Deep Visual Computing:

Deep Rendering
(or Neural Rendering)

Daniel G. Aliaga
Spring 2023
Sources:

- State of the Art on Neural Rendering

*Figure 1: Neural renderings of a large variety of scenes. See Section 6 for more details on the various methods. Images from [SBT* 19, SZW19, XBS* 19, KHM17, GLD* 19, MBPY* 18, XSHR18, MGK* 19, FTZ* 19, LXZ* 19, WSS* 19].*
Learning to Generate Chairs with Convolutional Neural Networks
Dosovitsky, Springenberg, Brox

• Early work in deep neural rendering (CVPR 2015)

Figure 1. Interpolation between two chair models (original: top left, final: bottom left). The generative convolutional neural network learns the manifold of chairs, allowing it to interpolate between chair styles, producing realistic intermediate styles.
• Architecture is basically a pair of inverted CNNs

• Trained on 809 chair models from 62 viewpoints (~50,000 training samples)
Learning to Generate Chairs with Convolutional Neural Networks
Dosovitsky, Springenberg, Brox

• Analysis:
  • They looked into activating single neurons

Figure 7. Images generated from single unit activations in feature maps of different fully connected layers of the 128 x 128 network. From top to bottom: FC-1 and FC-2 of the class stream, FC-3, FC-4.

Figure 8. The effect of increasing the activation of the ‘zoom neuron’ we found in the layer FC-4 feature map.
Learning to Generate Chairs with Convolutional Neural Networks
Dosovitsky, Springenberg, Brox

• Analysis:
  • They looked into activating single layers
Analysis:

- They looked into activating single neurons, single layers, and interpolating input viewing parameters.
Learning to Generate Chairs with Convolutional Neural Networks
Dosovitsky, Springenberg, Brox

• **Analysis:**
  • They looked into activating single neurons, single layers, and interpolating input viewing parameters
• Analysis:
  • They looked into activating single neurons, single layers, and interpolating input viewing parameters

Figure 12. Examples of interpolation between angles. In each pair of rows the top row is with knowledge transfer and the second - without. In each row the leftmost and the rightmost images are the views presented to the network during training, while all intermediate ones are not and hence are results of interpolation. The number of different views per chair available during training is 15, 8, 4, 2, 1 (top-down). Image quality is worse than in other figures because we use the 64 × 64 network here.
Analysis:
- They looked into activating single neurons, single layers, and interpolating input viewing parameters.
Neural Rendering

• So how to improve from the shown “baseline”?

• One next wave of methods improved neural-rendering quality by using perceptual similarity metrics...
Perceptual/Image Similarity Metrics

• L2 loss (=“square of pixel to pixel difference”)
  • Yields overall similarity

• L1 loss (=“absolute value of pixel to pixel difference”)
  • More sensitive to strong changes, thus focuses on details but does not do well with the overall picture

• PSNR (=“signal to noise ratio”)
  • Used lots of compression
Perceptual/Image Similarity Metrics

- SSIM (Structured Similarity Image Metric)
  - Combine luminance, contrast, and structure

\[
I(x, y) = \frac{2\mu_x \mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad C(x, y) = \frac{2\sigma_x \sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \\
S(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x \sigma_y + C_3} \\
SSIM(x, y) = I(x, y)^\alpha C(x, y)^\beta S(x, y)^\gamma
\]
Loss Functions for Image Restoration with Neural Networks, Zhao et al.

Fig. 5: Results for denoising-demosaicking for different approaches. The noisy patches are obtained by simple bilinear interpolation. Note the splotchy artifacts $\ell_2$ produces in flat regions. Also note the change in colors for SSIM-based losses. The proposed metric, MS-SSIM+$\ell_1$, referred to as Mix, addresses the former issues. Reference images are shown in Figure 4.
What else? How to use deep learning?
Generating Images with Perceptual Similarity Metrics based on Deep Networks
Dosovitskiy, Brox

• Proposed a class of loss functions, called deep perceptual similarity metrics (DeePSiM), to reduce over-smoothed results

• Losses are weighted sums of
  • Feature loss (C)
  • Adversarial loss (G)
  • Pixel-space loss (D)

Figure 2: Schematic of our model. Black solid lines denote the forward pass. Dashed lines with arrows on both ends are the losses. Thin dashed lines denote the flow of gradients.
Generating Images with Perceptual Similarity Metrics based on Deep Networks
Dosovitskiy, Brox

• Feature loss

\[ \mathcal{L}_{\text{feat}} = \sum_{i} \|C(G_{\theta}(x_i)) - C(y_i)\|_2^2. \]

• Generator loss

\[ \mathcal{L}_{\text{discr}} = -\sum_{i} \log(D_{\varphi}(y_i)) + \log(1 - D_{\varphi}(G_{\theta}(x_i))), \quad (3) \]

\[ \mathcal{L}_{\text{adv}} = -\sum_{i} \log D_{\varphi}(G_{\theta}(x_i)). \]

• Pixel-space loss

\[ \mathcal{L}_{\text{img}} = \sum_{i} \|G_{\theta}(x_i) - y_i\|_2^2. \]
Generating Images with Perceptual Similarity Metrics based on Deep Networks
Dosovitskiy, Brox

- Used with for compression with an autoencoder, generative model with a VAE, and inversion of AlexNet

Figure 3: Autoencoder qualitative results. Best viewed on screen.

Figure 7: Reconstructions from FC6 with some components of the loss removed.
Figure 1: Our two-stage adversarial framework translates an OpenGL rendering (a) to a realistic image (c). Compared to single-stage prediction with CycleGAN (b), our result has more realistic illumination and better preserves texture details, as shown in the insets. (Best viewed in digital.)
Deep CG2Real: Synthetic-to-Real Translation via Image Disentanglement
Bi et al.

Figure 2: The framework of our two-stage OpenGL to real translations.
Neural Rerendering in the Wild (CVPR 2019)

• https://www.youtube.com/watch?v=E1crWQn_kmY
Image-guided Neural Object Rendering

• https://niessnerlab.org/projects/thies2020ignor.html
Neural Point-Based Graphics

- https://dmitryulyanov.github.io/neural_point_based_graphics
Deferred Neural Rendering: Image Synthesis using Neural Textures (SIGGRAPH 2019)

- https://www.youtube.com/watch?v=z-pVip6WeyY
DeepView: View Synthesis with Learned Gradient Descent (CVPR 2019)

• https://www.youtube.com/watch?v=CQ0kdR3c4Ec