Learning Morphological Operators for Depth Completion

Martin Dimitrievski[1], Peter Veelaert[1], Wilfried Philips[1]

IMEC-IPI-Ghent University, Ghent, Belgium
(martin.dimitrievski,peter.veelaert,wilfried.philips@ugent.be)

Abstract. Depth images generated by direct projection of LiDAR point clouds on the image plane suffer from a great level of sparsity which is difficult to interpret by classical computer vision algorithms. We propose a method for completing sparse depth images in a semantically accurate manner by training a novel morphological neural network. Our method approximates morphological operations by Contraharmonic Mean Filter layers which are easily trained in a contemporary deep learning framework. An early fusion U-Net architecture then combines dilated depth channels and RGB using multi-scale processing. Using a large scale RGB-D dataset we are able to learn the optimal morphological and convolutional filter shapes that produce an accurate and fully sampled depth image at the output. Independent experimental evaluation confirms that our method outperforms classical image restoration techniques as well as current state-of-the-art neural networks. The resulting depth images preserve object boundaries and can easily be used to augment various tasks in intelligent vehicles perception systems.

1 Introduction

Recent advances in active depth sensing technologies such as high resolution LiDAR and Time of Flight cameras have extended the applications where robustness and accuracy have been a limiting factor in the past. This is especially true in the field of robotics and computer vision where solving high level problems such as autonomous navigation relies on a rich, multi-modal information. Until recently, most of such applications reinforced visible light images with per-pixel depth obtained using stereo cameras. Even though, stereo reconstruction has been widely researched and many high performance solutions do exist , e.g. close to 100 submissions on the KITTI Stereo 2015 benchmark[1], it still suffers from the effect of measurement correlated noise. As the 3D point is further away from the sensor, the perceived image disparity between the stereo cameras drops exponentially and so does the accuracy of it's estimate. Furthermore, the quality of reconstructed depth is directly coupled to the camera baseline which creates another set of challenges. A small autonomous robot cannot carry a wide baseline stereo rig, limiting the efficiency of any stereo reconstruction algorithm

[1] http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=stereo

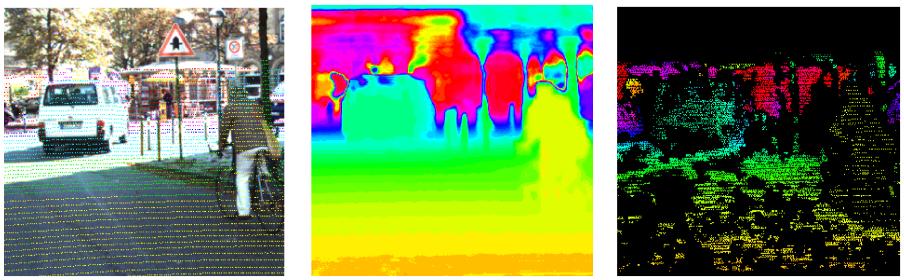

Fig. 1. Illustration of the depth completion problem. Left: input data is given in the form of registered RGB image and projected LiDAR points. Middle: Fully reconstructed depth image, output of the proposed method. Right: dense ground truth data used for training.

applied on the data. On the other hand, wide baseline stereo setups suffer greatly from disparity artifacts around object boundaries that are close by. Solving for these issues is a challenging task which more often than not requires additional computational load on the perception system. Active depth sensors contrast the principle of depth reconstruction by sending out well controlled infra-red pulses into the surroundings. This infra-red light reflects back from the environment and can be correlated to distances in a systematic way. At the time of writing, commercial real-time LiDAR sensors can reliably measure depth in the range of $50cm$ to $80m$ while at the same time sustain low noise levels that are uncorrelated with the measurements. Data is being produced in the form of 3D point clouds which have excellent depth resolution, but rather low spatial resolution. In order to achieve real time operation, the LiDAR electronics sample depth at predefined sparse azimuth and elevation latices. Usually, the spatial resolution is lower in the elevation axis, which compared to the contemporary camera sensors varies in the range of 1 depth scan-line for each 5-10 lines of RGB data. See the left image in figure 1 for illustration of this effect.

Consequently, a direct projection of a LiDAR point cloud on the camera image produces a very sparse depth image. Classical computer vision algorithms such as visual odometry, scene understanding, segmentation, object detection, etc. have difficulties extracting useful information from this sparse input. Many of the processing steps need to be specifically tuned to incorporate sparse depth pixels and ignore missing depth values which reduces the efficiency and in turn usefulness of state-of-the-art algorithms. The problem of achieving equal sampling resolution of the RGB camera and depth sensor is called depth completion and is the main topic of this paper. A non-sparse data cube consisting of reconstructed depth pixels, middle image on figure 1, can be easily interpreted by

classical computer vision algorithms. We have shown that pedestrian detection, in particular, can achieve much higher performance when operating on a RGB-D data reconstructed from a camera-LiDAR pair [1]. Even though many of the proposed depth completion methods produce dense and visually pleasing depth images, the depth completion problem is not entirely solved. A seemingly under explored track is the exploitation of contextual information in the camera image in order to produce more accurate depth in a semantically meaningful manner. In this paper we propose a novel neural network architecture which is capable to complete missing depth pixels by employing a mixture of morphological and convolutional layers. Learnable morphological operators provide robustness to the input sparsity, while multi-resolution convolutional layers extract contextual information about object shapes and boundaries. Our network reconstructs a depth image which is complete, accurate and preserves object edges.

In the following section we will make a brief overview of how the state-of-the-art handles context in the depth completion problem, then in section §3 we will define a novel neural network architecture suited for fusion of RGB information with sparse depth input, furthermore in section §4 we perform large scale evaluation of the proposed method and report the accuracy and performance, and we discuss the effectiveness and possible downsides of using our method in section §5.

2 Overview

One of the pioneering depth completion methods, [2] considered to estimating each pixel location in the sparse depth image by means of local interpolation within a square window. The authors analyzed various classical reconstruction techniques which rely on depth information alone such as inverse distance weighting, Shepard's Method, ordinary Kriging, Delaunay triangulation and bilateral filtering. Furthermore they introduced a modified bilateral filter which also considers depth dispersion within the interpolation window. This method can crudely model the appearance of an object edge or boundary into two categories: foreground and background. A local segmentation is performed on the depth pixels which produces two clusters from which only the points that belong to the dominant cluster contribute to the bi-lateral filter. These authors found out that even simple techniques such as the minimum and median filter can complete missing depth with comparable accuracy to the more complex bi-lateral filter. A major drawback in this work is the simple model of the environment which doesn't take into account the geometrical and contextual structure of objects.

Following the success of the bi-lateral filter, we have proposed a semantically aware multi-lateral filter, [1] that is guided by a segmentation image. The segmentation image is computed by segmenting the LiDAR point cloud in a pre-processing step and is independent on the filtering window size. In our modified multi-lateral filter we use both IR-reflectance and depth pixels originating from the dominant object within the reconstruction window. Although we report

state-of-the-art performance on a small scale database, we encountered difficulties obtaining an accurate segmentation image due to the point cloud sparsity. This effect is especially pronounced in distant objects that are sampled by only a few points. Contextual information originating from the RGB camera image remained untapped.

Ku et al. [3] propose a surprisingly simple yet efficient depth completion method using a sequence of morphological operations on the sparse depth image. In their experiments they show that a small set of fine tuned dilations and erosions is enough to reconstruct a high quality depth image. By experimenting with various kernel sizes and shapes they come to the conclusion that 5×5 diamond shaped morphological operators are able to outperform even some neural network based methods. However, higher level information about object types and shapes is completely ignored, which can potentially lead to even better reconstructions.

Recently, [4] proposed a method for semantically guided depth completion by means of local plane fitting. They use the assumption that the environment is locally smooth and can be piece-wise modeled by 3D planes. With the intention of preserving depth discontinuities and tiny structures, they introduce an novel edge and semantics aware geodesic distance metric. Additionally, they propose an outlier rejection scheme by utilizing labels from the state-of-the-art semantical segmentation algorithm, FCN [5]. Their reported qualitative results are promising, however, the method is not monolithic as it relies on different technologies and has been outperformed by special purpose neural networks.

Uhrig et al. [6] propose a depth completion method by processing the raw RGB-D data cube using a novel neural network. They propose a sparsity invariant convolutional layer which is built using an additional sampling mask. The mask holds binary information about which pixel is scanned by the LiDAR and is used to normalize the convolutional operations. Therefore, the network can easily handle varying degrees of input data sparsity without any adjustments or tweaking of the parameters. One downside of this method is that the network is based on the Fully Convolutional architecture which has a high computational load. Each inference produces a single depth pixel value and thus can not be employed in real time applications.

3 Method

3.1 General Architecture

Our depth completion system is a transfer function which takes sparse depth images and corresponding camera frames as inputs and outputs fully sampled depth images. The system consists of a fixed and a trainable part. In the fixed part, we make full use of previous state-of-the-art interpolation knowledge, as implemented in [3], and for the trainable part we propose a novel morphological CNN based on the principles of U-Net [7], SegNet [8] and ResNet [9]. We argue that local depth information alone is not enough to fill-in the missing depth

values in a semantically meaningful way. Therefore, our CNN processes RGB-D information at multiple scales, learning an optimal depth reconstruction function that in part is guided by color information. The fixed part of the pipeline can be seen as a pre-processing step used to better initialize the later CNN. Even though all of the network weights are actually initialized using the Xavier method [10], we noticed that convergence is reached much faster when the preprocessed data is introduced.

Standard 2D convolution operations have difficulties in learning sparse data input problems [6], [11]. This is especially true when it is necessary to distinguish between actual measurement values and invalid pixels. Therefore, the entry point of our CNN is a series of trainable morphological filters operating on the sparse depth images alone. The purpose of this morphological sub-network is to separate the low level RGB and depth processing pipelines in order to better learn an initial depth image estimate. We approximate morphological dilation and erosion operations by utilizing the limit behavior of the Contraharmonic Mean Filter (CHM). These filters can be easily implemented in most contemporary deep learning frameworks by means of standard convolutional layer building blocks. In the later layers, morphologically processed depth and RGB information is fused using standard convolutional layers. Instead of operating directly on the depth values, our morphological network operates on inverse depth (disparity) values which are then converted back to depth when computing the loss function. The network is designed to output a depth patch with equal spatial size of the input patch and can thus process high resolution data in reasonable time.

3.2 Morphological Filtering with a Contraharmonic Mean Filter

Morphological operators are the foundation of many image segmentation algorithms. Using so called "structuring elements" they represent non-linear operations which compute the minimum, maximum or the combination of both within the element. Morphological operations are also invariant to translation and are strongly related to Minkowski addition. In the context of depth completion, it is of interest for the system to learn the shape and the operation type that fits best the data. However, due to the non-differentiable nature of minimum and maximum filtering, only few approaches have been found to succeed in the literature. To this end, we find that the approximation of morphological operators by the contraharmonic mean (CHM) filter in [12] is the best founded technique which can easily be integrated in a deep learning framework. In this paper, we also use the CHM to approximate our basic learn-able morphological block. Following the analysis in [12], [13] and [14], we model the contraharmonic mean filter function $\psi_k(x)$ as the power-weighted 2D convolution of the image $f(x)$ and a filter w representing the structuring element:

$$\psi(x; W, k) = \psi_k(x) = \begin{cases} \frac{\left(f^{k+1} * w\right)(x)}{(f^k * w)(x)} & if\ k \in \mathbb{R} \\ \min\left(f_W(x)\right) & if\ k = -\infty \\ \max\left(f_W(x)\right) & if\ k = +\infty \end{cases}, \tag{1}$$

Algorithm 1 MATLAB snippet for initializing the morphological operator using AutoNN and MatConvNet framework

```
function  out = morphFilter(x,weights,biases,k)
          x1 = x.^(k+1);
          x1 = vl_nnconv(x1,weights,biases);
          x2 = x.^k;
          x2 = vl_nnconv(x2,weights,biases);
          out = x1./x2;
end
```

where the order k of the filter defines the desirable properties such as morphological erosion if $k \ll 0$ or dilation if $k \gg 0$, W defines the set of values in the support of w and f^k is simply the input image raised to the power k. When k is large, the filter tends to select the pixels x_i with the largest values within the support region W which in the limit case, $k \to \infty$ equates to the supremum, i.e. morphological dilation:

$$\lim_{k\to\infty} \psi_k(x) = \max_{x_i \in W} \left(f\left(x - x_i\right)\right) \equiv \psi_\infty(f). \tag{2}$$

Otherwise, when k is sufficiently small, the CHM filter will tend to select the smaller valued pixels which in the limit case $k \to -\infty$ equates to the infimum, i.e. the morphological erosion:

$$\lim_{k\to-\infty} \psi_k(x) = \min_{x_i \in W} \left(f\left(x - x_i\right)\right) \equiv \psi_{-\infty}(f). \tag{3}$$

In practice, the choice of k, and thus computing the derivative, will be limited by the computer number precision, but we found that a value of $k = 5$ produces the desired morphological filtering effect using single-precision floating point filter and pixel values. For a more detailed analysis of the filter properties and their proofs we advise the reader to the works of van Vliet [13] and Angulo [14]. This filter formulation is differentiable with respect to the input data x and both the filter mask w and the filter order k, as given in [12]. However, in our problem where we only encounter empty regions which we need to complete, we fix the order k to a positive value and are mainly focused on learning the structuring element w, hence the partial derivative of the filter with respect to the input is:

$$\frac{\partial \psi_k(x)}{\partial f} = \frac{f(x) - f(x)\left(f^{k+1} * w\right)(x)}{\left(f^k * w\right)(x)}, \tag{4}$$

and is used while applying the chain rule in back propagation. In practice we use the MatConvNet [15] framework with the AutoNN implementation of automatic differesntiation API which successfully handles the inference and back-propagation operations. Formally, the CHM filter is implemented using two convolution layers representing the denominator and numerator in (1). The convolution layers share the same filters and biases and have the same learning rates.

An extract of our initialization procedure for a single morphological operator is given in algorithm 1. The learned morphological operators can thus be visualized by taking the logarithm $\mathfrak{m}_i(x) = \log(w)$. We note that the input data for our morphological operators is the inverse depth (disparity) image rather than the raw depth values. This is done because of the nature of the morphological dilation operation in gray level images, where pixels with larger values are extended by the shape of the structuring element. In cases where an area to be dilated is completely filled with measurements, the resulting dilation will accentuate objects that are nearer to the camera (lower depth, greater disparity), rather than the background. This result is more desirable since it is safer to assume that no object with size less than half of the structuring element will be completely lost by applying the dilation.

3.3 Proposed Network Architecture

Depth completion using a regression neural network can be performed in three different ways. Depth can be reconstructed per pixel, per patch or per entire frame. While processing each pixel individually enables us to use a very deep CNN, in reality it's deployment is intractable due to long computing times. On the other hand, reconstructing the entire image in a single pass is most desirable, but it is difficult to achieve acceptable results since processing is limited by image resolution and GPU memory. Single image inference networks must be kept relatively simple which limits their performance in terms of error rates. In this paper we propose a medium sized CNN architecture which can process sparse patches of the input RGB-D images. A complete dense depth image is thus formed by processing overlapping patches over the input image. The general structure of the proposed network is loosely based on the auto-encoder with information leak proposed in [7] and [8] for high resolution segmentation tasks.

We exploit the knowledge from [3] and start with a fixed pre-processing morphological block consisting of a sequence of dilation, closing and two hole filling operators. In a separate step we use two learnable CHM filter layers to infer the optimal morphological operations from the sparse input, each with a support of 11×11 pixels. In the end, the RGB channels are concatenated with the output of [3] and the output of our CHM filters and are passed through a three stage U-Net, figure 2.

The novelty of this paper comes from the introduction of morphological layers before the contracting part which helps to eliminate the sparsity in the input. By applying a succession of convolutional filters we then double the number of channels and reduce the spatial resolution by half. Having learned the optimal dilation structuring elements in the CHM filter, the U-Net channels have no sparsity and can easily adapt to produce the desired output. At the end of each contracting block we specifically chose to use average pooling layers because they capture all of the information necessary for later depth reconstruction. In the expanding part, the network employs standard series of "deconvolution" layers that upsample their input by a factor of $2\times$ and concatenate outputs from the respective contracting layers. The purpose of the expanding blocks is to reduce

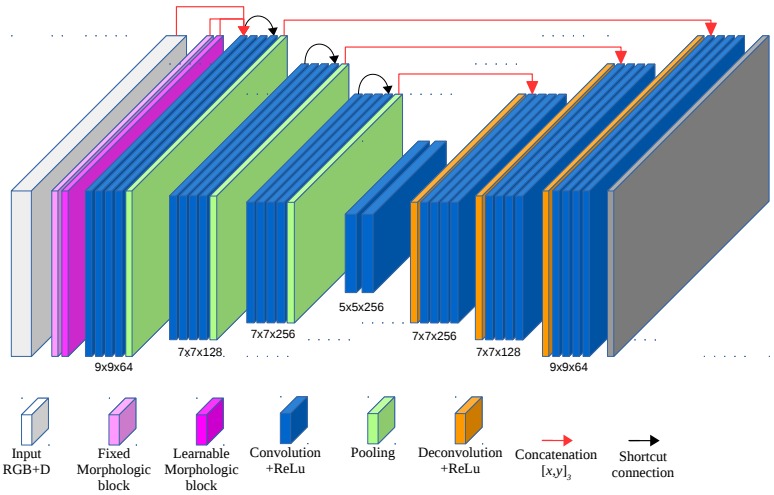

Fig. 2. General structure of the proposed network.

the number of contextual information stored within the channels while at the same time increase the spatial resolution. This way, information over varying resolutions and abstraction levels in both RGB and input depth is used to form high resolution depth output which adheres to object boundaries.

In order to learn the shape of the morphological and convolutional filters we use dense ground truth depth images to which we compare the output of our proposed network. Deviations from the ground truth can be quantified by a multitude of different metrics, such as absolute error, squared error, inverse absolute error, inverse squared error, absolute and squared relative error, percentage of outliers, etc. but in this paper we stick to the classical mean squared error as our loss function. We are motivated to do so because we are interested in accurate depth reconstructions which can later be used for the tasks of autonomous driving. Therefore, object distances need to be accurate regardless of their absolute distance to the sensor, i.e. we equally penalize error whether it is for a distant or a close object. For training the entire network we use the backpropagation chain rule and employ the stochastic gradient descent by adaptive moment estimation (ADAM) technique, [16]. This method computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients. We set an initial learning rate $\left(\alpha = 10^{-5}\right)$ and two hyper-parameters: decay for the first moment vector $(\beta_1 = 0.9)$ and decay for the second moment vector $(\beta_2 = 0.999)$.

4 Experiments

4.1 General

Learning a robust set of depth completion filters requires a large and variable set of input and ground truth data. The very recently published KITTI depth completion[2] data [6,17,18] is an excellent example of such dataset in the context of depth sensing for autonomous navigation. It consists of video sequences captured by a stereo RGB camera pair as well as point clouds from the Velodyne HDL-64E LiDAR. Each point cloud is projected on a virtual camera image creating a sparse depth image, previously visualized on the left image in figure 1. There is a total of 151 sequences with 93505 frames split into $\sim 92\%$ training and the remaining ones for model validation. Independent method evaluation is also provided by means of an on-line server which tests uploaded results to frames with unknown ground truth data.

4.2 Data Preparation

In order to reduce the computational burden and expedite the training process we only use a small sub-sample of the training set. We noticed that, due to the relatively high sampling rate, most of the 93K frames in the KITTI dataset contain temporally correlated information. Additionally, many of the sequences are recorded from a static vehicle and thus contain a large portion of the same content. Thus, in all our experiments we removed most of the static sequences and only sample every 6^{th} frame from the remaining data. We ended up with a training set of $\sim$ 4.3K samples. Input images are padded to a fixed resolution of 1280×384 pixels from which we randomly sample rectangular patches of size 96×96. Since our network uses 3 stage contraction, the lowest resolution of the input image inside the network is 12×12 with a channel depth of 256.

4.3 Training Procedure

Learning of the optimal network parameters is done by presenting the network with sub-batches of the labeled training set. After each inference, batch-average MSE is calculated from ground truth and the value is used as a loss to adjust the convolution filter parameters. Each successive layer is updated by backpropagation using the chain rule. We employ the ADAM optimization method and, since we train using small patches of images, we train until convergence for ~ 200 epochs. During training we keep the hyper-parameters α, β_1 and β_2 fixed, but adaptively change the batch size, starting from 4 increasing to 64. After each epoch, we also perform validation using a small *sub-set* of the validation dataset.

[2] http://www.cvlibs.net/datasets/kitti/eval_depth.php?benchmark=depth_completion

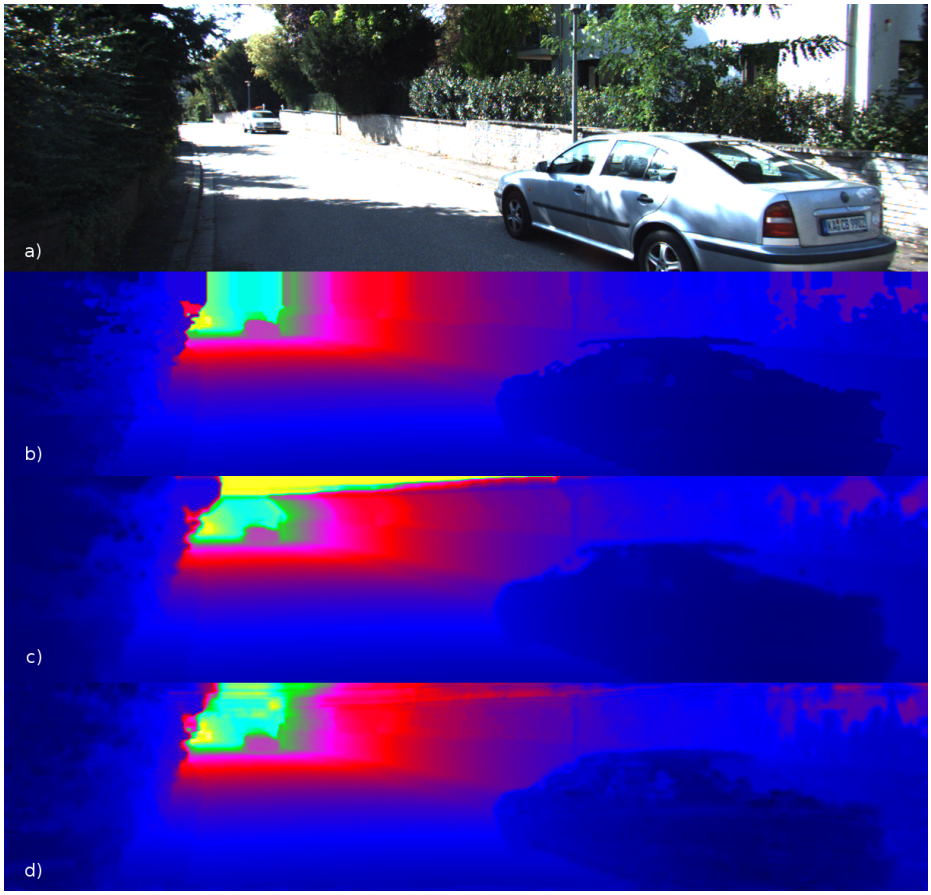

Fig. 3. Typical use case scenario in an urban environment. From top to bottom: a) RGB camera frame, b) completed depth image from [3], c) completed depth image from [6] and d) completed depth image from the proposed method.

4.4 Analysis

We deployed our trained neural network on the 1000 test samples from the KITTI depth completion benchmark. The accuracy of our method in terms of iRMSE, iMAE, RMSE and MAE is independently measured by the KITTI online server and compared to various techniques. This benchmark contains various anonymous submissions, summarized on Table 1, to which we can't fully compare since we don't know their exact details. To that end, we match our results only to results from publicly available techniques. In terms of raw RMSE error we outperform classical methods such as [4] and [3], as well as the only published CNN method [6]. Qualitatively, our method also better preserves object boundaries which is visible from the results shown on figure 3. Using the RGB information in the contracting and expanding network architecture, we are able

Table 1. Depth completion results on the KITTI benchmark.

Method	iRMSE	iMAE	RMSE	MAE	Runtime	Source
HMS-Net_v2	3.90	1.90	911.49	310.14	0.02 s / GPU	n/a
Sparse-to-Dense-2	3.21	1.35	954.36	288.64	0.04 s / GPU	n/a
HMS-Net	3.25	1.27	976.22	283.76	0.02 s / GPU	n/a
Morph-Net	**3.84**	**1.57**	**1045.45**	**310.49**	**0.17 s / GPU**	**Proposed**
IP-Basic	3.78	1.29	1288.46	302.60	0.011 s / 1 core	Ku [3]
ADNN	59.39	3.19	1325.37	439.48	0.04 s / GPU	n/a
NN+CNN	3.25	1.29	1419.75	416.14	0.02 s /	Uhrig [6]
SparseConvs	4.94	1.78	1601.33	481.27	0.01 s /	Uhrig [6]
NadarayaW	6.34	1.84	1852.60	416.77	0.05 s / 1 core	Uhrig [6]
SGDU	7.38	2.05	2312.57	605.47	0.2 s / 4 cores	Schneider [4]
NiN CNN	4.60	2.15	2378.79	685.53	0.01 s /	n/a
NiN+Mask CNN	4.63	2.40	2534.26	848.25	0.01 s / GPU	n/a

to effectively fill in missing object parts with the relevant depth information. This is especially noticeable in transparent objects such as house and car windows and glass displays. The inclusion of morphological layers makes the network flexible enough so that sparse data is handled in the initial layers, while the rest of the network is dedicated to better extracting contextual information.

5 Conclusion

Depth completion from sparse inputs has traditionally been solved by local image processing that handles sparsity using fine tuned filters. However, in instances where the level of sparsity varies spatially or parts of objects are completely missing, local processing is unable to accurately reconstruct depth information. Contextual information from the entire scene or parts of objects must be considered to better fill-in missing depth. We explored the idea of adding learnable morphological filters to a state-of-the-art multi-scale neural network in order to reduce the input sparsity. At the same time, these filters are adjusted for optimal reconstruction quality. We implemented morphological operators using the CHM filter which can be constructed out of standard deep learning building blocks. In terms of reconstruction accuracy, our method outperforms all published classical and neural network based approaches. It operates on image stripes which are concatenated to form the final depth image. The run-time for completing a single image (inference) of size 1280x384 pixels is on average $1.75s$. A future direction of our research will be to investigate which additional steps need to be taken in order to reduce the complexity of the network while not sacrificing reconstruction accuracy.

Acknowledgements

"The Titan Xp used for this research was donated by the NVIDIA Corporation through the Academic Grant Program."

References

1. M. Dimitrievski, P. Veelaert, and W. Philips, "Semantically aware multilateral filter for depth upsampling in automotive lidar point clouds," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1058–1063, June 2017.
2. C. Premebida, L. Garrote, A. Asvadi, A. P. Ribeiro, and U. Nunes, "High-resolution lidar-based depth mapping using bilateral filter," *CoRR*, vol. abs/1606.05614, 2016.
3. J. Ku, A. Harakeh, and S. L. Waslander, "In defense of classical image processing: Fast depth completion on the CPU," *CoRR*, vol. abs/1802.00036, 2018.
4. N. Schneider, L. Schneider, P. Pinggera, U. Franke, M. Pollefeys, and C. Stiller, "Semantically guided depth upsampling," in *Pattern Recognition* (B. Rosenhahn and B. Andres, eds.), (Cham), pp. 37–48, Springer International Publishing, 2016.
5. J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *CoRR*, vol. abs/1411.4038, 2014.
6. J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant cnns," *CoRR*, vol. abs/1708.06500, 2017.
7. O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015.
8. V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *CoRR*, vol. abs/1511.00561, 2015.
9. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.
10. X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (Y. W. Teh and M. Titterington, eds.), vol. 9 of *Proceedings of Machine Learning Research*, (Chia Laguna Resort, Sardinia, Italy), pp. 249–256, PMLR, 13–15 May 2010.
11. X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," *CoRR*, vol. abs/1611.07759, 2016.
12. J. Masci, J. Angulo, and J. Schmidhuber, "A learning framework for morphological operators using counter-harmonic mean," *CoRR*, vol. abs/1212.2546, 2012.
13. L. J. V. Vliet, "Robust local max-min filters by normalized power-weighted filtering," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 1, pp. 696–699 Vol.1, Aug 2004.
14. J. Angulo, "Pseudo-morphological image diffusion using the counter-harmonic paradigm," in *Advanced Concepts for Intelligent Vision Systems* (J. Blanc-Talon, D. Bone, W. Philips, D. Popescu, and P. Scheunders, eds.), (Berlin, Heidelberg), pp. 426–437, Springer Berlin Heidelberg, 2010.
15. A. Vedaldi and K. Lenc, "Matconvnet – convolutional neural networks for matlab," in *Proceeding of the ACM Int. Conf. on Multimedia*, 2015.
16. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.
17. A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, June 2012.
18. A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.