0.97
20%	0.98	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97
30%	0.98	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.98
40%	0.99	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
50%	0.99	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
60%	0.99	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
70%	0.99	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
80%	0.99	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
90%	0.99	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
100%	0.99	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98

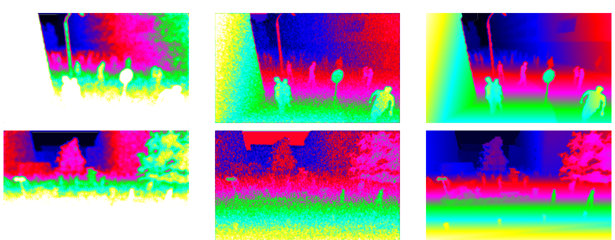
(c) SparseConvNet

Figure 4: Comparison of three different networks on the Synthia dataset [45] while varying the sparsity level of the training split (left) *and* the sparsity of the test split (top). From left-to-right: ConvNet, ConvNet with concatenated validity mask and the proposed SparseConvNet. All numbers represent mean average errors (MAE).



(a) Input (visually enhanced) (b) ConvNet (c) ConvNet + mask (d) SparseConvNet (ours) (e) Groundtruth

Figure 5: Qualitative comparison of our sparse convolutional network to standard ConvNets on Synthia [45], trained and evaluated at 5% sparsity. (b) Standard ConvNets suffer from large invalid regions in the input leading to noisy results. (c) Using a valid mask as input reduces noise slightly. (d) In contrast, our approach predicts smooth and accurate outputs.



(a) ConvNet (b) ConvNet + mask (c) SparseConvNet

Figure 6: Network predictions for scenes in Figs. 1 and 5, with all networks trained at 5% sparsity and evaluated at 20% sparsity. While ConvNets with and without visibility mask produce substantially worse results, the results of the proposed sparsity invariant CNN do not degrade.

we leverage the Synthia dataset of Ros et al. [45] which gives us full control over the sparsity level. To artificially adjust the sparsity of the input, we apply random dropout to the provided dense depth maps during training. The probability of a pixel to be dropped is set to different levels ranging from 0% to 95%.

We train three different variants of a Fully Convolutional Network (FCN) with five convolutional layers of kernel size 11, 7, 5, 3, and 3. Each convolution has a stride of one, 16 output channels, and is followed by a ReLU as nonlinear activation function. We use the Adam solver with momentum terms of 0.9 and 0.999 together with a fixed learning rate of $1 \cdot 10^{-3}$ and weight decay of $5 \cdot 10^{-4}$. The three variants we consider are: i) plain convolutions with only sparse depth as input, ii) plain convolutions with sparse depth and concatenated valid pixel map as input, and iii) the proposed *Sparse Convolution Layers*, cf. Fig. 2. We train separate networks for various levels of sparsity using the Synthia *Summer* sequences, whereas evaluation is performed on the Synthia *Cityscapes* dataset. To compare the performance of the different approaches we first evaluate them on the sparsity level they have been trained on. To test the generalization ability of the different models we further apply them to sparsity levels which they have not seen during training.

Fig. 4 shows our results. We observe that plain convolutions perform poorly with very sparse inputs as all pixels (valid and invalid) are considered in the convolution. This introduces a large degree of randomness during training and testing and results in strong variations in performance. Convolutions on sparse depth maps with the concatenated valid mask perform slightly better than using only the depth input. However, in contrast to our *Sparse Convolutions* they perform poorly, especially on very sparse input.

Invariance to the level of sparsity is an important property for depth completion methods as it increases robustness towards random perturbations in the data. Besides, this property allows to generalize to different depth sensors such

Table 3: Performance comparison (MAE) of different methods trained on different sparsity levels on Synthia and evaluated on our newly proposed KITTI depth dataset.

Sparsity at train:	5%	10%	20%	30%	40%	50%	60%	70%
ConvNet	16.03	13.48	10.97	8.437	10.02	9.73	9.57	9.90
ConvNet+mask	16.18	16.44	16.54	16.16	15.64	15.27	14.62	14.11
SparseConvNet	**0.722**	**0.723**	**0.732**	**0.734**	**0.733**	**0.731**	**0.731**	**0.730**

as structured light sensors, PMD cameras or LiDaR scanners. As evidenced by Fig. 4, all methods perform reasonably well at the performance level they have been trained on (diagonal entries) with the sparse convolution variant performing best. However, both baselines fail completely in predicting depth estimates on more sparse and, surprisingly, also on more dense inputs. In contrast, our proposed *Sparse Convolution Network* performs equally well across all levels of sparsity no matter which sparsity level has been observed during training. This highlights the generalization ability of our approach. Fig. 5 shows a qualitative comparison of the generated dense depth maps for the two baselines and our approach using 5% sparsity during training and testing. Note that the input in Fig. 5 (a) has been visually enhanced using dilation to improve readability. It thus appears more dense than the actual input to the networks. For the same examples, Fig. 6 shows the drastic drop in performance when training standard CNNs on 5% and evaluating on 20%, while our approach performs equally well. While ConvNets with input masks lead to noisy results, standard ConvNets even result in a systematic bias as they are unaware of the level of sparsity in the input.

5.1.1 Synthetic-to-Real Domain Adaptation

To evaluate the domain adaption capabilities of our method, we conduct an experiment where we train on the Synthia dataset and evaluate on our proposed KITTI validation set. Table 3 shows the performance of our network (SparseConv) as well as the two regular CNN baselines using the same number of parameters. Our experiments demonstrate that sparse convolutions perform as well on KITTI as on Synthia, while the dense baselines are not able to adapt to the new input modality and fail completely. We show qualitative results of this experiment in Fig. 7.

5.2. Comparison to Guided Upsampling

As discussed in the related work section, several approaches in the literature leverage a high resolution image to guide the depth map completion task which significantly facilitates the problem. Dense color information can be very useful to control the interpolation of sparse depth points, e.g., to distinguish between object boundaries and smooth surfaces. However, relying on camera information in multi-modal sensor setups, such as used in e.g. autonomous cars, is not always recommended. Bad weather and night scenes can diminish the benefit of image data or even worsen the result. Therefore, we target an approach which leverages depth as the only input in this paper.

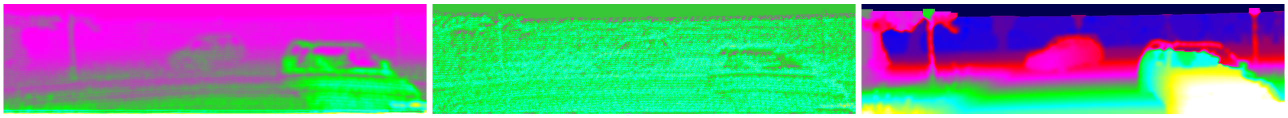
Table 4: Performance comparison of different methods on our KITTI depth dataset. Our method performs comparable to state-of-the-art methods that incorporate RGB (top), while outperforming all depth-only variants (bottom).

Method	RMSE [m]		MAE [m]	
	val	test	val	test
Bilateral NN [25]	4.19	5.233	1.09	1.09
SGDU [49]	2.5	2.02	0.72	0.57
Fast Bilateral Solver [1]	**1.98**	**1.75**	0.65	0.52
TGVL [9]	4.85	4.08	**0.59**	**0.46**
Closest Depth Pooling	2.77	2.30	0.94	0.68
Nadaraya Watson[38, 53]	2.99	2.86	0.74	0.66
ConvNet	2.97	2.69	0.78	0.62
ConvNet + mask	2.24	1.94	0.79	0.62
SparseConvNet (ours)	**2.01**	**1.81**	**0.68**	**0.54**

In this section, we show that despite not relying on guidance information, our approach performs on par with the state-of-the-art in guided depth completion and even outperforms several methods which use image guidance. Table 4 (top) shows a comparison of several state-of-the-art methods for guided filtering. In particular, we evaluated the methods of Barron et al. [1], Schneider et al. [49], Ferstl et al. [9], and Jampani et al. [25] which all require a non-sparse RGB image as guidance. For a fair comparison we added the same amount of convolutional layers as we use in our sparse convolutional network for Jampani et al. [26]. For the other baseline methods we optimized the hyper parameters via grid search on the validation split.

In addition, we compare our method to several depth-only algorithms in Table 4 (bottom). We first evaluate a simple pooling approach that takes the closest (distance to sensor) valid point to fill in unseen regions within a given window. Second, we apply the Nadaraya-Watson regressor [38, 53] using a Gaussian kernel on the sparse depth input. We optimized the hyperparameters of both approaches on the training data. We also compare our method to high-capacity baselines. In particular, we consider a standard ConvNet with and without visibility mask as additional feature channel.

It is notable that our approach performs comparable to state-of-the-art guided depth completion techniques despite not using any RGB information. In particular, it performs second in terms of RMSE on both validation and test split which we attribute to the Euclidean loss used for training.



(a) ConvNet (b) ConvNet + mask (c) SparseConvNet

Figure 7: Qualitative comparison of the best network variants from Table 3 trained on synthetic Synthia [45] and evaluated on the proposed real-world KITTI depth dataset. While our SparseConvNet adapts well to the novel domain, standard convolutional neural networks fail completely in recovering sensible depth information.

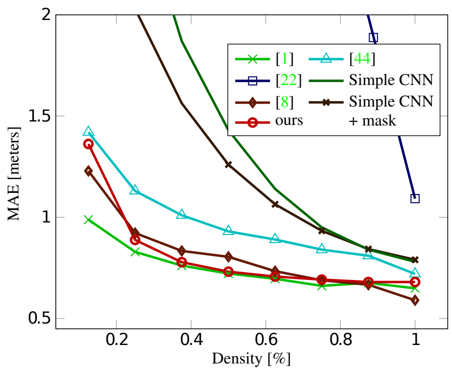


Figure 8: Quantitative results in MAE (meters) on our depth annotated KITTI subset for varying levels of input density. We compare our unguided approach to several baselines [1, 49, 9, 25] which leverage RGB guidance for upsampling and two standard convolutional neural networks with and without valid mask concatenated to the input.

5.2.1 Sparsity Evaluation on KITTI

In the KITTI dataset, a 64-layer laser scanner with a rotational frequency of 10 Hz was used to acquire ground truth for various tasks such as stereo vision and flow estimation. If projected to the image, the depth measurements cover approximately 5 % of the image. For industrial applications such as autonomous driving, often scanners with only 32 or 16 layers and higher frequencies are used. This results in very sparse depth projections. To analyze the impact of extremely sparse information, we evaluate the *Sparse Convolutional Network* and several baselines with respect to different levels of sparsity on our newly annotated KITTI subset. In particular, we train all networks using all laser measurements and evaluate the performance when varying the density of the input using random dropout. Our results in Fig. 8 demonstrate the generalization ability of our network for different levels of sparsity. Regular convolutions as well as several state-of-the-art approaches perform poorly in the presence of sparse inputs. Note that both Barron et al. [1] and Ferstl et al. [9] perform slightly better than our method on very sparse data but require a dense high-resolution RGB image for guidance.

Table 5: IoU performance of different network variants on the Synthia Cityscapes subset after training on all Synthia sequences (mean over all 15 known classes).

Network	IoU [%]
VGG - Depth Only	6.4
VGG - Depth + Mask	4.9
VGG - Sparse Convolutions	31.1

5.3. Semantic Labeling from Sparse Depth

To demonstrate an output modality different from depth, we also trained the well-known VGG16 architecture [35] for the task of semantic labeling from sparse depth inputs. We modify VGG16 by replacing the regular convolutions using our sparse convolution modules. Additionally, we apply the weighted skip connections presented in Section 3.2.1 to generate high-resolution predictions from the small, spatially downsampled FC7 layer, while incorporating visibility masks of the respective network stages.

Table 5 shows the mean performance after training on all Synthia "Sequence" frames (left camera to all directions, summer only) and evaluating on the Synthia "Cityscapes" subset. Again, we observe that the proposed sparse convolution module outperforms the two baselines. The comparably small numbers can be explained by the different nature of the validation set which contains more people and also very different viewpoints (bird's eye vs. street-level).

6. Conclusion

We have proposed a novel sparse convolution module for handling sparse inputs which can replace regular convolution modules and results in improved performance while generalizing well to novel domains or sparsity levels. Furthermore, we provide a newly annotated dataset with 93k depth annotated images for training and evaluating depth prediction and depth completion techniques.

In future work, we plan to combine the proposed sparse convolution networks with network compression techniques to handle sparse inputs while at the same time being more efficient. We further plan to investigate the effect of sparse irregular inputs for 3D CNNs [44].

References

[1] J. T. Barron and B. Poole. The fast bilateral solver. In *Proc. of the European Conf. on Computer Vision (ECCV)*, pages 617–632. Springer, 2016. 3, 7, 8

[2] D. Chan, H. Buisman, C. Theobalt, and S. Thrun. A Noise-Aware Filter for Real-Time Depth Upsampling. In *ECCV Workshops*, 2008. 3

[3] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3d object detection network for autonomous driving. *arXiv.org*, 1611.07759, 2016. 1, 2

[4] T. S. Cohen and M. Welling. Group equivariant convolutional networks. In *Proc. of the International Conf. on Machine learning (ICML)*, 2016. 2

[5] J. Diebel and S. Thrun. An application of markov random fields to range sensing. In *Advances in Neural Information Processing Systems (NIPS)*, 2005. 3

[6] J. Dolson, J. Baek, C. Plagemann, and S. Thrun. Upsampling Range Data in Dynamic Environments. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010. 3

[7] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *Proc. of the European Conf. on Computer Vision (ECCV)*, pages 184–199. Springer, 2014. 3

[8] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 38(2):295–307, 2016. 3

[9] D. Ferstl, C. Reinbacher, R. Ranftl, M. Ruether, and H. Bischof. Image Guided Depth Upsampling Using Anisotropic Total Generalized Variation. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2013. 3, 7, 8

[10] M. Figurnov, A. Ibraimova, D. P. Vetrov, and P. Kohli. Perforatedcnns: Acceleration through elimination of redundant convolutions. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 2

[11] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

[12] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research (IJRR)*, 32(11):1231–1237, 2013. 2, 4

[13] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012. 2, 4

[14] D. Glasner, S. Bagon, and M. Irani. Super-Resolution from a Single Image. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2009. 3

[15] B. Graham. Spatially-sparse convolutional neural networks. *arXiv preprint arXiv:1409.6070*, 2014. 2

[16] B. Graham. Sparse 3D convolutional neural networks. *Proc. of the British Machine Vision Conf. (BMVC)*, 2015. 2

[17] S. Han, J. Pool, J. Tran, and W. J. Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems (NIPS)*, 2015. 2

[18] S. Hawe, M. Kleinsteuber, and K. Diepold. Dense disparity maps from sparse disparity measurements. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2011. 3

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

[20] J. F. Henriques and A. Vedaldi. Warped convolutions: Efficient invariance to spatial transformations. *arXiv.org*, 1609.04382, 2016. 2

[21] H. Hirschmller. Stereo processing by semiglobal matching and mutual information. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 30(2):328–341, 2008. 4, 5

[22] M. Hornacek, C. Rhemann, M. Gelautz, and C. Rother. Depth super resolution by rigid body self-similarity in 3d. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013. 3

[23] T. Hui, C. C. Loy, and X. Tang. Depth map super-resolution by deep multi-scale guidance. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2016. 3

[24] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015. 2

[25] V. Jampani, M. Kiefel, and P. V. Gehler. Learning sparse high dimensional filters: Image filtering, dense crfs and bilateral neural networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 7, 8

[26] V. Jampani, M. Kiefel, and P. V. Gehler. Learning sparse high dimensional filters: Image filtering, dense CRFs and bilateral neural networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4452–4461, 2016. 7

[27] J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[28] R. Köhler, C. J. Schuler, B. Schölkopf, and S. Harmeling. Mask-specific inpainting with deep neural networks. In *Proc. of the German Conference on Pattern Recognition (GCPR)*, 2014. 2

[29] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele. Joint Bilateral Upsampling. *ACM Trans. on Graphics (SIGGRAPH)*, 26(3), 2007. 3

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012. 2

[31] D. Laptev, N. Savinov, J. M. Buhmann, and M. Pollefeys. TI-POOLING: transformation-invariant pooling for feature learning in convolutional neural networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[32] B. Li, T. Zhang, and T. Xia. Vehicle detection from 3d lidar using fully convolutional network. In *Proc. Robotics: Science and Systems (RSS)*, 2016. 1, 2

[33] B. Liu, M. Wang, H. Foroosh, M. Tappen, and M. Pensky. Sparse convolutional neural networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 2

[34] M. Liu, O. Tuzel, and Y. Taguchi. Joint geodesic upsampling of depth images. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013. 3

[35] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015. 4, 8

[36] O. Mac Aodha, N. D. Campbell, A. Nair, and G. J. Brostow. Patch based synthesis for single depth image super-resolution. In *Proc. of the European Conf. on Computer Vision (ECCV)*, pages 71–84. Springer, 2012. 3

[37] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015. 4, 5

[38] E. Nadaraya. On estimating regression. In *Theory of Probability and its Applications*, 1964. 7

[39] N.Mayer, E.Ilg, P.Häusser, P.Fischer, D.Cremers, A.Dosovitskiy, and T.Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

[40] J. Park, H. Kim, Y. Tai, M. S. Brown, and I. Kweon. High quality depth map upsampling for 3d-tof cameras. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2011. 3

[41] J. Park, S. R. Li, W. Wen, H. Li, Y. Chen, and P. Dubey. Holistic sparsecnn: Forging the trident of accuracy, speed, and size. *arXiv.org*, 1608.01409, 2016. 2

[42] G. Riegler, D. Ferstl, M. Rüther, and H. Bischof. A deep primal-dual network for guided depth super-resolution. *arXiv preprint arXiv:1607.08569*, 2016. 3

[43] G. Riegler, M. Rüther, and H. Bischof. ATGV-net: Accurate depth super-resolution. In *Proc. of the European Conf. on Computer Vision (ECCV)*, pages 268–284. Springer, 2016. 3

[44] G. Riegler, A. O. Ulusoy, and A. Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 8

[45] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 4, 6, 8

[46] A. Saxena, M. Sun, and A. Y. Ng. Make3D: learning 3D scene structure from a single still image. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 31:824–840, 2009. 4

[47] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nesic, X. Wang, and P. Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *Proc. of the German Conference on Pattern Recognition (GCPR)*, 2014. 4

[48] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision (IJCV)*, 47:7–42, 2002. 4

[49] N. Schneider, L. Schneider, P. Pinggera, U. Franke, M. Pollefeys, and C. Stiller. Semantically guided depth upsampling. In *Proc. of the German Conference on Pattern Recognition (GCPR)*, pages 37–48. Springer, 2016. 3, 7, 8

[50] P. Y. Simard, D. Steinkraus, and J. C. Platt. Best practices for convolutional neural networks applied to visual document analysis. 2003. 2

[51] X. Song, Y. Dai, and X. Qin. Deep depth super-resolution: Learning depth super-resolution using deep convolutional neural network. In *Proc. of the Asian Conf. on Computer Vision (ACCV)*, 2016. 3

[52] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015. 4

[53] G. Watson. Smooth regression analysis. In *Sankhy: The Indian Journal of Statistics*, 1964. 7

[54] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li. Learning structured sparsity in deep neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 2

[55] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[56] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010. 3

[57] Q. Yang, R. Yang, J. Davis, and D. Nister. Spatial-Depth Super Resolution for Range Images. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007. 3

[58] Y. Zhou, Q. Ye, Q. Qiu, and J. Jiao. Oriented response networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[59] S. Zweig and L. Wolf. Interponet, A brain inspired neural network for optical flow dense interpolation. *arXiv.org*, 1611.09803, 2016. 2