Inpainting of Depth Images using Deep Neural Networks for Real-Time Applications

Roland Fischer, Janis Roßkamp, Thomas Hudcovic, Anton Schlegel, and Gabriel Zachmann

University of Bremen, Bremen, Germany
r.fischer@uni-bremen.de

Abstract. Depth sensors enjoy increased popularity throughout many application domains, such as robotics (SLAM) and telepresence. However, independent of technology, the depth images inevitably suffer from defects such as holes (invalid areas) and noise. In recent years, deep learning-based color image inpainting algorithms have become very powerful. Therefore, with this work, we propose to adopt existing deep learning models to reconstruct missing areas in depth images, with the possibility of real-time applications in mind. After empirical tests with various models, we chose two promising ones to build upon: a U-Net architecture with partial convolution layers that conditions the output solely on valid pixels, and a GAN architecture that takes advantage of a patch-based discriminator. For comparison, we took a standard U-Net and LaMa. All models were trained on the publically available NYUV2 dataset, which we augmented with synthetically generated noise/holes.
Our quantitative and qualitative evaluations with two public and an own dataset show that LaMa most often produced the best results, however, is also significantly slower than the others and the only one not being real-time capable. The GAN and partial convolution-based models also produced reasonably good results. Which one was superior varied from case to case but, generally, the former performed better with small-sized holes and the latter with bigger ones. The standard U-Net model that we used as a baseline was the worst and most blurry.

Keywords: Image Inpainting · Depth Completion · Real-Time · Depth Images · Deep Learning · CNN · GAN · LaMa · U-Net · Azure Kinect.

1 Introduction

With the growing availability of low-cost depth sensors and RGB-D cameras (e.g., Azure Kinect), their popularity and employment increased throughout various research areas and industries. Typical use cases are SLAM and object detection in computer vision and robotic applications, or real-time capturing of point cloud avatars for telepresence systems. A long-lasting challenge is, however, handling the inherent sensor noise, as well as artifacts and holes that lead to an incomplete depth image. These issues are inevitable consequences of the time-of-flight principle many depth sensors use. Concretely, multipath inference, caused by repeated reflection of the infrared rays between objects, and signals that are too powerful or too weak lead to ambiguous or invalid depth values. Having accurate and dense depth maps is important for many downstream tasks such as safe motion planning, reliable vision in autonomous vehicles, or industrial inspection. One solution would be to adapt those downstream, domain-specific algorithms and models to work with incomplete input. However, experience shows that specialized algorithms and models that only focus on the specific task of denoising/inpainting the input data are more effective and lead to better overall results. Therefore, preprocessing and enhancing the depth images is an important task. Reconstructing the missing areas in real-time is not trivial, though, as there are strong spatial dependencies between the data points, both locally and globally. Additionally, previous work in the area of hole filling and image inpainting was mostly focused on regular color images and is not necessarily well-suited for direct application on depth images.

With this work, we propose an approach of real-time depth image inpainting using neural networks. Our main contribution is the investigation of the depth image reconstruction quality of two fast U-Net-based network models that were originally designed for color image inpainting, including a comparison with a basic U-Net and a more sophisticated state-of-the-art model. In contrast to many others, the models we use do not need any color images for guidance, which makes them more generally applicable as they can be also employed in use cases where no color information is available. The first model we chose uses partial convolutions, while the second one is based on a GAN architecture. Furthermore, we present a detailed quantitative and qualitative evaluation using two public datasets and a custom one we recorded ourselves.

2 Related Work

Traditionally, missing areas in pictures are reconstructed, or painted-in, using pixel- or patch-based exemplar methods [18], diffusion methods [25], or hybrids of the two [22]. In recent years, however, deep-learning-based methods outclassed traditional methods, especially when restoring larger areas, as they are able to learn and consider the semantics of the image. The most common CNN variants for image inpainting are FCN and U-Net. However, to avoid filling missing areas with noise and then convoluting this information further, some authors proposed non-standard convolutions. For instance, Liu et al. [13] proposed partial convolution layers that dynamically mask out invalid pixels and cope better with irregular holes. Yu et al. [30] introduced gated convolutions that generalize partial convolutions and provide a learnable dynamic feature selection mechanism across all channels and layers. Similarly, Xie et al. [27] suggested using learnable bidirectional attention maps. In order to better and effectively capture long-distance information, Ning et al. [16] proposed adding a multi-scale attention module. Yan et al. [28] introduced shift connection layers that shift features of known areas to serve as guidance for missing areas and Suvorov et al. [23] presented a combination of Fourier convolutions, a high receptive field perceptual loss, and large training masks for inpainting of large areas. Moreover, many approaches for deep-learning-based inpainting employ one of the various GAN architectures, such as the one proposed by Isola et al. [5], as they feature strong data generation capabilities. Other examples include the works by Shen et al. [21], Yeh et al. [29], and Shao et al. [20]. Similarly, diffusion-based networks such as the one by Rombach et al. [17] achieved impressive results in various image synthesis tasks. These models consist of a hierarchy of denoising autoencoders and can model complex, multi-modal distributions, however, inference tends to be very expensive.

Very recently, transformer networks, usually employed for natural language processing, were discovered to be very effective for computer vision and image processing tasks, such as denoising, too [2]. The main benefit is their ability to model long-range dependencies. Interestingly, Makarov and Borisenko [14] used vision transformers for color-guided depth completion and Li et al. [9] proposed a combination of convolutions and transformers for large hole inpainting. Similarly, Yu et al. [31] presented a bidirectional autoregressive transformer model for diverse inpainting, and Deng et al. [3] designed a transformer for inpainting with a focus on efficiency. However, transformers are usually rather slow.

Most research is focused on inpainting color images, and only very few works consider reconstructing depth images. Works that do consider depth images usually are situated in the field of RGB-D reconstruction or lidar-based depth completion and use the color image for guidance. For instance, Ma and Karaman [15] employed a deep regression network to predict depth images based on corresponding color images and sparse depth samples. Fujii et al. [4] used a late fusion GAN to simultaneously reconstruct color and depth images by exploiting the complementary relationship between RGB and depth information. Lee et al. [8] proposed multi-scaled and densely connected locally convolutional layers for depth completion, Tao et al. [24] use a neural network for the prediction of dense depth maps as well as uncertainty estimates, and Jeon et al. [6] performs depth completion based on line features by bridging the conventional and deep learning-based approaches. All these works require color input as well, though. Similarly, Zhang and Funkhouser [33], as well as Satapathy and Sahay [19], rely on color guidance. In contrast, Jin et al. [7] and Li and Wu [11] presented solutions for depth inpainting without color guidance, however, they are only designed to handle smaller holes. Other works that solely work on depth images can be found in the medical domain, i.e., to reconstruct and in-paint CT or MRI scans. Both, Li et al. [10] and Armanious et al. [1], for instance, presented promising solutions using patch-based GANs. For a more comprehensive overview, we refer to the excellent literature review by Zhang et al. [32].

3 Our Approach

To tackle the issue of (real-time) depth image inpainting, and after thoroughly experimenting with the current state of the art in deep color image inpainting, we decided to adopt two promising works that we considered suitable as a foundation. The first model we chose is the one by Liu et al. that introduced partial convolutions [13], and the second one is the GAN model proposed by Isola et al. [5]. As a baseline for comparison, we also took a standard U-Net model and the more sophisticated state-of-the-art model by Suvorov et al. [23], LaMa, which we expected to be significantly slower, though.

3.1 Datasets and Preprocessing Pipeline

For the training and evaluation of our models, we resorted to using two publicly available depth datasets, namely, the SceneNet RGB-D dataset by McCormac et al. and the NYU Depth V2 dataset by Silberman et al. The SceneNet dataset provides 5 million photo-realistic RGB-D images of synthesized indoor scenes. We only use the depth images. These are 16-bit encoded which is similar to real-world input, however, the image resolution is significantly lower than the ones of common depth sensors such as the Azure Kinect. In order to prevent upsampling artifacts from influencing the training, we use this dataset only for evaluation. The NYUV2 dataset was collected by capturing a wide range of indoor locations within a large city using a Kinect V1 RGB-D camera. Additionally, we created our own custom dataset consisting of mostly static and a few dynamic scenes using the Microsoft Azure Kinect RGB-D camera. As this data lacks a ground truth, we use it only for evaluation, too. In the end, we trained our models with a split of 44984 depth images for the training set, 654 for the validation set, and 5704 for the test set (NYUV2). For the evaluation, we used additional 23 scenes with 6900 images (SceneNet) and 23 scenes with 6739 images (custom dataset).

For the training procedure, the images go through a preprocessing pipeline. First, the images get resized to 512^2 and scaled to the range of 0-1 for compatibility purposes with the models. Then, an illumination mask similar to the one of the Azure Kinect is generated and applied to adapt the dataset's images to real-world input conditions. As the dataset used for training doesn't contain any holes, we generate synthetic ones as well as single outlier pixels. The synthetic holes are created by combining multiple random masks with different scales and frequencies that are generated with sci-kit-image. To guarantee a diverse input, the final noise masks are evenly drawn from multiple categories with varying percentages of invalid pixels and sizes of holes. Finally, we apply classical data augmentation techniques such as random flipping (90-degree angles) and homogeneous intensity shifts.

3.2 Network Details

In the following, the network details of our models get briefly described. For more details, we refer to the corresponding original papers.

Partial Convolution: Our first network model is based on the one presented by Liu et al. [13] and, like the original, follows a U-Net architecture with partial convolution layers. Our model features only one input and output channel, respectively, though. We chose an input resolution of 512^2, as it is the closest square number to the resolution of the Azure Kinect images. The kernel sizes for the partial convolutions in the encoder part are 7,5,5,3,3,3,3 and 3, following the presented layer order. The decoder uses filter sizes of 3 for all convolutions. For all convolutions in the network, stride values of 2 are used. The implementation of this network is based on the existing third-party implementation of Ryan Wongsa for the U-Net architecture [26] and loss functions. However, adjustments were made due to the fact that the crucial weight initializations as well as the input normalizations of the VGG-16 network were missing. Moreover, the implementation of the partial convolutional layer from the original authors was used [12], too.

GAN: Our second network model is based on the GAN architecture presented by Isola et al. [5] that uses a U-Net for the generator and a convolutional PatchGAN classifier for the discriminator. The latter penalizes structure at the scale of image patches. The generator part of the GAN is very close to the previous U-Net: The encoder consists of 8 identical blocks instead of 7, which are Conv-BN-LeakyRelu blocks that use the same filter sizes of 64, 128, 256, 512, 512, 512, 512, 512. The decoder consists of seven Upsampling-Concat-BN-Relu blocks. Additional dropouts of 50% are applied to the first three blocks after the normalization process. A final convolution maps the number of output channels. The input dimensions are $512^2 \times 3$, as three depth images are stacked. All convolutions of the network use filters of size 4 with a stride of 2. The discriminator consists of one Conv-LeakyReLu layer followed by 3 Conv-BN-LeakyReLU blocks and a single Conv-ZeroPadding-Sigmoid block. This outputs a 30^2 image patch that can classify a 70^2 portion of the input image.

Convolutional U-Net: As a baseline for comparison of the previous models, we utilize a CNN with a standard U-Net architecture, although models with normal convolutional layers that treat all image pixels the same and even share filter weights are not ideal for image inpainting. The architecture is similar to the one of the partial convolution model but with regular convolutions.

LaMa: To get a more complete picture and to compare the models with more sophisticated networks, we also adopted the LaMa network by Suvorov et al. [23]. It is specifically designed for the inpainting of large areas by using fast Fourier convolutions that provide a large receptive field, as well as an adapted perceptual loss and large training masks. However, as it is more complex, we expect it to be significantly slower and possibly not real-time capable. For details about the architecture, we refer to the original paper, from which we directly adopted it.

3.3 Training Procedure

For convenience, from now on, we abbreviate the models' names with Conv, PConv, GAN, and LaMa. The models were trained for 7 epochs (LaMa: 5) using a batch size of 2 (LaMa: 5), due to the huge memory load. As a loss function, we used, similarly to the partial convolution paper by Liu et al., a combination consisting of two per-pixel accuracy losses, a perceptual loss, two style losses, and a total variation loss. We experimented with different weights but found the ones used in the paper to be the best-performing ones. In the case of the GAN model, the generator loss is a combination of the previous total loss and the original generator loss as described in the paper by Isola et al. The losses for LaMa were directly adopted from the original paper.

4 Results

First, we measured the duration of inference needed for inpainting a 512^2 depth image, using an Intel Core i5-10400F CPU, 16 GB of RAM, and an NVIDIA GeForce RTX 2070. A fast inference is crucial for practical real-time applications, i.e., as a preprocessing step in a longer pipeline. As depth sensors usually capture with 30 Hz, the inference time must stay below 33 ms for real-time use. To replicate a data stream of images, the images were inpainted one after another, instead of as a batch. For the GAN method, we measured a pure inference time of 24.3 ms, for the Conv method 24.93 ms, and for the PConv method 9.37 ms. Including preprocessing, we get 27.69 ms, 26.29 ms, and 34.34 ms, respectively. The PConv model takes the longest for the preprocessing as it needs more steps than the other models, i.e., an extra input mask. However, the time for pure inference is the quickest. Generally, even though there is still potential for optimization, these models are quick enough for real-time application. In contrast, LaMa takes 60.02 ms and, thus, is significantly slower and not quite real-time capable. Out of interest, we also tested a diffusion-based model [17] but, as expected, the inference was extremely slow with 3-4 seconds for an image with 50 sampling steps (which was, as we found, a “sweet spot” for image legibility and speed). Unfortunately, the inpainting results were still comparatively poor. And although better output quality can be achieved with more sampling steps during inference, doing so only impacts inference time even more, which is why we did not consider latent diffusion-based models further.

To quantitatively evaluate the performance of our models, we calculated and compared the MAE, MSE, PSNR, and SSIM on the test sets of the NYUV2 and SceneNet RGB-D datasets (only depth used). Moreover, we separately computed the metrics for the different hole/mask categories, which bundle images with similar ratios of valid/invalid areas to get more detailed insights. The results on the NYUV2 dataset show that LaMa consistently performs best. Moreover, we see a better performance of the GAN method on the first four mask categories, especially if looking at the MAE and MSE, see Table 1 (left). The performance gradually decreases with each category, though, and after the fourth category, the PConv method overtakes the GAN performance in terms of SSIM and PSNR values. In comparison, the Conv method is (as expected) the worst-performing one. Generally, the PConv method seems to be the most consistent method and better at dealing with bigger holes than the GAN and Conv methods. Overall, the models seem to perform similarly on the SceneNet RGB-D dataset (see Table 1 (right)): For the lower mask categories, the GAN method outperforms the Conv and PConv methods, while the PConv method shows better results on the higher categories, and is the most consistent one. LaMa again performs most often the best. However, in terms of SSIM, here, GAN/PConv perform better.

Table 1. Inpainting results on the NYUV2 (left) and SceneNet RGB-D (depth only) (right) test sets using 6 hole categories (percent of invalid pixels; more/bigger holes to the right). The best value per block is marked in bold. Most often, LaMa performs best. The GAN method performs second best on smaller mask categories while the PConv method performs second best on bigger ones and has the most consistent results.

		NYUV2						SceneNet RGB-D					
	Model	.01/.1	.1/.2	.2/.3	.3/.4	.4/.5	.5/.6	.01/.1	.1/.2	.2/.3	.3/.4	.4/.5	.5/.6
MAE	PConv	4.89	5.24	5.10	5.32	5.79	7.61	66.89	81.85	77.27	65.85	63.67	110.62
	Conv	3.53	3.24	3.27	3.52	5.64	13.35	118.68	103.84	101.80	113.30	182.04	438.79
	GAN	1.79	1.77	1.93	2.46	4.48	11.18	66.49	65.07	72.18	89.61	176.30	414.24
	LaMa	**0.06**	**0.18**	**0.31**	**0.42**	**0.64**	**1.00**	**4.28**	**10.52**	**16.09**	**20.75**	**30.77**	**45.54**
MSE	PConv	47.82	54.00	54.21	60.67	77.99	154.54	21732	23122	21622	18157	**16787**	32051
	Conv	62.90	56.49	58.45	69.16	131.46	612.88	73128	66152	66116	78239	133677	669006
	GAN	6.79	7.34	10.93	16.68	67.83	415.20	8414	9023	13010	21732	95940	588678
	LaMa	**0.28**	**0.87**	**1.67**	**2.48**	**4.81**	**12.18**	**5044**	**8187**	**9521**	**10858**	17161	**29519**
PSNR	PConv	35.12	34.81	34.70	34.43	33.27	30.47	38.83	39.08	39.18	38.91	37.85	35.58
	Conv	32.29	32.40	32.13	31.64	28.59	22.40	35.50	35.89	35.77	35.41	32.41	26.33
	GAN	41.42	40.51	38.94	37.01	31.13	23.72	44.37	43.76	42.38	40.34	34.32	26.90
	LaMa	**55.04**	**50.15**	**47.38**	**45.74**	**43.01**	**39.22**	**57.31**	**53.02**	**50.59**	**49.23**	**46.93**	**43.82**
SSIM	PConv	.9799	.9771	.9746	.9701	.9630	.9385	.9881	.9876	.9867	**.9866**	**.9855**	**.9818**
	Conv	.9344	.9230	.9184	.9026	.8819	.8264	.9659	.9606	.9556	.9510	.9388	.8919
	GAN	.9935	.9874	.9815	.9759	.9480	.8814	**.9960**	**.9931**	**.9885**	.9834	.9674	.9166
	LaMa	**.9987**	**.9966**	**.9943**	**.9927**	**.9898**	**.9842**	.9958	.9899	.9855	.9829	.9793	.9755

We also did a qualitative evaluation of the inpainting performance based on a selection of test images from different mask categories. This evaluation is, naturally, subjective but possibly also more relatable. Fig. 1 shows the results using the NYUV2 dataset. For all three mask categories, LaMa produces the best results that are very close to the original. The PConv method is also able to create good results without apparent visual artifacts, apart from a slight blur in the last row with a mask of 40%-50% hole-to-image ratio. For the GAN method, the results for the small mask are very close to the ground truth image. However, on the medium and big masks, we can see slight deteriorations and then even more artifacts occurring, respectively. The Conv method visibly leads to the worst results throughout all mask categories, as can be seen by the increased blurriness and other (dark, cloudy) artifacts. Generally, we find that the qualitative results are consistent with the quantitative ones. The results using the SceneNet RGB-D dataset in Fig. 2 are similar: LaMa performs better than all the others, especially for the bigger mask categories. The PConv method creates reasonably good results for the small and medium mask categories, the GAN performs well in the small category, and the Conv method is the worst-performing method. However, on this dataset, all methods apart from LaMa have issues with artifacts in the form of too-bright or too-dark areas that get more severe with bigger masks. This could be because of systemic differences between this dataset and the one used for training (NYUV2). For instance, this dataset with synthetically created images generally has sharper edges/objects than the NYUV2 dataset, which also incorporated errors that slightly degrade the images.

To evaluate our models on real-world data, we first investigate the effects of the inpainting methods on the valid areas. Ideally, they should remain unchanged. As can be seen in Fig. 3, which shows the color-coded deltas between the original and inpainted images, this is mostly not the case. The PConv method leads to relatively small differences, mostly along the edges of objects, corners, or at thin shapes. This could be an effect of the model trying to prevent hard edges and instead favoring slow transitions. The GAN method performs better at far corners and edges and, generally, produces images with more even deltas. Moreover, it creates the sharpest results with more abrupt object transitions. An odd issue with the GAN method is the distinct artifacts that occur consistently in the upper right corner. We suspect this to be an issue with the value of the introduced weighting factor λ for the loss function, as the authors of the original method suggested that lower values lead to sharper results but, in turn, lead to more artifacts. The Conv method, again, leads to the worst results and produces the biggest deltas throughout the whole image. Interestingly, in contrast to the others, LaMa does not change the originally valid areas at all, which is the best result. For a final comparison of the models, we compare the resulting images after inpainting, again, using our own custom dataset. As visible in Fig. 4, all methods are able to create reasonable predictions for the missing areas, although the Conv method produces more blurry results. Interestingly, PConv and LaMa as well as Conv and GAN tend to have a similar behavior. Generally, LaMa tends to create the most plausible and visually pleasing results, followed by PConv. However, one drawback of these methods seems to be the prediction around outlier pixels. The advantage of LaMa on this real-world dataset is smaller as with the other datasets though. Moreover, in some cases, the GAN method produces better results, hence, there seems to be no method that is categorically superior.

5 Conclusion

We presented an approach for real-time reconstruction of missing or invalid areas in depth images using deep neural networks. In particular, our approach does not use any guidance by color images. We adopted two different U-Net-based models that originally were designed for color-image inpainting, one using partial convolutions, and the other one being a patch-based GAN. For comparison, we took also a basic U-Net and a more sophisticated state-of-the-art model, namely LaMa. The training was done using the public NYU Depth V2 dataset that we augmented with custom holes. Our quantitative and qualitative evaluations with the NYUV2 and SceneNet datasets showed that LaMa, overall, produces the best inpainting results, the GAN method performs especially well on images with smaller hole-to-image ratios, the partial convolution approach achieves consistently good results (images with various hole sizes and

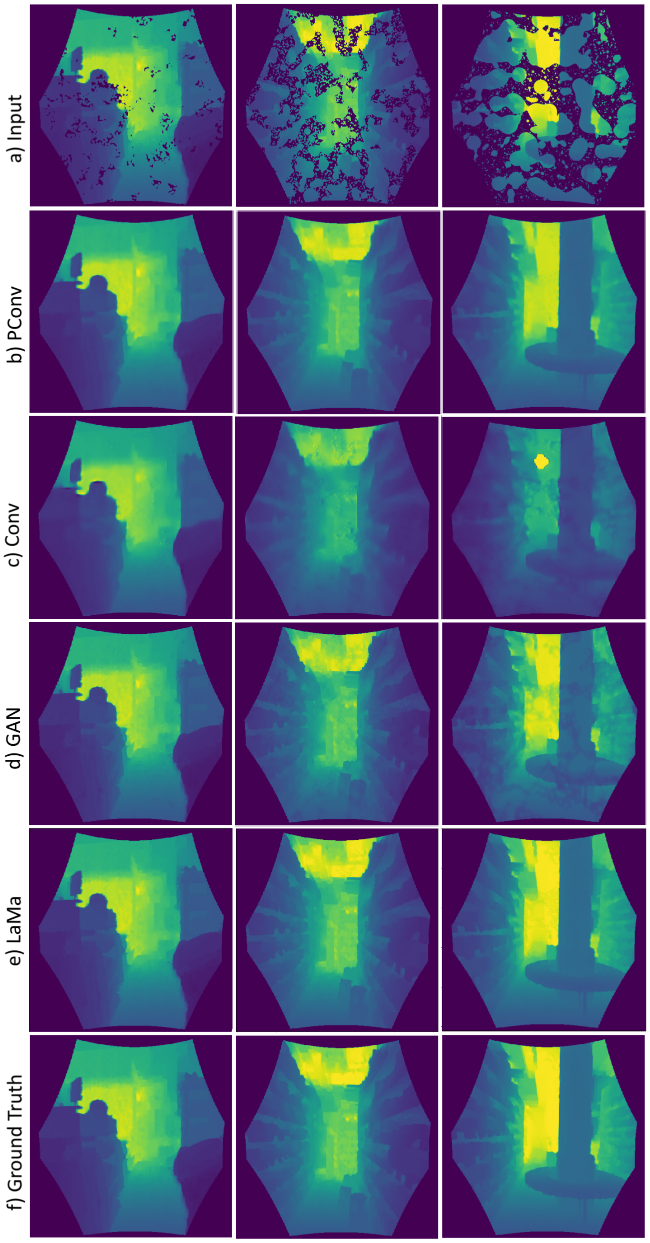

Fig. 1. Visual inpainting results on the NYUV2 test set using various hole categories (columns). LaMa performs best, the PConv method performs second best, the GAN struggles with bigger holes, and the Conv method is the worst.

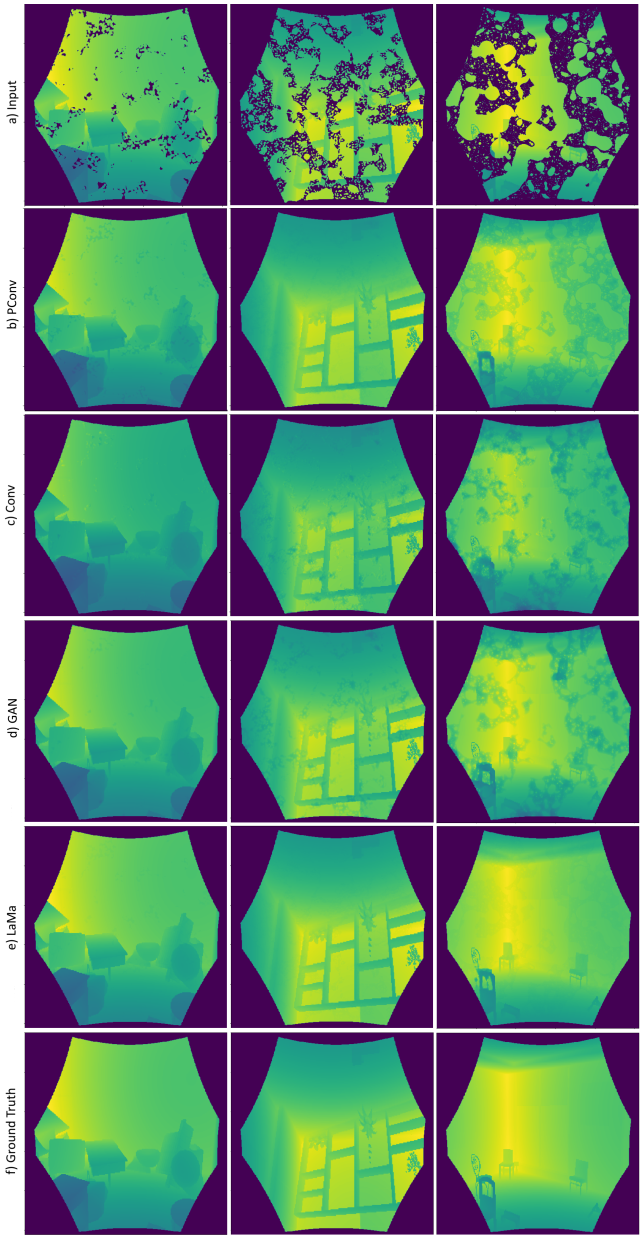

Fig. 2. Visual inpainting results on the SceneNet RGB-D test set (depth only) using various hole categories (columns). All methods apart from LaMa, which performs best, produce distinct artifacts. However, PConv and GAN perform reasonably well in the medium/small categories, and Conv is again the worst-performing method.

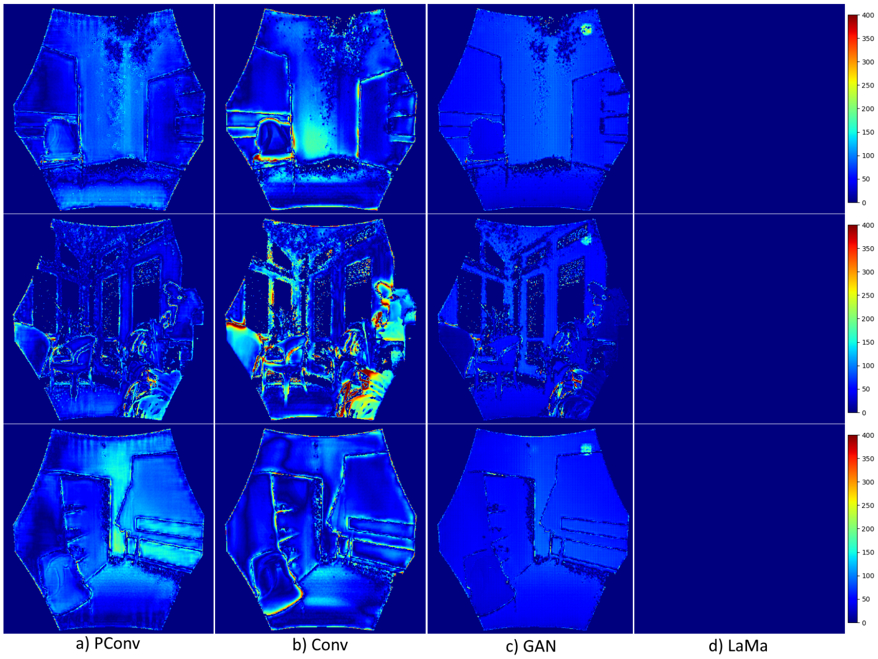

Fig. 3. Color-coded pixel-wise deltas of originally valid areas after inpainting using our dataset. The holes were reintegrated from the input data. The Conv method alters the original data around holes the most (for smoother transitions), GAN the least, and LaMa not at all.

ratios), and the regular convolution-based approach fares the worst. Applied to a custom dataset we recorded with an Azure Kinect sensor, we found that the LaMa model, on average, leads to the visually most pleasing inpainting results, although the PConv and GAN methods also achieve reasonably good and coherent results (the latter sometimes even being superior). To conclude, all methods are able to reconstruct holes of any shape, size, or location without any post-processing procedures, with reasonable to good visual quality. Also, apart from LaMa which is notably slower, they achieve this in a real-time fashion. In the future, we plan to also incorporate RGB data as additional input, if available, to enhance the inpainting results with this extra information. Other network architectures such as transformer models, originally from the natural language processing domain, should be investigated to also take advantage of temporal coherency between subsequent images. Moreover, producing ground truth data for our own dataset (recorded with the Azure Kinect) would be highly beneficial for the training and evaluation of the models. One approach for this challenging task would be to couple the Azure Kinect with another precisely, externally registered depth-sensing device, such as a stereo camera, from which the depth for the missing areas can be produced.

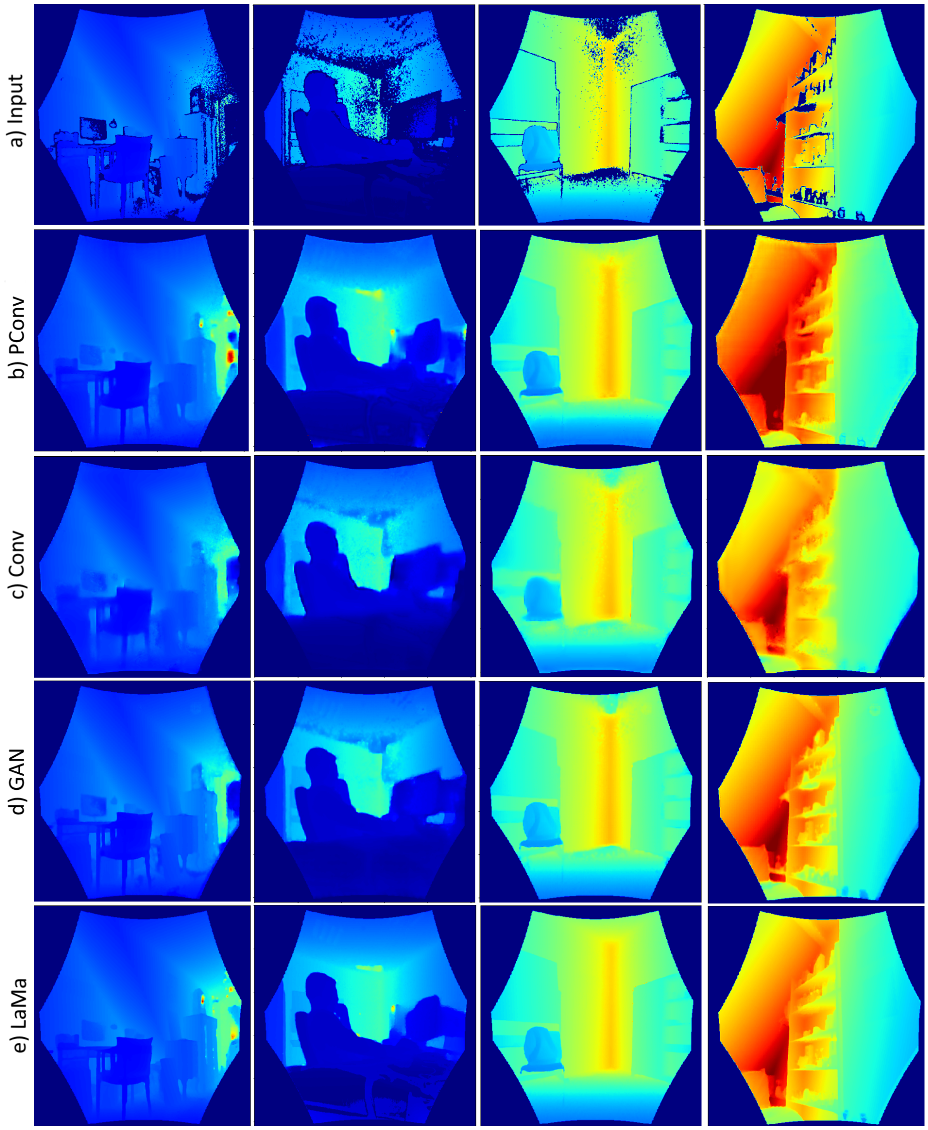

Fig. 4. Inpainting results with our own dataset. The LaMa method most often produces the best visual results. PConv behaves quite similarly, and both struggles with outliers. However, in some cases, the GAN performs the best.

References

1. Armanious, K., Mecky, Y., Gatidis, S., Yang, B.: Adversarial inpainting of medical image modalities. In: IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP). pp. 3267–3271 (2019)
2. Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., Gao, W.: Pre-trained image processing transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12299–12310 (2021)
3. Deng, Y., Hui, S., Zhou, S., Meng, D., Wang, J.: T-former: An efficient transformer for image inpainting. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 6559–6568. MM '22, Association for Computing Machinery (2022)
4. Fujii, R., Hachiuma, R., Saito, H.: Rgb-d image inpainting using generative adversarial network with a late fusion approach. ArXiv **abs/2110.07413** (2020)
5. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.: Image-to-image translation with conditional adversarial networks. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 5967–5976 (07 2017)
6. Jeon, J., Lim, H., Seo, D.U., Myung, H.: Struct-mdc: Mesh-refined unsupervised depth completion leveraging structural regularities from visual slam. IEEE Robot. and Autom. Letters **7**(3), 6391–6398 (2022)
7. Jin, W., Zun, L., Yong, L.: Double-constraint inpainting model of a single-depth image. Sensors **20**(6) (2020)
8. Lee, S., Yi, E., Lee, J., Kim, J.: Multi-scaled and densely connected locally convolutional layers for depth completion. In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 8360–8367 (2022)
9. Li, W., Lin, Z., Kun, Z., Qi, L., Wang, Y., Jia, J.: Mat: Mask-aware transformer for large hole image inpainting. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10748–10758 (06 2022)
10. Li, Z., Cai, A., Wang, L., Zhang, W., Tang, C., Li, L., Liang, N., Yan, B.: Promising generative adversarial network based sinogram inpainting method for ultra-limited-angle computed tomography imaging. Sensors **19**(18) (2019)
11. Li, Z., Wu, J.: Learning deep cnn denoiser priors for depth image inpainting. Applied Sciences **9**(6) (2019)
12. Liu, G.: Pytorch implementation of the partial convolution layer for padding and image inpainting. https://github.com/NVIDIA/partialconv (2018)
13. Liu, G., Reda, F.A., Shih, K.J., Wang, T.C., Tao, A., Catanzaro, B.: Image inpainting for irregular holes using partial convolutions. In: European Conference on Computer Vision (2018)
14. Makarov, I., Borisenko, G.: Depth inpainting via vision transformer. In: 2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct). pp. 286–291 (10 2021)
15. Mal, F., Karaman, S.: Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). pp. 1–8 (2018)
16. Ning, W., Li, J., Zhang, L., Du, B.: Musical: Multi-scale image contextual attention learning for inpainting. In: Proc. of the Twenty-Eighth Int. Joint Conf. on Artif. Intell. pp. 3748–3754 (08 2019)
17. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)

18. Ruzic, T., Pizurica, A.: Context-aware patch-based image inpainting using markov random field modeling. IEEE Transactions on Image Processing **24**(1), 444–456 (2015)
19. Satapathy, S., Sahay, R.R.: Robust depth map inpainting using superpixels and non-local gauss-markov random field prior. Signal Processing: Image Communication **98**, 116378 (2021)
20. Shao, M., Zhang, W., Zuo, W., Meng, D.: Multi-scale generative adversarial inpainting network based on cross-layer attention transfer mechanism. Knowledge-Based Systems **196**, 105778 (2020)
21. Shen, L., Hong, R., Zhang, H., Zhang, H., Wang, M.: Single-shot semantic image inpainting with densely connected generative networks. In: Proceedings of the 27th ACM International Conference on Multimedia. pp. 1861–1869. MM '19 (2019)
22. Starck, J.L., Elad, M., Donoho, D.: Image decomposition via the combination of sparse representations and a variational approach. IEEE Transactions on Image Processing **14**(10), 1570–1582 (2005)
23. Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., Lempitsky, V.: Resolution-robust large mask inpainting with fourier convolutions. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 2149–2159 (January 2022)
24. Tao, Y., Popovic, M., Wang, Y., Digumarti, S., Chebrolu, N., Fallon, M.: 3d lidar reconstruction with probabilistic depth completion for robotic navigation. In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 5339–5346 (10 2022)
25. Tschumperle, D., Deriche, R.: Vector-valued image regularization with pdes: a common framework for different applications. IEEE Trans. on Pattern Analysis and Machine Intell. **27**(4), 506–517 (2005)
26. Wongsa, R.: Pytorch implementation of the paper: Image inpainting for irregular holes using partial convolutions. https://github.com/ryanwongsa/Image-Inpainting (2020)
27. Xie, C., Liu, S., Li, C., Cheng, M.M., Zuo, W., Liu, X., Wen, S., Ding, E.: Image inpainting with learnable bidirectional attention maps. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 8857–8866 (10 2019)
28. Yan, Z., Li, X., Li, M., Zuo, W., Shan, S.: Shift-net: Image inpainting via deep feature rearrangement. In: European Conference on Computer Vision (2018)
29. Yeh, R., Chen, C., Lim, T.Y., Schwing, A., Hasegawa-Johnson, M., Do, M.: Semantic image inpainting with deep generative models. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6882–6890 (07 2017)
30. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.: Free-form image inpainting with gated convolution. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4470–4479 (10 2019)
31. Yu, Y., Zhan, F., Wu, R., Pan, J., Cui, K., Lu, S., Ma, F., Xie, X., Miao, C.: Diverse image inpainting with bidirectional and autoregressive transformers. Proc. of the 29th ACM Int. Conf. on Multimedia (2021)
32. Zhang, X., Zhai, D., Li, T., Zhou, Y., Lin, Y.: Image inpainting based on deep learning: A review. Information Fusion **90** (09 2022)
33. Zhang, Y., Funkhouser, T.: Deep depth completion of a single rgb-d image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)