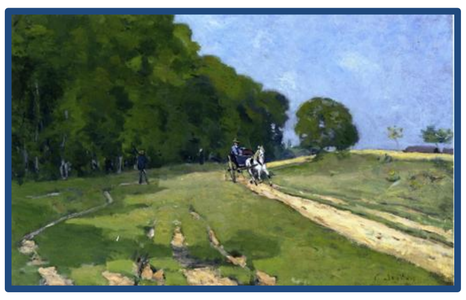
Inverse Rendering for 2D Art Relight

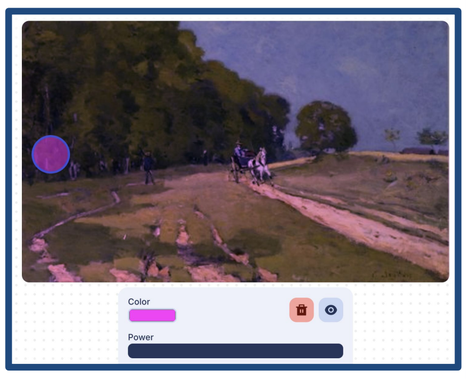
Carol Meng
CSC2529, University of Toronto

Motivation

original



ClipDrop Relight



Problem:

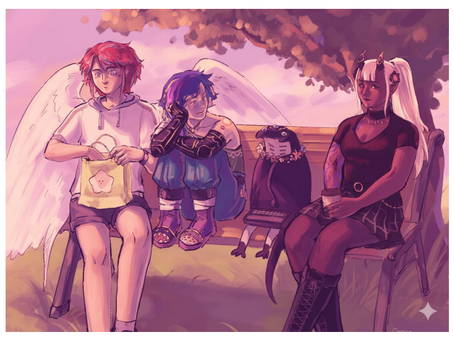
- Manually creating different versions of the same artwork (ie. day/night scenes) can be time consuming and tedious for artists
- Existing relighting services are mainly built for photographs instead of 2D, stylized art. GenAI solutions may not work or preserve style

My goal:

- Apply neural inverse rendering based on SotA techniques, like Nvidia's UniRelight, to infer 3D structure from 2D art in a similar way as how artists mentally parse art
- Relight 2D art while respecting the boundaries of artistic regions, and without affecting style

original

Gemini Nano Banana



prompt: change the light source to the right side and make it pink

Related Work

Nvidia UniRelight [1]: uses dual-stream diffusion to predict intrinsics (albedo, normals), made for photos but diffusion model approach applicable to art

Artistic relighting [6], a method based on using 3D models (proxies) to approximate image geometry

Proxies

References

[1] He, Kai, et al. "UniRelight: Learning Joint Decomposition and Synthesis for Video Relighting." arXiv preprint arXiv:2506.15673 (2025)
[2] Wang, Zhou, et al. "Image quality assessment: from error visibility to structural similarity." IEEE transactions on image processing 13.4 (2004)
[3] Perez, Ethan, et al. "Film: Visual reasoning with a general conditioning layer." Proceedings of the AAAI conference on artificial intelligence. Vol. 32. No. 1. 2018
[4] Zhang, Richard, et al. "The unreasonable effectiveness of deep features as a perceptual metric." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
[5] Chen, Zhifei, et al. "Uni-Renderer: Unifying Rendering and Inverse Rendering Via Dual Stream Diffusion." Proceedings of the Computer Vision and Pattern Recognition Conference. 2025.
[6] Henz, Bernardo, and Manuel M. Oliveira. "Artistic relighting of paintings and drawings." The Visual Computer 33.1 (2017): 33-46.

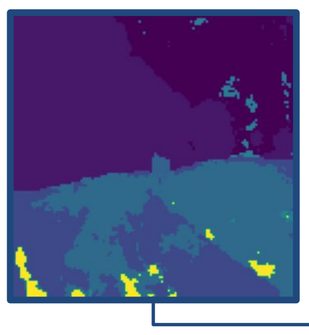
Image Intrinsics Prediction Pipeline

- Built on SegFormer and UniRenderer [5] since it implements a similar dual stream diffusion model as UniRelight
- Applies semantic segmentation as a preprocessing step, inspired by how artists parse artwork visually

input image (RGB)

SegFormer Model

segmentation map



Variational Autoencoder (VAE) **Encoder**

latent space

Dual Stream Diffusion Model
UNet Backbone, extends UniRenderer's UNet2DConditionModel

Feature-wise Linear Modulation (FiLM) layer [3] injected into each ResNet block of the base UNet2DConditionModel
Uses seg map as conditioning signal
applies affine transform $\gamma \odot x + \beta$ to input

Down block: encode latent features to intrinsics

Mid block: bottleneck

Up block: decode

Stream A: predicts albedo, roughness

Stream B: predicts surface normals, monocular depth

Variational Autoencoder (VAE) **Decoder**

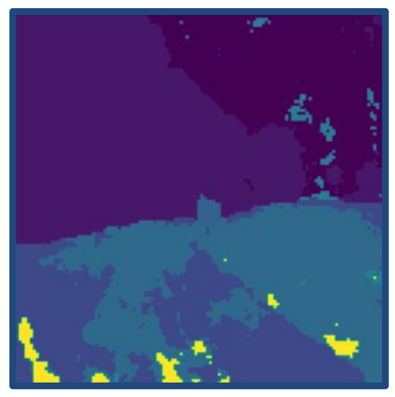
rough-ness | normal | albedo | light

RGB output

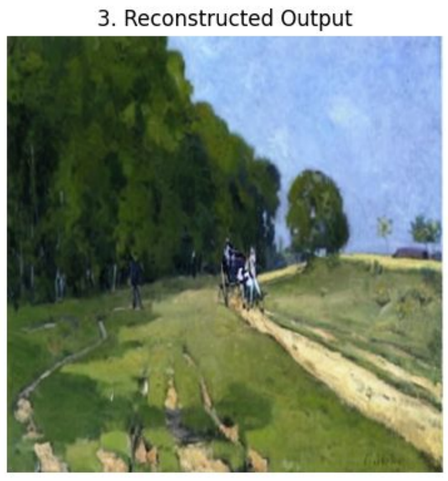
Experimental Results & Current Progress

- Not capable of relighting images yet, lacks the features for the user to define new lighting
- Segments into distinct classes

- VAE Encoding to Latent Space
  - On example image in previous section:
  - Latent shape: torch.Size([1, 4, 64, 64])
  - Latent range: [-9.40, 7.52]
- Extracted material (roughness), normal, albedo, and light as output from the diffusion model, can be reconstructed into the original image
  - VAE Decoding back to Image Space
  - Decoded shape: torch.Size([1, 3, 512, 512])



3. Reconstructed Output



Limitations and Future Plans

- Currently restrained by lack of datasets for training the dual stream diffusion model
- No ground truth intrinsics (albedo, normals) to fine tune diffusion model on

- Applying the lighting of one image (should be doable with the predicted lighting) to another image
- Possible user interface for letting the user define new light sources (like in ClipDrop Relight
- Pretrained SegFormer model works for doing semantic segmentation.