Reconstruction-based Pairwise Depth Dataset for Depth Image Enhancement Using CNN

Junho Jeon and Seungyong Lee

POSTECH
{zwitterion27, leesy}@postech.ac.kr

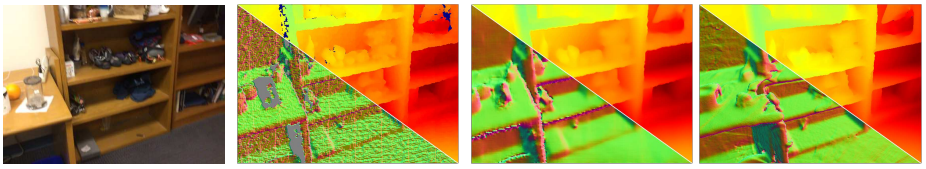
Abstract. Raw depth images captured by consumer depth cameras suffer from noisy and missing values. Despite the success of CNN-based image processing on color image restoration, similar approaches for depth enhancement have not been much addressed yet because of the lack of raw-clean pairwise dataset. In this paper, we propose a pairwise depth image dataset generation method using dense 3D surface reconstruction with a filtering method to remove low quality pairs. We also present a multi-scale Laplacian pyramid based neural network and structure preserving loss functions to progressively reduce the noise and holes from coarse to fine scales. Experimental results show that our network trained with our pairwise dataset can enhance the input depth images to become comparable with 3D reconstructions obtained from depth streams, and can accelerate the convergence of dense 3D reconstruction results.

Keywords: depth image dataset, depth image enhancement, 3D reconstruction, deep learning, Laplacian pyramid network

1 Introduction

With consumer RGB-D cameras, e.g., ASUS Xtion [2] and Occipital Structure sensor [34], depth images can be easily captured and have been utilized for improving the performance of vision algorithms, such as 3D reconstruction [32, 33, 7], object recognition [3, 11], and semantic segmentation [26, 14, 39, 6]. Nevertheless, the quality of depth images from those hand-held consumer RGB-D cameras is still limited because their important design goal was speed rather than precise acquisition of 3D geometry. The captured depth images suffer from heavy noise and missing values, due to physical limitations of the sensors and low processing power (Fig. 1b).

Several image processing methods have been developed for depth image enhancement. As the quality of concurrently captured RGB image is relatively better than the depth image, exploiting the correlation between color and geometry information, called sensor fusion, was investigated, mainly with local filter-based methods [37, 38, 47]. However, a single degraded depth image contains only partial information of the scene geometry, and previous single image-based methods have limited capability especially in resolving heavy noise and missing values.



(a) color image (b) noisy input depth (c) our result (d) 3D reconstruction

Fig. 1. Depth enhancement of a low quality raw depth image. In (b-d), the top right and bottom left parts show the depths and surface normals, respectively. Visualization of the normals clearly shows small but oscillating noise.

Recent advances of image processing using deep learning have achieved impressive improvements on color image enhancement and restoration, such as single image super-resolution (SISR) [23, 25], blur removal [45, 30] and image completion [36, 18]. In those techniques, deep learning networks are optimized using large datasets to automatically extract useful features and combine them to produce desired outputs. However, deep learning based depth image enhancement has not been actively researched so far due to the lack of a suitable dataset.

In the case of color image restoration, a network can be trained using a self-supervised dataset that can be easily generated by degrading high-quality images [1]. In contrast, a RGB-D camera can capture only low-quality depth images, and a self-supervised dataset for depth image enhancement cannot be built in the same way as for color image restoration. As a result, a large-scale dataset that enables deep learning based approaches for depth image enhancement has not been made available yet.

In this paper, we present a large-scale pairwise depth dataset that consists of noisy raw depth images and the corresponding clean depth images. To construct the raw-clean depth image pairs, we utilize dense 3D reconstruction from a RGB-D stream to estimate the clean and complete scene geometry. For an input raw depth image, we generate the corresponding clean depth image by rendering the reconstructed 3D scene at the estimated camera position. During the process, structure-based image similarity [42] is measured to filter out low quality depth image pairs caused by misalignments of camera positions and slight mismatches of exact scene geometry. This filtering effectively increases the quality of our dataset for depth image enhancement.

Using the dataset, we train a Laplacian pyramid based neural network to obtain a clean depth image from a given raw depth image. We introduce a gradient-based structure loss function to effectively preserve depth discontinuities around object boundaries. Our network can progressively reduce the noise and holes in the input by producing intermediate clean depth images from coarse to fine scales.

In the experiments, we show that our network trained with our dataset significantly reduces the noise and holes from a raw depth image, while preserving the desired discontinuities, e.g., between the foreground objects and the background. As an application of depth image enhancement, we demonstrate that the

convergence of dense 3D surface reconstruction can be drastically accelerated by pre-filtering the input depth stream with our enhancement method.

Our main contributions can be summarized as follows:

- we generate a large-scale raw-clean pairwise depth image dataset that can be used for supervised learning of depth image enhancement, by applying the state-of-the-art dense 3D surface reconstruction on RGB-D streams.
- we propose a deep Laplacian pyramid network with multi-scale skip connections for depth image enhancement which reduces noise and holes in a cascaded manner.
- Our loss functions for training the network enable original geometric structures to be preserved during depth image enhancement, and the property helps accelerating the convergence of dense 3D surface reconstruction.

2 Related Work

2.1 Depth image enhancement

The most common approach to refine a low quality depth image from a RGB-D camera is to incorporate the concurrently captured high quality color image. Besides the conventional joint bilateral filtering based methods [5, 24], various approaches have been tried to exploit the correlation between color and geometry. For example, low-rank matrix completion [29], multi-scale sparse representation learning [22], shape from shading [46, 43], and analysis representation model [13] have been used for depth map refinement.

Another line of researches to improve the depth image quality is depth image super-resolution. Similar to depth enhancement, high-resolution color images [28], dictionary learning [19, 41], and shape from shading [15] have been used. Although these techniques can enhance the quality of depth images from a consumer RGB-D camera, their main goal is to increase the spatial resolution, rather than noise reduction or hole filling.

2.2 CNN-based image processing

Convolutional neural network (CNN) based image processing methods have shown great performance on various problems, covering from low-level image restoration, such as single image super resolution [10, 25, 23] and image deblurring [45, 30], to high-level tasks, such as image completion [36, 18] and image generation [9]. Their successes are based on development of novel network architectures [16, 12] and availability of large-scale training datasets [8, 27].

In contrast, deep learning has not been intensively applied to depth image processing mainly due to the lack of large-scale training datasets. Recently, Hui et al. [17] proposed a CNN-based depth map super-resolution method. Their multi-scale guided network can upscale depth maps with high resolution color guidance images, but the network as well as the dataset cannot be directly used for enhancing depth images captured by consumer RGB-D cameras.

2.3 Dense 3D reconstruction and RGB-D dataset

Our approach for constructing a pairwise depth dataset utilizes a dense 3D reconstruction method and a large-scale RGB-D dataset. Based on the pioneering work of KinectFusion [32], a few following works have been proposed. Nießner et al. [33] drastically reduced the memory consumption for reconstruction using a sparse hash data structure, enabling large-scale reconstruction of an entire room or a whole floor. Dai et al. [7] proposed the BundleFusion algorithm that uses additional color features for registration and global bundle adjustment for obtaining more precise scene geometry.

By capturing depth streams using consumer RGB-D cameras, several RGB-D image dataset have been published for computer vision tasks. SUN RGB-D dataset [39] consists of 10K images with manually annotated semantic information. ScanNet dataset [6] contains 2.5 million images from more than 1500 scans. In our dataset construction, we use the ScanNet dataset [6], as it provides raw RGB-D streams and the corresponding scene geometries reconstructed using the state-of-the-art BundleFusion algorithm [7].

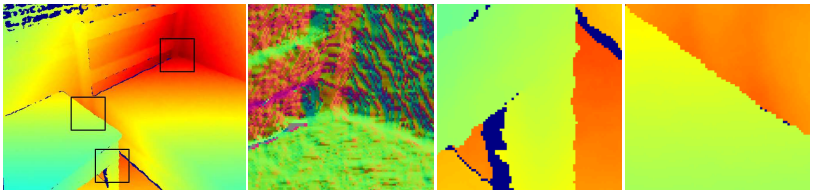
Concurrently and independently from our work, Zhang et al. [49] presented a pairwise depth image dataset generated using 3D reconstruction from a RGB-D stream. However, in contrast to our work, they mainly focused on estimating large unobserved depth values rather than removing noise and holes from low quality RGB-D images. Moreover, they did not address possible misalignments between the raw input and the rendered depth images caused by inaccurate 3D reconstruction, which should be resolved for effective training of a depth image processing network.

3 Our Approach

In this paper, we mainly address three key issues that should be considered when processing raw depth images captured by consumer RGB-D cameras: depth noise, depth hole, and depth discontinuity.

Depth noise A raw depth image usually contains strong non-uniform noise patterns. Since a RGB-D camera captures 3D geometry by analyzing projected patterns (structured light cameras) or measuring the traveling times of emitted lights (ToF cameras), noise distributions are affected by surface materials and distances from the camera (Fig. 2b). Therefore, a conventional image filter with a fixed kernel size, such as bilateral filter [40], would not be enough for processing a variety of noise. Instead, in our work, we use a deep CNN that can adaptively handle the noise by extracting multi-scale features from a given depth image.

Depth hole Similar to the depth noise, physical limitation of a RGB-D camera causes missing depth values, called holes. These holes are usually found around object boundaries, because of the visibility differences between the light emitter and the image sensor (Fig. 2c). In addition, too shiny or light-absorbing parts can



(a) raw depth image (b) noise (c) holes (d) discontinuity

Fig. 2. Key factors that lower the quality of a raw depth image. (a) a raw depth image, (b) spatially varying depth noise (visualized using surface normals), (c) depth holes near the edges (blue regions), (d) depth discontinuity at the object boundary.

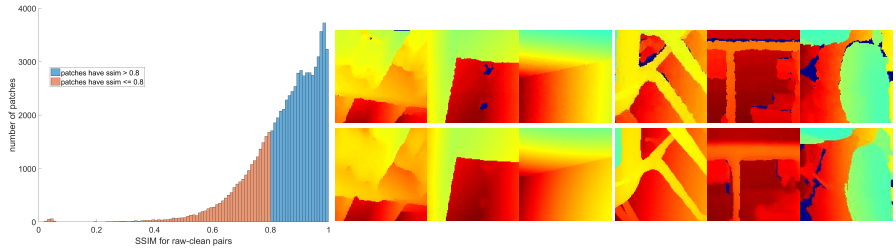
cause missing depth values. Predicting the missing values requires understanding of the local and global contexts of the input scene. To enable such prediction, our network architecture progressively enhances the depth image from coarse (1/4 size of the input) to fine scales.

Depth discontinuity The values in a depth image have strong discontinuities along depth edges (Fig. 2d). Unlike a color image having anti-aliased smooth pixels around edges, a depth image should not have anti-aliased depth values obtained by blending the foreground and background depths. Such blended depth pixels would result in small floating fragments of object boundaries between the foreground and the background. In our work, to preserve the original discontinuities in a depth image, we present a gradient-based structure preserving loss that can strongly penalize smoothing of depth edges.

Dataset quality To achieve the enhancement from raw noisy depths to clean depths using supervised learning, the quality of the dataset used for training is very important. Especially, hole filling and discontinuity preserving filtering require exact spatial alignments of geometric features, such as depth edges, for the raw and clean depth image pairs. In the dataset generation process, we check the quality of raw-clean depth image pairs by measuring the structural similarity, and filter out low quality pairs to improve the overall quality of the dataset.

4 Pairwise Depth Dataset Generation

To train a deep neural network for depth image enhancement, we need a large-scale pairwise depth image dataset that consists of raw-clean image pairs. Capturing a scene with a high-precision laser scanner, in addition to a RGB-D camera, could produce smooth and clean depth images for the dataset, but such an approach requires additional hardware. On the other hand, we can obtain clean depth image by rendering synthetically modeled high-quality 3D scenes, but in that case, degrading the rendered depth images to obtain real raw depth images is not straightforward as complex physical interactions between the capture setup and object materials should be reflected in the degradation process.



(a) SSIM histogram (b) good alignment (c) bad alignment

Fig. 3. Structure similarity histogram of the generated patch pairs and example patch pairs from good (blue) and bad (red) alignment sets. top: raw patch, bottom: rendered patch.

In this paper, without using any additional sensor or 3D models, we construct the raw-clean depth image pairs by taking a dense 3D surface reconstruction technique [7]. Given a depth stream, dense 3D surface reconstruction integrates multiple depth images into a single volumetric space. The integration reduces the noise and fills the missing geometry by aggregating geometric information captured at multiple views. Simultaneously, the reconstruction estimates the camera poses of input frames, so we can render the reconstructed geometry using the estimated camera poses to generate clean depth images that correspond to the input noisy depth images.

4.1 3D reconstruction dataset

For successful learning, the generated pairwise depth dataset should cover real world scenes as much as possible. We use ScanNet dataset [6] as the input for 3D reconstruction. ScanNet consists of more than a million of RGB-D images captured from hundreds of scenes. In addition, ScanNet [6] provides high-quality triangular mesh data and the estimated camera poses obtained by BundleFusion [7].

For input depth frames, we render the corresponding clean depth images using the reconstructed triangular mesh and the estimated camera poses. As adjacent frames include lots of overlapping geometry, we only sample a frame per 40 consecutive frames. In addition, we selected 40 scenes from ScanNet, avoiding redundant scene information. As a result, we obtain 4,000 depth image pairs for the dataset in total. Note that 3D scenes often consist of simple primitive shapes such as planes and curves, and are not as much as complex compared to color images. Hence, thousands of well-sampled frames could be enough for our depth enhancement framework. Finally, we slice the depth images into 128×128 patches as the training samples. This is for efficient network training and outlier handling, which will be discussed in the following sections. Fig. 3 shows rendered clean depth patches and the corresponding low quality raw depth patches.

4.2 Misaligned outliers filtering of dataset

Although rendered clean depth images are smooth and contain fewer holes compared to the corresponding raw depth images, there still exists an additional issue to generate a dataset good enough for training a deep neural network for depth image enhancement. BundleFusion [7] shows the state-of-the-art 3D reconstruction, but its camera tracking may contain some errors. Moreover, the geometric integration through a RGB-D stream sometimes misses sharp and thin structures, e.g., chair legs and clothes hangers. These errors may introduce misalignments between the geometry of an input depth image and the corresponding rendered clean depth image (Fig. 3c). Especially, the misalignments become prominent around object boundaries as depth values around depth edges change rapidly and are merged in the volumetric reconstruction. In network learning, these misalignments work as outlier samples and training becomes unstable (Fig. 8a). Consequently, we need a filtering process for the dataset to discard misaligned depth patch pairs.

To discard the misaligned depth patch pairs, we measure the structural similarity (SSIM) [42] between the raw input and corresponding clean label patches. SSIM can effectively measure the structural misalignments between two images. Fig. 3a shows the SSIM histogram of the originally constructed pairwise depth patch dataset. In Fig. 3, we can see that a large portion of patch pairs have low SSIM values caused by misalignments around depth edges and small missing objects. We discard such patch pairs that have SSIM values lower than 0.8, which are about 20% of the original dataset. We also discard incomplete pairs whose raw or clean patch contains a hole larger than 10% of the patch area. After the outlier filtering process, our pairwise depth dataset consists of 56,000 depth patch pairs, where 52,288 pairs are used for training and 3,712 for validation.

5 Laplacian Pyramid Depth Enhancement Network

5.1 Network architecture

As mentioned in Section 3, to handle spatially varying noise and holes by considering local and global contexts, our network progressively reduces the noise and fills the holes from coarse to fine scales. We choose the deep Laplacian pyramid network (LapSRN) as our base network architecture, which was proposed for image super-resolution [23]. LapSRN progressively upsamples the input low resolution image by predicting the residual image for the next finer level in an image pyramid. For more details, please refer the original paper [23].

By modifying LapSRN for depth image enhancement, we propose the deep Laplacian pyramid depth image enhancement network (LapDEN). Fig. 4 shows the overall architecture of LapDEN. Unlike super-resolution that directly upsamples the spatial resolution of a given input image, LapDEN first predicts the clean complete depth image at the coarsest scale, which has a quarter of the original resolution. Then the predicted quarter-sized clean depth image is progressively upsampled through the pyramid to predict the half and original-sized

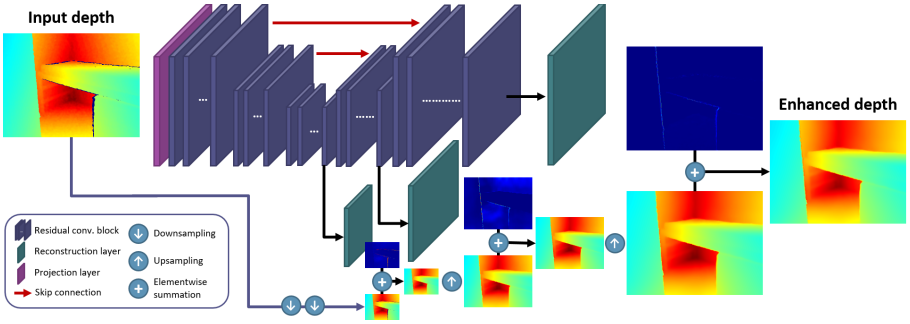


Fig. 4. Laplacian pyramid-based depth image enhancement network (LapDEN).

clean depth images. In addition, the features extracted during the downsampling are passed to the upsampling pyramid with skip connections to prevent loss of the original details in the input depth image during the upsampling.

The overall structure of LapDEN introduces two advantages. First, noise reduction and hole filling become easier when tried at a coarse scale. Downsampling the input depth image naturally reduces noise and holes, and the receptive field size of CNN becomes larger. Then the network can easily learn to predict the clean and complete depth image representing the overall structure of the scene. Second, since we estimate the overall structure and smooth surfaces in the coarse scale prediction, finer scale layers only need to learn to predict the residuals that sharpen depth edge discontinuities and fine details.

Network architecture details As shown in Fig. 4, LapDEN predicts an enhanced depth image through a 3-level image pyramid. After the input depth image is projected onto a 64 channel feature map using a convolutional layer with 7×7 kernels, we extract the multi-level features using a stack of multiple convolutional layers with local residual skip connections. For each level of the image pyramid, a long skip connection directly passes the extracted features to the later corresponding part of the network to enable a fusion of the features extracted in different scales (red arrows).

At the coarsest level, we predict the quarter size residual depth image from the extracted features using an image reconstruction layer. Noise and holes could be almost removed at this level. After that, the features are upsampled and transformed further to predict fine-scale sub-band residuals for the upper levels. We use 20 and 40 convolutional layers for residual blocks at the mid- and high-levels of the pyramid, respectively.

Every convolutional layer except the layers predicting the residuals (i.e., reconstruction layers) has a following leaky rectified linear unit (LReLU) with a negative slope of 0.2. Following the original LapSRN architecture [23], all convolutional layers use 64 filters with size of 3×3. The downsampling and upsampling are performed by convolutional and transposed convolutional layers using 64 filters with size of 4×4.

5.2 Loss functions for training

Our goal is to train a transform function $\hat{\mathbf{y}} = f(\mathbf{x};\theta)$ for estimating an enhanced depth image $\hat{\mathbf{y}}$ from a given noisy raw depth image $\mathbf{x}$ with network parameters θ. Let $\mathbf{y}$ be the ground truth clean depth image corresponding to $\mathbf{x}$. Then our network training is to find the set of parameters θ that minimize the loss function $\sum_{(\mathbf{x},\mathbf{y})\in\mathbf{D}} L(f(\mathbf{x};\theta),\mathbf{y})$ for the training samples in the pairwise depth image dataset $\mathbf{D}$. Our overall loss function is defined as a combination of data loss L_D and structure preserving loss L_S:

$$L(\hat{\mathbf{y}} = f(\mathbf{x};\theta),\mathbf{y}) = L_D(\hat{\mathbf{y}},\mathbf{y}) + 10L_S(\hat{\mathbf{y}},\mathbf{x}). \tag{1}$$

Multi-scale data losses We first define L_D by the $\mathcal{L}_1$ distances of the depth and the depth gradient between $\hat{\mathbf{y}}$ and $\mathbf{y}$ as in usual CNN-based image regression [44, 23]. In addition, we use the $\mathcal{L}_1$ distance of surface normal map between them. Surface normal direction is highly sensitive to the oscillating noise of depth values, so minimizing the surface normal distance is effective for removing small depth noise compared to the previous two measures. Overall, we define the data loss L_D as follows:

$$L_D(\hat{\mathbf{y}},\mathbf{y}) = \frac{1}{N}\sum_p \Big(\rho(\hat{y}_p - y_p) + \lambda_g\rho(\nabla\hat{y}_p - \nabla y_p) + \lambda_n\rho(\hat{n}_p - n_p)\Big), \tag{2}$$

where p is a pixel position of $\hat{\mathbf{y}}$, y_p and n_p is a depth and a surface normal of pixel p, respectively. N is the number of pixels, ∇ is the gradient operator, and $\rho(x) = \sqrt{\|x\|^2 + \epsilon^2}$ is the Charbonnier loss function [4], which is a differentiable form of $\mathcal{L}_1$ norm. We used $\mathcal{L}_1$ norm because of its robustness against to misaligned outlier pairs that might still remain in our training data. λ_g and λ_n are balancing parameters. We set $\lambda_g = \lambda_n = 2$ in our experiments.

Structure preserving loss As we discussed in Section 3, a depth image has clear discontinuity and strong aliasing along the edges between foreground and background regions. Conventional loss functions such as $\mathcal{L}_2$ or $\mathcal{L}_1$ for the depth values can hardly preserve this discontinuity. In this work, we propose a gradient-based structure preserving loss L_S to preserve the original geometric structure and discontinuity of a depth image. Mathematically, depth discontinuity introduces strong gradient magnitudes at the edge pixels. If anti-aliasing or blending happens, *the maximum gradient magnitude around the edge* becomes small as the steep edge is spreaded out to multiple pixels. Based on this observation, L_S is defined as:

$$L_S(\hat{\mathbf{y}},\mathbf{x}) = \frac{1}{N}\sum_p \left(\max_{q\in\Omega(p)}|\nabla\hat{y}_q| - \max_{q\in\Omega(p)}|\nabla x_q|\right)^2, \tag{3}$$

where $\Omega(p)$ is a local window centered at pixel p. L_S calculates the maximum gradient magnitude around pixel p, and measures the distance between those

maximums for $\hat{\mathbf{y}}$ and $\mathbf{x}$. Therefore, minimizing L_S enforces $\hat{\mathbf{y}}$ and $\mathbf{x}$ to have similar depth discontinuity structures. In our experiments, we set $\Omega(p)$ as a 5×5 window for all levels of the image pyramid.

Differently from the previous data loss L_D, L_S uses the input depth image $\mathbf{x}$ as a supervision. As the training sample pairs might not be exactly aligned despite of the dataset filtering, promoting strong discontinuities following the target depth image could increase the ambiguity of the transform to be trained. Instead, we use the input depth image as our supervision to guide the network output to preserve the *original structure* of the given input depth image.

In addition, instead of giving a strong penalty to a misaligned edge, we allow rooms for edge positions by comparing the maximum gradient magnitude around edge pixels. It enables a predicted edge to take structural information from neighboring pixels even if the depth pixel is missing at that position in the input image. As a result, by training the network with the structure preserving loss L_S as well as the data loss L_D, the original depth discontinuity of an input image is effectively preserved while its noisy and missing depth values are significantly enhanced.

6 Experimental Results

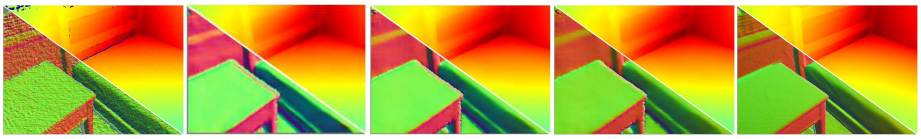
For experiments, we tested our trained network LapDEN on two datasets: ScanNet [6] and NYU-Depth V2 dataset [31]. For the ScanNet dataset, we evaluate our results by comparing them with clean depth images.

6.1 Training details and parameters

LapDEN contains more than 90 convolutional layers and the multi-scale pyramid supervisions are also included in the network. It was hard to train the entire network in a single session. Instead, we used three-stage strategy for training.

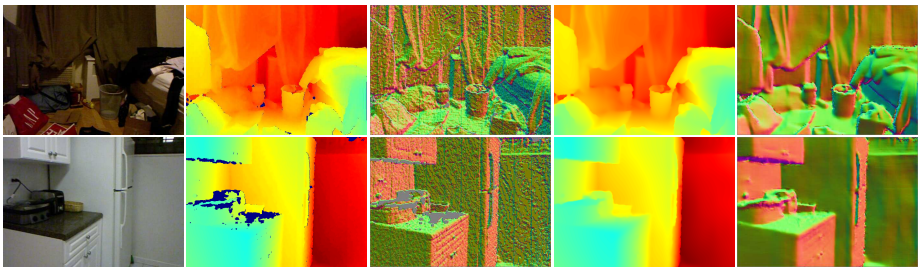
In the first stage, we train the network only with the coarsest level supervision. In other words, the network is trained to predict the overall structure and smooth surfaces by reducing noise and filling the holes in the quarter-sized spatial resolution. After that, we initialize the network with pre-trained parameters for the second stage training. In this stage, we use both the first and second level supervisions to retain the prediction ability for the scene structure at the coarset level. Similarly, in the final third stage, we train the entire network initialized with pre-trained parameters from the second stage using all three levels of supervisions to predict the result in the original spatial resolution.

We used NVIDIA Titan Xp GPU to train the network. We built our framework on the Pytorch library [35]. For optimization, we use Adam optimizer [20] with $\beta = 0.9$. For the first and second stages, we trained the network with a learning rate of 10^{-4} for 30 epochs. For the last stage, we used a learning rate of 10^{-4} for 30 epochs, and then decayed it to 10^{-5} for additional 20 epochs. We used a batch size of 64 for the first and second stages, and 32 for the last stage training.



(a) input image (b) 1st level (c) 2nd level (d) 3rd level (e) target

Fig. 5. Progressive depth enhancement results of our method.



(a) input color (b) input depth (c) input normal (d) our depth (e) our normal

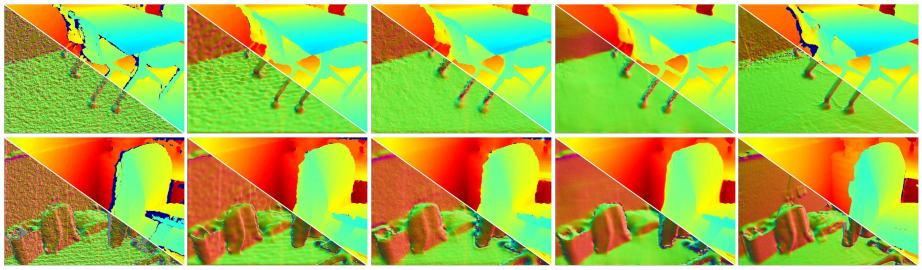
Fig. 6. Depth enhancement results on NYU-Depth v2 dataset [31]

6.2 Enhancement results

Progressive depth image enhancement LapDEN progressively enhances a depth image through the 3-level image pyramid. Fig. 5 shows intermediate enhancement results at the pyramid levels and the target clean depth image. This example demonstrates that depth noise and holes are refined at the coarsest scale, and the details and sharp edges are recovered through the two finer scales. The target clean depth image has been generated by integrating tens of RGB-D frames with a state-of-the-art 3D reconstruction method [7]. LapDEN only takes a single depth image as the input, but still it can produce a clean and sharp depth image which is comparable to the target image. Additional enhancement results are given in Fig. 1.

We also tested our method on depth images from the NYU dataset [31]. Fig. 6 shows that our method significantly reduces the noise of the given raw depth images, and well predicts missing depth values.

Comparison with previous methods Fig. 7 shows comparison results of our method with previous approaches. For the baseline method, we choose the rolling guidance filter (RGF) [48], which was originally proposed for image texture decomposition. Since oscillating depth noise can be treated as a texture pattern, RGF reduces the noise with a few iterations of filtering. We also compared our results with the recent joint filtering method [38], where the mutual-structures among color and depth images are exploited to enhance a depth image. As these two methods are not good at hole filling, before applying the methods, we filled



(a) input image (b) RGF (c) joint filter (d) our result (e) target

Fig. 7. Visual comparison of our results with (b) rolling guidance filter [48] and (c) mutual-structure joint filter [38]. Note that the target images contain holes due to incomplete 3D reconstruction. Additional examples are in the supplementary material.

Table 1. Performance comparison with a test set. We measured SSIM and RMSE between the noisy input depth and the ground truth depth for a baseline.

	Raw input depth	RGF [48]	Joint filter [38]	Ours
SSIM	0.8620	0.9065	0.9152	**0.9229**
RMSE	0.3450	0.2401	0.2360	**0.2148**

the holes with joint bilateral upsampling [21] with the guidance of the corresponding color image.

As shown in Fig. 7, our method outperforms the previous methods on removing the depth noise as well as preserving the depth discontinuity. RGF [48] uses a fixed-size filter kernel, and requires a large kernel size to remove severe depth noise in the regions far from the camera, resulting in loss of geometric structures. The joint filtering method [38] shows relatively good results on reducing the noise but it introduces some waving artifacts around the edges of the desk, which seems to be caused by misaligned color information. The artifacts can be seen more clearly in the surface normal.

For quantitative comparison, we measured the average SSIM and RMSE between the enhancement results and the ground truth depth images on a test set that consists of 355 depth images sampled from a subset of ScanNet scenes [6]. Table 1 shows our results reported the highest performance in the experiment. More comparisons can be found in the supplementary material.

6.3 Component analysis

Dataset filtering As shown in Fig. 8(a), if we do not filter the dataset to remove misaligned depth pairs, the training process becomes unstable and converges to a higher loss compared to the filtered dataset. As a result, the output patches of a network trained without the filtering show blurry and noisy depth values around the edges (Fig. 8c). In contrast, our dataset filtering improves the quality of the dataset, and enables clean and sharp results to be obtained (Fig. 8e).

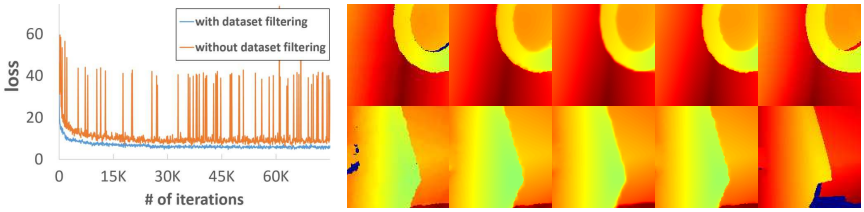


Fig. 8. Component analysis of our network. (a) training loss curves, (b) input patches, (c) without dataset filtering, (d) without L_S, (e) our result, (f) ground truth.

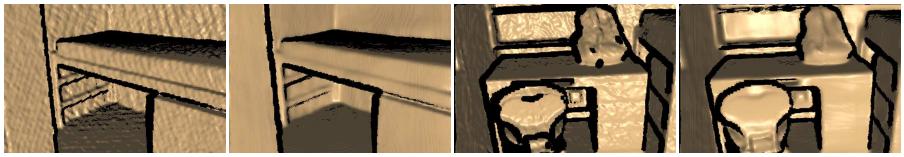
Structure preserving loss To verify the effectiveness of our structure preserving loss L_S, we trained the network only using the data loss L_D with other settings unchanged. As shown in Fig. 8d, although the network can reduce the noise of smooth surface well, it introduces blurry depth boundaries compared to the result of the complete network. These blurry pixels cause 3D floating points in the space around object boundaries, which would act as outliers in the applications of depth images, such as 3D reconstruction.

6.4 Application: 3D reconstruction with pre-filtered depth images

As we described in Section 4, we use dense 3D reconstruction for the pairwise dataset generation. As an application of our depth image enhancement, now we demonstrate that our method can drastically accelerate the convergence of dense 3D surface reconstruction by enhancing the input depth stream.

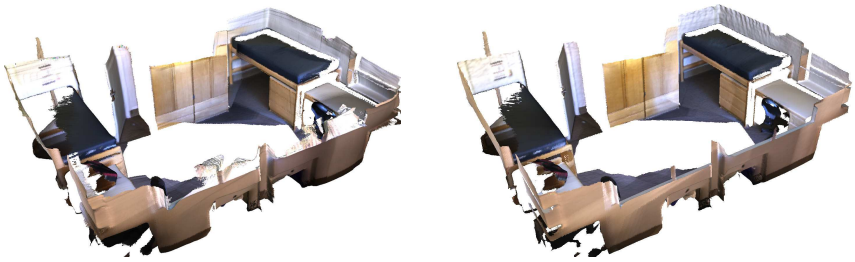
Convergence acceleration of depth integration In dense 3D surface reconstruction, input depth images from multiple viewpoints are integrated to reduce noise and complete the 3D geometry. In this experiment, we pre-filtered the input depth stream using our enhancement method and provided it to a 3D reconstruction method [7]. Fig. 9a shows the results. By integrating only a few frames, we could already obtain a converged smooth surface, which would have needed more frames to be integrated if the raw stream was used. This example shows that the 3D reconstruction process could be made more effective and time-saving with our enhancement method, as we do not need to wait until many frames are integrated to produce smooth surfaces of the scene.

Reconstruction with frame-skipped stream We also show that the pre-filtered depth stream can reconstruct the complete geometry even if we omit every other frame to reduce the frame rate of the original stream into a half. As shown in Fig. 9b, the reconstructed mesh has not been degraded with the frame-skipped stream, due to the accelerated convergence of depth integration with our enhancement method. This experiment implies that we can move the RGB-D camera twice faster than usual while still preserving the quality of 3D reconstruction using our depth image enhancement method.



Raw 5 frames Filtered 5 frames Raw 5 frames Filtered 5 frames

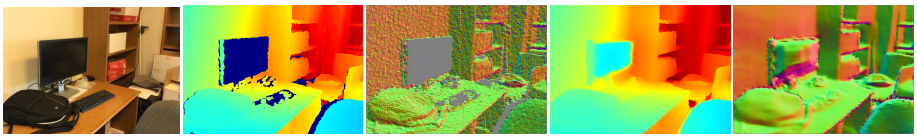
(a) Convergence acceleration



Frame-skipped pre-filtered stream Full raw stream

(b) Reconstruction with a frame-skipped stream

Fig. 9. 3D reconstruction experiments with pre-filtered depth images.



input image input depth input normal output depth output normal

Fig. 10. Failure case of the proposed algorithm.

7 Conclusions

This paper presented a pairwise depth image dataset generation method, which is based on dense 3D surface reconstruction from a RGB-D stream. We also presented a Laplacian pyramid-based neural network and gradient-based structure preserving loss for depth image enhancement. With experiments, we demonstrated that our method can produce clean and sharp depth images from raw depth images, which can be utilized for accelerating 3D reconstruction process.

There remain few limitations of our method. First of all, the speed is not real-time, and the method cannot be applied for real-time applications. In addition, depth holes larger than the patch size used for network training may not be clearly recovered (Fig. 10). Resolving these limitations with more light-weighted and advanced network structures would be interesting future work.

Acknowledgements We appreciate the constructive comments from the reviewers. This work was supported by the Ministry of Science and ICT, Korea, through IITP grant (IITP-2015-0-00174), Giga Korea grant (GK18P0300), and NRF grant (NRF-2017M3C4A7066317).

References

1. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 1122–1131 (2017)
2. Asus Xtion Pro Live: `https://www.asus.com/3D-Sensor/Xtion_PRO_LIVE/`
3. Blum, M., Springenberg, J.T., Wülfing, J., Riedmiller, M.: A learned feature descriptor for object recognition in rgb-d data. In: Proc. IEEE International Conference on Robotics and Automation (ICRA). pp. 1298–1303 (2012)
4. Bruhn, A., Weickert, J., Schnörr, C.: Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. International Journal of Computer Vision (IJCV) **61**(3), 211–231 (2005)
5. Chen, L., Lin, H., Li, S.: Depth image enhancement for kinect using region growing and bilateral filter. In: Proc. International Conference on Pattern Recognition (ICPR). pp. 3070–3073 (2012)
6. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2432–2443 (2017)
7. Dai, A., Nießner, M., Zollhöfer, M., Izadi, S., Theobalt, C.: Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. ACM Transactions on Graphics (ToG) **36**(3), 24 (2017)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 248 – 255 (2009)
9. Denton, E.L., Chintala, S., Fergus, R., et al.: Deep generative image models using a laplacian pyramid of adversarial networks. In: Proc. Advances in Neural Information Processing Systems (NIPS). pp. 1486–1494 (2015)
10. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: Proc. European Conference on Computer Vision (ECCV). pp. 184–199 (2014)
11. Eitel, A., Springenberg, J.T., Spinello, L., Riedmiller, M., Burgard, W.: Multimodal deep learning for robust rgb-d object recognition. In: Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 681–687 (2015)
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Proc. Advances in Neural Information Processing Systems (NIPS). pp. 2672–2680 (2014)
13. Gu, S., Zuo, W., Guo, S., Chen, Y., Chen, C., Zhang, L.: Learning dynamic guidance for depth image enhancement. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 712–721 (2017)
14. Gupta, S., Girshick, R., Arbeláez, P., Malik, J.: Learning rich features from rgb-d images for object detection and segmentation. In: Proc. European Conference on Computer Vision (ECCV). pp. 345–360 (2014)
15. Han, Y., Lee, J.Y., Kweon, I.S.: High quality shape from a single rgb-d image under uncalibrated natural illumination. In: Proc. IEEE International Conference on Computer Vision (ICCV). pp. 1617–1624 (2013)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)

17. Hui, T.W., Loy, C.C., Tang, X.: Depth map super-resolution by deep multi-scale guidance. In: Proc. European Conference on Computer Vision (ECCV). pp. 353–369 (2016)
18. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and Locally Consistent Image Completion. ACM Transactions on Graphics (ToG) **36**(4), 107 (2017)
19. Kiechle, M., Hawe, S., Kleinsteuber, M.: A joint intensity and depth co-sparse analysis model for depth map super-resolution. In: Proc. IEEE International Conference on Computer Vision (ICCV). pp. 1545–1552 (2013)
20. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proc. International Conference on Learning Representations (ICLR) (2015)
21. Kopf, J., Cohen, M.F., Lischinski, D., Uyttendaele, M.: Joint bilateral upsampling. ACM Transactions on Graphics (ToG) **26**(3), 96 (2007)
22. Kwon, H., Tai, Y.W., Lin, S.: Data-driven depth map refinement via multi-scale sparse representation. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 159–167 (2015)
23. Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 624–632 (2017)
24. Le, A.V., Jung, S.W., Won, C.S.: Directional joint bilateral filter for depth images. Sensors **14**(7), 11362–11378 (2014)
25. Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4681–4690 (2017)
26. Lin, D., Fidler, S., Urtasun, R.: Holistic scene understanding for 3d object detection with rgbd cameras. In: Proc. IEEE International Conference on Computer Vision (ICCV). pp. 1417–1424 (2013)
27. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Proc. European Conference on Computer Vision (ECCV). pp. 740–755 (2014)
28. Lu, J., Forsyth, D.: Sparse depth super resolution. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2245–2253 (2015)
29. Lu, S., Ren, X., Liu, F.: Depth enhancement via low-rank matrix completion. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3390–3397 (2014)
30. Nah, S., Kim, T.H., Lee, K.M.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 257–265 (2017)
31. Nathan Silberman, Derek Hoiem, P.K., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: Proc. European Conference on Computer Vision (ECCV). pp. 746–760 (2012)
32. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A.: Kinectfusion: Real-time dense surface mapping and tracking. In: Proc. IEEE international Symposium on Mixed and Augmented Reality (ISMAR). pp. 127–136 (2011)
33. Nießner, M., Zollhöfer, M., Izadi, S., Stamminger, M.: Real-time 3d reconstruction at scale using voxel hashing. ACM Transactions on Graphics (ToG) **32**(6), 169 (2013)
34. Occipital Structure Sensor: `https://structure.io/`
35. Paszke, A., Gross, S., Chintala, S., Chanan, G.: Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration (2017)

36. Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., Efros, A.: Context encoders: Feature learning by inpainting. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2536–2544 (2016)
37. Schmeing, M., Jiang, X.: Edge-aware depth image filtering using color segmentation. Pattern Recognition Letters (PR) **50**(C), 63–71 (2014)
38. Shen, X., Zhou, C., Xu, L., Jia, J.: Mutual-structure for joint filtering. In: Proc. IEEE International Conference on Computer Vision (ICCV). pp. 3406–3414 (2015)
39. Song, S., Lichtenberg, S.P., Xiao, J.: Sun rgb-d: A rgb-d scene understanding benchmark suite. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 567–576 (2015)
40. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proc. IEEE International Conference on Computer Vision (ICCV). pp. 839–846 (1998)
41. Tosic, I., Drewes, S.: Learning joint intensity-depth sparse representations. IEEE Transactions on Image Processing (TIP) **23**(5), 2122–2132 (2014)
42. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing (TIP) **13**(4), 600–612 (2004)
43. Wu, C., Zollhöfer, M., Nießner, M., Stamminger, M., Izadi, S., Theobalt, C.: Real-time shading-based refinement for consumer depth cameras. ACM Transactions on Graphics (ToG) **33**(6), 200 (2014)
44. Xu, L., Ren, J., Yan, Q., Liao, R., Jia, J.: Deep edge-aware filters. In: Proc. International Conference on Machine Learning (ICML). pp. 1669–1678 (2015)
45. Xu, L., Ren, J.S., Liu, C., Jia, J.: Deep convolutional neural network for image deconvolution. In: Proc. Advances in Neural Information Processing Systems (NIPS). pp. 1790–1798 (2014)
46. Yu, L.F., Yeung, S.K., Tai, Y.W., Lin, S.: Shading-based shape refinement of rgb-d images. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1415–1422 (2013)
47. Zhang, L., Shen, P., Zhang, S., Song, J., Zhu, G.: Depth enhancement with improved exemplar-based inpainting and joint trilateral guided filtering. In: Proc. IEEE International Conference on Image Processing (ICIP). pp. 4102–4106 (2016)
48. Zhang, Q., Shen, X., Xu, L., Jia, J.: Rolling guidance filter. In: Proc. European Conference on Computer Vision (ECCV). pp. 815–830 (2014)
49. Zhang, Y., Funkhouser, T.: Deep depth completion of a single rgb-d image. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)