Supplementary Material: Free-form 3D Scene Inpainting with Dual-stream GAN

Ru-Fen Jheng[1]
ruby2332ruby@cmlab.csie.ntu.edu.tw
Tsung-Han Wu[1]
tsunghan@cmlab.csie.ntu.edu.tw
Jia-Fong Yeh[1]
jiafongyeh@ieee.org
Winston H. Hsu[1,2]
whsu@ntu.edu.tw

[1] National Taiwan University
[2] Mobile Drive Technology

Abstract

In this supplementary material, we detail our free-form 3D mask generation algorithm for the FF-Matterport dataset in Sec. 1 with pseudo code in Algorithm 1. We then provide the implementation details of our designed network dual-stream generator and the NN-based edge detector in dual-stream discriminator in Sec. 2. Furthermore, we show additional results in Sec. 3, including the qualitative (perceptual) improvement of our edge discriminator (Sec. 3.1) and the limitations of our proposed model (Sec. 3.2).

1 Free-form 3D Mask Generation Algorithm

As stated in Sec. 3.1 of the main paper, our designed free-form 3D mask generation algorithm aims to mimic human drawing trajectories in 3D space and randomly generate diverse free-form masks for efficient training and evaluation. To avoid covering the empty space in the scene into the masked areas and to flexibly draw arbitrary shapes of 3D objects, we utilize the characteristic of TSDF and dynamically decide the length and direction of strokes based on the TSDF values. We show our algorithm in Algorithm 1.

The input of the algorithm is an original scene S_o from a real-world scanned scene containing T_o TSDF and C_o color voxelized values, and three hyper-parameters $diameter$, $maxStrokeStep$, and $totalStep$ to control the mask distribution. The output is a masked scene S_m for training and inferring consisting of masked T_m TSDF and C_m color voxelized values with corresponding binary mask map M_m. This algorithm is for the 64x64x128 chunk size in the training dataset. For the test dataset which contains whole indoor rooms, We run this algorithm several times in equal proportions to the size of the indoor rooms.

© 2022. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

Algorithm 1 Free-form 3D Mask Generation Algorithm

Input: Original scene: $S_o = \{T_o, C_o\}$, and $diameter$, $maxStrokeStep$, $totalStep$.
Output: Masked scene: $S_m = \{T_m, C_m, M_m\}$.
Init: Copy from original scene $T_m \leftarrow T_o$, $C_m \leftarrow C_o$, (#step, #strokeStep) $\leftarrow (0,0)$.
function RANDOMMAXSTROKE(s)
    return random.randint(s, s+10)
end function
function FINDVALIDBALL(c, d, t) ▷ c is center, d is diameter, and t is threshold
    return $X \leftarrow \{x | x \in B_{c,d} \cap |T_m(x)| \leq t\}$ ▷ $B_{c,d}$ is a ball centered at c with diameter d
end function
center $O_c \leftarrow$ random.choice($|T_m| \leq 1$) ▷ Random a starting point on occupied voxels
max stroke step $L \leftarrow$ RANDOMMAXSTROKE($maxStrokeStep$)
while #step $\leq TotalStep$ **do**
    Mask out $B_{O_c, diameter}$ in $\{T_m, C_m, M_m\}$
    #step $\leftarrow$ #step $+ 1$
    #strokeStep $\leftarrow$ #strokeStep $+ 1$
    if #strokeStep $\geq L$ **then** ▷ Restart a new stroke
        center $O_c \leftarrow$ random.choice($|T_m| \leq 1$)
        max stroke step $L \leftarrow$ RANDOMMAXSTROKE($maxStrokeStep$)
        #strokeStep $\leftarrow 0$
    else if $X \leftarrow$ FINDVALIDBALL($O_c, diameter//2, 1) \neq \emptyset$ **then** ▷ Move a small step
        $O_c \leftarrow$ random.choice(X)
    else if $X \leftarrow$ FINDVALIDBALL($O_c, diameter, 5) \neq \emptyset$ **then** ▷ Move a big step
        $O_c \leftarrow$ random.choice(X)
    else ▷ Dead end, Restart a new stroke
        center $O_c \leftarrow$ random.choice($|T_m| \leq 1$)
        max stroke step $L \leftarrow$ RANDOMMAXSTROKE($maxStrokeStep$)
        #strokeStep $\leftarrow 0$
    end if
end while
return $\{T_m, C_m, M_m\}$.

2 Network Architecture

We show the implementation details of our proposed network architectures: dual-stream generator in Fig. 1 and the NN-based edge detector of dual-stream discriminator in Fig. 2. The implementation of the patch-based discriminator of dual-stream discriminator follows the one in SPSG [2]. In the figures, GatedConv and Conv stand for the 3D gated convolutional layer and 3D convolutional layer. Conv2D denotes the 2D convolutional layer. The parameters of convolution are given by (in-channel, out-channel, kernel size, stride, padding), and all GatedConv layers are paired with a 3D batch normalization and a Leaky ReLU.

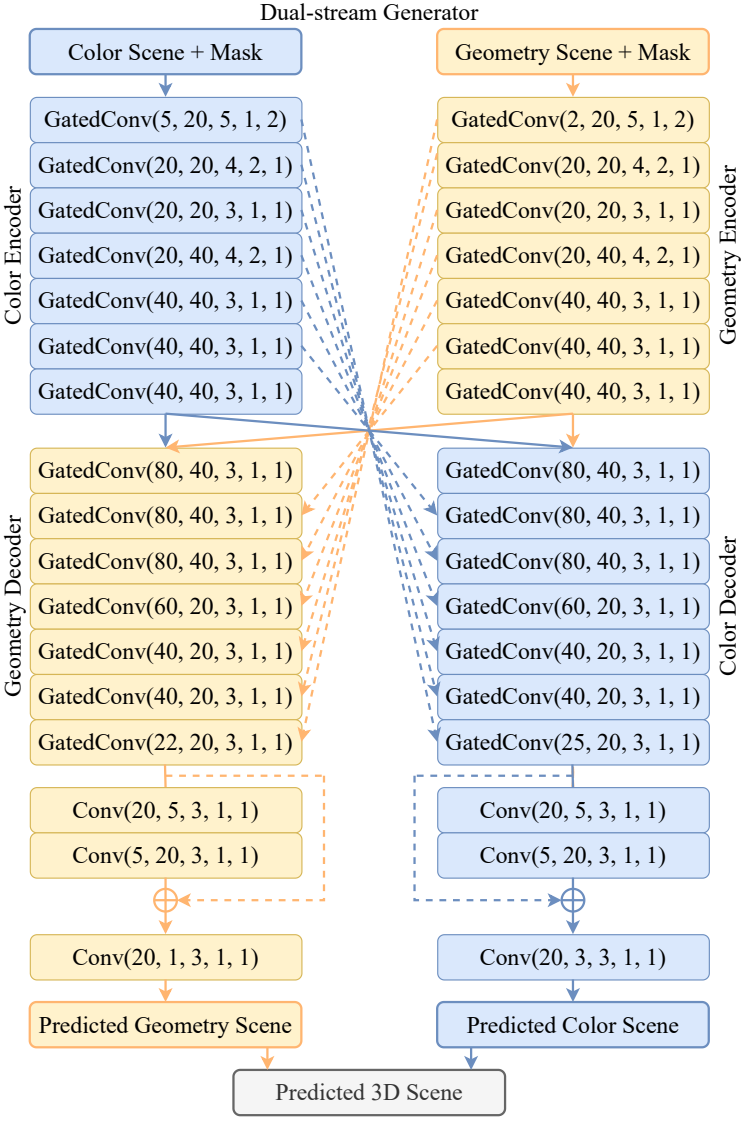


Figure 1: Details of dual-stream generator architecture, where $\oplus$ denotes add and dashed line denotes skip connection. We process the geometry and color scenes in two streams and concatenate the predicted geometry and color scenes to get the final result.

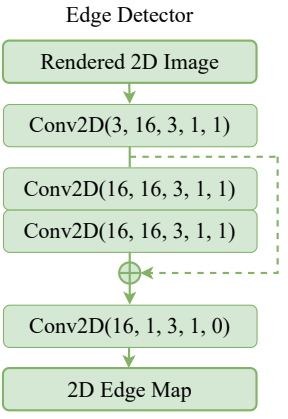


Figure 2: Details of edge detector architecture of dual-stream discriminator, where $\oplus$ denotes add and dashed line denotes skip connection. The edge detector takes the rendered 2D image from the differentiable 2D rendering and extracts the corresponding edge map for our edge-stream discriminator.

3 Additional Results

3.1 Edge Discriminator Qualitative Result

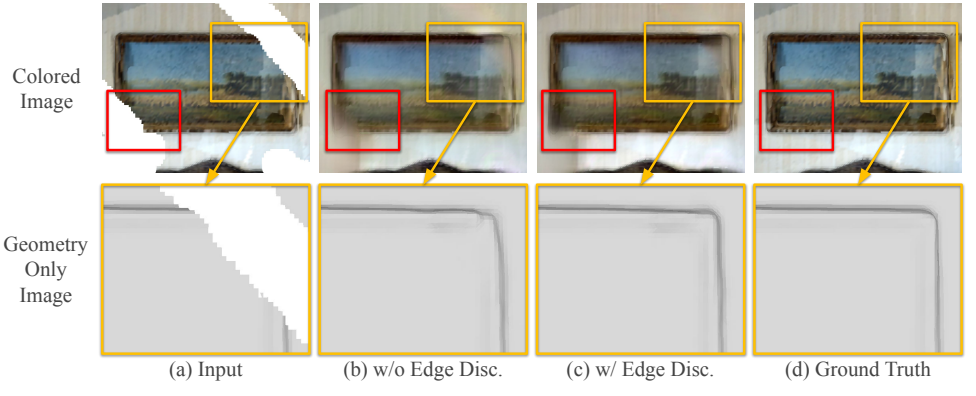
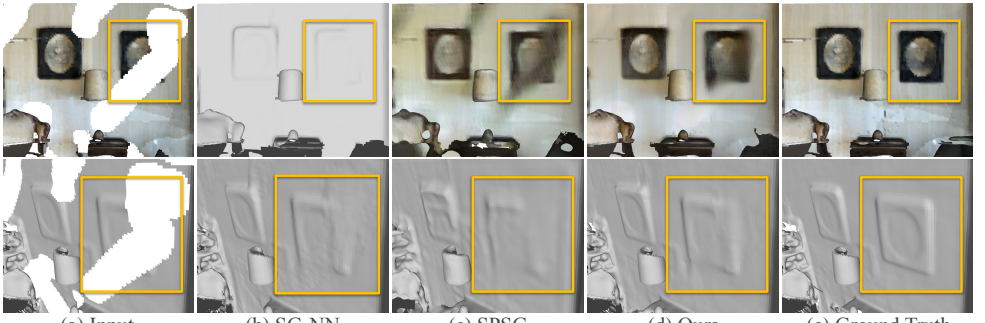


Figure 3: Additional qualitative results of the dual-stream discriminator on FF-Matterport. By adding the edge stream to the conventional color stream discriminator, the color boundary becomes less blurred (shown in red frames), and the geometry shapes become more straight and sharper (shown in zoomed-in pictures in the 2nd row with yellow frames).

In the Sec. 3.3 of the main paper, we state that our additional discriminator aims to enhance the sharpness and details of the results for better human perception. As shown in Fig. 3, we provide additional qualitative results on FF-Matterport to demonstrate the efficacy of our dual-stream discriminator. Compared with (b) the model without our edge discriminator, the full model (c) performs better on both geometry and color. In (c), the bottom left corner (red frame) of the color image becomes less blurred after adding the edge stream, and the top right corner (yellow frame) of the color image zoomed in as the geometry only



(a) Input (b) SG-NN (c) SPSG (d) Ours (e) Ground Truth

Figure 4: A failure case of the frame on the right side in (a) losing most of the edges and color. Even though our model fails to predict distinct structures and edges in the missing parts, our results (d) still outperform the baselines SG-NN [1] (b) and SPSG [2] (c) in both geometry and color visual performance.

image shows that the geometry shapes become sharper after adding our edge discriminator. This phenomenon echoes our motivation that 2D edge loss can simultaneously guide the 3D geometry and color streams to generate delicate edges.

3.2 Limitation

Although our proposed method can generate realistic geometry and color results for the real-world 3D scene inpainting task, we still find its limitations and unsolved challenges as shown in Fig. 4. For example, in the 1st row (a), the right side of an incomplete picture frame loses most of the edges and color; also, the 2nd row (a) shows that the structure of the frame is very close to the wall and hard to be distinguished. Therefore, even with the help of our dual-stream GAN design, the model fails to predict the correct edges of the top right and bottom left corners, resulting in blurred color boundaries in the comparison of (d) and (e). Still, our predicted results contain more details than SG-NN [1] (b) and SPSG [2] (c).

Moreover, due to the natural limit of CNN models and voxel representation, the output resolution is restricted. Even though some new 3D data representations are proposed recently and claimed to support the unlimited resolution, such as implicit function, it is still challenging to properly handle the mask information in the 3D scene inpainting task.

References

[1] Angela Dai, Christian Diller, and Matthias Nießner. Sg-nn: Sparse generative neural networks for self-supervised scene completion of rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[2] Angela Dai, Yawar Siddiqui, Justus Thies, Julien Valentin, and Matthias Nießner. Spsg: Self-supervised photometric scene generation from rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.