2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

# Real-time Semantic Segmentation-based Depth Upsampling using Deep Learning

Vlad-Cristian Miclea and Sergiu Nedevschi

***Abstract*— We propose a new real-time depth upsampling method based on convolutional neural networks (CNNs) that uses the local context provided by semantic information. Two solutions based on convolutional networks are introduced, modeled according to the level of sparsity given by the depth sensor. While first CNN upsamples data from a partial-dense input, the second one uses dilated convolutions as means to cope with sparse inputs from cost-effective depth sensors. Experiments over data extracted from Kitti dataset highlight the performance of our methods while running in real-time (11 ms for the first case and 17 ms for the second) on a regular GPU.**

## I. INTRODUCTION

Depth perception is a key aspect in autonomous driving. Stereo reconstruction is the traditional method for depth measurement providing very accurate solutions at relatively low cost. Due to the apparition of LiDAR (Light Detection and Ranging), a more robust and trustworthy sensor, the ubiquitous usage of stereo has gradually decreased. Although extremely accurate for depth measurements, LiDAR has the disadvantage of being more expensive and of giving sparse results.

With the increased prosperity of deep learning, convolutional neural networks have been employed for depth map enhancement. CNNs enable methods dealing with stereo cost computation [1], optimization [2], post-processing [3], end-to-end stereo [4] [5], LiDAR-based depth upsampling [6] [7] or even single image depth estimation [8].

According to their completeness [9], depth maps have been classified as:

- Sparse – Generated by a cost-effective (4-ray or 16-ray) LiDAR, or by a feature-based reconstruction system [10]. These methods provide quite few, but extremely reliable depth points;
- Partial-Dense (Semi-Dense) – Generated by a 64-ray, 128-ray LIDAR sensor or by a low resolution stereo reconstruction method. Although still being sparse, the final depth map given by these method has an improved structure, image objects being better delimited one from another;
- Full-Dense – Generated by either a very accurate stereo reconstruction method [1], by means of tracking and fusion of multiple images, or by structured-light systems (eg. Microsoft Kinect). Such methods produce a value for each pixel in the depth image.

The authors are with the Department of Computer Science, Technical University of Cluj-Napoca, Cluj-Napoca, Romania, E-mails: Vlad.Miclea@cs.utcluj.ro, Sergiu.Nedevschi@cs.utcluj.ro

We tackle here the problem of depth image upsampling by using convolutional neural networks as mechanisms to densify depth maps with low resolution. A CNN-based upsampling method that directly converts from sparse to full dense might lead to less robust or too dateset-dependent results. Therefore we split the upsampling problem in two more feasible tasks, proposing two upsampling solutions. Both methods take advantage of the information provided by (i) the RGB image and (ii) semantic segmentation of the RGB image. The two methods can upsample the depth image:

- From Partial-Dense to Full-Dense. We propose a CNN architecture that receives simultaneously the depth, RGB and the semantic maps; it extracts the most relevant information from each particular feature map and it combines this data generating an improved depth image;
- From Sparse to Partial-Dense. We rely on a similar convolutional network that additionally uses dilated convolutions to deal with the lack of structure found in very sparse depth images.

The paper starts with presenting the state of the art in depth upsampling and semantic segmentation-based depth generation solutions. Next section describes the proposed ConvNet-based partial-dense to full-dense architecture. Moreover, it describes the required datasets along with parameters and other training details. In section 4 we discuss the particular enhancements we propose when creating the sparse to partial-dense CNN architecture. Section 5 presents a thorough evaluation of the improvements given by our methods in driving scenarios. Finally, we conclude the paper in section 6.

## II. RELATED WORK

### A. Depth densification methods

The problem of depth upsampling was initially studied in the context of sparse stereo reconstruction. The authors of [11] propose a method to increase the resolution of disparity maps by incorporating prior disparity knowledge into their current frame that consists in only very reliable points.

Classic approaches that deal with small depth inconsistencies (partial-dense to dense category) generally apply (edge-aware) filtering techniques such as median filters [12], bilateral [13] [14] or guided [15]. An edge-aware deep learning-based guided filter is proposed by [16]. The method introduces a convolutional neural network that enhances a target image by using priors extracted from its RGB counterpart.

978-1-5386-4451-5/18/$31.00 ©2018 IEEE

For sparser depths, upsampling methods also rely on RGB images, using them as guides for better scene understanding. For instance, the method presented in [17] relies on the cosparse analysis models and makes use of analytic operators, requiring no training data.

### B. Improving depth perception by semantic segmentation

Several methods that rely on image segmentation to improve depth accuracy have been developed during the last years [18] [19]. Stereo reconstruction has been combined with segmentation to increase disparity accuracy either in the post-processing step [20], [19] or by using semantic info as a guide for better matching [21].

In the context of high resolution depth image generation, LiDAR sensors also benefit from semantic information [22]. In their paper [6] the authors use the semantic map as a guide that provides both local context and edge information. Our method resembles this approach, but instead of formulating the upsampling problem as a global energy minimization, we choose to directly improve resolution by deep learning.

## III. Partial-Dense to Dense

### A. Problem formulation

Let $D_1 \in R_{h \times w}$ be a depth map having $n = h \times w$ entries and assume that $m \ll n$ pixels are known. $D1$ is obtained after the transformation from the real to the camera coordinate system and then to the image plane using the calibration matrix. Therefore, it is aligned with $I$ - RGB and $S$ - semantic images. The main goal is to compute a higher resolution map $D_2$, in which $m'$ pixels are known, where $m' \approx n$.

In contrast to methods that rely on particular stereo camera/LiDAR set-ups, the $m$ depth values can be computed by any method. However, depending on the relation between $m$ and $n$, we can separate the upsampling problem in two:

- from sparse unstructured data to dense
- from sparse structured data to dense

In the case of upsampling data from partial-dense to dense case, $m > 1/2 \times n$, so more than half of the pixels are known from sensor measurements. This is quite important, because with such a resolution we can rely on depth image structure.

We need a solution of estimating the $k = m' - m$ of the unsampled positions in $D_1$. One of the major difficulties for this estimation is to compute the correct depth around boundaries regions (edges). For this purpose, besides intensity values we rely on semantic information to provide relevant boundary information.

### B. Semantic segmentation of the image

In order to acquire relevant object infromation, a first step in our solution is to compute a semantic segmentation of the scene. Classic segmentation methods have been combined with depth image correction [19], generally being used as a post-processing [18]. However, we can now take advantage from the boost that semantic segmentation lately received with the introduction of deep neural networks. Cityscapes dataset [23] enables methods such as [24], [25] or [26] to accurately classify object categories at pixel level.

One of the most robust approaches in semantic segmentation is Erf-NET [27], having one of the best trade-offs in terms of accuracy (69.7 for IoU) vs speed (around 25 ms). The method uses an encoder-decoder architecture, with 23 layer blocks. The key for speed and precision is their novel layer block – a mixture of residual connections and factorized convolutions that preserves the structure in the image and reduces computational costs. The method classifies the scene into 19 foreground and background classes.

### C. Dataset generation

Since we try to optimize the problem by means of machine learning, a reliable dataset is extremely important. The main prerequisites for the training set are:

1) RGB images for semantic segmentation with driving scenarios
2) sparse depth image acquired either from stereo (need left and right images) or from LiDAR
3) dense accurate depth ground truth

Although Kitti 2015 stereo dataset [28] is adequate for the first two needs, the GT it provides is sparse (given by LIDAR). DispNetC [4] provides the dense GT we need, but its synthetic nature reduces the inherent difficulties found in real situations (eg. unexpected illumination, scene complexity). A different option is to choose the disparity obtained with a top stereo method (MC-CNN acrt [1]) as our dense ground truth. While this method has a low error rate on the evaluated Kitti pixels, it is adapted to real-life situations. We choose a mixture of these two ($60\%$ of images from Kitti and $40\%$ images from DispNetC), benefiting from both pixel-wise accurate ground truths and real-life driving scenarios. We further extract patches at various positions and randomly generate around 200.000 patches, with RGB, semantic information, sparse depth and dense depth.

### D. CNN Architecture

For the learning-based upsampling we employ a 3-input ConvNet architecture (Fig. 1) that follows the principles of the guided filter proposed by [16].

The first part of the network consists in two similar branches, with the role of extracting reliable features from both depth and RGB image. A 41x41 patch from the Left RGB image is the input to the first branch, while a patch from the incomplete depth map is plugged in to the second one. Each branch consists of three residual Non-bottleneck1D blocks, followed by a Batch Normalization layer. The first block contains 64 feature maps, the second 128, while the third produces just one feature map, that incorporates the most relevant features extracted from each branch.

The convolution layers are designed using the speed-up techniques presented in [27]. A Non-Bottleneck1D (Fig. 2) block is therefore shaped by:

- Residual connections – important information extracted from initial layers is preserved throughout the entire network so that later layers can benefit from it;

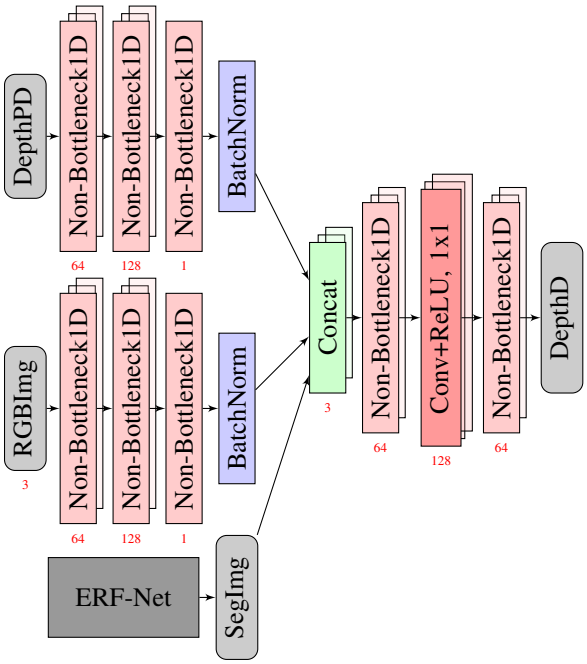



Fig. 1: Architecture of the Proposed Partial-Dense to Dense Upsampling

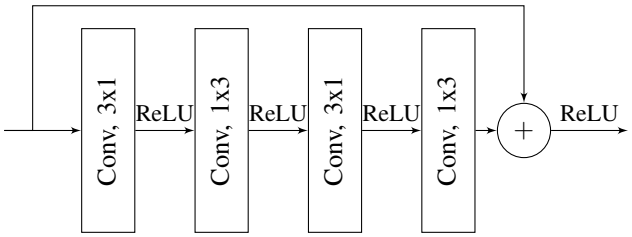



Fig. 2: Non-bottleneck1D block having a receptive field size of 5x5

- 2D convolution layers approximated by two 1D convolutions – this trick reduces the number of convolution weights by more than a half while preserving stability and accuracy;
- ReLU units inserted after each convolution – used to zero the gradients on negative input values.

Exhaustive testing showed us that RGB features are not enough to provide effective guidance. This problem is mainly caused by the mixture and the variety of information RGB maps carry. Although the first part of the network tends to extract more effective features and provides relevant information, we consider useful to aid this process with an additional term. Therefore results of the two branches are also concatenated with a patch extracted from the segmentation image. The semantic segmentation map is the third set of important features for our network, containing information about object boundaries and linking together similar structures.

The second part of the network consists in two additional Non-Bottleneck-1D blocks, interleaved by a layer of 1x1 convolutions. The role of the second part is to join together the three maps and generate a more reliable depth image, simulating a non-linear regression. This way we will integrate the knowledge extracted from the three aforementioned feature maps. Numerically, a pixel-wise mean squared error is computed between the resulting upsampled depth patch and the ground truth, thus estimating the degree of convergence for our method.

### *E. Parameters and Training Details*

All patches are normalized by subtracting the mean and dividing with the maximum image intensity. Similar learning rates have been given to both depth and image branches. Experimental testing showed that our network converged only when the segmentation learning rate was set to 1/5 of the learning rate for RGB and Depth. In other scenarios segmentation features became too powerful, and depth information was dropped. We tried two optimization methods: Stochastic Gradient Descent and Adaptive Moment Estimation (Adam). Adam seemed to properly control the learning so we chose it as our optimizer.

We trained the network for 400 epochs, with a batch size of 128, decreasing the learning rate with a factor of 0.1 at the interval of 100 epochs.

## IV. Sparse to Semi-Dense

### *A. Dataset generation*

In the case of sparse to partial-dense upsampling, the dataset is generated from the Kitti [29] raw dataset. The set consists in RGB images, over which the semantic segmentation is applied. The set also contains 3D points obtained with a 64-ray Lidar. We project the 3D points to the image plane and further select only those points that fall inside our image frame to generate the depth ground truth (semi-dense). For each image we extract around 4000 points.

In order to generate sparse depth points we carefully extract points from ground truth image, downsampling with a rate of 4. Particularly, we want to emulate the results of a 16-ray LiDAR. After projecting an entire row of LiDAR points the downsampling method skips the following three rows. An example of input and ground truth image can be seen in Figure 6 b) and c).

To sum up, from each image we extract:

- a 61x61 sparse patch, simulating results given by a 16-ray LiDAR. Larger patches are required due to the scattered depth pixels
- a patch with RGB intensities;
- a patch with semantic information;
- a groung truth partial dense patch, given by a 64-ray LiDAR.

### *B. CNN Architecture*

The sparse to partial-dense CNN architecture also consists in several sub-networks joined together. Intensity and semantic sub-networks will take the guiding images and extract two feature maps. The main difference between this architecture and the aforementioned one consists in the way the target image is processed: instead of extracting only one feature

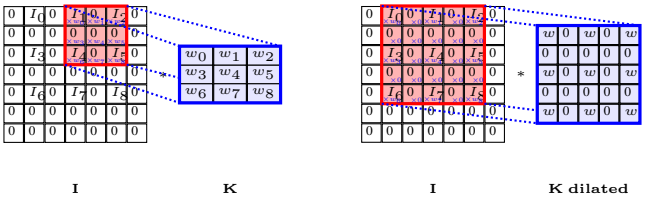



Fig. 3: Dilated convolutions in context of upsampling; Left - Regular convolution over sparse data; Right - Dilated convolution over sparse data

map, we apply multiple convolutions, with various dilation coefficients.

Figure 3 left presents a regular convolution, applied over sparse data (with a sparsity rate of $1/4$). Even with such a high rate (for sparse images obtained form a 16-ray LiDAR this rate can get to $1/100$) it can be easily seen that more than half of the weights are not required, covering only a small part of the image. Larger convolutions might lead to better image covering, but they will still suffer from the same problem of memory waste.

Dilated convolutions have been initially proposed by [30] in the context of semantic segmentation and proved to work as ideal features extractors when various scales are required. In our case (Figure 3 right), dilated convolution are enlarging the receptive field of our convolutions, without increasing the memory or computational load. Moreover, by mixing semantic guidance with dilated convolutions, the shortcomings of convolutions with large receptive fields around object boundaries will be reduced.

Our architecture consists in three depth-related sub-networks. Each sub-network applies dilated convolutions, increasing the receptive field at each step. All convolution layers follow the same construction rules as Non-bottleneck1D presented above, with the major difference of changing the second pair of $3\times1$ and $1\times3$ convolutions for a pair of dilated 1D convolutions. Residual connections presented in each Non-bottleneck1D ensures that the important information extracted from initial layers is preserved throughout the entire network.

Finally, all resulting feature maps are joint together and passed to the non-linear regression sub-network for optimization. The CNN architecture can be seen in Figure 4.

## V. Evaluation

### A. Semi-dense to dense

The partial dense depth input consists in the ground truth from Kitti2015 stereo method [28]. The input data consists in LiDAR points projected from 11 consecutive frames. To determine the percentage of erroneous depth points, we employ a threshold of three pixels and we compute the number of misclassified pixels with respect to the aforementioned ground truth.

*1) Accuracy of Depth upsampling methods:* For this part we use the same depth generation method, and see how our method behaves with respect to other approaches. We

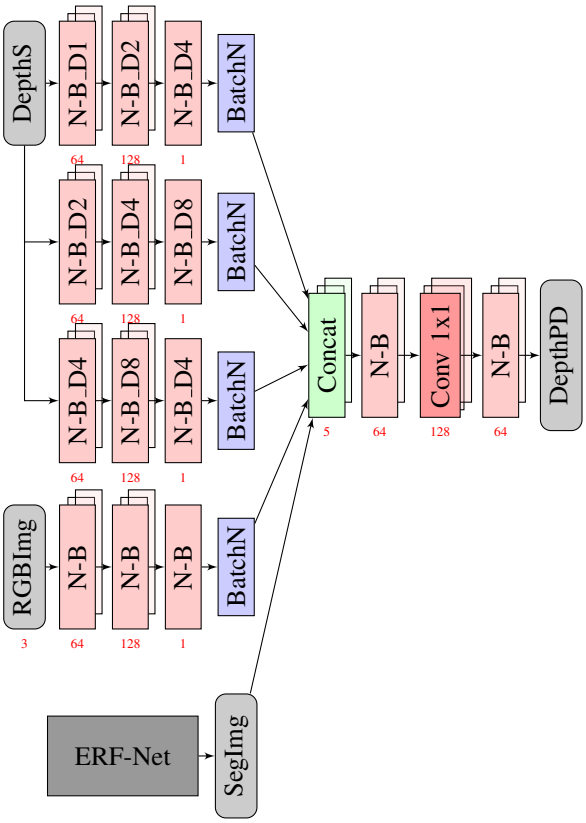



Fig. 4: CNN Architecture for Sparse to Partial-Dense Upsampling

TABLE I: Performance of various upsampling techniques for Partial-Dense to Dense case

| Method | Error | Speed | Platform |
|---|---|---|---|
| Without | 10.19% | - | - |
| Median filter [12] | 9.37% | 0.3 ms | GPU (CUDA) |
| Bilateral filter [13] | 9.22% | 8 ms | CPU (C++) |
| Guided filter [15] | 9.07% | 20 ms | CPU (C++) |
| Fast bilateral [14] | 9.17% | 180 ms | CPU (C++) |
| DeepJoint filter [16] | 7.72% | 100 ms | GPU (Matlab) |
| **Proposed** | **5.25%** | **11 ms** | **GPU (CUDA)** |

chose some of the most commonly used upsampling filters: median [12], bilateral [13], guided [15], fast bilateral [14] and DeepJoint [16]. We implemented our own median and fast bilateral filters and used the OpenCV implementations for the bilateral and guided filters. DeepJoint filter has been trained using images from Kitti2015 dataset. Our method is shown to outperform its counterparts by a large margin while maintaining a relatively low time consumption. This evaluation also reveals the improvement we obtained (around 2.5% wrt the joint filter) by introducing the segmentation map into the CNN. Table I shows numerical results. Visual results can be seen in Figure 5 c) - d) for stereo and e)-f) for LiDAR.

*2) Accuracy obtained with our upsampling method when changing the input depth estimation:* We are interested to see how our upsampling method behaves when the underlying depth estimation method is changed. We chose the

TABLE II: Accuracy of the upsampling when applied to various depth construction methods

| Method | Error | Error(+Upsampling) |
|---|---|---|
| Stereo: Census | 57.27% | 69.27% |
| Stereo: BM | 30.89% | 22.86% |
| Stereo: SGM | 10.19% | 5.25% |
| Stereo: MC-CNN fast | 3.79% | 3.23% |
| 64-ray merged LiDAR | 21.75% | 10.65% |

TABLE III: Performance of upsampling techniques in Sparse to Partial-Dense case

| Method | Error | Speed | Platform |
|---|---|---|---|
| Without | 75.14% | - | - |
| DeepJoint [16] | 73.42% | 100 ms | GPU (Matlab) |
| Proposed, without dilation | 77.72% | 11 ms | GPU (CUDA) |
| Proposed, without semantics | 41.15% | 14 ms | GPU (CUDA) |
| Proposed, with dil+sem | 18.84% | 17 ms | GPU (CUDA) |

following depth generation methods as input to upsampling: Stereo matching using Census, Stereo Block Matching (BM) and Semi-Global Block Matching (SGBM) from OpenCV, feature-based streo matching using convolutional neural networks – MC-CNN fast from [1] and 64-ray LiDAR – the stereo GT from Kitti. For each of the stereo methods we performed left-right consistency check, removing inconsistent pixels to generate incomplete disparity maps. Due to the locality of LiDAR points, we only tested the lower part of the images. The densification network was trained only once, with patches from a subset of Kitti and DispNetC images.

Since the goal for our first network is to deal only with structured data, it can not cope with the large errors found in Census-only case. Stereo solutions with an average error rate can really benefit from our upsampling method, the error largely decreasing for LiDAR and for Semi-Global Matching [31]. On the other hand, stereo solutions with low error and high resolution can only marginally gain from our solution, since they use other approaches to deal with this problem. Numerical results are shown in Table II.

### *B. Sparse to partial-dense*

Since the methods we used for comparison in partial-dense to dense case will not work (too few information), and (to the best of our knowledge) there are no benchmarks for the upsampling, we could only compare our architecture against the results obtained with DeepJoint filter [16], against our architecture without introducing any dilation in layers and against our architecture without semantic information. This experiment shows the importance of dilated convolutions – upsampling methods without such layers behave poorly (this can be seen even at training phase). For such a complex problem as upsampling in sparse setting we show that out proposed upsampling method is the best solution among the tested ones, semantic information giving additional accuracy. Figure 6 shows our densified depth image, together with the given sparse data and ground truth. Numerical results can be seen in Table III.

## VI. CONCLUSIONS

Learning methods such as ConvNets are becoming more and more popular in the depth measurement domain. We present here two ConvNets for upsampling low resolution depth images provided by either stereo or LiDAR sensors. Both ConvNets use intensity and semantic information as guidance. The initial convolutional architecture receives a structured depth map and filters it to complete the measuring information. Since depth information lacks structure in the sparse case, we employ a new architecture for the sparse to partial-dense case. The network uses multiple dilated convolutions as basic means to cope with the unstructured data. We performed multiple tests on different types of data with best positive results for the proposed approach.

We intend to continue our work by developing other real-time upsampling convolutional architectures that are capable to directly infer the complete dense map from sparse data.

## ACKNOWLEDGEMENT

This work was supported by UEFISCDI (Romanian National Authority for Scientific Research and Innovation) in the national research project Multispectral Environment Perception by Fusion of 2D and 3D Sensorial Data from the Visible and Infrared Spectrum (MULTISPECT), project code PN-III-P4-ID-PCE2016-0727, contract number 60/2017.

## REFERENCES

[1] J. Zbontar and Y. LeCun, "Computing the stereo matching cost with a convolutional neural network," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 1592–1599.

[2] A. Seki and M. Pollefeys, "SGM-Nets: Semi-Global Matching With Neural Networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[3] J. Pang, W. Sun, J. S. J. Ren, C. Yang, and Q. Yan, "Cascade residual learning: A two-stage convolutional neural network for stereo matching," *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 878–886, 2017.

[4] 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," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4040–4048, 2016.

[5] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, "End-to-End Learning of Geometry and Context for Deep Stereo Regression," *CoRR*, vol. abs/1703.04309, 2017. [Online]. Available: http://arxiv.org/abs/1703.04309

[6] N. Schneider, L. Schneider, P. Pinggera, U. Franke, M. Pollefeys, and C. Stiller, "Semantically guided depth upsampling," in *GCPR*, 2016.

[7] J. Dolson, J. Baek, C. Plagemann, and S. Thrun, "Upsampling range data in dynamic environments," *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1141–1148, 2010.

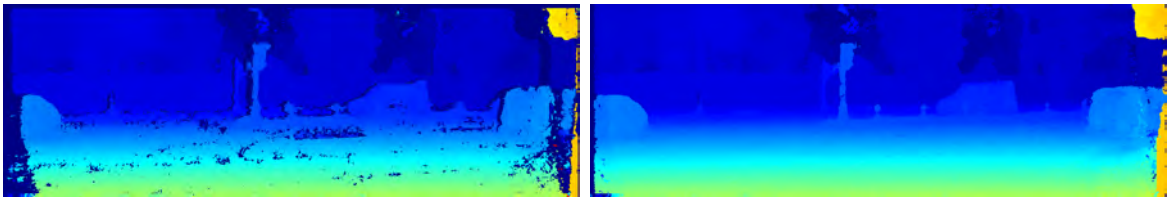
[8] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2024–2039, Oct. 2016. [Online]. Available: https://doi.org/10.1109/TPAMI.2015.2505283

[9] X. Huang, L. Fan, J. Zhang, Q. Wu, and C. Yuan, "Real time complete dense depth reconstruction for a monocular camera," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2016.

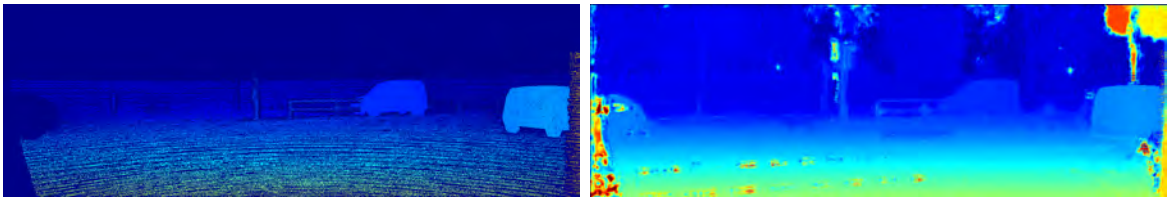
[10] W. E. L. Grimson, "Computational experiments with a feature based stereo algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-7, no. 1, pp. 17–34, Jan 1985.

[11] S. Hawe, M. Kleinsteuber, and K. Diepold, "Dense disparity maps from sparse disparity measurements," in *2011 International Conference on Computer Vision*, Nov 2011, pp. 2126–2133.

(a) Left Image (b) Segmentation Image

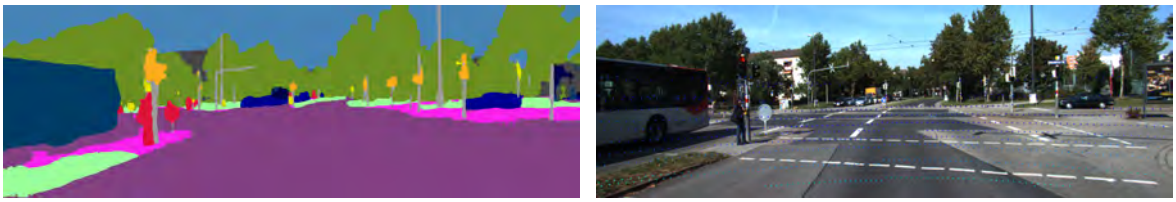


(c) Incomplete Depth Image – from Stereo (d) Upsampled Depth Image – from Stereo



(e) Incomplete Depth Image – from LiDAR (f) Upsampled Depth Image – from LiDAR

Fig. 5: Depth maps obtained with our partial-dense to dense method on traffic images



(a) Segmentation image (b) Sparse image (16-ray LiDAR)

(c) Ground truth image (64-ray LiDAR) (d) Densified image (Partial Dense)

Fig. 6: Depth maps obtained with our sparse to partial-dense architecture on traffic images


[12] T. Huang, G. Yang, and G. Tang, "A fast two-dimensional median filtering algorithm," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 27, pp. 13–18, 1979.

[13] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," in *Proceedings of the Sixth International Conference on Computer Vision*, ser. ICCV '98. Washington, DC, USA: IEEE Computer Society, 1998, pp. 839–. [Online]. Available: http://dl.acm.org/citation.cfm?id=938978.939190

[14] S. Paris and F. Durand, *A Fast Approximation of the Bilateral Filter Using a Signal Processing Approach*, 2006.

[15] K. He, J. Sun, and X. Tang, "Guided Image Filtering," in *Proceedings of the 11th European Conference on Computer Vision: Part I*, ser. ECCV'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 1–14. [Online]. Available: http://dl.acm.org/citation.cfm?id=1886063.1886065

[16] Y. Li, J.-B. Huang, N. Ahuja, and M.-H. Yang, *Deep Joint Image Filtering*, 2016.

[17] X. Gong, J. Ren, B. Lai, C. Yan, and H. Qian, "Guided depth upsampling via a cosparse analysis model," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2014, pp. 738–745.

[18] M. Humenberger, T. Engelke, and W. Kubinger, "A census-based stereo vision algorithm using modified Semi-Global Matching and plane fitting to improve matching quality," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, June 2010, pp. 77–84.

[19] K. Yamaguchi, D. McAllester, and R. Urtasun, "Efficient joint segmentation, occlusion labeling, stereo and flow estimation," in *European Conference on Computer Vision*. Springer, 2014, pp. 756–771.

[20] M. Humenberger, T. Engelke, and W. Kubinger, "A census-based stereo vision algorithm using modified semi-global matching and plane fitting to improve matching quality," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, June 2010, pp. 77–84.

[21] V.-C. Miclea and S. Nedevschi, "Semantic segmentation-based stereo reconstruction with statistically improved long range accuracy," in

*Intelligent Vehicles Symposium Proceedings, 2017 IEEE*, 06 2017, pp. 1795–1802.

[22] M. Liu, M. Salzmann, and X. He, "Semantic-aware depth super-resolution in outdoor scenes," *CoRR*, vol. abs/1605.09546, 2016. [Online]. Available: http://arxiv.org/abs/1605.09546

[23] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[24] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 3431–3440.

[25] G. Ghiasi and C. C. Fowlkes, "Laplacian reconstruction and refinement for semantic segmentation," *CoRR*, vol. abs/1605.02264, 2016. [Online]. Available: http://arxiv.org/abs/1605.02264

[26] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *CoRR*, vol. abs/1606.02147, 2016. [Online]. Available: http://arxiv.org/abs/1606.02147

[27] E. Romera, J. M. Alvarez, L. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation," vol. PP, pp. 1–10, 10 2017.

[28] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[29] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[30] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *CoRR*, vol. abs/1511.07122, 2015. [Online]. Available: http://arxiv.org/abs/1511.07122

[31] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 2, pp. 328–341, Feb 2008.