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]

References


SketchUp. URL: https://3dwarehouse.sketchup.com/search/?q=airplane.

2. Model

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

4. Result Examples

• 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.

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.

5. Accuracy Measures

<table>
<thead>
<tr>
<th>dataset</th>
<th>MSE</th>
<th>5-5000</th>
<th>10-10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5000</td>
<td>6.377</td>
<td>6.968</td>
<td>7.571</td>
</tr>
<tr>
<td>49-14700</td>
<td>7.493</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• 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.