University of Glasgow | School of Computing Science

Relighting assets using deep learning and neural rendering

Sarthak Ahuja

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

A dissertation presented in part fulfilment of the requirements of the Degree of Master of Science at The University of Glasgow

2 September 2022

Abstract

Animation video content production has been on the rise in recent years but to date, a lot of animation studios spend a lot of labour to sketch, colour, relight and render every single frame one by one. Though certain lighting presets exist for the same, it is believed it would be great to have an artificial intelligence-based algorithm/software to mimic a photo's light setting – in terms of colour, softness, direction, emotions, situations etc so it saves time. Hence, this project aims to look closely at the problem of relighting using total relighting, relightable neural radiance fields, and relightable and editable neural rendering. It tries to explore relighting in images and applications in neural rendering, which is currently an open challenge. It draws inspiration from traditional workflows used to do relighting. In the end, it aims to explore the possibilities of relighting scenarios in augmented reality environments.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: _____ Signature: _____

Acknowledgements

I would like to thank my professor Nicolas Pugeault for helping me ideate in this research area and further guiding me to fine-tune my topic. I am grateful for all the support provided through multiple learning resources and publications. Follow-up meetings and discussions were quite vital in shaping my dissertation and helped me to produce objective, coherent and comprehensive content.

I would further like to thank professor Jose Cano Reyes for seamlessly explaining the requirements and expectations of this dissertation in addition to catering to any requests.

In the end, I would like to thank my family and friends, who all were patient and supportive throughout these three months.

Contents

List of Figures

Chapter 1: Introduction

1.1 Motivation

Animation video consumption has only been on the rise for the last few years, but animation video content production has been struggling to meet up with the increasing demand. To date, most animation studios especially small-to-medium scale ones spend a lot of time doing labour-intensive work to sketch, colour, relight and render every single frame one by one. If some components of this content cycle could be automated or produced to a certain degree of accuracy using artificial intelligence, it could partially ease the labour-intensive work for the animation studio workers.

After a brief study and survey with relevant teams, some areas were identified which can help improve the classical animation graphics rendering pipeline.

1.2 Context

Rendering is the process of producing final assembled animation scenes or parts from the computer in the format of a sequence of individual frames. It is used both in 2D and 3D animation to generate a series of individual pixel-based frames or a video clip. It is the final stage in the production phase and a very technically complex aspect. Unsurprisingly, the maximum time taken by the production pipeline is the final frame/video rendering.

While currently, there is some real-time rendering possible for graphic-heavy game development but there is no real-time rendering for animation that can be exported as a stream with passes, which would reduce the time to render.

Another area that requires effort is when the studio needs to produce a similar frame as the current one. Both the current and next frames would only have a small difference, which ideally is the only part to be worked out by the studio animators. Currently, this problem can be alleviated by using already existing interpolation techniques for computing the next frames given two keyframes.

Generating a 3D render model of a face/character directly from a few images would be a good upgrade to minimise the effort in the content production pipeline. This still depends on how photorealistic or editable the generated 3D model would be, as there might again be some specific changes that might be needed to be tinkered with on top of the initial render. Shahrukh et al [3] have worked on the same in-depth to reproduce animate faces, but there is still a lot of other related research happening.

1.3 Problem Overview

A significantly important aspect of such 2D and 3D animation content production is the effort spent in lighting on subjects and their corresponding backgrounds. Though certain lighting presets exist for the same, it is believed it would smoothen the pipeline to have an artificial intelligence-based algorithm or software to mimic a photo's light setting – in terms of colour, softness, direction, emotions, situations, etc so it saves overall production time. This report takes an attempt to explore current progress in the same and attempts to alleviate the process to perform relighting in images, which then can at least be directly applied in 2D animation production if not in a much more computationally intensive 3D animation.

1.3.1 Challenges

Using classical computer graphics techniques, creating photorealistic assets is pretty taxing. Real-world objects are highly complex as they have many transparent surfaces, are glossy and have smooth structures etc.

Even modelling humans is tricky as human skin shows subsurface scattering effects and humans also have a large variety of appearances in terms of different clothing. Furthermore, the human head has a fine scarce structure in its hair, while the teeth and the eyes are specular and transparent.

More challenges in performing realistic image synthesis are discussed and contrasted in depth in chapter 2.2.1.

1.4 Aims and Objectives

This project aims to look closely at the problem of relighting photorealistic assets using image relighting, neural radiance fields, and editable neural rendering. By taking cues from conventional relighting techniques used in such situations, it attempts to investigate the potential of relightable neural rendering. In the end, it seeks to investigate the viability of relighting scenarios in environments using augmented reality. The main objectives for the project would be:

- To study and examine the various rudimentary and advanced image relighting techniques.
- To implement and analyze different approaches for image relighting.
- To reformulate the existing neural network-based approaches and propose an effective architecture to tackle improperly applied image relighting
- To perform a qualitative analysis of the reconstructed relit images in terms of multiple image quality assessment metrics.

1.5 Document Overview

In Chapter 2, a brief introduction of the previous and recent work in novel image relighting along with deep learning approaches and multiview relighting is discussed. The methods are contrasted and their advantages along with certain limitations are discussed.

Chapter 3 focuses on the experiments performed during the project. It outlines the software and hardware used, the datasets and their preprocessing. Certain inspired loss functions are talked about apart from the architecture of experiments performed. The methodology of experiments along with the changes proposed is also discussed here.

Chapter 4 focuses on the evaluation aspect, first showing the evaluation metrics used and their importance. Following the same, the result is discussed with some qualitative analysis.

Chapter 5 discusses the current limitations and reflects on possible improvements, and concludes the project by exploring possible applications with future works.

Chapter 2: Survey

2.1 Background Methods

2.1.1 Generative Adversarial Networks

A generator(G) network is used to train a discriminator(D) network to maximize the probability of assigning a correct label to both training examples and samples from generator G. Developed initially by Ian J. Goodfellow et. al [7, 2] , this is performed with simultaneously training generator *G* to minimize the function log(1 - *D(G(z))*); where z represents an input noise variable.

It can be said that simply, a discriminator D and a generator G represent a two-player minimax game with a value function V(G, D):

$$\min_G \max_D V(D,G) = E_{x \sim p_{data}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z))] \quad (2.1)$$

where *D(x)* represents the probability that *x* belongs to the sample data provided rather than the generator's distribution and $p_z(z)$ are the prior defined on the input noise variables.

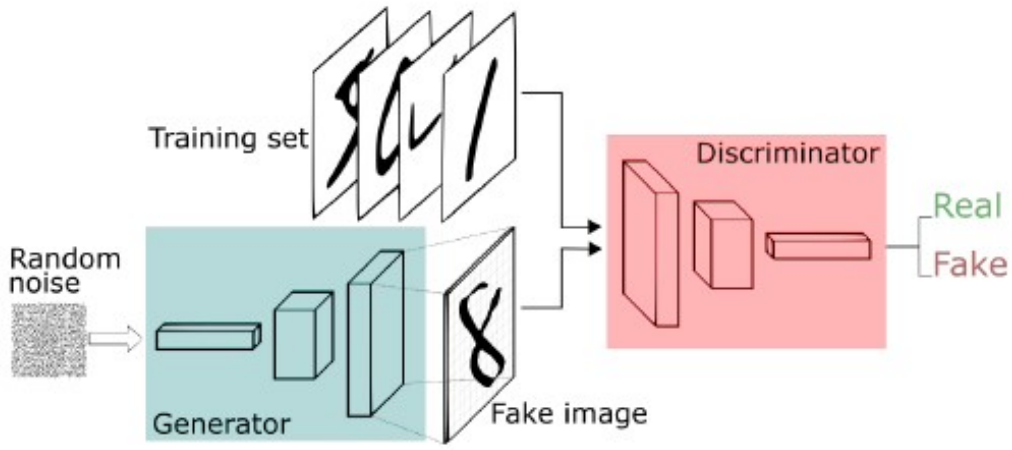


Figure 2.1: Training process for Generative Adversarial Networks ([21], figure source)

Figure 2.1 depicts the process to train a GAN network. Such generative adversarial networks have been widely used before performing the relighting process, just like Wang et al [25] do so while performing general scene reconversion. Further, they use the discriminator to differentiate the shadow regions from the ground-truth images.

2.1.2 Neural radiance fields

Neural radiance fields or commonly known as NeRFs [14], have become an active area of research. It is inspired by following a rudimentary computer graphics approach. It is a technique for view synthesis that provided a set of observed images of some object or scene, along with the corresponding camera parameters.

Initially, it uses ray marching to generate 3D sample points in space. For each of such the coordinates of these sample points, positional encoding is performed, before letting them run through a Multilayer Perceptron that outputs color and opacity. Following this, another computer graphic technique like volumetric aggregation scheme and alpha blending is used to blend all these colors based on the opacity and then generate the final pixel color. Figure 2.2 gives a good overview of the input, output, rendering and loss in neural radiance fields.

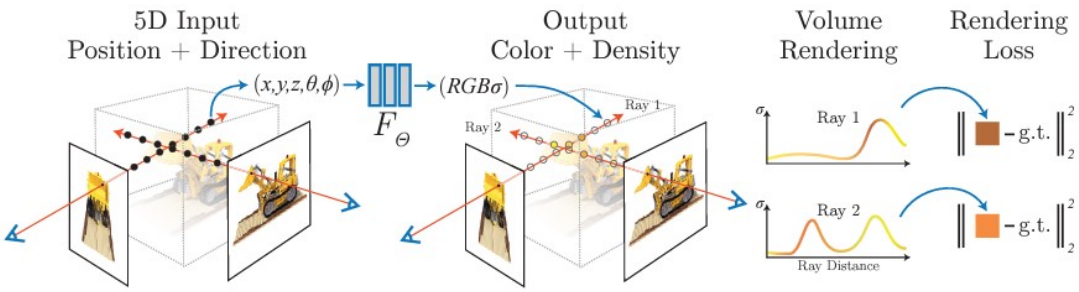


Figure 2.2: Overview for neural radiance fields ([14], figure source)

This recovers a neural radiance field, which is a three-dimensional representation of the scene that can render photorealistic images of the scene from novel, unobserved viewpoints.

NeRFs represent a scene as a continuous volume of particles that emit light. Rendering a ray that passes through a NeRF representation is simply just sampling points along the ray. The final colour is a sum of light that's emitted by particles at each point, multiplied by how much that point is blocked by the closer samples. Although NeRFs work well for rendering the same scene from novel viewpoints, it does not give a way to simulate how any point's appearance will change with new lighting conditions.

2.1.3 Relighting

A major challenge in computer graphics and computer vision is modelling the appearance of real scenes from captured images and rendering new images under novel conditions. This has been extensively researched and studied for many years now. An ideal model for appearance acquisition should allow for photorealistic rendering with variable camera angles, variable lighting, and even flexible editing. Contrary to appearance view synthesis techniques, which concentrate on replicating the same appearance and its original environment, appearance acquisition techniques genuinely aim to also change the same appearance under the given circumstances. One core problem of it is to render a scene under a novel lighting condition, which is known as relighting. Figure 2.3 depicts the process flow for image based relighting, while the Figure 2.4 shows some examples of relighting from different camera positions and viewpoints.

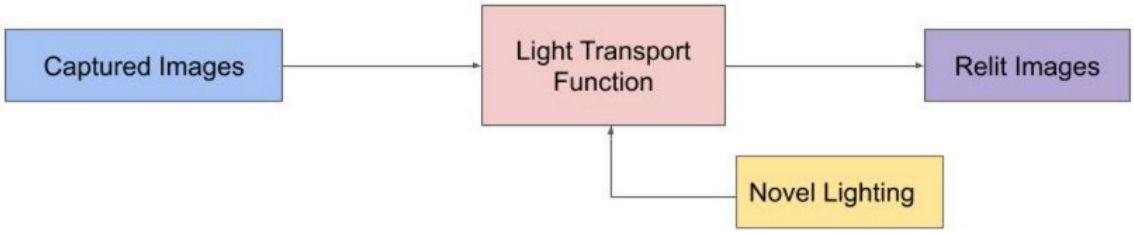


Figure 2.3: Image based relighting

Figure 2.4: An example of image relighting from different light directions [25]

2.1.4 Relighting with neural radiance fields

To enable relighting, instead of all points being light emitters, the NERF model now has to be based on a model where all points have surface normals and a reflectance function that describes how the particles reflect incoming light. Light from external sources can thereby be simulated and possibly attenuated and reflected by the particles in the scene.

In a neural reflectance field, the network outputs a surface normal direction and BRDF parameters at each point in the volume. Direct illumination considers the effects of emitted light that has just been reflected by particles along the camera ray back toward the camera. After initially querying the network, the volume density at each point along the camera ray is noted to figure out how much each of these points is blocked by closer points. For each of the points, the visibility is computed between that point and each light source by densely querying the network for volume density along each light ray. Hence, each of the dots represents a full evaluation of the fully-connected neural network. Performing this procedure for every ray in the training batch would be roughly a million training iterations per scene. Hence, simulating direct lighting with this brute force visibility sampling is way too slow during training [22]. Figure 2.5 depicts the same with only direct lighting.

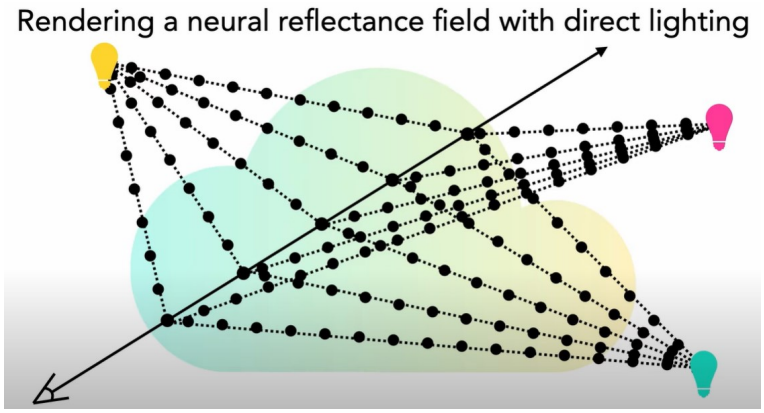


Figure 2.5: Neural radiance field with direct lighting, ([22])

A possible improvement for this formulated from recent neural reflectance fields research work is that if the lighting is controlled and the object is only illuminated by a single light source, and that light source is co-located with the camera, then all the volume densities are already being calculated to estimate the light visibility with the queries already been done along the ray's render each pixel.

Although, the training time is still not faster than a standard NeRF, it can reproduce impressive relightable models. It still lacks clarity with models under more general lighting conditions where there are many more light sources.

2.2 Related work

2.2.1 Initial developments for creating photorealistic assets

Initially, there were two proposed different methods for creating realistic synthetic imagery. For instance, one needs a high-quality asset that takes a lot of effort to generate to render highly realistic images using computer graphics (Figure 2.6). Additionally, in computer graphics, every visible parameter, such as the position of the light source or the camera, is completely controllable. On the other hand, generative machine learning methods (Figure 2.7) need a large amount of training data to build their models. However, the 3D representation can be automatically learned with enough training data, negating the need for modelling. However, there is no fine-grained control of the previously mentioned parameters in generative machine learning. Both computer graphics-based approaches and generative machine learning have particular advantages and disadvantages (Figure 2.8). So combining these two paradigms leads to a new class of approaches called neural rendering. [15]

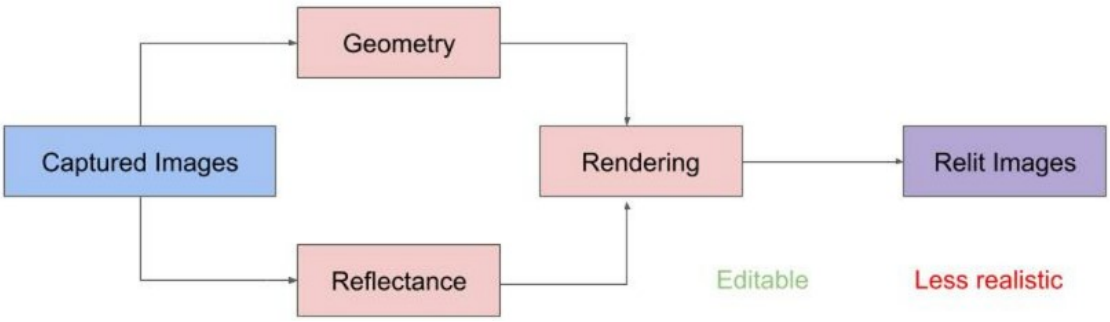


Figure 2.6: Reconstruction based imagery methods

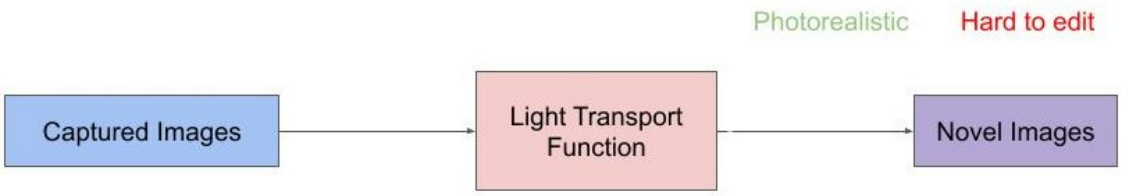


Figure 2.7: Generative machine learning based

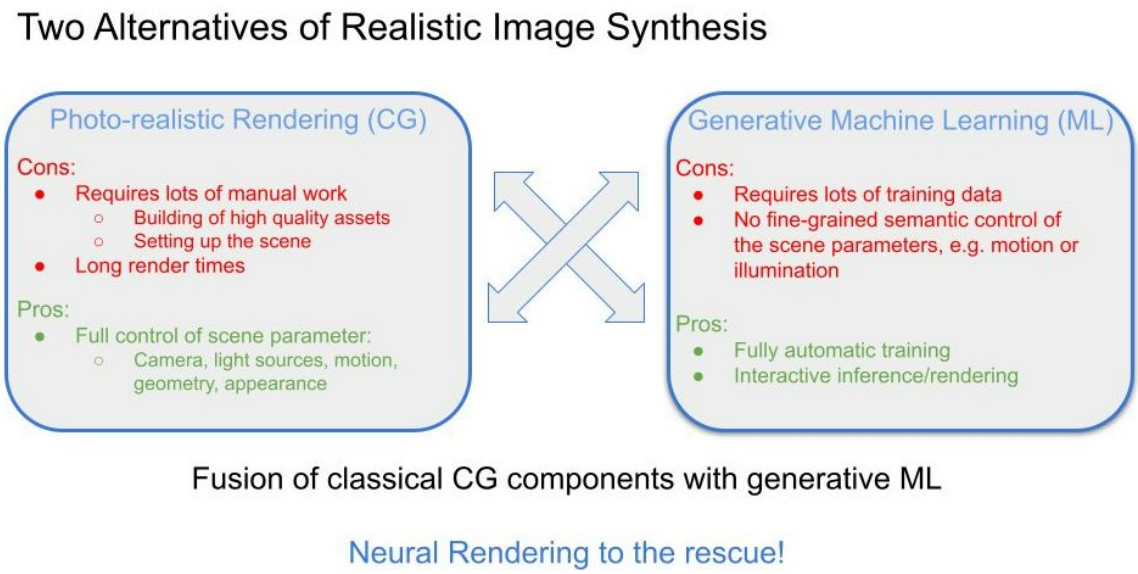


Figure 2.8: Comparison for photo-realistic rendering and generative machine learning

In the last few years, a lot of progress has been made in the research area of neural rendering. The progress made is majorly in relighting arbitrary images, which allows changing the illumination of arbitrary renderings and photographs with the help of machine learning. There has been great success combining classical computer graphics pipelines and certain learnable components. However, the main goal of previous works does not change, which is to generate compelling photorealistic imagery.

Photorealistic assets can be learned using neural rendering, as opposed to directly from image data, which is generally pretty challenging.

Deep neural networks are required to generate images or videos that can explicitly or implicitly control the properties that are being observed. A three-step process is followed. Initially, raw pixel output must be produced to generate images. Second, controlled generated images are necessary to maintain controllability. In the end, the physical parameters, such as camera position, lighting, geometry, etc., must ultimately be managed.

2.2.2 Initial approaches in relighting

Flexible appearance models [1] can carry out editing tasks like material editing, shape editing, and optic insertion. The appearance acquisition [26] and relighting are generally performed by two types of methods. By recreating the scene's geometry and material reflectance in its entirety, one can perform a full reconstruction of the scene. Once all of these components have satisfied the requirements, the entire scene can then be rendered. In other cases with image-based methods, renderings are output directly from a black box function rather than fully explicit reconstruction.

Typically, in this case, a black box function is referred to as a light transport function (Figure 2.7). Usually referred to as a light transport function, this black function. Such image-based techniques can produce photorealistic results, but editing the same content is very challenging. While the initial approach, reconstruction-based modelling has additional geometry and reflection elements that naturally permit scene editing, but not the trade-off is that the rendered images are typically less realistic as a result.

Relighting involves images which generally consider a fixed viewpoint. It first captures some images of the scene under certain lighting conditions and then synthesizes new images of the scene under a specific novel lighting condition. The requirement is to keep the same content unchanged and only change the lighting in the image.

This was first achieved using a sophisticated device called "Light Stage" in 2000 by Debevec et al. [6] (Figure 2.9) that presented a similar relighting work to capture a lot of images under different lighting conditions. This was performed using a typical light stage which then could direct generate relit images by linearly combining the captured images. The relighting results were very realistic, but they required more than 2000 input images to produce satisfactory results. After that, a lot of traditional and neural methods have been presented to improve the rendering quality and acquisition efficiency of image-based relighting.

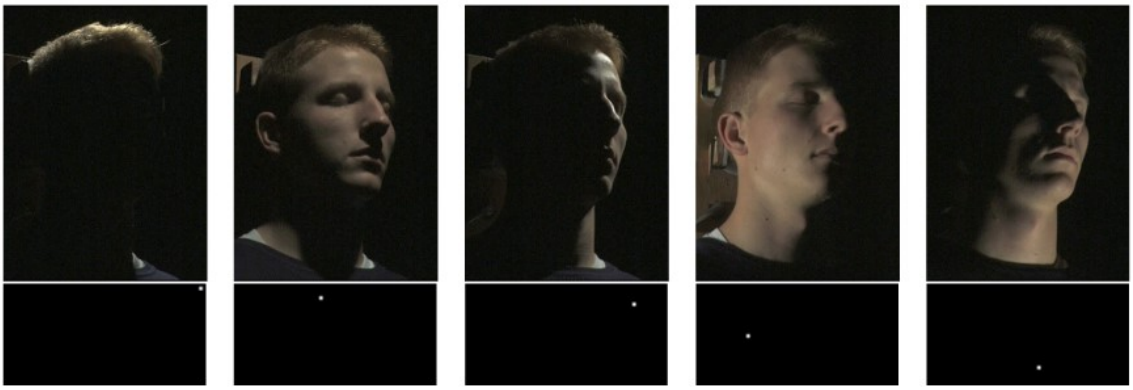


Figure 2.9: Light stage images to produce a relit output ([6], figure source)

2.2.3 Deep learning approaches with relighting

The first relighting work with neural networks was presented in 2015 by Ren et al.[18] (Figure 2.10). It distributed a lot of small multi-layer perceptrons (MLPs) in the image space and performs a scene-wise optimization of the MLPs on input images in multiple hundreds for relighting. Similar works have used localized MLPs for image or scene representation, but still use thousands of images, making it very expensive.

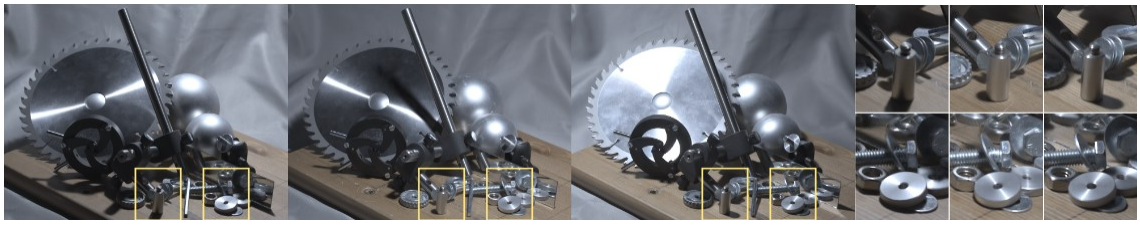


Figure 2.10: Relighting results with neural networks implementation ([18], figure source)

Xu et al.[28] proposed the first relighting technique that can generalize across multiple scenes. It proposed to use deep convolutional neural networks to do relighting. It could relight a scene from only five input images and outputs a new image under any normal lighting condition as specified. It can synthesize photorealistic relighting results on a diverse range from these five captured images. It first linearly combines them and then relit(s) the image(s) under different lighting conditions to enable relighting in complex environments. It even does well to reproduce complex shading effects like challenging specularities and soft shadows.

Moreover, under a fixed viewpoint, the input and output of the relighting are well aligned. This initial approach has proven to be very suitable by applying the convolutional neural networks (CNNs) and since then, a lot of CNN-based relighting papers have been presented focusing on doing human relighting, especially portrait relighting [24] (Figure 2.11)

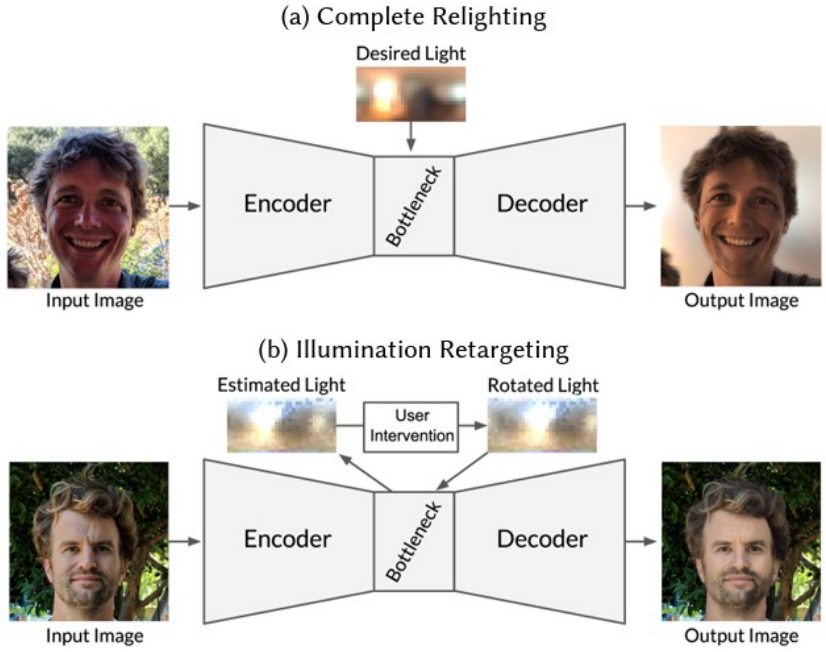


Figure 2.11: Illumination retargeting and human potrait complete relighting ([24])

When performing portrait relighting, due to the strong category-specific priors, these scene-based methods often handle lighting estimation jointly and do relight from a single input image. While in classical image relighting the viewpoint is fixed to an extent in most of the research works.

The more challenging problem is when the viewpoint changes jointly while the relighting is being performed. A potential solution for ht same is simply running view synthesis and relighting techniques in a sequence together. Again, Xu et al. [27] works on this, and it now performs the view synthesis technique combined with the relighting technique and hence performs novel view relighting from a sparse set of images captured under different lighting and viewpoints (Figure 2.12).

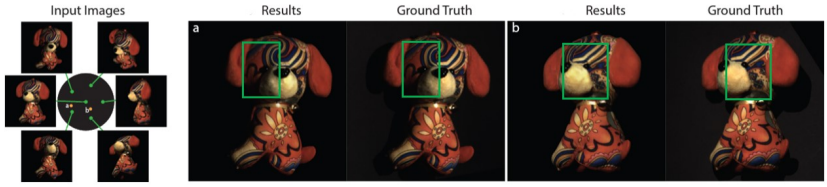


Figure 2.12: Novel View Synthesis from single-light network ([27], figure source)

2.2.4 Multiview relighting and appearance acquisition

Instead of performing relighting and view synthesis, recent work focuses on designing specific techniques for multiview relighting. These were inspired by using a pre-computed mesh as a geometry-based proxy, that would align the multiview input images to a novel viewpoint and then use a deep neural network to learn the relighting function for any viewpoint possible [17] (Figure 2.13. Ideally, this approach combines concepts from image-based rendering and combines them with image-based relighting using deep neural networks. These methods are still image-based, but they leverage some level of geometry reconstruction.

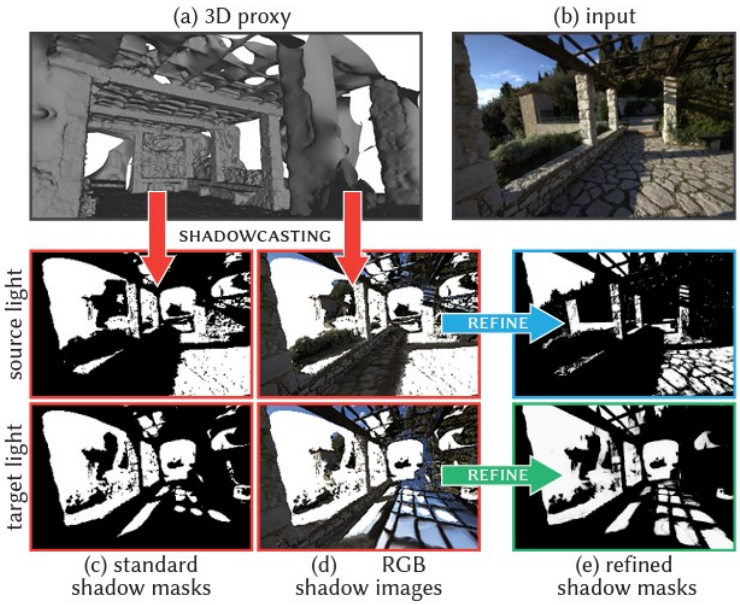


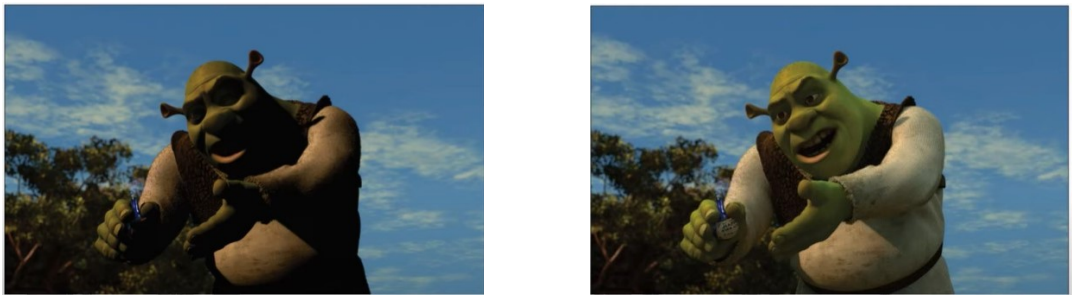
Figure 2.13: Using geometry proxy shadow refinement to improve relighting results ([17])

As seen earlier, another way of relighting and appearance acquisition is to do an explicit reconstruction of the geometry and reflectance. There has been previous work in the reconstruction space but a lot of recent deep learning-based reflection acquisition methods [5] often consider only flat material samples without any complex geometry. Other advanced works study complete appearance acquisition with a complex geometry, which is then a more challenging task to achieve.

Kang et al.[11] proposed to use a neural network to drive the acquisition process of a specifically defined acquisition setup. It could recover the complex appearance and shape of real objects and do realistic view synthesis and relighting with very high quality. The limitation was the need to recreate their expensive acquisition setup and it could only work in very specific conditions.

2.3 Relightable NERFs

Srinivasan et al. [23] were among the first ones to attempt relighting and view synthesis with neural radiance fields. A major problem faced in the same was simulating indirect illumination for recovering accurate 3D representations. Indirect lighting can have a significant impact on the appearance observed in the images, so that needs to take take into account at the same time.



(a) Direct illumination

(b) Indirect illumination

Figure 2.14: Effect of considering indirect illumination in a 3D Shrek (Credit: Eric Tabellion, Dreamworks)

The famous animation character Shrek looks like with just direct environment illumination (Figure 2.14 (a)), in comparison to what it looks like if one model's the indirect illumination that is bouncing off the character's body and the ground. Indirect illumination has a significant impact on the appearance and it makes it very important to model the same into the training process to recover the representations. If not considered, it might lead to estimating incorrect geometry and incorrect reflection parameters when trying to simulate the difference in appearance as indirect lighting can arrive at a scene point from all directions.

In Figure 2.15, it can be seen the differences in results in experiments performed by Srinivasan et al. [23] using indirect illumination and direct illumination.

As shown above, in Figure 2.16, Guo et al [8] discusses in detail object-level light and modelling how light transports for each object. It further elaborates on once there is a learned model for each object, the main aim is to compose these different objects into a scene using scene-level light.

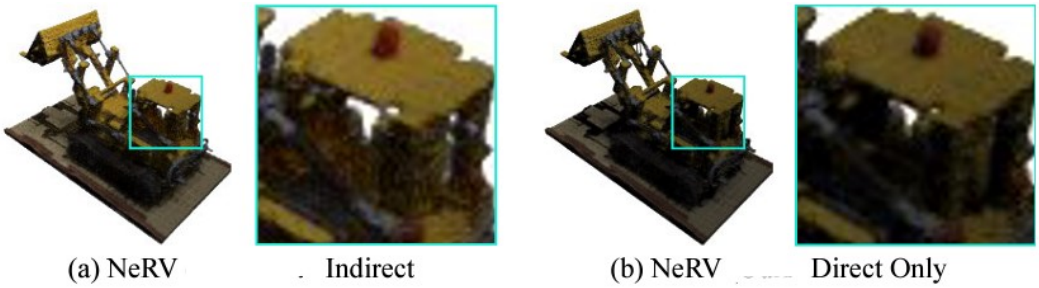
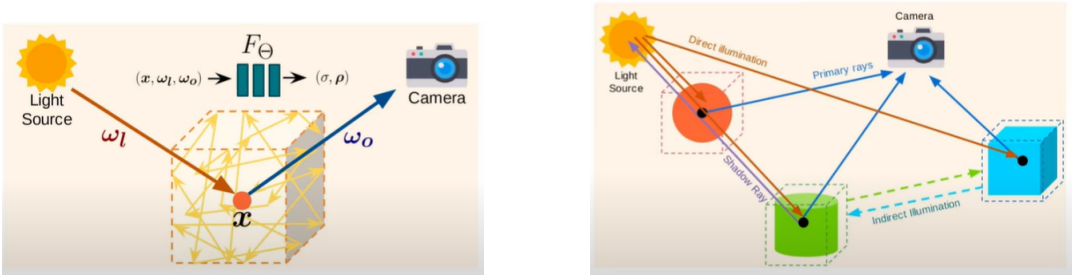


Figure 2.15: Difference with indirect and direct illumination [23], figure source



(a) Object-Level Light

(b) Scene-Level Light

Figure 2.16: Difference with indirect and direct illumination ([8], figure source)

In one of the more advanced deep neural approaches, Meka et al [13] shows how to depict deep relightable textures by leveraging deep neural networks. The implicit features are learnt by modelling the classical rendering process. The view-dependent appearance of the subject, independence from the geometry layout, allowance for generalization to unseen subject poses and novel subject identification is represented by these features.

Chapter 3: Experiments

3.1 Software and Hardware

The project mainly had python 3.7 as the main coding language, while bash scripting was used a little bit to do dataset and image preprocessing. The deep learning libraries used were majorly PyTorch and TensorFlow 1.3.

The hardware used while performing the experiments was initially a workstation laptop with NVIDIA MX 450, an Intel processor with 16 GB RAM. This was not very sufficient in the end, and then the other experiments were performed using Google Colab Pro+ machines.

3.2 Datasets

The VIDIT 2.0 (Virtual Image Dataset for Illumination Transfer [9]) was initially used for experimentation. It has three hundred ninety virtual scenes with various scene elements like wood and metal that are generated by the Unreal gaming engine to create high-resolution visuals. Forty five scenarios each were used for testing and validation, leaving about three hundred scenes for training. The core idea behind this VIDIT dataset is basic illumination manipulation. Each scene initially gets rendered with eight light directions or even five possible colour temperatures, which in turn possibly produces forty images with a resolution of 1024*1024. They can be further used to make shadow-free photos. The overall goal is to have the network architecture estimate the amount of illumination for any particular light source given a picture taken under any sort of illumination.

Another data source was via Sang et al. [19] where they performed single shot neural rendering. They use a large-scale rendered synthetic dataset. They built their synthetic dataset based upon the DiLiGent dataset introduced by Shi et al. [20].

3.3 Loss functions

There has been a multitude of loss functions used in generative machine learning and various methods of image relighting approaches discussed in chapter 2. Although, a lot looks promising, for the experiments performed the major loss functions used were directly inspired and reprocessed from a portrait image relighting approach [16].

It consists of a relighting module, which takes an input of a target illumination map and a predicted foreground from a mapping module. This design along with the deep relighting module includes several U-Net architectures in a chain. Their initial network estimates the subject's geometry; the second network estimates the subject's diffused albedo (reflectance property). It has been proven by such recent prior work on relighting, that using a supervising network training with these intrinsic components can improve relighting results.

All of the loss functions are developed to be compared with the ground truth images. A few common ones are geometry-image loss (between ground truth surface normals and predicted image normals) and relit image loss. There are similar VGG shaded and VGG albedo loss functions on the relit images, wherein the squared L2 distance between target features/albedo is calculated with predicted features/albedo images using a VGG network pre-trained on ImageNet classification task [29]. Furthermore, a network can be trained to focus on particular regions that are likely to contain specular information in the images. These are usually challenging for the neural network to synthesize and this thereby acts as an attention mechanism for the network. This is term as specular-weighted relit image loss as it indirectly encourages the network to produce specularities.

Hence, it came to a conclusion using such loss functions would be more valuable as they would take care of per-pixel reconstruction error and perceptual differences, hence taking care of relighting too.

3.4 Architecture

One of the main changes in the initial experiment was directly inspired by Bi et al. [4]. They discussed the advantages of using a late-conditioned model that incorporates the additivity of light in the network architecture (Figure 3.1). In general, it adds the light parameters and properties after an initial convolutional neural network fuses these parameters along with a second network pass with extracted features, right before producing the colour of each texel. It is believed this performs better than using a direct early condition model (wherein lighting information is passed in the initial pass itself). Although this was not completely adopted in the experiments, it was inspired by the "Additive Lighting infrastructure" and a part of the same ideology was implemented as it is expected to generalize better.

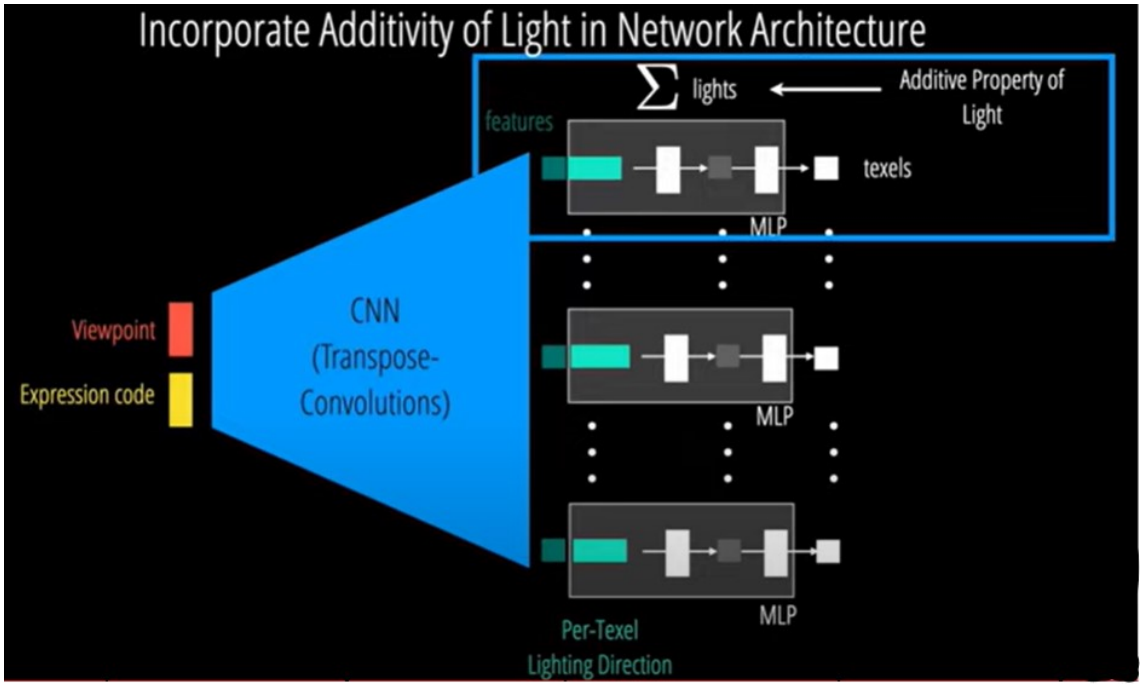


Figure 3.1: Late conditioned model with additive light, ([4], figure source)

3.5 Reconstruction of images with relighting

The idea here, as discussed in Chapter 3.2, is trying to use the high-resolution synthetic dataset and explore illumination areas. While the reconversion scene network was not giving good reconstruction results, initially the problem was focused on using generative adversarial networks to create a dataset using data augmentation. It was being done parallelly in attempts to generate new images and subsequently create a new dataset as it could potentially be used to train/generate new images and would have more value. This was then added to the initial dataset and the experiment was reperformed.

Different approaches have been tried to fix up some problems faced will like the CUDA memory issue. Initially, the images have been tried to be loaded one by one in the NVIDIA MX450 GPU due to fewer GPU RAM constraints. Another thing that has been ensured is that all variables have been written properly. In the end, what solved some issues was changing the high resolution of the image that was changed from 1024*1024 to 512*512 in comparison. Additional precautions were taken to ensure good code quality and good image outputs to be produced by solving current and potential linking issues. Figure 3.2 depicts some output results from the experiment on VIDIT dataset. Due to unsatisfactory results in the reconstruction and not much improvement in illumination results, there was a need to perform an experiment with a good reconstruction base network and build upon with a more varied illumination network. The same is performed in the following section.

Figure 3.2: Relighting after reconstruction with VIDIT 2.0 dataset

3.6 Single shot neural rendering and relighting

In this experiment, initially the shapes and the BRDF parameters of the synthetic dataset from Li et al [12] are adopted. These are used to render a new dataset altogether. A pre-rendering layer is implemented using the BRDF model and equations defined. It is the coded using the PyTorch deep learning framework with NVIDIA CUDA acceleration.

An interesting empirical test in the same dataset, inspired from DiliGENT [12], the images now are rendered in an online manner instead of pre-rendering all the training set of images. The rendering of the relighting the target image(s) is done under random point light sources for every single iteration. This ensures that the model ideally sees more varied samples due to the ground truth being rendered with a larger set of lights in comparison with an offline rendering method (pre-rendering).

Initially the experiment is performed with the late conditioned model discussed in section 3.4 and then it is performed with this single shot neural rendering approach. Figure 3.3 depicts the input images for both experiments. Only the latter experiment results are depicted in Figure 3.4 and are considerably better than the first late conditioned model experiment.

The green rectangles in Figure 3.4 clearly show the differences in the illumination in certain backgrounds and side views of the animation character in the figure.

Figure 3.3: Initial input images for network

Figure 3.4: Relighting output

Chapter 4: Evaluation

4.1 Evaluation Metrics

Some common evaluation metrics in our application are Structural similarity index metric (SSIM), Mean square error (MSE), Mean SSIM (MSSIM), Peak Signal to Noise Ratio (PSNR) and Colour Image Quality Measure (CQM) [2, 10]. They can be used as metrics to measure the similarity between the original image and the corresponding reconstructed image.

PSNR is measured in decibels(dB).

$$PSNR = 10 \times \log_{10}(\frac{MAX^2}{MSE}) \tag{4.1}$$

Here, MAX is the upper range of the pixels of the input image.

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \|O(i,j) - D(i,j)\|^2 \tag{4.2}$$

where $O(i,j)$ is the original image pixel and $D(i,j)$ is the regenerated image pixel.

$$CQM = (PSNR_Y \times R_W) + (\frac{PSNR_U + PSNR_V}{2}) \times C_W \tag{4.3}$$

where the Y, U and W channel components are extracted from the images and their respective PSNR values are found to compute CQM. Also, $C_w = 0.0551$ and $R_w = 0.9449$, which are the constants corresponding to the rods and cons. CQM is also measured in dB decibels like PSNR.

A greater value of CQM represents greater image similarity. Despite CQM being used due to better chrominance and luminance perception, the SSIM metric is considered superior to other metrics like PSNR and MSE as SSIM considers image degradation as perceived change in structural information (stronger inter dependencies on spatially close pixels) - more important information, whereas PSNR just estimates perceived errors. The range of SSIM is between -1 and 1, with a maximum of 1 possible with similar images.

4.2 Results and discussions

Table 4.1: Metrics for different datasets

Dataset	MSE	SSIM	MSSIM	PSNR	CQM
VIDIT Exp1	0.019	0.912	0.9321	30.89	35.21
VIDIT Exp2	0.024	0.9042	0.9192	29.53	36.01
DiLiGent 2.0 Exp1	0.015	0.939	0.963	31.63	37.32
DiLiGent 2.0 Exp2	0.026	0.881	0.939	30.15	34.37

Table 4.2: PSNR and SSIM values for DeblurGAN

Experiment	**Experiment1**	**Experiment2**	**Experiment3**
Metrics	Synthetic	LateCon	Single Shot
PSNR	24.9	24.23	26.16
SSIM	0.809	0.781	0.812

Figure 4.1: Initial Output from network

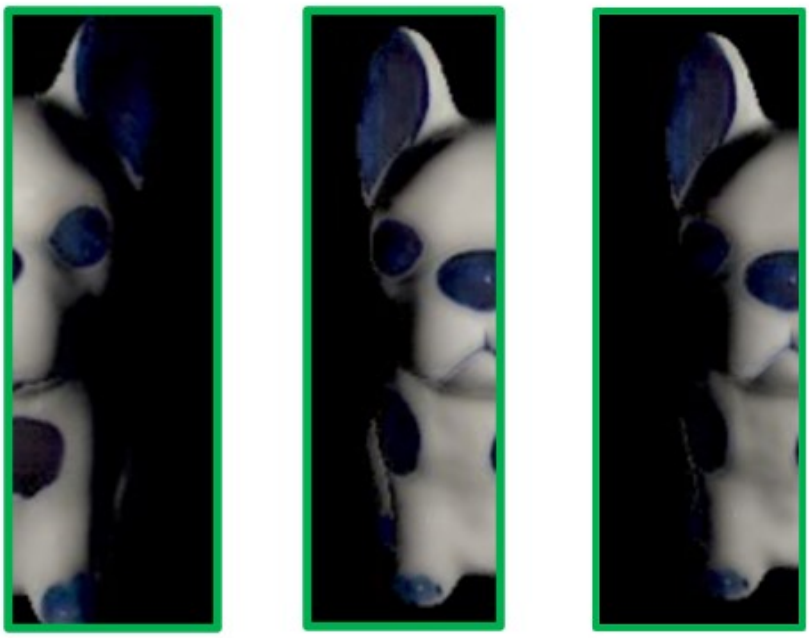


Figure 4.2: Zoomed output for illumination differences in output from character

As seen in Table 4,1 the results are much better on the DiLiGent 2.0 dataset, this can be due to a clean black background with easy reconstruction of image (unlike VIDIT dataset) and more focus on illumination regions.

In Table 4.2, we can clearly see the best results of PSNR and SSIM came with the single shot rendering approach, instead of the late conditioned model or synthetic VIDIT reconstruction approach. This should be attributed to best illumination single shot algorithm for experiment three.

In Figure 4.1 and Figure 4.2, we can see the initial output and zoomed out output from experiment 3.

Chapter 5: Conclusion

5.1 Limitations

In cases wherever a human suspect was present, the lighting was improper specifically in clothing parts like sections of a shirt. It is caused by the shading effect rather than surface reflectance. Additionally, the relighting network is not able to generate the specular highlights in a character's eye region. For portrait relighting, this is very crucial and needs to be fixed. In a side experiment, when the network was applied to a small video sequence, it exhibited temporal inconsistencies and flickering in the prediction.

5.2 Reflection

The initial experiment performed on the VIDIT dataset did perform well for the illumination aspect, but the major drawback was the poor reconstruction of the high-resolution synthetic images. Multiple approaches were attempted to fix the same, but despite performing good illumination, the image reconstruction results were quite variable.

Other experiments performed with the late conditioning model and new reconstructed data using a generative adversarial network only had minimal improvement. Hence, it would be insufficient to justify the high increase in compute time.

In the end, the experiment with single-shot neural rendering produced very satisfactory relighting results. The pre-rendering of the images was done in an online manner and when combined with the late conditioning model, this was able to perform relighting with a good range of metric values like a PSNR of 26.16.

5.3 Applications

These neural rendered outputs can be used for data augmentation as they provide more viewpoints possible captured. In addition, they can give arbitrary lighting conditions and arbitrary backgrounds where compositing with alpha masks, hence adding more diversity to training data. Hence, it can be said that this neural rendering pipeline could potentially be better than a state-of-the-art geometric pipeline in terms of photorealism, geometry and alpha masks.

Other applications where neurally rendered data could be used apart from relighting lighting estimation are alpha matting, depth estimation, pose estimation, virtual reality and augmented reality.

5.4 Future Work

In general, these neural renderings use underlying representations to influence the output space. The majority of facial and full-bodied re-enactment methods, for example, assume a fixed view. This allows them to reuse static content like the background.

There is a strong need to be able to change the viewpoint of virtual reality or augmented reality applications. The ability to synthesize images and videos of humans is of paramount importance for teleconferencing applications in virtual reality or augmented reality. It also allows us to implement powerful movie editing and post-production tools for both capturing and synthesis. Such artificial intelligence-based processes show the promising state-of-the-art results and are quite reliable.

Furthermore, some neural renderers have additional learnable features in the form of neural textures, vertex textures, or multilayer perceptron that is either defined on the surface or in 3D space. There is the additional disentanglement of foreground and background, but the major significance is the disentanglement of illumination that plays an important role.

Lastly, there needs to be further work on bringing new rendering primitives to the same level of editability and animation capabilities that when working with generic animation polygonal meshes. The main focus has to be on combining editing, appearance and animating that can lead to high-quality editable neural rendering.

Moreover, under certain conditions, one can edit lighting, camera, location, materials and other properties but there still exists a gap when one compare's to traditional rendering. To perfect relightable neural rendering, we need to gain inspiration from more traditional editing workflows and investigate how to reproduce them under a different toolset - neural rendering. This research is quite pivotal on how it can revolutionise the animation industry.

Appendix A: Appendix

All the code for this project and image data is present at this github link :
https://github.com/sarthakahuja11/Relighting$_{A}ssets$
(Now it is a public repository, unlike a private one during the project)

Bibliography

[1] *Flexible Appearance Model*, pages 61–73. Springer US, Boston, MA, 2004.

[2] Sarthak Ahuja, C. Udaya Kumar, and S. Hemalatha. Competitive coevolution for color image steganography. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pages 719–723, 2019.

[3] ShahRukh Athar, Zhixin Shu, and Dimitris Samaras. Flame-in-nerf: Neural control of radiance fields for free view face animation. *arXiv preprint arXiv:2108.04913*, 2021.

[4] Sai Bi, Stephen Lombardi, Shunsuke Saito, Tomas Simon, Shih-En Wei, Kevyn Mcphail, Ravi Ramamoorthi, Yaser Sheikh, and Jason Saragih. Deep relightable appearance models for animatable faces. *ACM Trans. Graph.*, 40(4), jul 2021.

[5] Sai Bi, Zexiang Xu, Kalyan Sunkavalli, David Kriegman, and Ravi Ramamoorthi. Deep 3d capture: Geometry and reflectance from sparse multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5960–5969, 2020.

[6] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, page 145–156, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[8] Michelle Guo, Alireza Fathi, Jiajun Wu, and Thomas A. Funkhouser. Object-centric neural scene rendering. *CoRR*, abs/2012.08503, 2020.

[9] Majed El Helou, Ruofan Zhou, Johan Barthas, and Sabine Süsstrunk. VIDIT: virtual image dataset for illumination transfer. *CoRR*, abs/2005.05460, 2020.

[10] S. Hemalatha, U. Dinesh Acharya, and A. Renuka. Wavelet transform based steganography technique to hide audio signals in image. *Procedia Computer Science*, 47:272–281, 2015. Graph Algorithms, High Performance Implementations and Its Applications (ICGHIA 2014).

[11] Kaizhang Kang, Cihui Xie, Chengan He, Mingqi Yi, Minyi Gu, Zimin Chen, Kun Zhou, and Hongzhi Wu. Learning efficient illumination multiplexing for joint capture of reflectance and shape. *ACM Trans. Graph.*, 38(6), nov 2019.

[12] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM Trans. Graph.*, 37(6), dec 2018.

[13] Abhimitra Meka, Rohit Pandey, Christian Haene, Sergio Orts-Escolano, Peter Barnum, Philip Davidson, Daniel Erickson, Yinda Zhang, Jonathan Taylor, Sofien Bouaziz, Chloe Legendre, Wan-Chun Ma, Ryan Overbeck, Thabo Beeler, Paul Debevec,

Shahram Izadi, Christian Theobalt, Christoph Rhemann, and Sean Fanello. Deep relightable textures - volumetric performance capture with neural rendering. volume 39, December 2020.

[14] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[15] Ren Ng, Ravi Ramamoorthi, Pratul Srinivasan, and Jon Barron. Advances in Neural Rendering (SIGGRAPH 2021 Course) Part 1 of 2 — youtube.com. `https://www.youtube.com/watch?v=otly9jcZ0Jg`, 2021. [Accessed 12-Sep-2022].

[16] Rohit Pandey, Sergio Orts-Escolano, Chloe LeGendre, Christian Haene, Sofien Bouaziz, Christoph Rhemann, Paul Debevec, and Sean Fanello. Total relighting: Learning to relight portraits for background replacement. volume 40, August 2021.

[17] Julien Philip, Michaël Gharbi, Tinghui Zhou, Alexei A. Efros, and George Drettakis. Multi-view relighting using a geometry-aware network. *ACM Trans. Graph.*, 38(4), jul 2019.

[18] Peiran Ren, Yue Dong, Stephen Lin, Xin Tong, and Baining Guo. Image based relighting using neural networks. *ACM Trans. Graph.*, 34(4), jul 2015.

[19] Shen Sang and M. Chandraker. Single-shot neural relighting and svbrdf estimation. In *ECCV*, 2020.

[20] Boxin Shi, Zhipeng Mo, Zhe Wu, Dinglong Duan, Sai-Kit Yeung, and Ping Tan. A benchmark dataset and evaluation for non-lambertian and uncalibrated photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):271–284, 2019.

[21] Thalles Santos Silva. A short introduction to generative adversarial networks. *https://sthalles.github.io*, 2017.

[22] Pratul Srinivasan. Relightable neural radiance fields, advances in neural rendering, siggraph, 2021.

[23] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7495–7504, 2021.

[24] Tiancheng Sun, Jonathan T Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul E Debevec, and Ravi Ramamoorthi. Single image portrait relighting. *ACM Trans. Graph.*, 38(4):79–1, 2019.

[25] Li-Wen Wang, Wan-Chi Siu, Zhi-Song Liu, Chu-Tak Li, and Daniel P. K. Lun. Deep relighting networks for image light source manipulation. In *Computer Vision – ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part III*, page 550–567, Berlin, Heidelberg, 2020. Springer-Verlag.

[26] Tim Weyrich, Jason Lawrence, Hendrik P. A. Lensch, Szymon Rusinkiewicz, and Todd Zickler. Principles of appearance acquisition and representation. *Found. Trends. Comput. Graph. Vis.*, 4(2):75–191, feb 2009.

[27] Zexiang Xu, Sai Bi, Kalyan Sunkavalli, Sunil Hadap, Hao Su, and Ravi Ramamoorthi. Deep view synthesis from sparse photometric images. *ACM Trans. Graph.*, 38(4), jul 2019.

[28] Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, and Ravi Ramamoorthi. Deep image-based relighting from optimal sparse samples. *ACM Transactions on Graphics (TOG)*, 37(4):126, 2018.

[29] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.