Real-time 3D-aware Portrait Video Relighting

Dhruv Misra

November 5, 2024

1 Introduction

This implementation is part of a project for the course EE798R. It is based on the research paper titled **"Real-time 3D-aware Portrait Video Relighting"** by Ziqi Cai et al. The project aims to synthesize realistic talking faces in portrait videos under custom lighting conditions and novel viewing angles using Neural Radiance Fields (NeRF). The GitHub repository for the implementation can be found here: `https://github.com/dhruvmisra1007/PortraitRelighting`.

The proposed method in the paper provides the first real-time solution for relighting portrait videos, generating high-quality 3D reconstructions with consistent lighting and viewpoint adjustments. The implementation utilizes dual-encoders to infer albedo and shading, as well as a temporal consistency network to reduce flickering artifacts. The colab implementation file can be found here: `https://colab.research.google.com/drive/19ipGj6IZV0H0cOoeveHpnlWW5ODL4n4Y`

2 Methodology

2.1 Technical Approach

The methodology used in the paper includes several key components:

- **Tri-plane Representation**: The model uses a tri-plane representation for 3D facial reconstruction, treating albedo and shading separately. This enables effective relighting under novel viewpoints.
- **Dual-Encoder Network**: The model employs dual-encoders to separately infer albedo and shading, ensuring better disentanglement of the features for improved relighting quality.
- **Temporal Consistency Network**: A temporal consistency network is used to minimize flickering artifacts between consecutive frames, resulting in smoother relighting in portrait videos.

2.2 Loss Functions

The model utilizes several loss functions to optimize the performance:

- **Albedo Loss**: Measures the difference between predicted and ground-truth albedo to ensure accurate color reproduction.
- **Shading Loss**: Measures the disparity between predicted and ground-truth shading features to improve lighting consistency.
- **Temporal Consistency Loss**: Consists of short-term and long-term consistency measures to minimize flickering and maintain consistency between frames.
- **Adversarial Loss**: The loss function includes an adversarial loss that helps in differentiating between real and generated images, improving the realism of the output.

2.3 Training Strategy

The training strategy follows three main stages:

- **Stage 1:** The albedo encoder is trained independently from others to reconstruct the given portraits.
- **Stage 2:** Both the albedo and shading branches are trained separately to learn non-entangled features.
- **Stage 3:** The two branches are merged, and training is jointly performed. Adversarial losses are used after it converges initially to improve output quality.

3 Implementation

3.1 Code Overview

The implementation is organized into several modules, each handling specific tasks:

- `networks/relighting.py`: Contains the implementation of the relighting network.
- `third_party/wrappers.py`: Includes wrappers for external models and utilities used for facial reconstruction and lighting estimation.
- `examples/`: Contains example images and videos used for testing the relighting process.

3.2 Environment Setup

The environment setup was done using Google Colab, leveraging `conda` for managing dependencies. The `requirements.txt` file provided in the repository was used to install all necessary packages. T4 GPU acceleration was used to speed up the training and inference processes.

3.3 Challenges Encountered

During the implementation, several challenges were faced, including:

- Resolving dependencies related to CUDA for GPU processing.
- Setting up the correct versions of external libraries to ensure compatibility.
- Adjusting the training parameters to match those specified in the original paper.

4 Dataset Description

4.1 Dataset Overview

The implementation uses pre-existing data from the original GitHub repository, which includes:

- **Example Images**: A sample portrait image (`example.png`) used for reconstruction and relighting.
- **Video Frames**: Individual frames extracted from a video, used for video relighting.
- **Pre-trained Models**: `NeRFFaceLighting.pkl` and BFM models used for facial reconstruction and lighting estimation.

4.2 Evaluation Dataset

The INSTA dataset was used to evaluate the performance on metrics like relighting accuracy, lighting instability, and identity preservation. Additionally, synthetic data was generated for training to enhance temporal consistency using augmentation techniques tailored for dynamic viewing angles.

5 Results

5.1 Quantitative Metrics

The following metrics were calculated to evaluate the performance of the model:

- **Lighting Error (LE)**: Measures the accuracy of the lighting estimated by the model.

- **Lighting Instability (LI)**: Evaluates the stability of the lighting across consecutive frames.
- **Identity Preservation (ID)**: Assesses how well the identity of the person is preserved after relighting.
- **Warping Error (WE)**: Measures the distortion in the relighted images.
- **LPIPS**: Measures the perceptual similarity between consecutive frames to evaluate flickering.
- **Time Cost**: The average time taken per frame during inference.

5.2 Qualitative Examples

The implementation generated several visual examples comparing the input frames and the relighted frames produced by the model. These examples illustrate the high-quality reconstructions and smooth temporal transitions achieved by the model.

6 Conclusion

6.1 Summary

The project successfully implemented the real-time 3D-aware portrait video relighting as described in the original paper. The model generated high-quality relighted videos with consistent lighting and viewpoints, achieving comparable results to those presented in the paper. The results are saved in the examples/ directory of the GitHub repository.

6.2 Challenges

Major challenges included setting up the environment, resolving dependencies, and fine-tuning the training process to match the performance reported in the paper.

6.3 Future Work

Possible future improvements include:

- Using a more sophisticated temporal consistency model to further reduce flickering.
- Optimizing the dual-encoder network to achieve better feature disentanglement.
- Improving runtime performance to make the model suitable for real-time applications.

7 References

- Ziqi Cai et al., "Real-time 3D-aware Portrait Video Relighting", CVPR 2024.
- `https://github.com/GhostCai/PortraitRelighting`