E-Journal of Advanced Maintenance Vol.17 (2025) No.2
Japan Society of Maintenology

Method for Creating Large Datasets for Deep Learning to Improve Image Depth Accuracy

Masahiro MURAYAMA[1,*], Yuki HARAZONO[1], Hirotake ISHII[1], Hiroshi SHIMODA[1], and Yasuyoshi TARUTA[2]

[1] *Graduate School of Energy Science, Kyoto University, Sakyo-ku Yoshidahonmachi, Kyoto-shi, Kyoto 606-8501, Japan*
[2] *Fugen Decommissioning Engineering Center, Japan Atomic Energy Agency, 3 Myojin-cho, Tsuruga-shi, Fukui, 914-8510, Japan*

ABSTRACT

High quality depth images are required for accurate 3D modeling of a facility. However, depth images captured using a typical commercially available RGB-D camera include much noise. Recently, methods using deep learning for depth enhancement have been developed. As described herein, we developed a novel method to create a high-quality dataset by generating high-quality depth images with pixel-wise depth enhancement, which is less affected by camera pose estimation errors. Furthermore, our method improves the quality of the entire dataset by post-processing suitable for our depth enhancement process. Comparison with the dataset created using the existing method showed that datasets created using the proposed method are suitable for training a network for depth enhancement. Depth images taken inside the Fugen Decommissioning Engineering Center are processed by a network trained on the dataset. The network completed the missing areas more correctly and removed the noise while maintaining the detail shapes.

KEYWORDS

dataset creation, deep learning, depth image, noise removal

ARTICLE INFORMATION

Article history:
Received 18 November 2024
Accepted 9 June 2025

1. Introduction

In scientific maintenance, technologies such as environmental measurement and recognition are essential. Among the crucial methods, one involves using RGB-D cameras for environmental measurement. In recent years, depth images, which have information about the distance from the camera to the object at each pixel, have come to be captured inexpensively and easily using RGB-D cameras. However, depth images captured using commodity RGB-D cameras are noisy and have inaccurate and missing values at some pixels. As described in this paper, filling in the missing values and improving the accuracy of the measured values of depth images are designated as "depth enhancement".

Recently, deep learning-based methods have been developed as methods for depth enhancement [1-8]. Self-supervised and unsupervised learning methods for depth enhancement are still less accurate for depth images than supervised learning methods. Supervised learning methods can enhance depth images by learning noise features in advance using a dataset with large amounts of depth images paired with and without noise. Before training a deep learning network, many pairs of depth images must be prepared with much noise and missing values (input images) and corresponding highly accurate depth images (correct images). Existing methods using deep learning often use a special dataset for training the network. It is created by adding noise that simulates the degradation caused by RGB-D cameras on images synthesized by computer graphics (CG). However, those methods are unable to enhance depth images sufficiently because of the noise features of individual RGB-D cameras. To overcome this difficulty, Jeon [2] created a dataset that includes the noise features by enhancing noisy images captured using actual RGB-D cameras. However, Jeon's dataset creation method entails the important difficulties

*Corresponding author, E-mail: murayama@ei.energy.kyoto-u.ac.jp

of producing a low-quality dataset because its depth enhancement process is susceptible to the estimation error of the camera pose.

As described in this paper, we propose a novel method to create a high-quality dataset based on depth images captured using an actual RGB-D camera for deep learning to enhance depth images. Using the proposed method, a high-quality dataset is created by generating high-quality depth images with pixel-wise depth enhancement method [9], which was developed by the authors for earlier studies. Then additional post-processing is applied. The depth images taken for evaluation inside the Fugen Decommissioning Engineering Center of the Japan Atomic Energy Agency (JAEA) are processed by a network trained on the dataset.

The salient contributions of this paper are presented below.

- We demonstrate that the depth enhancement method [9] is more suitable for creating a dataset for deep learning to enhance depth images than that of existing methods.
- We propose post-processing for dataset creation that is suitable for our depth enhancement process.
- We apply the proposed method to depth images of actual sites and demonstrate that it is more effective than existing methods.

This paper comprises six chapters, including the Introduction. Chapter 2 describes existing studies of dataset creation methods. Chapter 3 presents the proposed dataset creation method. Chapter 4, we compare the generated dataset with that of the existing dataset creation method. In Chapter 5, we train a network with a dataset created using the existing and the proposed dataset creation method respectively and compare the network outputs. Finally, Chapter 6 presents conclusions of this study and expectations for future work.

2. Related Works

In many existing studies, datasets used to train networks for depth enhancement are created by processing the Middlebury Stereo Dataset [10-13], which is a representative depth image dataset. The Middlebury Stereo Dataset is a dataset originally used to evaluate the performance of stereo matching algorithms. It includes the corresponding true depth images along with RGB images. As an example of a dataset for deep learning from Middlebury Stereo Dataset, Lu et al. created an input image by adding noise to a depth image from the datasets to simulate degradation caused by an RGB-D camera [14]. However, the noise added by this method does not fully simulate the noise features caused by an actual RGB-D camera. Therefore, the inference results are often insufficiently enhanced when this method is used as a dataset for deep learning to process depth images captured in an actual environment. In addition, the Middlebury Stereo Dataset contains only 95 images, which is insufficient to be used for training the network. The number is very small compared to typically used datasets used for deep learning, which can include tens of thousands of images.

To train the features of noise caused by actual RGB-D cameras, some methods create datasets from depth images captured using actual RGB-D cameras, such as those included in the NYU Depth Dataset [15] and ScanNet Dataset [16]. As a typical method, Jeon uses the mesh and camera pose obtained by BundleFusion [17] from the depth images of ScanNet Dataset, generates rendered depth images, and post-processes them to produce correct images for the dataset [2]. BundleFusion consists of two processes: camera pose estimation and mesh generation. However, the camera pose obtained by BundleFusion includes errors, which tend to misalign the object in the image considerably from its original position during rendering. These misalignments around object boundaries can cause incorrect learning, leading to unstable learning.

By contrast, Xian propose a method for creating datasets aimed mainly at complementing missing depth images [18]. Using this method, depth images with small areas of missing areas are selected from the NYU Depth Dataset. By repeatedly applying the expansion operator to the missing areas, depth images with different sizes of missing areas are created and used as input images. Furthermore, correct images are created by filling in small missing areas of selected images. However, the missing areas created using this method differ in location and size from those created using an actual RGB-D camera. Moreover, only images with a small missing area are useful. Therefore, only about 0.3% of the NYU Depth Dataset is available, which not only leads to a small number of images of the dataset, but also biases the captured targets, making it unsuitable for deep learning for depth enhancement.

3. Dataset Creation Method based on Pixel-Wise Depth Enhancement

The dataset creation method proposed herein generates highly accurate depth images from captured depth images using a method that is less susceptible to errors in camera pose estimation. To use these highly accurate depth images as the correct images for the dataset, a new post-processing must be used.

This chapter presents the processing flows of the existing and proposed methods. Then the depth enhancement and post-processing of the proposed dataset creation method are explained, as well as differences between the depth enhancement and post-processing of Jeon's method and the proposed method.

3.1. Overview of dataset creation method

Dataset creation using the depth images captured using an actual RGB-D camera in the proposed method consists of a depth enhancement process for the depth images using our depth enhancement method [9] and post-processing. Deep learning-based depth enhancement methods input a single depth image, whereas our depth enhancement method [9] requires multiple depth images taken consecutively as input. By reducing random errors for each pixel, our depth enhancement process creates highly accurate depth images from captured images. Although this pixel-wise depth enhancement method is unsuitable for real-time processing because of its slow processing, no difficulty arises even if creating a dataset takes time. However, even when high-precision processing is applied, the accuracy might remain not enough, or pixels might be missing in some cases. Therefore, as a post-processing step, a higher quality dataset is created by removing missing areas or insufficiently accurate depth images after depth enhancement has been applied.

The flows of dataset creation for Jeon's method [2] and the proposed method are portrayed in Fig. 1. In Jeon's method, BundleFusion is used for depth enhancement. It is a type of SLAM that processes images sequentially. However, for this study, to prioritize improvement of the accuracy of the resulting depth images, we apply the depth enhancement method [9] we developed in an earlier study.

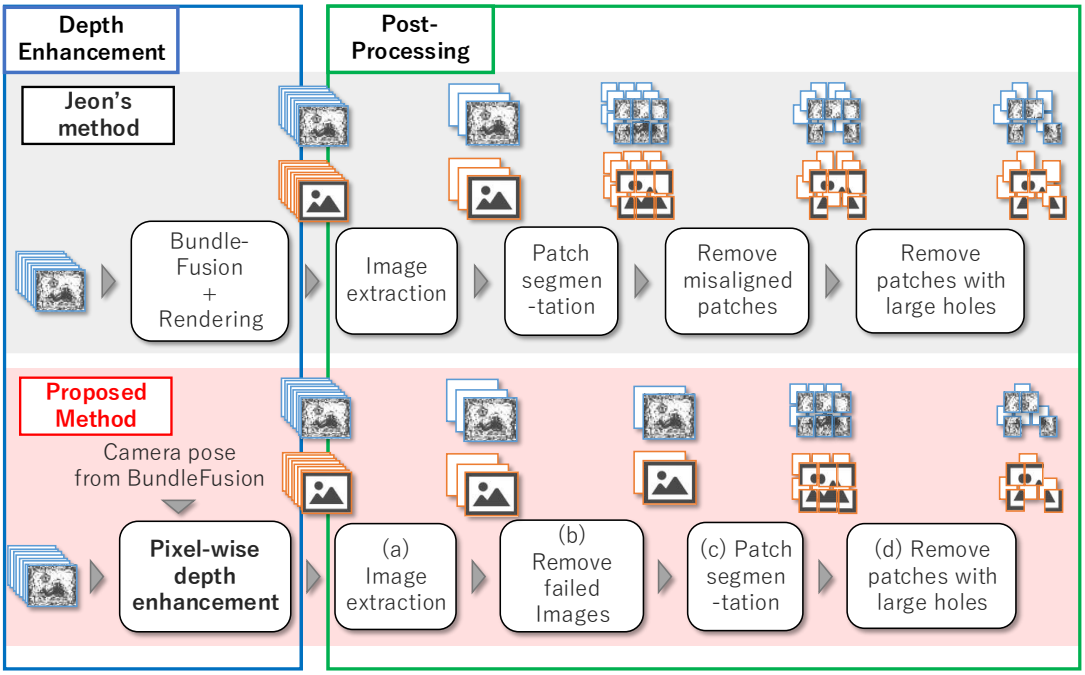

Fig. 1. Flows of existing and proposed methods for creating datasets

Jeon's method requires image extraction, patch segmentation, removal of patch pairs with misaligned edges, and removal of patches with large missing areas as post-processing after depth enhancement. For the removal of patch pairs with misaligned edges, the Structural Similarity Index

Measure (SSIM) [19] is calculated for the patch pairs. Patch pairs with low SSIM are removed from the dataset. By contrast, we propose to remove depth enhancement failed images before patch segmentation. For the removal of the depth enhancement failure images, the SSIM of the entire image, not the patch, is calculated. Then image pairs with low SSIM are removed from the dataset. The post-processing of Jeon's method specifically examines the local structure to remove misaligned edge patches. However, this removal might also remove patch pairs that are suitable for training and which contain misaligned edges because of correct depth enhancement. Therefore, post-processing of the proposed method specifically examines the overall image structure to remove low-quality depth images.

3.2. Details of the dataset creation method

For the proposed dataset creation method, our depth enhancement method [9] is first applied to all captured images using the camera poses. In this paper, the camera poses are results obtained from BundleFusion in accordance with Jeon's method to compare the depth enhancement of the proposed method and Jeon's method, but other methods such as COLMAP [20-21] can be used. Our depth enhancement method can generate highly accurate depth images by reducing random noise for each pixel of depth images using multiple captured depth images and camera poses. It is noteworthy that this method is less affected by the estimation error of the camera pose, making it more robust than existing methods for depth enhancement.

After depth enhancement, (a) image extraction is applied to the depth images with our depth enhancement. The images are used for dataset creation. Images used as input for our depth enhancement [9] must be taken consecutively so that adjacent images are mutually similar. With deep learning, the learning efficiency might worsen if the dataset includes many similar depth images. Therefore, images are extracted at regular intervals for the captured images and for images with depth enhancement. For this study, we used a method of extracting an image from every 40 images, in accordance with Jeon's method.

Then, (b) removal of failed images for depth enhancement is executed. If a large error is included in the estimation of camera pose, then the accuracy might not be improved correctly. The depth enhancement process might actually increase noise. These images are removed from the dataset because they adversely affect training. Specifically, SSIM is calculated for the image pair. Then images with a small SSIM are removed as failed depth enhancement images. SSIM, a metric used to evaluate the similarity between images, is calculated using Equation (1).

$$SSIM(\boldsymbol{x}, \boldsymbol{y}) = \frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)} \quad (1)$$

In the equation above, $\boldsymbol{x}$ and $\boldsymbol{y}$ respectively denote the two images to be evaluated. In this case, they are images before and after the depth enhancement. Also, μ_x and μ_y are the respective averages of pixel values of $\boldsymbol{x}$ and $\boldsymbol{y}$, and σ_x and σ_y respectively represent the standard deviations of pixel values of $\boldsymbol{x}$ and $\boldsymbol{y}$. σ_{xy} stands for the covariance of pixel values of $\boldsymbol{x}, \boldsymbol{y}$. In addition, C_1 and C_2 are constants, defined respectively as $C_1 = (K_1L)^2$ and $C_2 = (K_2L)^2$, where L is the maximum value of pixels, using constants $K_1, K_2 \ll 1$. Also, $K_1 = 0.01$ and $K_2 = 0.03$. When calculating SSIM between images, SSIM is usually calculated for each small area (window) in the image. The average of SSIM for all windows in one image pair is calculated while shifting the window. That is, the SSIM for the entire image is calculated using Equation (2).

$$SSIM_N(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{N}\sum_{i=1}^{N} SSIM(\boldsymbol{x_i}, \boldsymbol{y_i}) \quad (2)$$

In that equation, $\boldsymbol{X}$ and $\boldsymbol{Y}$ respectively stand for the image pairs before and after depth enhancement. Also, N represents the number of times SSIM is calculated for the entire image (number of windows). SSIM facilitates comparison of the differences in image intensity, contrast, and structure. The closer the SSIM value is to 1, the more similar the two images are from all three perspectives. Therefore, it is expected that the SSIM values are larger among images for which the camera pose has been estimated

successfully and for which the depth enhancement process has been applied appropriately. In the proposed method, the window size is set as 8×8. Then SSIM is calculated by shifting the window by 8 pixels vertically and horizontally on the image. As a criterion for failure to achieve depth enhancement, image pairs with SSIMs of 0.95 or fewer between images before and after depth enhancement are removed from the dataset.

However, the depth image obtained at this stage might also include missing pixels. One possible method which could be used is to remove the correct images with missing pixels using only the images with no missing pixels in the dataset or by application of an additional completion process. However, the former of those methods reduces the number of images in the dataset. The latter method does not always lead to correct values. Therefore, (c) patch segmentation is used to divide the image into small regions (patches). Also, (d) removal of patches with large missing pixels is used to remove depth images containing large missing pixels. This removal allows the maximum use of the dataset by selecting only those areas from the depth images which are useful for training. By dividing the image into patches, the resolution of the image input to the network can be reduced, which also reduces memory usage during training. If the network used for training is a convolutional neural network (CNN), which extracts the features of the input image while sliding the kernel, the resolution of the processable input image does not decrease even if the resolution of the input image at training is reduced because of patch segmentation. In Jeon's post-processing, if the missing areas of either the enhanced patch or the captured patch are large, then the patch pair is removed from the dataset. However, post-processing of the proposed method removes the patch pair from the dataset if the missing areas of only the enhanced patch are large. In the proposed method, patch segmentation is applied with a patch size of 128×128 in accordance with Jeon's method. The enhanced patches which contain missing pixels with a patch size of 10% or more are removed from the dataset. Only the remaining patches are used as the training dataset. During training, to avoid learning missing areas of the depth images after the depth enhancement, a mask is used so that the missing areas are not used in the loss calculation.

4. Example of Dataset Creation and Comparison with Existing Method

In this chapter, to confirm the effects of using the depth enhancement of the proposed method [9] instead of that of Jeon's methods, we compare the results of processing ScanNet Dataset with (1) Jeon's depth enhancement and Jeon's post-processing and (2) the proposed depth enhancement and Jeon's post-processing. The camera pose of the captured images was estimated using BundleFusion in the ScanNet Dataset. The ScanNet Dataset captures a variety of indoor environments commonly seen in everyday life. This dataset includes 2.5 million pairs of RGB and depth images from approximately 1,500 scenes, camera pose estimation results from BundleFusion, and mesh data for each scene. From these images, 707 scenes (about 1.15 million images) of training data and 100 scenes (about 210,000 images) of test data are extracted to avoid scene overlap. The respective depth enhancement process and patch segmentation are applied to the training data. Then 337,764 pairs of patches are obtained. Subsequently, by removing patches in the post-processing of Jeon's method for both data, 178,994 pairs of patches are obtained using Jeon's depth enhancement and 214,033 pairs using the proposed depth enhancement.

A histogram of SSIMs for the patch pairs after patch segmentation is presented in Fig. 2. Although Jeon's post-processing removes patches with SSIM less than 0.8, the number of patch pairs with SSIM 0.8 or higher is 271,579 when using Jeon's depth enhancement and 305,173 when using the proposed depth enhancement. The patch pairs with larger SSIM values after patch segmentation using the proposed depth enhancement are more numerous than those of Jeon's depth enhancement, which demonstrates that the structural features of the original image are maintained. In Jeon's refinement process, the depth images are generated by rendering from the mesh data obtained by BundleFusion and the camera pose. Because of the errors in the camera pose obtained by BundleFusion, there are errors in the mesh shape. Also, the object in the image during rendering is likely to misalign considerably from its original position, leading to a smaller value of SSIM. In contrast, the proposed depth enhancement calculates depth values directly from some of the multiple depth images captured in sequence. Therefore, even if errors are included in the camera pose, they are less likely to be affected by the errors if they are similarly included in the successive images.

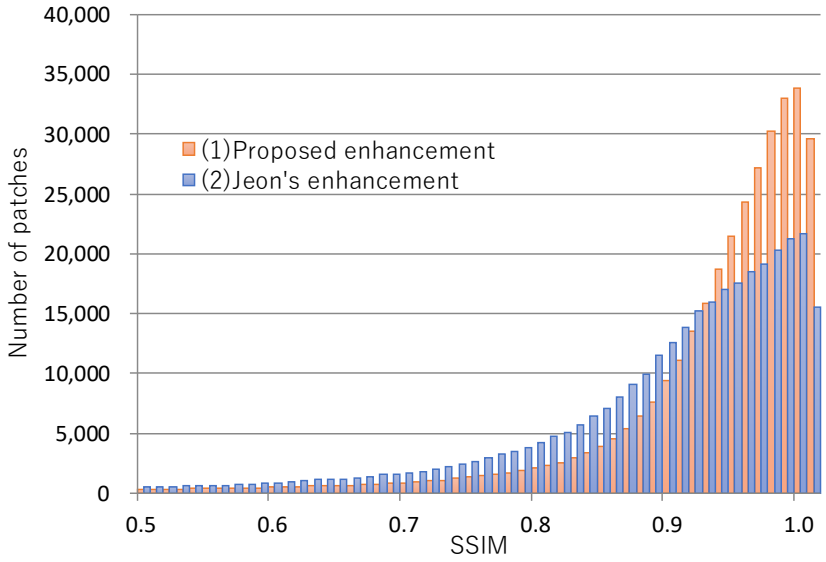

Fig. 2. Histogram of SSIM for noisy and clean patches using Jeon's and the proposed method

To ascertain whether the depth enhancement can maintain the original edge structure, a histogram of the amount of edges in the enhanced patches after patch segmentation is presented in Fig. 3. The histograms in Fig. 3 are obtained by application of the following process to enhanced patches after patch segmentation.

1. For edges detected using the Laplacian filter, the edge values are squared.
2. Edge dilation is applied using a kernel with 5×5 size.
3. The edge image is binarized using Otsu's binarization method [22].
4. Edges generated by the missing areas are removed.

Depth images have clear discontinuities along the edges between the object and the region behind it. Consequently, the edges should be sharp, but the depth enhancement can blur the edges. If the edges in the enhanced patch of the dataset are blurred, then the edges are also blurred in the processing of the trained network. The edge amount is the number of pixels in the edge image obtained by edge detection for the depth image. If the edges are blurred, then the edge amount is smaller. From Fig. 3, results demonstrate that the number of enhanced patches with a large amount of edges is larger. Consequently, sharper edges are retained when using the proposed depth enhancement than when using Jeon's depth enhancement, even though both are processed from the same depth images.

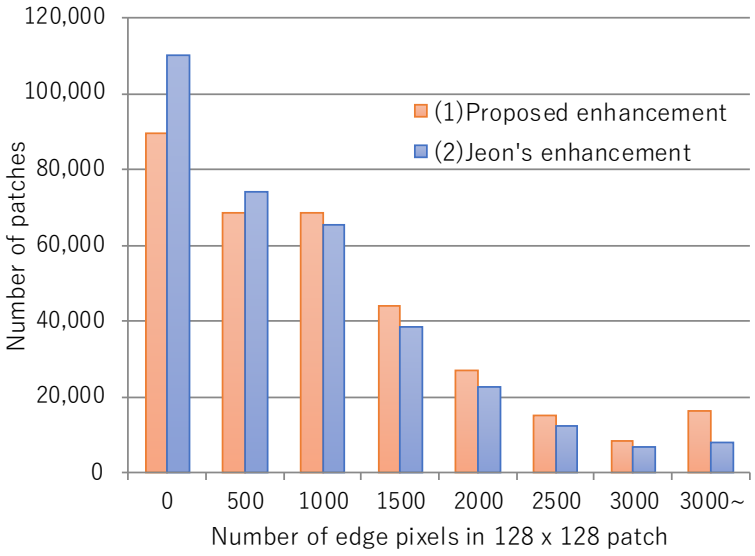

Fig. 3. Histogram showing the amount of edge in clean patches for Jeon's and the proposed method

Next, examples of patch pairs processed by each depth enhancement are depicted in Fig. 4. Depth images in the figure are displayed with contrast adjusted for each image to improve visibility. As shown in the depth image of the red box and the normal image in Fig. 4, the edges are blurred and details of the object are lost in Jeon's depth enhancement. In Jeon's depth enhancement, mesh data are created once. Therefore, small edges in the depth dimension are readily collapsed. The corners of objects are easily rounded. However, the proposed depth enhancement is able to retain even small edges and corners of objects. The patch using the Jeon's depth enhancement in the blue box in Fig. 4 has an upward misalignment from the captured patch because of the estimation error of the camera pose. This SSIM is 0.79, whereas the patch created using the proposed depth enhancement has less misalignment, with the SSIM of 0.96.

From these results, it can be said that the quality of the dataset is higher using the proposed depth enhancement than when using Jeon's depth enhancement.

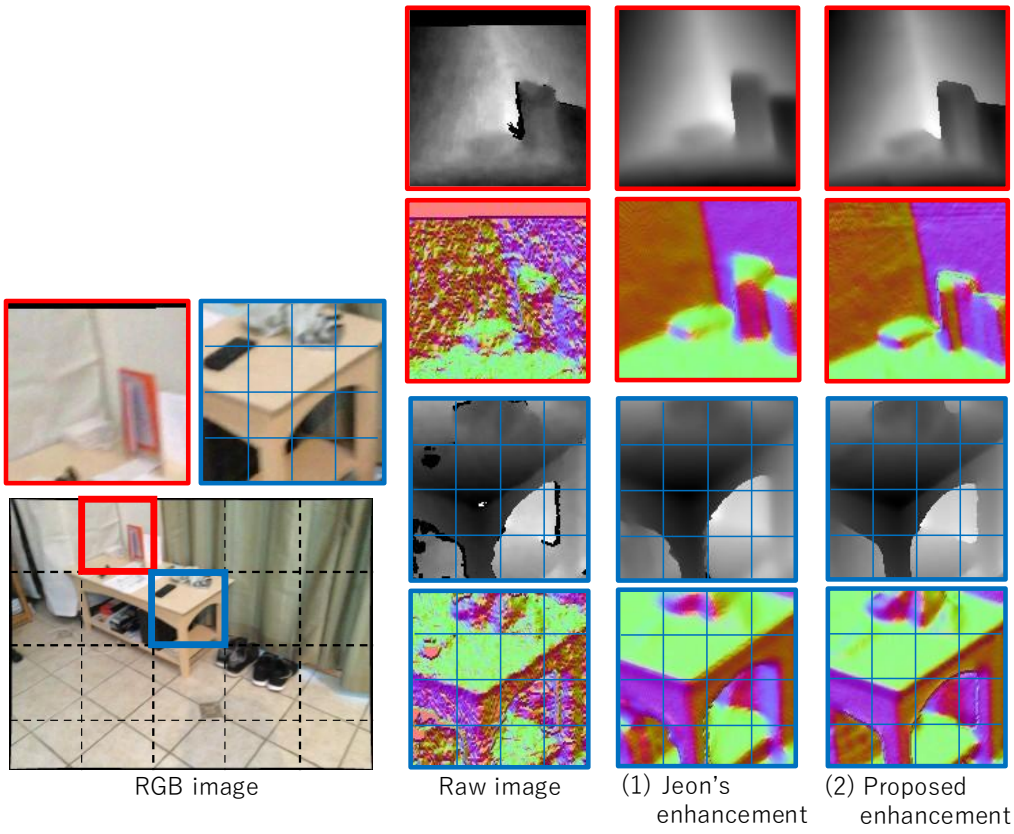

Fig. 4. Examples of patches split along black dashed line in the RGB image

5. Comparison of Training Results

As described in this chapter, after we train the same network using the following three datasets, we compare the results of depth enhancement for the networks.

(1) Dataset created using Jeon's depth enhancement and Jeon's post-processing.
(2) Dataset created using the proposed depth enhancement [9] and Jeon's post-processing.
(3) Dataset created using the proposed depth enhancement [9] and the proposed post-processing described in Chapter 3.

The target depth images for evaluation are taken inside the Fugen Decommissioning Engineering Center of the Japan Atomic Energy Agency (JAEA). This environment is different from one of the training datasets. Quantitative evaluations such as root mean squared error (RMSE) and peak signal-to-noise ratio (PSNR), which directly compare the processed image to the true image, are often used as accuracy measures. Nevertheless, it is extremely difficult to obtain depth images with the true values in the captured environment. If the depth images obtained using the proposed depth enhancement are used

as true-value images, then it is clear that training results obtained with the dataset using the proposed depth enhancement are better. Therefore, qualitative evaluation through visual comparison of depth images is conducted. Specifically, the comparison emphasizes the object edges shown by the depth image and the object surface shape shown by the normal image.

Aside from the dataset, the three trainings are the same. That is, the networks and loss functions used for training are Jeon's network (LapDEN) and loss function. The training algorithm is implemented using Python 3.8 and Tensorflow 2.9.0. Also, Adam (beta=0.9) is used as the optimization algorithm and is trained for 300 epochs.

Examples of the processing results from the networks trained on each dataset are presented in Fig. 5. The first row of Fig. 5 shows RGB images of two scenes in the Fugen Decommissioning Engineering Center. Rows two through five show the depth and normal images of the raw images, as well as the results processed using networks trained on each dataset. The Fugen Decommissioning Engineering Center has many pipes and complex shapes, and depth images taken in the facility are noisier than the depth images used for the dataset due to the characteristics of the RGB-D camera. However, the processing results from the networks trained on the dataset (3) are more accurate than the dataset (1) and (2).

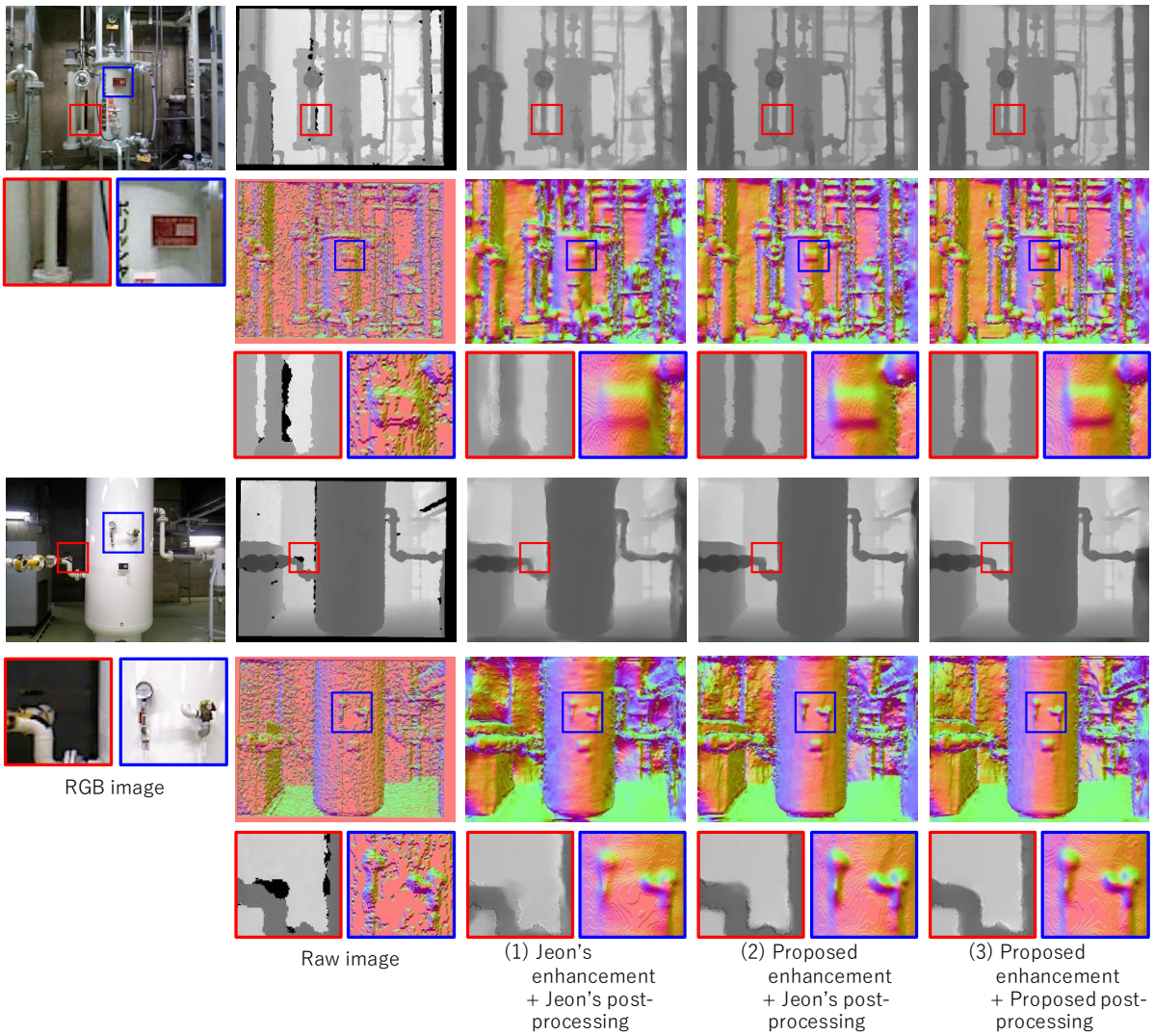

Fig. 5. Visual comparison of depth and normal images of Jeon's method and the proposed method

Examination of the area surrounded by the red box in Fig. 5 shows that the network trained on the dataset (3) completes the missing area of the depth image more accurately than the dataset (1) and (2). Jeon's post-processing removes the patch pair from the dataset if either the enhanced patch or the

captured patch has a large missing area. For that reason, the dataset is less likely to include input patches with large missing areas. The proposed post-processing includes input patches with larger missing areas than those with Jeon's post-processing because the patch pairs are removed from the dataset only when missing areas of the enhanced patches are large. Therefore, the network trained on the dataset created using the proposed method more correctly complete the missing area of the depth image.

The area surrounded by the blue box of the normal images depict that the network trained on the dataset created using the proposed method more accurately maintain the detailed shape while smoothing other surfaces. This finding is attributable to the fact that in Jeon's depth enhancement, details of the object are lost during reconstruction from the depth image by BundleFusion. The original detailed shape is not retained in the correct image.

6. Conclusion

In this paper, as a method to create a high-quality dataset for deep learning that includes the noise features of an actual RGB-D camera, we proposed a novel method to create a dataset based on depth images captured using an actual RGB-D camera. The proposed method generates highly accurate depth images using pixel-wise depth enhancement processing [9], and further improves the quality of the entire dataset by post-processing suitable for the depth enhancement process of the proposed method. Compared to the depth enhancement and post-processing necessary for Jeon's method, a typical existing method, those of the proposed method create a dataset with more patches that correctly completed large missing areas while retaining detailed shapes and accurate edges. We then trained the network on the dataset and applied it to the depth images taken inside the Fugen Decommissioning Engineering Center. We confirmed that the network trained on the dataset created using the proposed method can remove noise while retaining details such as small diameter piping in the facility, compared to the network trained on the dataset created using the existing method.

Acknowledgement

This work was supported by Japan Society for the Promotion of Science KAKENHI Grant Number 23K11166.

References

[1] X. Gu, Y. Guo, F. Deligianni, G. Z. Yang: "Coupled real-synthetic domain adaptation for real-world deep depth enhancement", IEEE Trans. Image Process., Vol. 29, pp. 6343-6356 (2020).
[2] J. Jeon, S. Lee: "Reconstruction-based pairwise depth dataset for depth image enhancement using CNN", Eur. Conf. Comput. Vis., pp 438-454 (2018).
[3] L. Kuan-Ting, L. En-Rwei, Y. Jar-Ferr, H. Li: "An image-guided network for depth edge enhancement", EURASIP Journal on Image and Video Processing, Vol. 2022, No. 6 (2022).
[4] X. Liao, X. Zhang: "Multi-scale mutual feature convolutional neural network for depth image denoise and enhancement", IEEE Vis. Commun. Image Process., pp 1-4 (2017).
[5] V. Sterzentsenko, L. Saroglou, A. Chatzitofis, S. Thermos, N. Zioulis, A. Doumanoglou, D. Zarpalas, P. Daras: "Self-supervised deep depth denoising", IEEE/CVF Int. Conf. Comput. Vis., pp 1242-1251 (2019).
[6] J. Wang, P. Liu, F. Wen: "Self-supervised learning for RGB-guided depth enhancement by exploiting the dependency between RGB and depth", IEEE Trans. Image Process., Vol. 32, pp. 159-174 (2023).
[7] W. Wang, F. Wen, Z. Yan, P. Liu: "Optimal transport for unsupervised denoising learning", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 45, No. 2, pp. 2104-2118 (2023).
[8] J. Zhu, J. Zhang, Y. Cao, Z. Wang: "Image guided depth enhancement via deep fusion and local linear regularizaron", IEEE Int. Conf. Image Process., pp. 4068-4072 (2017).
[9] M. Murayama, T. Higashiyama, Y. Harazono, H. Ishii, H. Shimoda, S. Okido, Y. Taruta: "Depth image noise reduction and super-resolution by pixel-wise multi-frame fusion", IEICE Trans. Inf. Syst., Vol. E105.D, No.6, pp. 1211-1224 (2022).
[10] H. Hirschmuller, D. Scharstein: "Evaluation of cost functions for stereo matching", IEEE Conf. Comput. Vis. Pattern Recognit., pp. 1-8 (2007).
[11] D. Scharstein, R. Szeliski: "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms", Int. J. Comput. Vis., Vol. 47, pp. 7-42 (2002).

[12] D. Scharstein, R. Szeliski: "High-accuracy stereo depth maps using structured light", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1, pp. 195-202 (2003).
[13] D. Scharstein, C. Pal: "Learning conditional random fields for stereo", IEEE Conf. Comput. Vis. Pattern Recognit., pp. 1-8 (2007).
[14] S. Lu, X. Ren, F. Liu: "Depth enhancement via low-rank matrix completion", IEEE Conf. Comput. Vis. Pattern Recognit., pp. 3390-3397 (2014).
[15] N. Silberman, D. Hoiem, P. Kohli, R. Fergus: "Indoor segmentation and support inference from RGBD images", Eur. Conf. Comput. Vis., pp. 746-760 (2012).
[16] A. Dai, A. Chang, M. Savva, M. Halber, T. Funkhouser, M. Nießner: "Scannet: richly-annotated 3D reconstructions of indoor scenes", IEEE Conf. Comput. Vis. Pattern Recognit., pp. 2432-2443 (2017).
[17] A. Dai, M. Nießner, M. Zollhofer, S. Izadi, C. Theobalt: "Bundlefusion: real-time globally consistent 3D reconstruction using on-the-fly surface reintegration", ACM Trans. Graph. Vol. 36, No. 4 (2017).
[18] C. Xian, D. Zhang, C. Dai, C.C.L. Wang: "Fast generation of high-fidelity RGB-D images by deep learning with adaptive convolution", IEEE Trans. Autom Sci. Eng., Vo. 18, No. 3, pp. 1328-1340 (2021).
[19] W. Zhou, A.C. Bovik, H.R. Sheikh and E.P. Simoncelli: "Image quality assessment: from error visibility to structural similarity", IEEE Trans. Image Process., Vol. 13, No. 4, pp. 600-612 (2004).
[20] J.L. Schönberger, J.M. Frahm: 'Structure-from-motion revisited", Conf. Comput. Vis. Pattern Recognit., pp. 4104-4113 (2016).
[21] J.L. Schönberger, E. Zheng, M. Pollefeys, J.M. Frahm: "Pixel-wise view selection for unstructured multi-view stereo", Eur. Conf. Comput. Vis., pp. 501-518 (2016).
[22] N. Otsu: "A threshold selection method from gray-level histograms", IEEE Trans. Syst. Man. Cybern., Vol. 9, No. 1, pp. 62-66 (1979).