

RenderGAN: GAN Based Texture Rendering

(Youtube Link: <https://youtu.be/UsRMuAzGutU>)

Joon Jung (joonjung@stanford.edu)

Stanford University

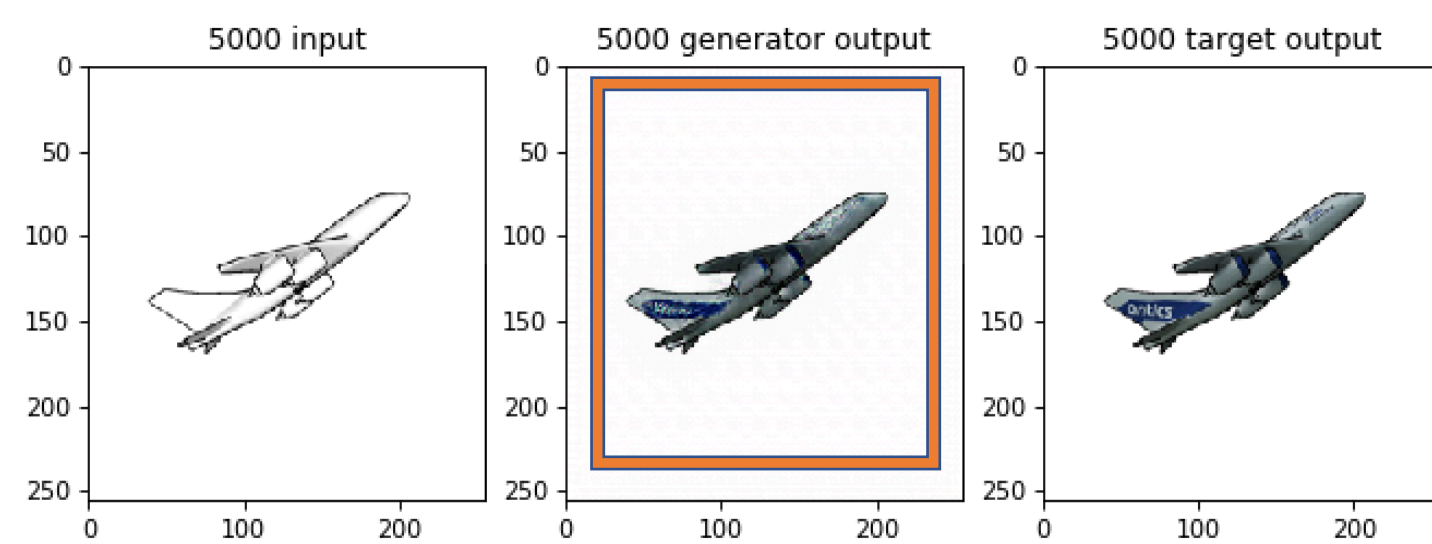
1. Motivation

Modern graphics rendering requires intensive computations, performing high resolution texture sampling and shading in million times fold. As the result, the memory related operations have become the most performance constraining components in the graphics pipeline.

In recent, various researches on extracting and fusing content representations and style representations from different image domains, using deep learning, have indeed significantly progressed. [GEB16][Iso+17] The content and style extraction and fusion is also called as 'texture transferring'.

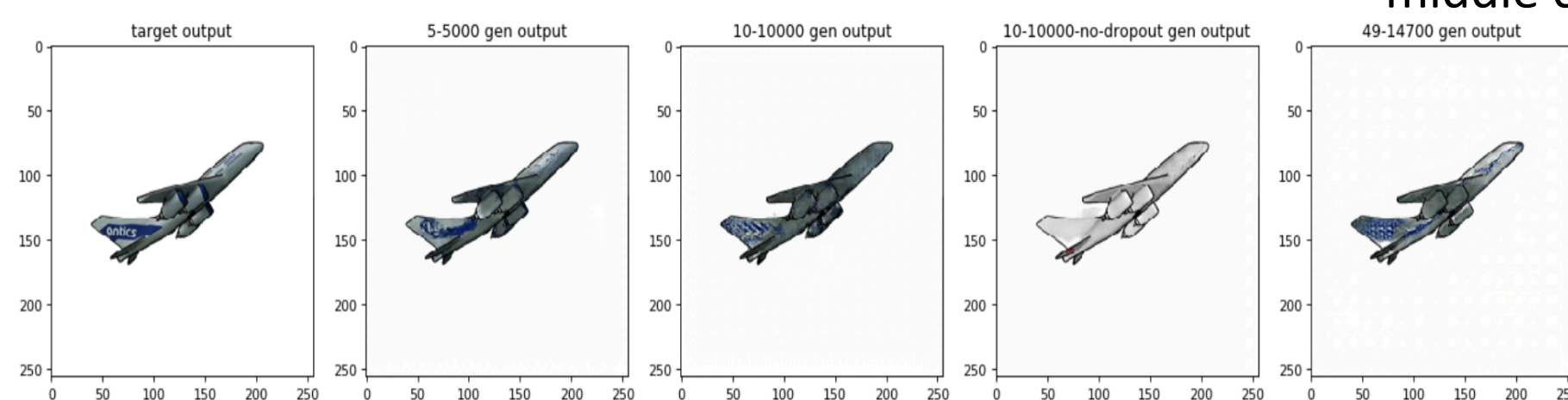
What if we can extract, from a high resolution texture, the style representation and fuse it with a polygon only 3D model acting as the content representation source. Then we can think of the 3D rendering process as the texture transferring process. If we can achieve this fusion in a seamless manner, this becomes equivalent to the texture rendering process of a 3D rendering pipeline. Therefore, it would be possible to completely replace the pipeline with a deep learning model. This project starts from this motive. We are replacing the texture rendering process in 3D pipeline with the generative texture transferring using the generator model from Pix2Pix. [Iso+17]

4. Result Examples



Losses

G_GAN: 4.812 G_L1: 0.370 D_real: 0.000 D_fake: 0.015



- Unique to this project, overfitting to the training data is actually desired.
- That is, the input model in the actual deployment environment comes from the training distribution.
- Many experiments are performed to search the right dataset.
- Multiple categories vs. single category dataset, multiple models vs. single model dataset, small vs. big dataset etc.
- The best result comes from a single model with 5000 random view captured dataset as shown in the figure left, in the middle column.



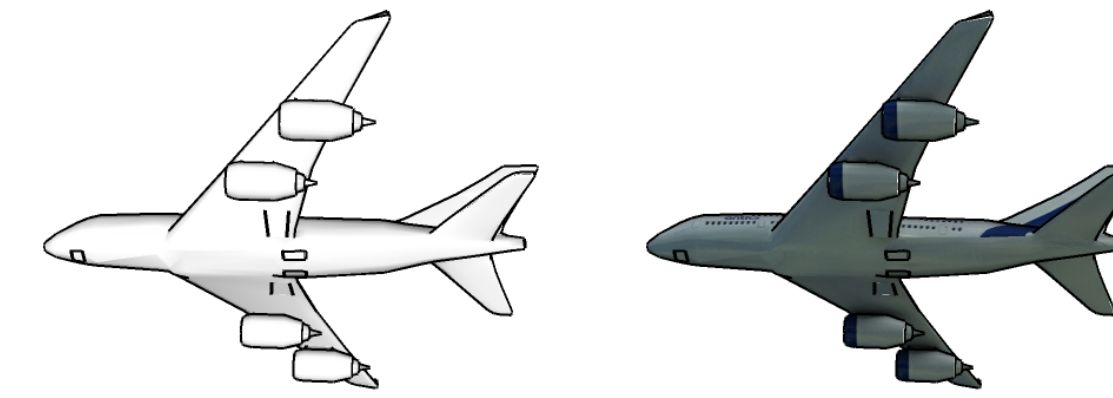
An example of the best draws trained in the heterogeneous dataset (10-1000)

- The images on the above left show the rest of the datasets results.
- **Dataset naming codes:** The first part number of the dataset name indicates the number of models used and the second part number indicates the number of random perspective capture images.
- **Heterogeneous dataset example:** The middle column image on the above right shows an example of the best draws produced by the generator trained with 10-10000 dataset. In this case quite impressively the generator successfully has drawn the different parts of the plane.

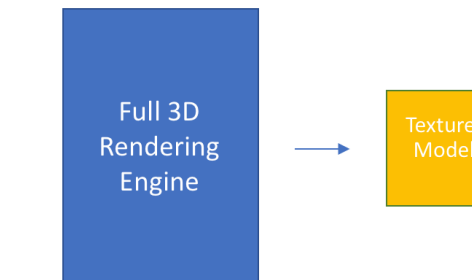
References

[GEB16] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. "Image Style Transfer Using Convolutional Neural Networks". In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). June 2016.
[Iso+17] Phillip Isola et al. "Image-to-Image Translation with Conditional Adversarial Networks". In: Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on. 2017.
[Ske] SketchUp. URL: <https://3dwarehouse.sketchup.com/search/?q=airplane>.

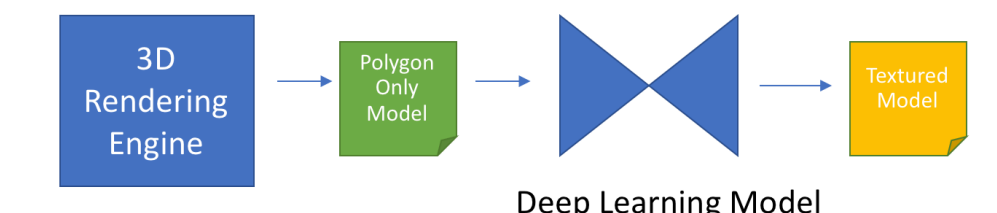
2. Model



Left: an input image. Right: a target output image.

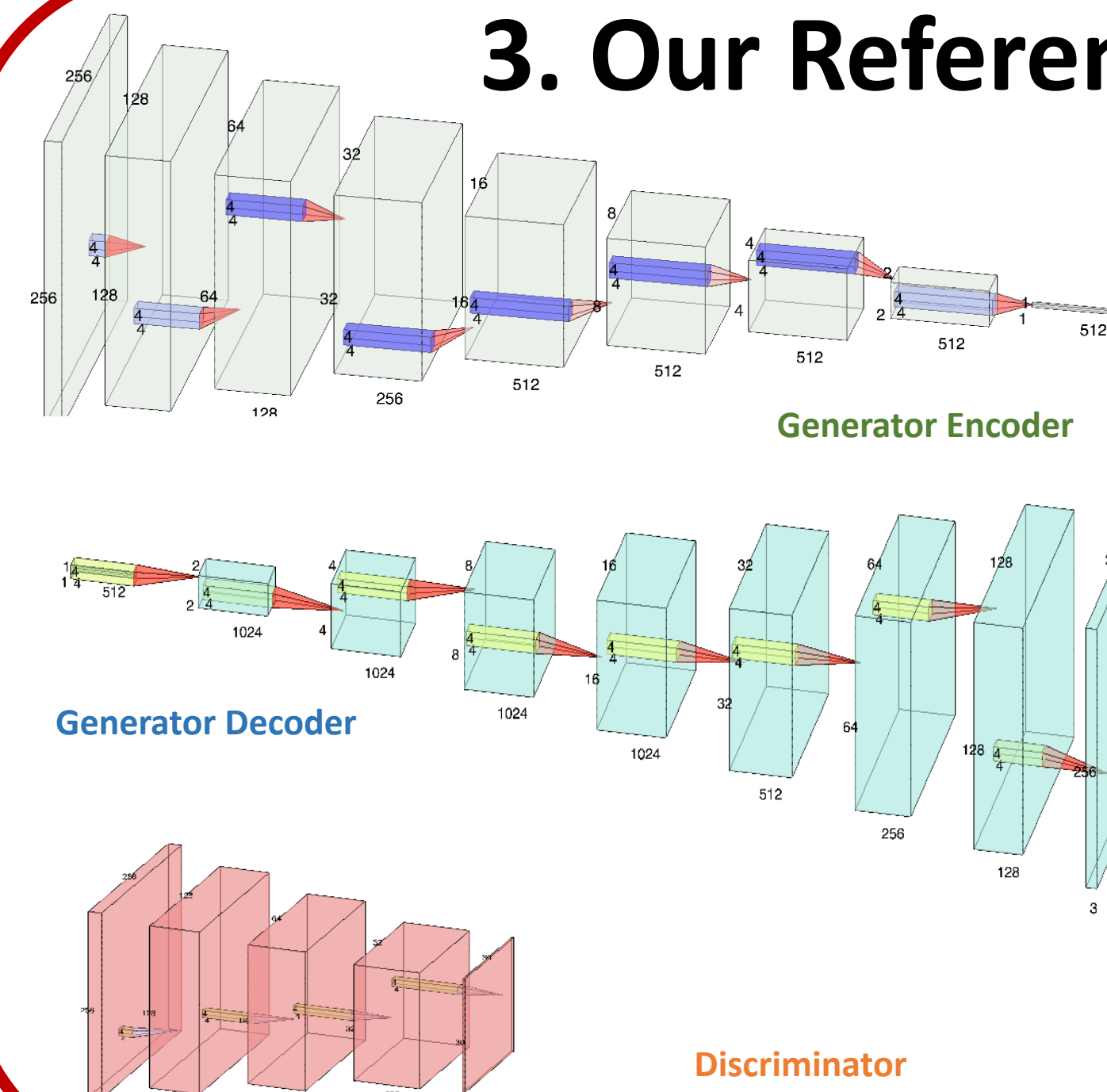


Normal Rendering Pipeline



Generative Model Based Rendering Pipeline

3. Our Reference Method: Pix2Pix



- Uses generative adversarial model to optimize the objectives.
- Generator G is a CNN encoder-decoder with skip connections to pass relevant features from encoder layers to the corresponding decoding layers.
- Discriminator D is a 70x70 PatchGan in our project.

Objective Functions

The overall objective function of Pix2Pix is

$$G^* = \arg \max_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G)$$

for the generator G and the discriminator D. The conditional GAN loss is

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y} [\log D(x, y)] + \mathbb{E}_{x,z} [\log D(x, G(x, z))].$$

This conditional term differs from the usual GAN's in such that the discriminator uses the input image x in addition to using the generated output image G(y) and y.

The L1 loss is

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z} [\|y - G(x, z)\|_1].$$

5. Accuracy Measures

dataset	1-5000	5-5000	10-10000	49-14700
MSE	6.377	6.968	7.571	7.493

- **Two metrics:** For the quantitative metric, we've used Mean Squared Error. For the qualitative metric, human eye inspection, which can be subjective, is used.
- **MSE:** The dataset 1-5000 achieves the best result with MSE 6.377.
- **Human Eye Inspection:** By inspecting, we can see the generator perfectly regenerated the shape of the plane among all the datasets presented here. (This is not the case if we use more than a single category datasets. That is, for example if we mix the models with passenger planes with fighter planes, the generator starts to corrupt the shape of the input model.) Also the generator performs pretty well mimicking the shading of the target image. We can even see it generating pretty convincing tail painting which includes letters, even though it is from a distance looking.