Semi-Dense Depth Interpolation using Deep Convolutional Neural Networks

Ilya Makarov
National Research University Higher School of Economics
School of Data Analysis and Artificial Intelligence
Moscow, Russia
iamakarov@hse.ru

Vladimir Aliev
National Research University Higher School of Economics
School of Data Analysis and Artificial Intelligence
Moscow, Russia
twoleggedeye@yandex.ru

Olga Gerasimova
National Research University Higher School of Economics
School of Data Analysis and Artificial Intelligence
Moscow, Russia
olga.g3993@gmail.com

ABSTRACT

With advances of recent technologies, augmented reality systems and autonomous vehicles gained a lot of interest from academics and industry. Both these areas rely on scene geometry understanding, which usually requires depth map estimation. However, in case of systems with limited computational resources, such as smartphones or autonomous robots, high resolution dense depth map estimation may be challenging. In this paper, we study the problem of semi-dense depth map interpolation along with low resolution depth map upsampling. We present an end-to-end learnable residual convolutional neural network architecture that achieves fast interpolation of semi-dense depth maps with different sparse depth distributions: uniform, sparse grid and along intensity image gradient. We also propose a loss function combining classical mean squared error with perceptual loss widely used in intensity image super-resolution and style transfer tasks. We show that with some modifications, this architecture can be used for depth map super-resolution. Finally, we evaluate our results on both synthetic and real data, and consider applications for autonomous vehicles and creating AR/MR video games.

CCS CONCEPTS

• **Human-centered computing** → **Mixed / augmented reality**; *Virtual reality*; Systems and tools for interaction design; • **Computing methodologies** → **Vision for robotics**; **Reconstruction**; **Neural networks**; *Shape analysis*; Image-based rendering; • **Software and its engineering** → **Virtual worlds software**; • **Information systems** → *Multimedia content creation*; • **Applied computing** → *Computer games*;

KEYWORDS

Computer Vision; Depth Map; Autonomous Vehicles; Augmented Reality; Mixed Reality; Convolutional Neural Networks

MM'17, , October 23–27, 2017, Mountain View, CA, USA.

ACM ISBN 978-1-4503-4906-2/17/10…$15.00
https://doi.org/10.1145/3123266.3123360

1 INTRODUCTION

Nowadays increasing computational power and advances in computer vision and decision making algorithms have greatly impacted such growing fields as augmented reality, robotics and self-driving cars, changing their status from emerging technologies to actively studied research areas.

In order to construct an augmented reality environments one needs to estimate scene geometry for proper interaction of virtually projected objects with the real world. In case of autonomous vehicles or mobile robots, estimating geometry is essential to make them safe from collisions and serves advanced scene understanding.

Although there exists special 3D-sensing hardware, such as time of flight or structured lights sensors, it is not always possible to use them. Such depth sensors can suffer from various environment conditions like background illumination [11], may have high cost or large size. As another example, augmented reality applications are usually designed for modern smartphones with only standard RGB cameras. Various environmental conditions, such as a lack of illumination or glares, can also degrade performance of 3D sensor-based hardware.

Whenever special 3D sensors can not be applied, scene depth can be estimated using stereo setup or monocular video. However, for mobile systems, such as smartphones, real-time, high-quality dense depth estimation using monocular video can be challenging task due to the constrained computational resources. For this case one can try to estimate semi-dense or low resolution depth map with its consequent interpolation. Semi-dense depth maps arise even in modern direct visual localization and mapping or odometry methods. Such methods as LSD-SLAM [8], REBVO [37], DSO [7] estimate semi-dense, edge-based, or even sparse depth maps to find the camera pose. In particular, DSO method measures depth maps filtering points with low confidence that leads to sparse depth mostly along the intensity gradient. To sum up, all these methods estimate partial depth maps, where depth values are distributed along intensity image gradient with high estimation confidence.

Any real application involving 3D-reconstruction requires to know the camera pose, which is exactly one of the problems solved by SLAM methods. Using semi-dense SLAM methods with an effective interpolation methods can ease 3D reconstruction by providing dense depth maps from semi-dense ones.

In this paper, we present a convolutional neural network that can be trained to interpolate semi-dense depth maps or low-resolution depth, supporting different spatial distributions of the input depth map. The network architecture induces almost no constraints on

the depth estimation algorithm. Finally, we show that with small architecture changes one can train the network to perform dense depth super-resolution.

In our method, along with MSE loss, which is traditional for regression tasks, we also apply the perceptual loss function. The perceptual loss is one of the ingredients of style transfer methods [39]. We use it to get better interpolation of object shapes and edges, which are usually oversmoothed by the MSE loss.

To train our networks we use SYNTHIA [31] dataset. This dataset provides a large amount of synthetic outdoor road scenes data. It includes full semantic labeling, stereo RGB images, and high-resolution depth maps. The dataset can easily be used in tasks related to road scene understanding by autonomous vehicles. To evaluate generalization power of our neural network we use two additional datasets: Sintel[3] and NYU Depth[34]. For the super-resolution evaluation we use Middlebury [32] dataset, which is the common benchmark for depth super-resolution methods. Finally, we consider the work-in-progress applications of our method to AR/MR video games and road accidents detection.

2 RELATED WORK

Depth interpolation is actively studied in the literature, but most of the works focus on low resolution depth upsampling methods, which are usually divided into two groups: guided upsampling and single image depth upsampling.

Guided upsampling methods use high-resolution intensity image, to provide depth clues. In [19], the authors applied joint bilateral filtering method proposed in [38]. Another local filtering method was proposed in [23] taking into account geodesic distances to pixels with known depth for computing depth for the high resolution image. In [25], Lu et al. developed a method to reconstruct depth structures inside each cluster of the segmented intensity image.

Another group of guided methods used the optimisation-based approaches. In [5], Deibel et. al. proposed MRF-based method with the penalized smoothness term along the edges of intensity image. Park et. al used regularization on non-local means in order to preserve thin details [29]. Anisotropic diffusion tensor was proposed in [9] for regularizing depth upsampling.

Single image depth upsampling methods do not need the aligned image for depth upsamling. For example, a dictionary-based method was proposed in [42], where the low resolution and the corresponding high resolution depth patches were used for upsampling low-resolution depth. In [1], the authors formulated similar task in terms of the MRF labeling problem. In [17], a guided residual interpolation was adapted for the single image upsampling problem with the help of depth interpolation by displacement fields upsampling method [40] as a guidance.

Apart from traditional methods, there also exist depth interpolation methods based on convolutional neural networks. Song et al. [36] suggested a model of convolutional neural network that was trained to produce high resolution depth using bicubic upsampled input with the following refinement. In [14], the authors proposed networks for both guided and unguided upsampling. They used early spectral decomposition and trained their networks using only high frequency part of low resolution depth and intensity images. The authors applied guidance through multiscale feature extraction from high-frequency part of intensity image.

Convolutional neural networks are also widely used in intensity image super-resolution. Dong et al. [6] presented the end-to-end learnable deep convolutional neural network for super-resolution. Shi et al. [33] made use of subpixel convolutional layers also known as pixelshuffle layers, to train a subpixel convolutional neural network for image and video super-resolution. Finally, in [21], the authors used Generative Adversarial Network to improve perceptual quality of interpolated images.

The latter set of methods is usually used to reconstruct depth from sparse set of samples. These methods are close to our model in the sense of using different spatial configurations of input depth samples, not limiting only to the dense low-resolution depth case. In [12], Hawe et. al. formulated an optimization problem based on the theory of Compressed Sensing, which allows to recover disparity map using only 5% of known disparity values. In [22], Liu et. al used ADMM method [28] to solve the reconstruction problem instead of optimization method from [12]. They also introduced optimal sampling scheme to improve depth reconstruction quality.

In contrast with previous techniques, our method considers neural network training problem instead of explicitly stating depth reconstruction optimization problem. Once trained, depth reconstruction is achieved by forward pass through the network. However, our method needs 15-20% of valid depth values for reconstruction, which is higher compared to 2-5% in [12], [22].

3 OUR METHOD

3.1 Interpolation Problem and Network Architecture

We formulate our task as a regression problem: for the input semi-dense map $\mathbf{x} \in R^{m \times n}$ find its interpolation $\hat{\mathbf{y}} \in R^{m \times n}$ that minimizes the loss function $L(\hat{\mathbf{y}}, \mathbf{y})$, where $\mathbf{y} \in R^{m \times n}$ is the original depth map. Since we suppose that known depth values of the input map represent true depth values, it is natural to implement a residual reconstruction mapping:

$$\hat{\mathbf{y}} = \mathbf{x} + \mathcal{K}(\mathbf{x}),$$

where $K(x)$ is the output of the proposed network. For the super-resolution task we use bicubic upsampling to get $\mathbf{x}_h$ and then sum it with network predictions. Similar idea was used in [14], where the authors used upsampled low-frequency part of depth map and sum it with high-frequency predictions:

$$\hat{\mathbf{y}} = \mathbf{x}_h + \mathcal{K}(\mathbf{x}).$$

3.2 Neural Network Overview

Our architecture for semi-dense depth map interpolation is a fully-convolutional encoder-decoder based architecture. Such architectures are common for problems where network output should be the same size as the network input, such as, for example, semantic segmentation [24], [30].

The encoder parts serves as information extractor, gradually decreasing the size of the input map. It reduces spatial redundancy of semi-dense depth map, in which only a small number of values

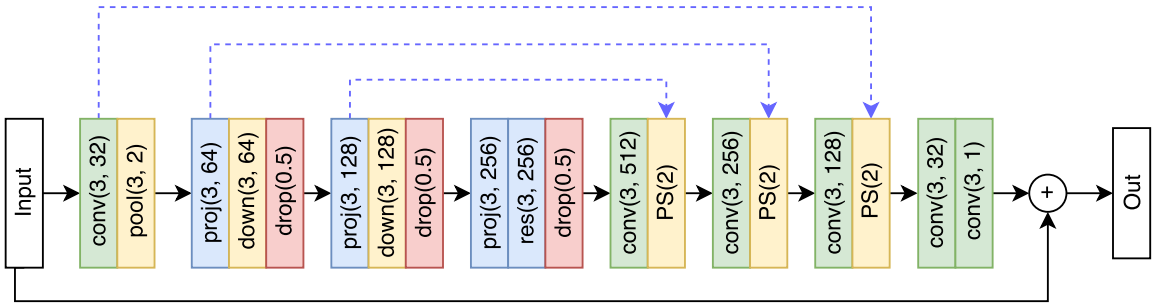

Figure 1: Network architecture for semi-dense depth maps interpolation. The sizes of the filter and the output features are given in parenthesis for convolutional, projection, downscale, and residual blocks. For pixelshuffle and dropout, the upscale factor and the probability are given, respectively. Dashed lines denote concatenation. The average pooling with window 3 and stride 2 was considered.

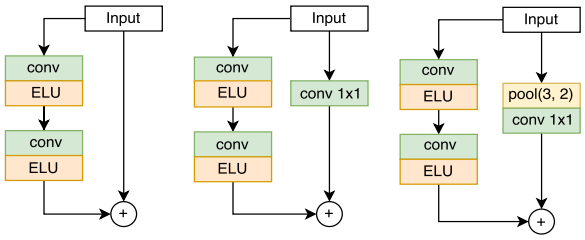

Figure 2: Main blocks layout. Left to right: standard residual block, projection block, and downscaling block.

are valid. The encoder part performs upsampling of the reduced input, based on the extracted low-resolution feature maps.

The decoder and encoder parts share information by feature maps concatenation. This type of connections was introduced in [30] and now is common for semantic segmentation architectures [15]. The purpose of feature map concatenation is to pass high-frequency information from the decoder to the encoder part, since this information can be lost during spatial downsampling. The feature maps concatenation improves detailing of the produced depth map.

For all activation functions in our network we use the exponential linear unit [4]. This activation also allows negative values to pass, which shifts mean activation to zeros. The ELU activation without batch normalization reduces the training time of the network and leads to a slightly better interpolation quality in our case.

3.3 Encoder Architecture

In the encoder part, we use several types of core building blocks, each one including residual connections. Residual blocks were introduced in [13]. These blocks include additional identity data path along with convolutions. It was shown in [13] that such types of blocks allow training very deep networks and achieve the state-of-the-art results on classification tasks. In our case, using residual blocks also gives a considerable boost in depth maps quality. When amount of feature maps is doubled, we add a 1×1 linear convolution layer in the identity path to match the output feature dimension. Downscaling blocks use strided convolutions in the convolution path and average pooling followed by 1×1 linear convolution layer in the identity path. We found that average pooling performs much better in our task than max pooling, since we extract the information from the sparse input. For regularization we use dropout with $p = 0.5$ after each downsampling stage.

3.4 Decoder Architecture

The decoder architecture, in opposite, does not contain residual blocks. The common practice is to use deconvolutional layers for upsampling. Recently, [33] has introduced subpixel convolution layers (also called pixelshuffle layers) that are computationally more effective and lead to better super-resolution. The deconvolutional layers upsample the feature map and then apply convolution. The subpixel layer arranges pixels from several feature maps in interleaving manner to match the output spatial size. In our decoder, we choose subpixel convolution layer providing better reconstruction. So, the decoder blocks are just plain convolutional layers followed by pixelshuffle rearrangement.

3.5 Architecture for Super-resolution

With slight modifications, our architecture can be used for dense depth-map upsampling. We just replace the encoder by several residual blocks. Here, we do not need the contracting part, as the input is already dense and we do not need to eliminate spatial redundancy. The encoder part remains the same. Similar network layouts are used in [21] and [14].

3.6 Loss Function

The depth and intensity image super-resolution problem generally considered low-resolution depth upsampling as a regression problem and used MSE error [14], [6]. We define a loss function consisting of two components:

$$L = L_{square} + \alpha L_{VGG}$$

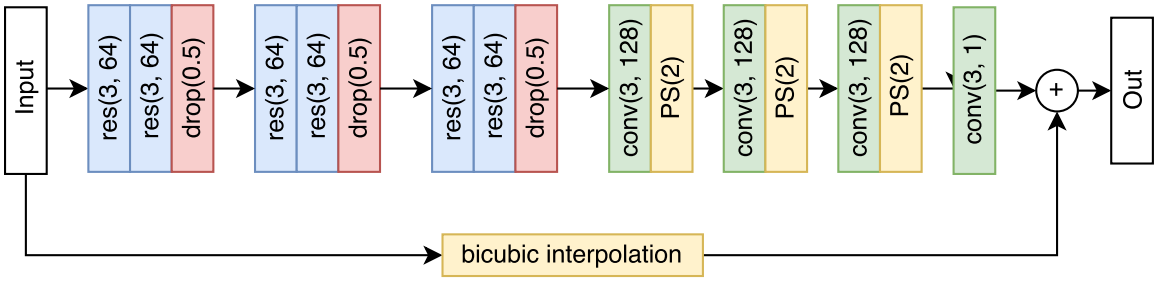

Figure 3: Network architecture for super-resolution, all notations are the same, as on figure 1. This architecture was inspired by [21].

The first component is the traditional MSE loss, which ensures that the interpolated depth map will be consistent with the ground truth:

$$L_{square} = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H}(y_{i,j} - \hat{y}_{i,j})^2$$

MSE loss function tends to give oversmoothed results and often has poor perceptual quality. Recently, many works proposed using Euclidean distance in the feature space of some pretrained neural network in the application to image inpainting [43], super-resolution[16], [21], and style transfer[39], [16]. Computing loss in the feature space of VGG-16/19 [35] networks leads to better performance in all tasks mentioned above.

We also make use of this perceptual loss, since it allows to transfer semantic knowledge from the loss network and achieve better interpolation of the object shapes and higher depth map detailing. This loss is written as:

$$L_{VGG} = \frac{1}{WHC}\sum_{i=1}^{W}\sum_{j=1}^{H}\sum_{k=1}^{C}(\Phi(\mathbf{y})_{i,j,k} - \Phi(\hat{\mathbf{y}})_{i,j,k})^2,$$

where $\Phi(\mathbf{y})$ is the output of the convolution layer of the loss network.

Following the previous works on super resolution and image style perception we use VGG-16 or VGG-19 networks to compute perceptual loss. In our work, we use the smaller VGG-16 network, as we found almost no difference comparing training results. VGG-16 consists of several groups of convolution layers separated by max pooling layers that perform spatial downsampling. We use the output of the first convolutional layer of the third group (conv3.1 layer). We do not use higher levels since they extract mostly low-level information such as edges. Deeper levels do not fit also, as they induce artifacts on the reconstructed depth map.

One can notice that in [21] or [16] the perceptual loss is used standalone. However, we found that in the depth interpolation problem, a network trained with perceptual loss only catches the geometry but fails to produce correct depth values. It leads to high errors in the sense of MAPE and RMSE metrics. We can explain it by the fact, that VGG networks are image classifier networks, and depth map input values distribution differs with intensity image brightness values.

The loss has a hyperparameter α. This hyperparameter should be tuned in log-scale to achieve a trade-off between depth error and perceptual quality of resulting depth maps. In our case we used $\alpha = 5 \times 10^{-5}$. Such low value is explained by the fact that in our case the perceptual loss value has order of 10^4 while MSE loss does not exceed 10.

4 EXPERIMENTS

4.1 Description of Datasets

SYNTHIA [31]. SYNTHIA is a synthetic dataset of road scenes, originally collected for semantic segmentation tasks. This dataset consists of large number of high-resolution intensity and depth images from the town landscape from the road view. SYNTHIA also provides dense semantic labeling, although, we do not use semantic segmentation at the current stage of the research. We use this dataset as our main training dataset, since it contains large number of high-quality depth maps. For training we used '02' and '06' sequences of this dataset. Other sequences were used for test.

Sintel [3]. Sintel dataset also provides high quality images and depth map from 3D rendered scenes. This dataset consists of very diverse set of environments that are quite different from SYNTHIA data. We use Sintel dataset for semi-dense depth interpolation evaluation.

Middlebury 2005 [32]. Middlebury dataset is a set of image pairs with ground truth disparity. We use this dataset to evaluate our depth super-resolution network performance. We chose this particular version among the others in Middlebury dataset series to compare our results with traditional single image depth upsampling methods considered in [17]

NYU Depth [34]. NYU Depth is a real indoor dataset, collected by means of Kinect depth sensor. This dataset provides preprocessed data with inpainted depth maps, synchronised with intensity images and dense semantic labeling. We use this dataset only for evaluation of our network ability to generalize to unknown data.

4.2 Data Preprocessing and Training Details

Neural Network was implemented using TensorFlow framework. We trained it on 8000 depth images from scratch, using only '02' and '06' sequences from SYNTHIA dataset. No additional fine-tuning on other datasets was performed. To reduce training time and memory consumption we resized all the depth images to 152×256 resolution for a semi-dense depth map interpolation. For depth

	MAPE, %			RMSE, meters		
Sampling type	SYNTHIA 01	NYU Depth	Sintel	SYNTHIA 01	NYU Depth	Sintel
Uniform	4.9	5.8	28.2	52.1	0.54	4.66
Regular	24.6	23.7	48.1	71.7	0.69	5.20
Along gradient	8.7	11.5	36.3	54.9	0.62	4.38
Gradient + uniform	**3.8**	**4.8**	**24.5**	**46.1**	**0.35**	**4.00**

Table 1: Semi-dense depth interpolation results for different types of input distributions.

	SYNTHIA		NYU DEPTH	
	RMSE	SSIM	RMSE	SSIM
MSE +αVGG, $\alpha = 1e-2$	1246	0.93	1.01	0.85
MSE +αVGG, $\alpha = 5e-5$	46.1	**0.96**	0.36	**0.89**
MSE only	46.9	0.94	**0.35**	0.87
MSE+αVGG + βTV, $\alpha = 5e-5, \beta = 1e-5$	**44.3**	0.92	0.45	0.86

Table 2: Comparison of different losses and hyperparameters. TV stands for Total Variation regularization.

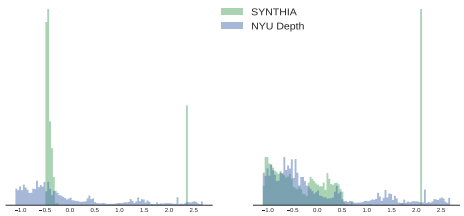

Figure 4: Histograms of depth values of two typical samples from SYNTHIA and NYU Depth datasets. Left to right: without logarithm transform, logarithm transform applied to SYNTHIA sample.

super-resolution tasks we used 38×64 depth maps as the input upsamling them to the 304×512 resolution, which is equivalent to $\times 8$ upscaling factor. Training was performed for 56k iterations using Adam optimizer with initial learning rate 10^{-4}, with batch size of 5. After 48k iterations of training learning rate was decreased to 10^{-5}. We used dropout regularization with $p = 0.5$.

To provide input data for the network we generated masks with desired non-zero values distributions and applied them to ground truth depth. Gradient masks were generated using adaptive thresholding of intensity image gradient; for combined masks we used clipped sum of gradient and uniform masks. We tuned the parameters of thresholding to achieve 15-25% density of input masks.

Another important preprocessing step was taking the logarithm from the input depth map. We noticed that SYNTHIA dataset depth maps have rather low dynamic range, so the typical depth values distribution was significantly different from the other datasets (see Figure 4). This difference leads to significant overfitting while training on the SYNTHIA dataset. Interpolating the logarithm of SYNTHIA depth maps solved the problem.

Further, we removed non-zero pixels mean from the input semi-dense depth map. Finally, we used random horizontal and vertical flips augmentation during training.

5 RESULTS

5.1 Semi-dense depth interpolation

The results of evaluation of our method on the datasets are given in Table 1. We can see that the best input distribution was obtained by the sampling along the intensity gradient combined with the uniform sampling. The uniform sampling adds information about homogeneous regions while intensity gradient adds object shapes information. We can also see that on the Sintel dataset our method gives rather large error. This can be explained by the fact, that this dataset contains images with objects very close on foreground combined with high depth values of background. Such combination gives very large depth variance within the frame and network fails to leverage it. From the error maps in Figure 8 we can indeed see that on the Sintel and NYU Depth datasets the error is large in the far areas, or on the entire background, in case of the Sintel. We should also note that our method run time is 760ms on Intel Core I7 4790K CPU and 88ms on NVIDIA GTX960 GPU for 400×400 semi-dense depth map. To compare, a similar method for sparse depth interpolation proposed in [22] has run times from 10 to 20 seconds on CPU depending on the number of available depth values.

In the Table 2, we have provided the comparison of different loss function on the semi-dense depth map interpolation quality. We can see that proposed loss is the best in terms of SSIM metric. It shows that the proposed loss function reconstructs structural details better than the others. We have also evaluated our result versus the loss function with the total variation term with weight $\beta = 10^{-5}$ from [16]. In fact, smaller beta values had almost no impact on the resulting quality, so we removed this term from the further research.

5.2 Depth super-resolution

The results for depth super-resolution with upscale factor $\times 8$ are given in Table 3. To be consistent with [17] we also provided structural similarity score (SSIM) [41]. We can see that in the sense of SSIM metrics our networks outperforms the traditional methods. However, there are images with very high RMSE, which is similar

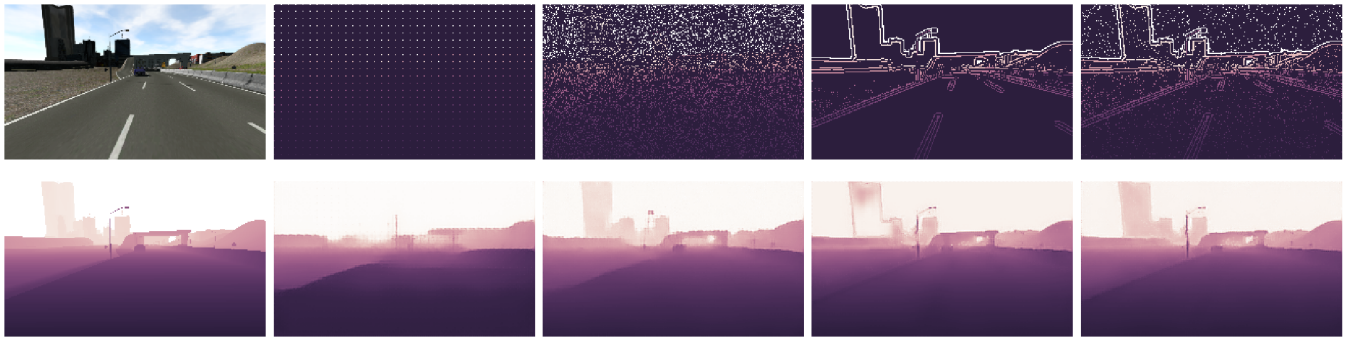

Figure 5: Intensity image and inputs (upper) and corresponding outputs (lower) of neural network on a sample from SYNTHIA dataset. Left to right: ground truth, regular grid, uniform, only gradient, and gradient combined with uniform spatial distributions.

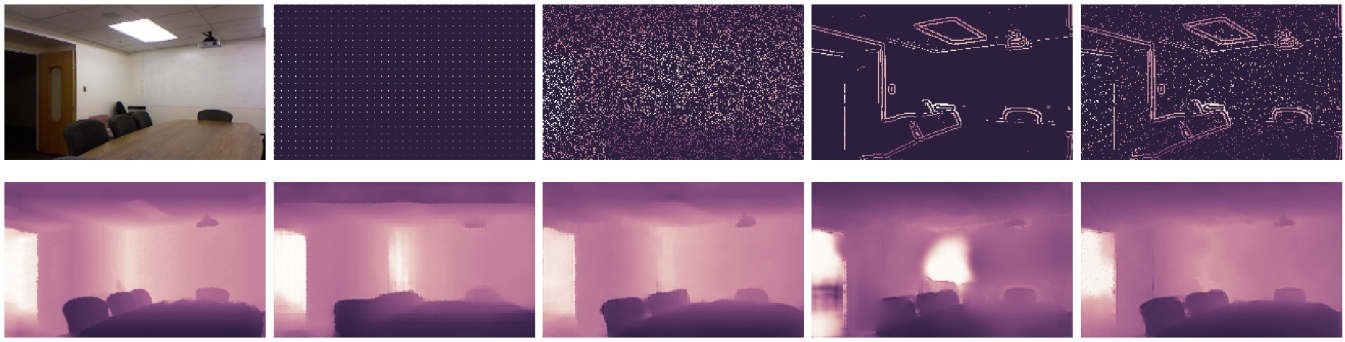

Figure 6: The processing order for the NYU Depth dataset is the same as in Fig. 5.

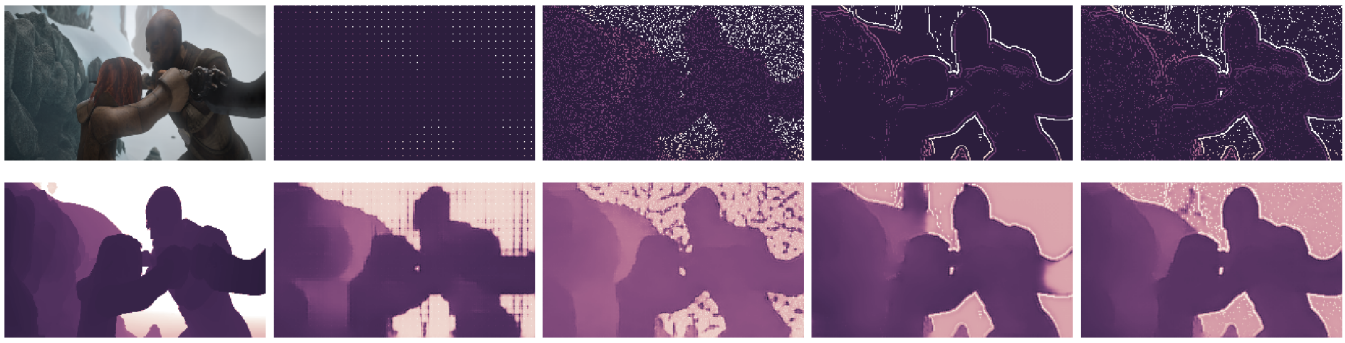

Figure 7: The processing order for the Sintel dataset is the same as in Fig. 5.

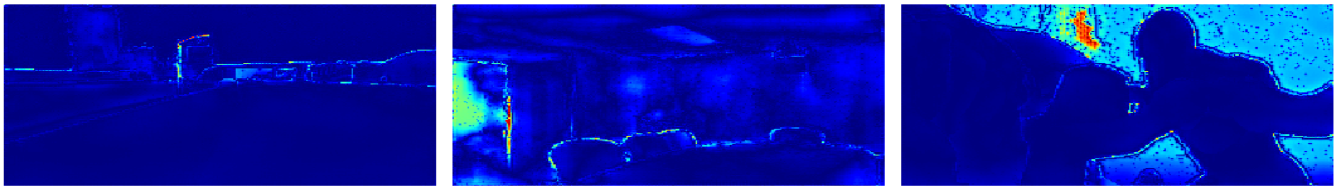

Figure 8: Error maps of combined sampling input examples given in Figures 5, 6, 7. Brighter colors mean higher errors. Left to right: SYNTHIA, NYU Depth, Sintel.

	Art		Laundry		Moebius		Dolls	
	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE
Yang [42]	0.43	46.21	0.41	33.87	0.41	19.01	0.54	16.42
Wang [40]	0.55	47.07	0.53	32.56	0.54	19.76	0.63	15.16
Konno et al. [17]	0.63	30.01	0.59	**21.31**	0.61	12.09	0.70	**10.86**
Our Method	**0.68**	**28.02**	**0.79**	21.92	**0.87**	**8.95**	**0.86**	26.98

Table 3: Evaluation of performance of the super-resolution network. Results are given for ×8 upsampling; for other methods values obtained from [17]

	RMSE					
	Art	Books	Moebius	Reindeer	Laundry	Dolls
Proposed	3.92	1.96	1.39	**1.61**	2.01	1.81
MS-Net [14]	**2.77**	**1.07**	**1.14**	1.97	**1.62**	**1.17**
Bilinear	5.99	2.39	2.19	2.31	3.12	1.89
Bicubic	5.29	2.07	2.01	3.99	3.45	1.86
Lanczos	5.50	2.08	2.03	4.92	4.01	1.84

Table 4: Comparison with unguided deep neural network based method proposed in [14] and classic interpolation methods, ×8 upsampling. Values for RMSE are given with scaling according to [14]. While [14] has better results, our methods run time is 62ms compared to 248ms in [14]

Figure 9: Example of depth super-resolution. Left to right: ground truth, input, ×8 upsampled output

to the case with semi-dense depth interpolation: the output depth values of the network have low dynamic range. In Table 4, we show the comparison with one of the state-of-the art unguided depth super resolution method MS-Net proposed in [14], along with standard interpolation methods. We scale our data to match results given in [14]. Besides our method had not outperformed MS-Net, our run time is almost 4 times less compared to MS-Net (62ms vs 248ms on NVIDIA TITAN X GPU as stated in [14]). In Figure 9 we show the example of inverse depth map from the Middlebury dataset. We can indeed see the background of reconstructed depth map is brighter than ground truth, which means that output has lower depth values in this area.

5.3 Discussion

From the results we can see that our method is able to provide meaningful interpolation of both semi-dense and low resolution depth maps. However, there are several drawbacks.

First, our method fails on scenes with high dynamic range as in the Sintel dataset, producing wrong values in the remote areas. This problem can be solved by segmenting the scene to the foreground and background and discarding invalid depth values for far or background segments. Another solution is to clip output depth values to some predefined range.

Second, the method tends to generate hole-like artifacts when there is no data in some large image region. While this effect can be seen on gradient-only sampling outputs, it prevents us from direct application to semi-dense depth maps generated by direct SLAM methods. In order to reduce this effect we need to change sampling patterns or to introduce hole-filling preprocessing procedure.

6 CONCLUSIONS AND FUTURE WORK

We have presented the end-to-end learnable method for semi-depth dense maps interpolation. Our method allows to reconstruct dense maps from a semi-dense map with small relative error, and to get high-quality low-resolution depth map upsampling. However, there

are still the unsolved issues. The problem with low dynamic range should be solved to provide reliable reconstruction. We will work on integrating our method with the existing semi-dense SLAM systems. Future work may also include optimisation of the network architecture for running on low-performance embedded devices.

We aim to provide a ready-to-use efficient method for scene reconstruction that can be used in autonomous vehicles to build environment model or in AR/MR games for scene reconstruction.

In [26], the authors described another application of fast depth map interpolation, which could be used for reconstructing navigable surfaces and operating computer controlled intellectual agents. With the help of the image representation with the depth map information, similar to [20], we use deep reinforcement learning methods for training on a visual input and extracting game features, such as the enemy recognizing and tracking, items collection, and learning tactical navigation. In what follows, we aim to create game application in Unreal Engine 4 that will unify the described method under a real-time multiplayer first-person shooter game application. We are looking forward merging our model of intelligent shooter game for VR [27] with Mixed/Augmented Reality, which will lead to a new level of game experience.

The detection of road accidents is also one of important tasks for the future work. As we could see on Sintel dataset, there are two situations, in which we could misclassify a close object to be a far one, thus increasing the error of the depth map interpolation. Such situations may cause a false depth estimation leading to car accidents. We plan to use the proximity data generated by our method in the context of ontology-based data access over temporal, streaming, and spatial data [18], [2] with disjunctive ontology language [10], where temporal and spatial ontologies define complex situations the users are interested to detect (for example, fast driving, over speed, pedestrian detection, incorrect speed counter malfunction, etc.) so that even non-expert users can easily formulate queries over processed streaming data.

ACKNOWLEDGMENTS

The work was supported by the Russian Science Foundation under grant 17-11-01294 and performed at National Research University Higher School of Economics, Russia.

REFERENCES

[1] Oisin Mac Aodha, Neill D. F. Campbell, Arun Nair, and Gabriel J. Brostow. 2012. Patch Based Synthesis for Single Depth Image Super-resolution. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part III (ECCV'12)*. Springer-Verlag, Berlin, Heidelberg, 71–84. https://doi.org/10.1007/978-3-642-33712-3_6

[2] Sebastian Brandt, Elem Güzel Kalayci, Roman Kontchakov, Vladislav Ryzhikov, Guohui Xiao, and Michael Zakharyaschev. 2017. Ontology-Based Data Access with a Horn Fragment of Metric Temporal Logic. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, Satinder P. Singh and Shaul Markovitch (Eds.). AAAI Press, NY, 1070–1076. http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14881

[3] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. 2012. A Naturalistic Open Source Movie for Optical Flow Evaluation. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part VI (ECCV'12)*. Springer-Verlag, Berlin, Heidelberg, 611–625. https://doi.org/10.1007/978-3-642-33783-3_44

[4] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2015. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289* 1511.07289 (2015), 1–14.

[5] James Diebel and Sebastian Thrun. 2005. An Application of Markov Random Fields to Range Sensing. In *Proceedings of the 18th International Conference on Neural Information Processing Systems (NIPS'05)*. MIT Press, Cambridge, MA, USA, 291–298. http://dl.acm.org/citation.cfm?id=2976248.2976285

[6] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2014. *Learning a Deep Convolutional Network for Image Super-Resolution.* Springer International Publishing, Cham, 184–199. https://doi.org/10.1007/978-3-319-10593-2_13

[7] Jakob Engel, Vladlen Koltun, and Daniel Cremers. 2016. Direct sparse odometry. *arXiv preprint arXiv:1607.02565* 1607.02565 (2016), 1–17.

[8] Jakob Engel, Thomas Schöps, and Daniel Cremers. 2014. *LSD-SLAM: Large-Scale Direct Monocular SLAM.* Springer International Publishing, Cham, 834–849. https://doi.org/10.1007/978-3-319-10605-2_54

[9] D. Ferstl, C. Reinbacher, R. Ranftl, M. Ruether, and H. Bischof. 2013. Image Guided Depth Upsampling Using Anisotropic Total Generalized Variation. In *2013 IEEE International Conference on Computer Vision*. IEEE, New York, NY, USA, 993–1000. https://doi.org/10.1109/ICCV.2013.127

[10] Olga Gerasimova, Stanislav Kikot, Vladimir Podolskii, and Michael Zakharyaschev. 2017. On the Data Complexity of Ontology-Mediated Queries with a Covering Axiom. In *Description Logics*. CEUR-WP (in Print), Montpellier, France, 1–12.

[11] M. Gupta, Q. Yin, and S. K. Nayar. 2013. Structured Light in Sunlight. In *2013 IEEE International Conference on Computer Vision*. IEEE, New York, NY, USA, 545–552. https://doi.org/10.1109/ICCV.2013.73

[12] S. Hawe, M. Kleinsteuber, and K. Diepold. 2011. Dense disparity maps from sparse disparity measurements. In *2011 International Conference on Computer Vision*. IEEE, New York, NY, USA, 2126–2133. https://doi.org/10.1109/ICCV.2011.6126488

[13] K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, New York, NY, USA, 770–778. https://doi.org/10.1109/CVPR.2016.90

[14] Tak-Wai Hui, Chen Change Loy, and Xiaoou Tang. 2016. *Depth Map Super-Resolution by Deep Multi-Scale Guidance.* Springer International Publishing, Cham, 353–369. https://doi.org/10.1007/978-3-319-46487-9_22

[15] Simon Jégou, Michal Drozdzal, David Vázquez, Adriana Romero, and Yoshua Bengio. 2016. The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation. *CoRR* abs/1611.09326 (2016), 1–9. http://arxiv.org/abs/1611.09326

[16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. *Perceptual Losses for Real-Time Style Transfer and Super-Resolution.* Springer International Publishing, Cham, 694–711. https://doi.org/10.1007/978-3-319-46475-6_43

[17] Yousuke Konno, Masayuki Tanaka, Masatoshi Okutomi, Yukiko Yanagawa, Koichi Kinoshita, and Masato Kawade. 2016. Depth Map Upsampling by Self-Guided Residual Interpolation. In *Proceedings of the 23rd International Conference on Pattern Recognition (ICPR2016)*. IEEE, New York, NY, USA, 1395–1400.

[18] Roman Kontchakov, Laura Pandolfo, Luca Pulina, Vladislav Ryzhikov, and Michael Zakharyaschev. 2016. Temporal and Spatial OBDA with Many-Dimensional Halpern-Shoham Logic. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, Subbarao Kambhampati (Ed.). IJCAI/AAAI Press, NY, 1160–1166. http://www.ijcai.org/Abstract/16/168

[19] Johannes Kopf, Michael F. Cohen, Dani Lischinski, and Matt Uyttendaele. 2007. Joint Bilateral Upsampling. *ACM Trans. Graph.* 26, 3 (July 2007), 1–5. https://doi.org/10.1145/1276377.1276497

[20] Guillaume Lample and Devendra Singh Chaplot. 2016. Playing FPS Games with Deep Reinforcement Learning. *CoRR* abs/1609.05521 (2016), 7. http://arxiv.org/abs/1609.05521

[21] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2016. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802* 1609.04802 (2016), 1–19.

[22] L. K. Liu, S. H. Chan, and T. Q. Nguyen. 2015. Depth Reconstruction From Sparse Samples: Representation, Algorithm, and Sampling. *IEEE Transactions on Image Processing* 24, 6 (June 2015), 1983–1996. https://doi.org/10.1109/TIP.2015.2409551

[23] M. Y. Liu, O. Tuzel, and Y. Taguchi. 2013. Joint Geodesic Upsampling of Depth Images. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, New York, NY, USA, 169–176. https://doi.org/10.1109/CVPR.2013.29

[24] J. Long, E. Shelhamer, and T. Darrell. 2015. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, New York, NY, USA, 3431–3440. https://doi.org/10.1109/CVPR.2015.7298965

[25] Jiajun Lu and D. Forsyth. 2015. Sparse depth super resolution. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, New York, NY, USA, 2245–2253. https://doi.org/10.1109/CVPR.2015.7298837

[26] Ilya Makarov, Vladimir Aliev, Olga Gerasimova, and Pavel Polyakov. 2017. Depth Map Interpolation using Perceptual Loss. In *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct) (in Print)*. IEEE, Nantes, France, 1–2.

[27] Ilya Makarov, Mikhail Tokmakov, Pavel Polyakov, Peter Zyuzin, Maxim Martynov, Oleg Konoplya, George Kuznetsov, Ivan Guschenko-Cheverda, Maxim Uriev, Ivan Mokeev, Olga Gerasimova, Lada Tokmakova, and Alexey Kosmachev. 2016. First-Person Shooter Game for Virtual Reality Headset with Advanced Multi-Agent

Intelligent System. In *Proceedings of the 2016 ACM on Multimedia Conference (MM '16)*. ACM, New York, NY, USA, 735–736. https://doi.org/10.1145/2964284.2973826

[28] Christopher J. Pal, Jerod J. Weinman, Lam C. Tran, and Daniel Scharstein. 2012. On Learning Conditional Random Fields for Stereo. *International Journal of Computer Vision* 99, 3 (2012), 319–337. https://doi.org/10.1007/s11263-010-0385-z

[29] Jaesik Park, Hyeongwoo Kim, Yu-Wing Tai, Michael S. Brown, and Inso Kweon. 2011. High Quality Depth Map Upsampling for 3D-TOF Cameras. In *Proceedings of the 2011 International Conference on Computer Vision (ICCV '11)*. IEEE Computer Society, Washington, DC, USA, 1623–1630. https://doi.org/10.1109/ICCV.2011.6126423

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. Springer International Publishing, Cham, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

[31] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. 2016. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, New York, NY, USA, 3234–3243. https://doi.org/10.1109/CVPR.2016.352

[32] Daniel Scharstein and Chris Pal. 2007. Learning conditional random fields for stereo. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, IEEE, New York, NY, USA, 1–8.

[33] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. 2016. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, New York, NY, USA, 1874–1883.

[34] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. 2012. *Indoor Segmentation and Support Inference from RGBD Images*. Springer Berlin Heidelberg, Berlin, Heidelberg, 746–760. https://doi.org/10.1007/978-3-642-33715-4_54

[35] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* 1409.1556 (2014), 1–14.

[36] Xibin Song, Yuchao Dai, and Xueying Qin. 2017. *Deep Depth Super-Resolution: Learning Depth Super-Resolution Using Deep Convolutional Neural Network*. Springer International Publishing, Cham, 360–376. https://doi.org/10.1007/978-3-319-54190-7_22

[37] Juan Jose Tarrio and Sol Pedre. 2015. Realtime Edge-Based Visual Odometry for a Monocular Camera. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV) (ICCV '15)*. IEEE Computer Society, Washington, DC, USA, 702–710. https://doi.org/10.1109/ICCV.2015.87

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

[39] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky. 2016. Texture Networks: Feed-forward Synthesis of Textures and Stylized Images. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48 (ICML'16)*. JMLR.org, New York, NY, USA, 1349–1357. http://dl.acm.org/citation.cfm?id=3045390.3045533

[40] L. Wang, H. Wu, and C. Pan. 2014. Fast Image Upsampling via the Displacement Field. *IEEE Transactions on Image Processing* 23, 12 (Dec 2014), 5123–5135. https://doi.org/10.1109/TIP.2014.2360459

[41] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.

[42] Jianchao Yang, John Wright, Thomas S. Huang, and Yi Ma. 2010. Image Super-resolution via Sparse Representation. *Trans. Img. Proc.* 19, 11 (Nov. 2010), 2861–2873. https://doi.org/10.1109/TIP.2010.2050625

[43] Raymond Yeh, Chen Chen, Teck Yian Lim, Mark Hasegawa-Johnson, and Minh N Do. 2016. Semantic Image Inpainting with Perceptual and Contextual Losses. *arXiv preprint arXiv:1607.07539* 1607.07539 (2016), 1–10.