Creating a Photorealistic World from Casual Lighting Capture

Yifan Wang

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2024

Reading Committee:

Steven M. Seitz, Chair

Brian Curless, Chair

Richard Szeliski

Program Authorized to Offer Degree:
Computer Science & Engineering

University of Washington

Abstract

Creating a Photorealistic World from Casual Lighting Capture

Yifan Wang

Co-Chairs of the Supervisory Committee:
Steven M. Seitz
Computer Science & Engineering

Brian Curless
Computer Science & Engineering

Light is the most crucial phenomenon for us to perceive and interact with the world. It plays a fundamental role in how we navigate, recognize, and interpret our surroundings. Throughout human history, we have sought to capture and record different lighting effects, from the earliest forms of painting to the invention of photography and the development of video technology. These mediums have allowed us to document and study the world in increasing detail. Many psychological studies have shown that the human visual system excels at deducing depth, shape, and motion from lighting effects such as shading and shadows. This ability underscores the importance of accurately simulating these effects in the creation of a photorealistic world.

Creating photorealistic images and virtual environments has been a long-standing goal in the field of computer graphics, driven by applications in virtual reality, film production, and architectural visualization. Achieving high levels of realism requires accurately simulating the interaction of light with various objects and surfaces, as well as generating detailed highlights and realistic shadows. Casual capture — everyday photos and videos taken with consumer devices — offer a rich source of data for understanding and replicating real-world lighting effects. In this thesis, I explore techniques for enhancing photorealistic image synthesis by leveraging casual lighting capture. The ultimate goal is to create highly realistic and immersive visual experiences from novel viewpoints, novel scene configurations,

and novel illumination.

I begin with an overview of the psychophysics of light, focusing on human perception and the use of shading techniques in western art, providing a foundational understanding of how lighting effects influence visual perception. This foundational knowledge is critical for the subsequent development of methods that accurately replicate the nuances of lighting and shading in synthetic imagery.

The core contributions of this thesis are as follows: First, I present *People as Scene Probes*, a method that infers depth, occlusion, lighting, and shadow information from video sequences captured from a single camera viewpoint. This technique enables realistic image composition by accurately modeling scene geometry and shading effects. Second, I introduce *Repopulating Street Scenes*, a framework that uses learned scene properties from image collections to automatically reconfigure street scenes by populating, depopulating or repopulating them with objects such as pedestrians or vehicles. It enables the realistic removal of existing objects along with their shadows and the insertion of new objects by accurately matching the lighting and casting shadows. This method enhances privacy and generates diverse training data for autonomous driving applications.

Next, I introduce *SunStage*, a lightweight capture setup that replicates the functionality of a light stage using only a smartphone camera and the sun as the light source. With a video of an individual rotating in-place under the sun, SunStage reconstructs a physical model of the subject and the scene lighting, which enables applications such as relighting the subject with realistic reflections and cast shadows. SunStage allows arbitrary lighting and reflectance control in the reconstructed physical space, which can be rendered to produce photo-realistic results. I demonstrate several applications such as editing skin reflectance, relighting, and view synthesis.

Finally, I present *Infinite Texture*, a method for generating arbitrarily large texture images from a text prompt. This technique supports applications in 3D rendering and texture transfer, ensuring consistent shading and depth through the use of a minimal dataset. I demonstrate the effectiveness of this approach in generating high-resolution, high-quality

textures that can be seamlessly integrated into various downstream tasks. This work represents significant progress towards the goal of generating high-quality graphics assets from natural language descriptions.

TABLE OF CONTENTS

Page

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my wonderful advisors, Steve Seitz and Brian Curless. Their unwavering support, guidance, and encouragement have been the bedrock of this journey. Steve, your profound insights and your ability to see the big picture have been invaluable. Brian, your meticulous attention to detail and your knack for solving challenging problems have greatly shaped my approach to research. Both of you have not only been exceptional mentors but also inspiring role models. Your dedication to my success has been a source of immense motivation, and I am incredibly fortunate to have had the opportunity to work with you both.

I would also like to extend my sincere thanks to my committee members, Rick Szeliski and Mark Haselkorn, for their constructive feedback and thoughtful suggestions, which have significantly enriched the quality of this thesis. Rick has been my role model and source of inspiration since my early days of research. Being a prominent figure in computer graphics, it has been an honor to work with him and have him on my committee. I have learned a great deal from my experience with Rick, including essential research skills such as documenting all my ideas and maintaining a list of written summaries of related papers. These practices have been invaluable in organizing my research and ensuring thoroughness in my work.

A heartfelt thank you goes to my close collaborator and friend, Aleksander Hołyński. He played a pivotal role in my research, and his pursuit of perfection has set a golden standard for me. Aleks is my go-to whenever I am stuck, need advice, or feel down. His encouragement has been crucial in helping me overcome challenges and persist through difficult times. Without his support and guidance, my work would not have reached its current level of depth and quality. He has inspired me to always try harder and do better.

I am also grateful for the opportunities and experiences I gained during my internships at ByteDance, Google Research, Adobe, and Project Starline. These experiences have

broadened my horizons and provided me with practical skills that have been crucial to my research. I would like to thank Jianchao Yang, Linjie Luo, Xiaohui Shen, Xing Mei, Andrew Liu, Richard Tucker, Jiajun Wu, Noah Snavely, Shangzhe Wu, Angjoo Kanazawa, Xiuming Zhang, Xuaner Zhang, Marc Levoy, Dor Verbin, Stephen Lombardi, Lukas Murmann, Pratul Srinivasan, Clément Godard, Philipp Henzler, Peter Hedman, Ricardo Martin-Brualla, and Ryan Overbeck for their assistance and encouragement. Their contributions made my experience unforgettable, significantly enriching my learning and professional development.

I would also like to express my gratitude to my labmates and fellow researchers: Qi Shan, Supasorn Suwajanakorn, Aditya Sankar, Edward Zhang, Soumyadip Sengupta, Konstantinos Rematas, Haisen Zhao, Chenming Wu, Ricardo Martin-Brualla, Aleksander Hołyński, Xuan Luo, Keunhong Park, Yashraj Narang, Beibin Li, Wenjun Wu, Jeong Joon Park, Adam Fishman, Alice Gao, Nikita Haduong, Vivek Jayaram, Teerapat Jenrungrot, Ben Jones, Johanna Karras, Benlin Liu, Jingwei Ma, Yuxuan Mei, James Noeckel, Roy Or-El, Mengyi Shan, Meng-Li Shih, Mehmet Saygin Seyfioglu, Isaac Tian, Bowei Chen, Chung-Yi Weng, Xiaojuan Wang, Kuo-Hao Zeng, and Luyang Zhu.

Lastly, and most importantly, I would like to thank my family for their unwavering love and constant support. This thesis would not have been possible without your encouragement and belief in me. I would also like to extend my gratitude to Congjing Zhang for her support. Thank you for always being there for me, providing strength and motivation throughout this journey. A very special shoutout to the two cutest Ragdolls, Lil CC and Big Jamie, for their companionship and the joy they brought into my life during this journey.

Chapter 1

INTRODUCTION

Light, essentially a stream of mass-energy units known as photons emitted from sources such as the sun or artificial lights, is the most crucial tool for humans to perceive and interact with the world. These photons are byproducts of atomic and subatomic processes, and span a spectrum of energy. Only those within a specific range are discernible to the human eye as visible light. This range corresponds to wavelengths that trigger responses in the retinal cells, enabling vision.

Among our five senses, humans rely heavily on vision (sense of sight) to interact with the world. Light grants us the ability to navigate, recognize, and interpret our surroundings: shadows help us discern the position and contours of objects, aiding in spatial awareness and movement; bright and dim areas indicate obstacles and open paths, guiding our navigation through complex spaces; and the interplay of light and shadow also gives us information about the time of day and weather conditions, further informing our decisions and actions as we move through different environments.

Human perception of these lighting effects is so effortless that we take it for granted. However, consider the complexity involved in observing even the most basic visual scene. Our eyes receive two small, inverted images, yet our brain interprets these as a coherent, three-dimensional space. This process is "nothing short of a miracle", as described by the neuropsychologist Richard Gregory [48]. This processing involves adjusting to varying light levels, discerning subtle changes in color and brightness, and compensating for different light sources and conditions. The human visual system's ability to adapt to different lighting scenarios is a testament to its complexity and efficiency, highlighting the importance of accurately simulating these effects in computer graphics to achieve a similar level of realism and immersion.

Humans have a long history of capturing lighting effects, from the earliest forms of

painting to the invention of the camera and the development of video technology. Artists have studied and mimicked the effects of light for centuries, striving to replicate the way it interacts with surfaces and creates shadows, highlights, and depth. The invention of the camera in the 19th century revolutionized our ability to capture light accurately and reproducibly, allowing for the documentation and study of the world in unprecedented detail. The advent of motion pictures and video further extended this capability, enabling the dynamic capture of light over time and opening new avenues for storytelling and visual communication. These advancements have continually pushed the boundaries of creating a photorealistic world, leading to ever more sophisticated methods for achieving photorealism in visual media.

However, existing capture setups have two main shortcomings: lack of *spatial* context and lack of *editability*. Still photographs, the most common form of visual media today, capture an immense amount of light information about a scene at a fixed viewpoint within a fixed field-of-view. They fail to capture the full experience of observing a particular scene because they cannot capture light that falls outside of the field-of-view and beyond line-of-sight. Once the image is taken, users can only edit a small amount of information, limited to simple operations such as crop, exposure, and white balance. Although modern image editors are increasingly powerful, the fact that they operate on 2D arrays limits their efficacy in 3D editing operations, *i.e.*, removing or inserting new objects into the scene.

In computer graphics, there have been attempts to capture lighting effects through sophisticated devices such as light probes [27] and light stages [28]. Light probes typically employ a highly reflective sphere placed in a scene that reflects the environment light from all directions, providing valuable information for accurately rendering scenes. Despite their relatively inexpensive setup, light probes only capture lighting effects at one location (their placement). Light stages, on the other hand, use a set of synchronized cameras and lights to capture high-detail shape and material properties of a subject, often used in facial scanning for movies and video games. These prior techniques in capturing light have been instrumental in advancing the realism of synthetic renderings. However, light stages are often expensive and technically complex, requiring specialized equipment and expertise.

Creating photorealistic images and virtual environments has been a long-standing goal

in the field of computer vision and graphics. The desired outcome is for the resulting creations to have a rich sense of *spatial* context, and are easy to *edit* the scene elements, *e.g.*, scene configurations, lighting, and viewpoints. The ability to render such scenes that are indistinguishable from real-life photographs is crucial for various applications, including virtual reality, gaming, film production, and architectural visualization. One of the core challenges in achieving high levels of realism is accurately simulating the interaction of light with objects and surfaces in a scene. The perception of realism heavily depends on how well lighting effects, such as shadows, reflections, and textures, are replicated.

In this thesis, I start with an overview of the psychophysics of light, including human perception under different lighting and the depiction of light in western art. I then delve into my work on learning from casual lighting capture in order to create photorealistic images and virtual environments. In particular, my contributions include: 1. a method to infer depth, occlusion, lighting, and shadow information from video taken from a single camera viewpoint (Section 1.1.1); 2. a framework using this learned information to automatically reconfigure images of street scenes by populating, depopulating, or repopulating them with objects such as pedestrians or vehicles (Section 1.1.2); 3. a lightweight capture setup alternative to a light stage that captures high quality facial geometry, reflectance and lighting using only a smartphone camera and the sun (Section 1.2); and 4. a method for generating arbitrarily large texture images from a text prompt, which also enables re-texturing an input image with the generated textures (Section 1.3). Finally, I conclude with directions for open problems and future work.

1.1 Photorealistic Image Composition

Websites such as Google Street View enable users to explore places around the world through street-level imagery. These sites can provide a rich sense of what different locales — neighborhoods, parks, tourist sites, *etc.* — are really like. However, the imagery provided by such sites also has key limitations. A given image might be full of cars and pedestrians, making it difficult to observe the environment. Alternatively, a user might want to see how a scene appears at a certain time of day, *e.g.*, lunchtime, but only have access to a morning image. And, importantly, the fact that the imagery records real people and vehicles may require

Figure 1.1: **Photorealistic image composition**. On the left, compared to the shadow-free composite image, our method learns from a static video capture to cast realistic shadows in the correct direction relative to the sun, which properly conform to the scene geometry. This is an application of populating an empty scene. On the right, I further demonstrate two applications: emptying the city and repopulating the scene. Our pipeline first removes all objects along with their shadows, then selects people matching the scene's lighting, places them randomly on sidewalks and roads, and synthesizes realistic shadows. This method learns from a collection of images and does not require a video capture.

anonymization efforts to protect privacy, *e.g.*, by blurring faces and license plates [39] or by removing pedestrians from images by leveraging multiple views [38].

Removing existing objects and inserting new objects into an image are two long-standing problems in computer vision. For object removal, standard methods such as image inpainting requires an additional input of the object mask, and only removes objects within. This often results in leftover shadows and reflections, making the image composition unrealistic. To insert a new object, the foreground object must have proper shape, size, occlusion order, and lighting based on its placement on the background. Furthermore, the interaction of the foreground object and the background, *e.g.*, casting shadows and reflections, has to be modeled in order to create a realistic image composition.

In this section, I summarize two of my works on leveraging deep networks that learn from casual captures to model the lighting effects between the foreground and the background. As shown in Figure 1.1, the resulting removal network removes the selected objects along with their shadows, while the insertion network casts realistic shadows in the correct direction.

1.1.1 Casting Shadows in Image Composition

In Chapter 3.2, I introduce a method [167] that analyzes the motion of people and other objects in a scene, to infer depth, occlusion, lighting, and shadow information from video taken from a single camera viewpoint. This information is then used to composite new objects into the same scene with a high degree of automation and realism. In particular, when a user places a new object (2D cut-out) in the image, it is automatically rescaled, relit, occluded properly, and casts realistic shadows in the correct direction relative to the sun, and which conform properly to scene geometry. On the left of Figure 1.1, I show results for object insertion. Our method produces crisper shadows in the correct direction and properly conforms to scene geometry, whereas the shadow-free composite image has no shadow at all.

1.1.2 Repopulating Street Scenes

The aforementioned method requires a long video of a scene as input, which limits its applications. To mitigate the limitation, I introduce a framework [170] in Chapter 3.3 for automatically reconfiguring images of street scenes by populating, depopulating, or repopulating them with objects such as pedestrians or vehicles. The framework is learned from data with minimal ground truth annotations, by making creative use of large-numbers of short image bursts of street scenes. I demonstrate two potential applications using the framework on the right of Figure 1.1: (1) emptying the city by removing all objects within an image and (2) repopulating the scenes with anonymized people. These applications are designed to enhance the privacy of a street image while preserving the realism.

1.2 SunStage: Portrait Reconstruction and Relighting

A light stage [28] acquires the shape and material properties of a face in high detail using a series of images captured under synchronized cameras and lights. This captured information can be used to synthesize novel images of the subject under arbitrary lighting conditions or from arbitrary viewpoints. This process enables a number of visual effects, such as creating digital replicas of actors that can be used in movies [1] or high-quality postproduction

relighting [173].

In many cases, however, it is often infeasible to get access to a light stage for capturing a particular subject, because light stages are not easy to find: they are expensive and require significant technical expertise (often teams of people) to build and operate. In these cases, hope is not lost — one can turn to methods that are *trained* on light stage data, with the intention of generalizing to new subjects. These methods do not require the subject to be captured by a light stage but instead use a machine learning model trained on a collection of previously acquired light stage captures to enable the same applications as a light stage, but from only one or several images of a new subject [114, 150, 191, 13, 88, 144, 193].

Unfortunately, these methods have difficulty faithfully reproducing and editing the appearance of new subjects, as they lack much of the signal necessary to resolve the ambiguities of single-view reconstruction, *i.e.*, a single image of a face can be reasonably explained by different combinations of geometry, illumination, and reflectance.

In Chapter 4, I present SunStage [169] — a system that allows for personalized, high-quality capture of a given subject, but without the need for expensive, calibrated capture equipment. SunStage uses only a handheld smartphone camera and the sun to simulate a minimalist light stage, enabling the reconstruction of individually-tailored geometry and reflectance without specialized equipment. The capture setup only requires the user to hold the camera at arm's length and rotate in place, allowing the face to be observed under varying angles of incident sunlight, which causes specular highlights to move and shadows to swing across the face. This provides strong signals for the reconstruction of facial geometry and spatially-varying reflectance properties. As demonstrated in Figure 1.3 (d), the reconstructed face and scene parameters estimated by SunStage can be used to realistically render the subject in new, unseen lighting conditions — even with complex details like self-occluding cast shadows, which are typically missing in purely image-based relighting techniques, *i.e.*, those that do not explicitly model geometry. In addition to relighting, I also show applications in modifying the lighting conditions (Figure 1.3 (b)), softening harsh shadows (Figure 1.3 (c)), and editing skin reflectance (Figure 1.3 (e)).

(a) Reconstruction (b) Edit lighting (c) Soften shadows (d) Swap lighting (e) Edit texture

Figure 1.3: **SunStage.** Given a selfie video rotating under the sun, SunStage reconstructs geometry, material, camera pose, and lighting information. This recovered information can be used to (a) realistically re-render the input images, (b) modify the lighting conditions by adding / removing lights, (c) soften harsh shadows by changing the size of the reconstructed light sources (d) render the person in an entirely new environment, and (e) edit the albedo or material properties to add freckles, makeup, or stickers that realistically interact with scene lighting.

1.3 Shading in Texture Transfer

Textures, defined as statistically repeating image content, are fundamental primitives in computer graphics. They effectively describe a wide variety of surfaces, ranging from tree bark to human skin, capturing fine surface details. The use of textures enables the efficient generation of photorealistic imagery, surpassing what can be achieved with geometry alone. Realistic textures can significantly enhance the overall visual experience.

Texture assets are often created by artists, either using manual design tools or finding and capturing reference content. Developments in texture synthesis [35, 157, 171, 34] propose to simplify this process by generating larger, randomly varying samples of a texture, from a smaller reference patch. Texture synthesis methods have evolved significantly over the years, from statistical or patch-based models [35, 157, 171, 34] to more modern deep-learning

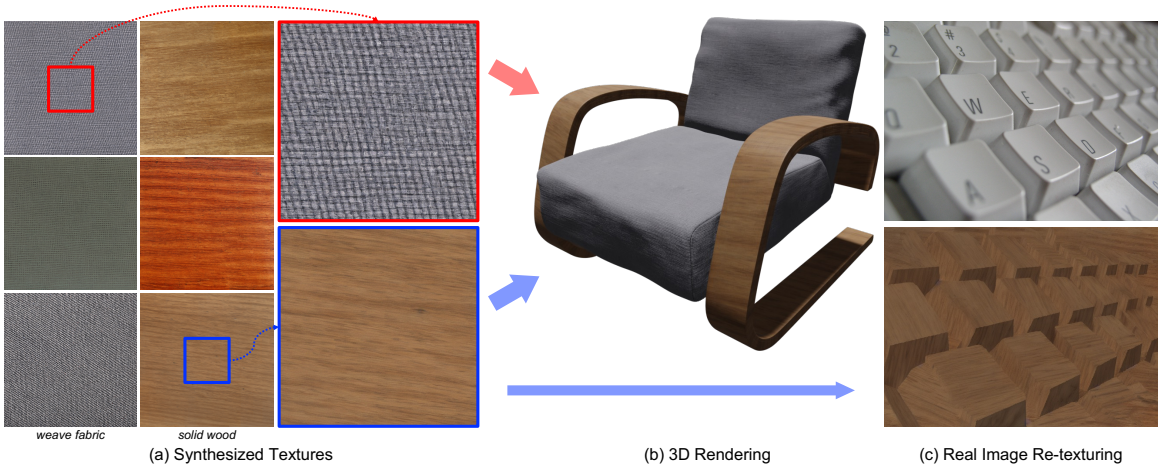


Figure 1.4: **Infinite Texture**. Given a text prompt describing the desired texture, Infinite Texture can (a) generate a varying collection of arbitrarily large, high-quality textures on a single GPU. The generated textures can (b) be readily applied in 3D rendering pipelines and (c) be used to construct a minimal dataset to ensure consistent shading and shape in texture transfer.

solutions [42, 5, 195]. Most existing methods have been demonstrated only on low resolution textures, *i.e.*, up to 512×512. Furthermore, all these methods involve an arduous process, as an artist must still capture or design an input reference texture.

In Chapter 5, I introduce Infinite Texture [168] which focuses on an alternative guiding signal for texture synthesis: text. Infinite Texture is a prompt-based approach for high-resolution texture generation using a text-to-image diffusion model. As shown in Figure 1.4 (a), it takes a text prompt as input, and generates a varying collection of arbitrarily large, high-quality textures on a single GPU. Moreover, it is capable of generating multiple distinct textures from the same text prompt, offering users a range of texture variations. The generated texture images have direct applications in 3D rendering pipelines (Figure 1.4 (b)).

Additionally, in Figure 1.4 (c), I showcase an application enabled by Infinite Texture: a diffusion-based approach for re-texturing an input image with the generated texture. The key insight here is that learning on a minimal dataset constructed from texture-mapped primitives can generalize to nature images. The proposed method successfully transfers the

appearance of the example texture onto various surfaces, while sharing consistent shading and shape with the input image.

Thesis Overview

I begin the following chapter with an overview of the psychophysics of light, including human perception of different lighting. This serves as a foundation for understanding how our visual system interprets light and how it influences our perception of the world. Furthermore, I review the depiction of light in western art, showcasing how artists throughout history have studied and utilized lighting techniques to enhance the realism and emotional impact of their work. In Chapter 3, I introduce *People as Scene Probes* [167] and *Repopulating Street Scenes* [170]. These two methods use a static video or a collection of street images as a casual capture to enable photorealistic image composition with physically correct geometry and shadows. In Chapter 4, I introduce *SunStage* [169], a capture setup alternative to a light stage that captures high quality geometry, reflectance and lighting. In Chapter 5, I present *Infinite Texture* [168] and its application in texture transfer, demonstrating that a minimal dataset can be used to guarantee consistent shading in the outputs of diffusion-based models. Finally, I conclude in Chapter 6 with directions for open problems and future work.

Chapter 2

PSYCHOPHYSICS OF LIGHT

Light is essentially composed of a stream of mass-energy units known as photons emitted from sources such as the sun or artificial lights. These photons are by-products of atomic and subatomic processes, and span a spectrum of energy. However, only those within a specific range are discernible to the human eye as visible light. This range corresponds to wavelengths that trigger responses in the retinal cells, enabling vision. In contrast, photons outside of this energy spectrum, either too low or too high, do not stimulate visual perception.

The propagation of light is traditionally modeled as straight lines in rendering. This simplified approach, while ignoring quantum effects, is sufficient for most practical purposes in computer graphics. In reality, quantum electrodynamics attempts to describe the probabilities and make statistical predictions of photon behaviors, often involving complex and counter-intuitive calculations. When photons interact with various types of matter such as opaque surfaces, transparent substances, holes, and sharp edges, they engage in intricate exchanges with local electrons rather than simply bouncing off surfaces. For our purposes, assuming that photons follow straight-line paths is a generalized approach that works well. This path is typically straightforward in uniform media such as air or water but becomes more complex in heterogeneous media or at interfaces between different media.

Within the scope of this thesis, I note two points that are most important. First, photons often travel in straight lines. Second, there are many surfaces through which their energies are not transmitted as visible light. In a real-world scenario, there are various objects disrupting and blocking the flow of photons. These disruptions, as described by the Philosopher Roy Sorenson "holes in the light" [148], are shadows.

In this chapter, I first discuss the different types of lighting effects that occur in the real world and the information they convey. I then delve into the human perception of lighting

Figure 2.1: The fall of light on a face c.1488 by Leonardo da Vinci. Credit: Wikipedia.

and analyze how humans perceive, depth, illumination source, space, motion, material, and texture from lighting. This thesis also touches upon on how human perception can be manipulated with shadows. Lastly, I examine lighting in artwork and their different use cases.

2.1 Different Types of Lighting Effects

To better understand the propagation of light and its effects on visual perception, it is essential to classify different types of lighting effects based on the shadows they produce.

Shadows are an integral part of how we perceive the shape, texture, and spatial relationships of objects. By examining shadows, we can infer the properties of the light sources and the objects they illuminate. This classification helps in understanding how different lighting affects our visual perception.

Shadows are areas that receive less light than their surroundings — they could be the result of an opaque object blocking the incoming light or a reduced amount of light reflected from the surface to the eye. As introduced by Baxandall [4], there are three distinct types of shadows. As shown in Figure 2.1, they are demonstrated clearly in one of Leonardo da Vinci's diagrams. This diagram inspired scientists in the eighteenth century to think about shadow and vision.

In Figure 2.1, A is a light source radiating onto the man's face, with angles marked from B to M. The light source is abnormally close to the face to better illustrate the effects. In the sector on the lower nose, the nose causes an occlusion — the tip of the nose blocks the light from reaching the upper lip. This is the first type of shadow, known as cast shadow.

The bottom of the nose is also not lit by the light source A. This is not because the light is blocked. Instead, this part of nose is facing away from the light source. This is the second type of shadow, known as attached shadow.

The third type of shadow is partial shadow. A surface facing towards the light source will receive more photons than an angled surface. For example, in Figure 2.1, the front of the man's nose will take more light than the top of his head. The less light received, the less available to reflect, so the head will appear to be darker than the nose. This type of shadow is known as shading.

These three types of shadows are theoretical constructs. In reality, the actual intensities observed on surfaces often comprise a mixture of these different shading and shadow effects. Real-world lighting involves intricate interactions between multiple light sources, surfaces, and materials, creating a dynamic and multifaceted visual environment. In the following sections, we will delve into human perception of different types of lighting, exploring how our visual system interprets these complex lighting scenarios and what we can infer from them.

Figure 2.2: Artists use shadows in sketches to induce a sense of three-dimensional surface shape [78] — much more surface structure is evident in the left image than in the right, in which the shadows have been removed.

2.2 Human Perception of Lighting

Human perception of the world is so effortless that we take it for granted. However, consider the complexity involved in observing even the most basic visual scene. Our eyes receive two small, inverted images, yet our brain interprets these as a coherent, three-dimensional space. This process is "nothing short of a miracle", as described by the neuropsychologist Richard Gregory [48].

Shadows play a vital role in this process, as they contain rich information about the three-dimensionality. Artists have long understood the importance of shadows, and leveraged them for generating an impression of three-dimensionality in paintings [24]. Figure 2.2 demonstrates an example of how shadows can be used to depict surface shape in scenes. Although both images are 2D sketches, shadows induce strong precepts of 3D structure. In this section, I make a detailed list of different aspects of information that humans can infer from different lighting.

2.2.1 Shape and Depth Perception

Lighting provides a strong source of information about the shapes of surfaces. Figure 2.3 shows two examples of how shadows enable shape and depth perception. On the left of Figure 2.3, what are initially seen as random black fragments soon crystallize into 3D letters of the alphabet. This is due to the single-light-source rule, the assumption that

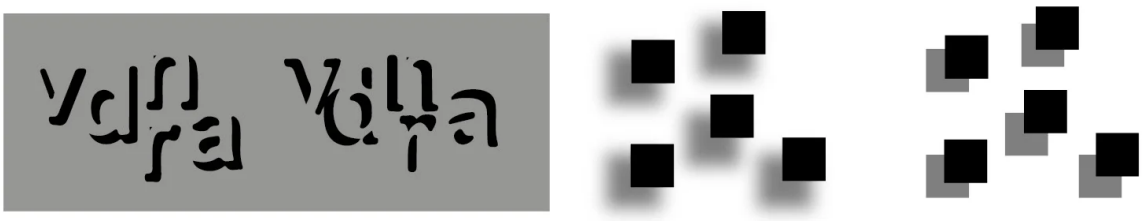


Figure 2.3: Left: attached shadows help human perceive 3D letters under a single light source, but hard to perceive as 3D under random lighting. Right: cast shadows help depth perception — the squares with blurred-edged shadows appear nearer to the observer than those with sharp-edged shadows. Credit: Scientific American.

in interpreting shaded images, the brain assumes that the entire scene is illuminated by a single light source. Once the lighting for each letter is randomized, the same letters are more difficult to perceive as 3D letters. But one can still infer those letters individually.

On the right of Figure 2.3, cast shadows enable the perception of depth, as the black squares seem to be popping out of the frame. Cast shadows with penumbrae, *e.g.*, the softer-edged shadows, are more realistic than those with sharp edges. This was first observed by German physiologist Ewald Hering in the nineteenth century. Even though the shadow area is located at the same distance from the square, the squares with blurred-edged shadows appear nearer to the observer than those with sharp-edged shadows.

Knill *et al.* [78] analyzed the local geometric structure of shadow contours on piecewise smooth surfaces. They found that points along intrinsic shadow boundaries provide constraints on surface shape, which can be used as boundary conditions for surface interpolation within shadow regions. The results suggest that intrinsic shadows can be directly used to infer global surface structure and the concavity/convexity of a surface.

2.2.2 Illumination Direction

Shadows serve as strong cues for the illumination direction. Two examples are shown in Figure 2.4. On the top left, a simple circle with a gradient suggests one side is lit and the other is in shadow. However, the perception depends on the convexity/concavity of

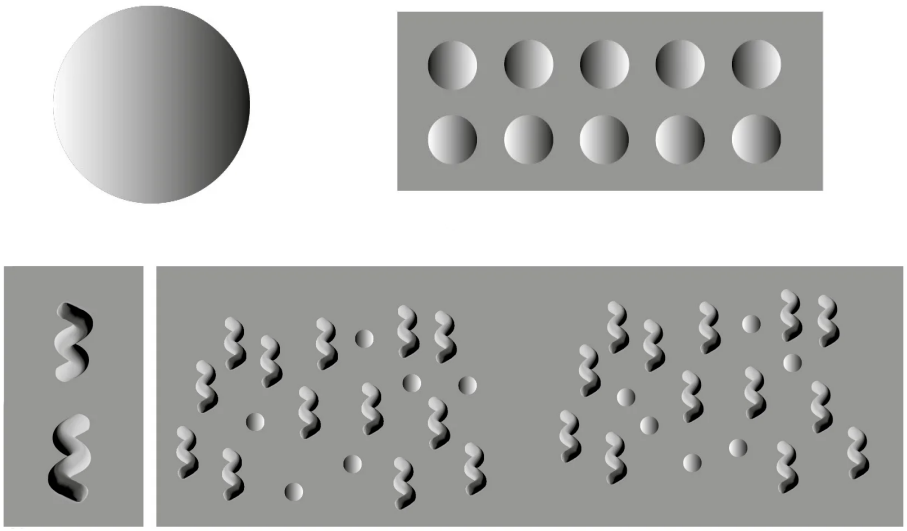


Figure 2.4: Top: the disk can be viewed as a sphere lit from the left or a cavity lit from the right. Right: with strong convexity demonstrated by the 'worm' shape, shadows reveal only one possible illumination direction. Credit: Scientific American.

the object. The same disk can be viewed as a convex ball lit from the left or a cavity lit from the right. This demonstration also reveals the first rule of shape from shading: given similar conditions, convexity is preferred. This preference might exist because most objects encountered in nature are usually convex, as they are solid objects.

On the top right of Figure 2.4, there is a strong tendency to see the bottom row as cavities, and vice versa. This is due to the single-light-source rule, which is the assumption that in interpreting shaded images, the brain assumes the entire scene is illuminated by a single light source. Additionally, human perception also prefers a top-down lighting, given that our planet is lit the sun from above.

Human perception of illumination direction does not work well when the convex ambiguity is present. Therefore, on the bottom left of Figure 2.4, the disk is replaced by vertical "worms" which shows strong convexity and would never concave in this illustration. With the strong convexity displayed, it is much easier for human brain to tell where the light comes from. Even with the ambiguous disks mixed in on the bottom right, they appear to

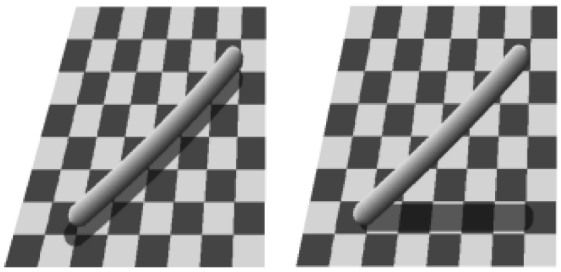


Figure 2.5: Knill *et al.* [78] demonstrated that shadows provide crucial information about the spatial relationship between objects. In these images, the only difference is the orientation of the shadow cast by the pole, yet the pole in the right image appears more upright relative to the background surface than the pole in the left image does.

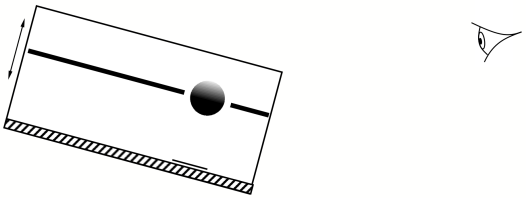


Figure 2.6: Kersten *et al.* [75] showed observers videos of a ball and its shadow moving with a matching trajectory in a box. Observers reported seeing the ball inflate as it moved to the back of the box and then shrink as it moved forward.

be seen as convex or concave to conform to the light source from the left, as suggested by the worms. The brain, therefore, uses the presence of unambiguous objects to determine where light is coming from and then interpret the more ambiguous details of an image.

2.2.3 Spatial Perception

Lighting can provide information about the spatial relationship between objects. Figure 2.5 is an example of extrinsic shadows: shadows cast on one object by another. Extrinsic shadows provide particularly salient cues to the relative positions and orientations of objects [181, 74]. By merely changing the orientation of the shadow cast by the pole in

Figure 2.5, the spatial relationship between the pole and the background also changes. The pole appears to be more upright when the shadow is detached from it. Yonas *et al.* [181] discovered that cast shadows can specify for the observer the spatial relations between an object and its surroundings. This sensitivity to such information is present even in three-year-old children. Such sensitivity also improves with age.

2.2.4 Motion Perception

Moving shadows can give clues about the movement of objects or light sources. Kersten *et al.* [75] shows the motion of a shadow can dramatically alter the perceived trajectory of the object casting it, overriding other sources of information and perceptual biases, such as the assumption of constant object size and a general viewpoint. In cases where the image of a ball remained the same size and had an identical trajectory in the image plane, the trajectory of the cast shadow was sufficient to make observers perceive the ball rising above or receding along the floor, overriding the constant size constraint. As shown in Figure 2.6, when the shadow's trajectory matched that of the ball, observers reported seeing the ball inflate as it moved to the back of the box and then shrink as it moved forward. This observation was made even though the ball's image size remained constant.

2.2.5 Texture and Material Perception

Lighting can also convey information about the texture and material properties of surfaces. This effect is widely used by artists in paintings and photography. Hogarth [55] introduced over ten different types of illumination in painting and how shadows are used under each type. In the setting of textural light, it reveals the surface qualities of three-dimensional forms. Shadows are used to emphasize the details of the textile surface.

The water, rocks, and trees in Figure 2.7 establish a series of tactile experiences, all communicated by textural light — the dense, terraced stone; the splashing, cascading stream; and the intricate foliage. These elements are revealed and enhanced by the strong contrasts of light and shadow characteristic of textural light. The tops of the rocks are brilliantly lit from a high source, creating tiny pits of shadow that convey texture. The shadows are

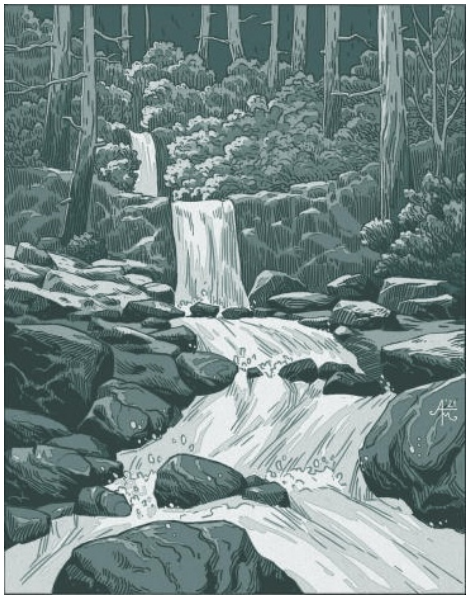


Figure 2.7: Lighting conveys information about the texture and material properties of surfaces. In this landscape, the shadows are strong and dark, yet rich in textural detail due to a significant amount of secondary, reflected light. Credit: Andead.

strong and dark, yet rich in textural detail due to a significant amount of secondary, reflected light. Similarly, clearly defined lights, shadows, and reflected lights sharply accentuate the textural details of the trees and foliage.

2.3 Lighting in Western Art

Lighting holds significant importance in western art [46], playing a crucial role in creating realistic, expressive, and visually compelling works. In this section, a few key aspects of the roles of lighting in western art are listed.

2.3.1 Creating Realism

One of the primary uses of lighting in western art is to achieve a sense of realism. The accurate depiction of light and shadow in paintings or sculptures creates a lifelike appearance, making the artwork appear more three-dimensional and realistic. This is particularly evident in the works of the Renaissance and Baroque periods, where mastering light and

Figure 2.8: Benedetto Diana, *Salvator Mundi*, 1510-20. Credit: The National Gallery, London.

shadow was essential to achieving realism.

The 'cartellino', the piece of paper or parchment, is a device frequently employed in Renaissance portraits. It is often the closest thing to the viewer, except in cases where the sitter's hand projects beyond a ledge that is parallel to the picture plane. Here too, a

Figure 2.9: Gerrit Berckheyde, *The Market Place at Haarlem, Looking towards the Grote Kerk by Gerrit Adriaenszoon Berckheyde*, 1665. Credit: Wikipedia.

cast shadow could add drama as well as reality; as in Benedetto Diana's *Salvator Mundi* (Figure 2.8), where both hands project, and the raised hand casts a shadow on the chest. This painting, depicting the risen Christ rather than a portrait in the usual sense, effectively emphasizes the physical reality of his resurrected body by these means.

2.3.2 Enhancing Depth

Shadows are used to give depth and volume to the subjects of art. By effectively using shadows, artists can convey the roundness of forms and the depth of space. This technique, known as chiaroscuro, was famously employed by artists to create dramatic effects in their paintings.

The strong attached shadows in the paintings by Beckheyde (Figure 2.9) not only add to the impression of depth but also enhance the effect of sunlight flooding the marketplace and square.

Figure 2.10: Claude, *A Seaport*, 1644. Credit: The National Gallery, London.

2.3.3 Creating Mood

Lighting can be used to set the mood or atmosphere of a piece. Artists often softens shadows to suggest the mellow light of morning or evening. These paintings illustrate the creation of a particular mood using shadows. The lengthening of shadows caused by the rising or setting sun is so memorably depicted by Claude in his harbor scene (Figure 2.10). Later, the Impressionists placed increasing emphasis on the observation that shadows are rarely grey but exhibit varying hues due to their contrast with the colors in their environment.

2.3.4 Directing the Viewer's Focus

Artists frequently use strong contrast lighting to guide the viewer's eye towards the focal points of their artwork. Strategic employment of light and shadow can highlight the most crucial elements of a composition, effectively controlling the viewer's experience of the art-

Figure 2.11: Follower of Rembrandt, *A Man Seated Reading at a Table in a Lofty Room*, 1628-30. Credit: The National Gallery, London.

work. A notable example of this technique is the *tenebroso* style of the seventeenth century, often linked with Caravaggio, which used extreme measures to enhance the radiance of light through tonal contrast. This effect is exemplified in the painting of a hermit or scholar reading at a table, attributed to a follower of Rembrandt or his school (Figure 2.11). This painting perfectly illustrates the use of dramatic light and shadow to focus the viewer's

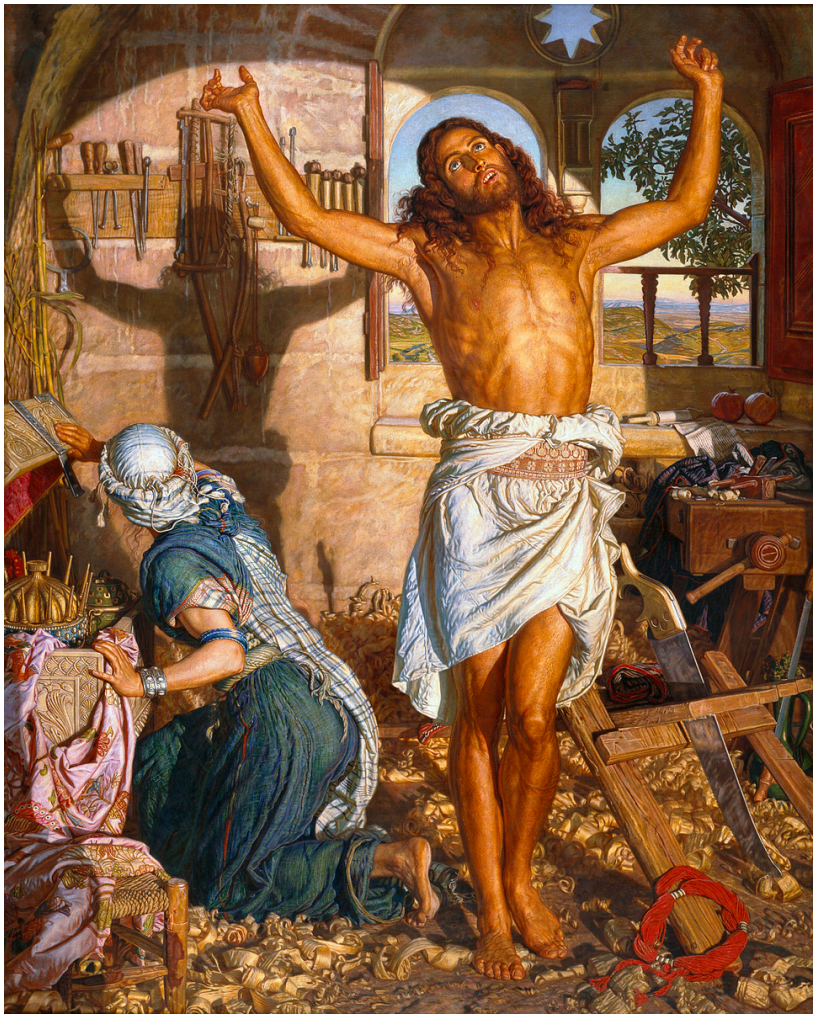


Figure 2.12: William Holman Hunt, *The Shadow of Death*, 1870-3. Credit: The Art Institute of Chicago.

attention.

2.3.5 Symbolic Meaning

In various instances, lighting and shadows in western art serve as symbols, representing concepts such as the passage of time, the presence of something unseen, or philosophical ideas like the transience of life or the duality of human nature. As a Symbolist, William Holman Hunt likely embraced the use of shadows to symbolize future events. In his work,

represented in Figure 2.12, he transformed the shadow of the young Christ into a forewarning of His ultimate death on the cross. This use of shadow not only enhances the artistic narrative but also imbues the artwork with deeper, symbolic significance.

Chapter 3

CREATING PHOTOREALISTIC IMAGE COMPOSITION

Image composition is the task of assembling various visual elements into a single coherent image. This process involves the careful placement and integration of objects, ensuring that they appear naturally within the scene. A crucial aspect of achieving realism in image composition is maintaining consistent lighting conditions across all elements. Proper lighting ensures that shadows, reflections, and highlights align correctly, contributing to the overall photorealism of the image.

Existing methods for image composition often fall short in this regard. While they can effectively place objects within a scene, they frequently fail to account for the correct lighting interactions between the foreground objects and their environment. Specifically, these methods do not accurately simulate shadows cast by foreground objects, nor do they adequately reflect the influence of ambient light on these objects. As a result, the composed images lack the depth and realism necessary to convincingly mimic real-world scenes.

In this chapter, I introduce two papers to overcome these limitations: People as Scene Probes [167] and Repopulating Street Scenes [170]. Both methods leverage learning based methods to ensure that composed images not only place objects accurately but also simulate correct lighting conditions, including the casting of realistic shadows. By addressing these challenges, these techniques push the boundaries of what is possible in photorealistic image composition, enabling the creation of scenes that are virtually indistinguishable from real-life images.

3.1 Related Work

Conditional Image Synthesis Deep generative models can learn to synthesize images, including generative adversarial networks (GANs) [47] and variational autoencoders (VAE) [77]. Conditional GANs [15, 105, 106, 107, 112] are used to synthesize images given category

labels. [116, 166, 65] focus on converting segmentation masks to photo-realistic images. They offer users an interactive GUI to draw their own segmentation masks and output a realistic image based on the given segmentation masks. However, these GANs do not leverage scene-specific geometry and lighting information, derived from many images. Our work embeds the scene's geometry and lighting into the GAN, to generate more realistic compositions.

Image Composition Lalonde *et al.* [81] proposed a system for inserts new objects into existing photographs by querying a vast image-based object library. Several authors have explored use of GANs to transform a foreground object to better match a background. ST-GAN [92] learns a homography of a foreground object conditioned on the background image. Compositional GAN [2] additionally learns the correct occlusion for the foreground object. SF-GAN [184] warps and adjusts the color, brightness, and styles of the foreground objects and embeds them into background images harmoniously. However, a realistic composition should also consider the foreground object's effect on the background (including shadows).

Some approaches aim to compose an object by rendering its appearance. [60] inserts an object into a scene based on a specified location and bounding box. [85] learns the joint distribution of the location and shape of an object conditioned on the semantic label map. PS-GAN [113] replaces a pedestrian's bounding box by random noise and infills with a new pedestrian based on the surrounding context. [86] blends the object with the background image in the bounding box, and learns a mapping to synthesize realistic images using both real and fake pairs. These works all train on images without hard shadows, and focus on person rather than shadow synthesis. For example, they only synthesize an area around the person's bounding box (not including long shadows), and don't take into account shadow casting information from other images of the same scene.

Shadow Matting Matting [121] is an effective tool to handle shadows. [22] enables synthesizing correct cast shadows, by estimating a shadow displacement map obtained by *manually* waving a shadow-casting stick over every part of the scene. Given an object to be composited, they can then synthesize correct shadows based on the object shape and shadow displacement map. The related problem of shadow *removal* has also been explored by a number of authors, e.g., [49, 186, 83]. We present the first shadow matting (synthesis)

method that is completely *passive*, i.e., does not require manually waving a stick, but instead learns from the movement of objects (people and cars) in the scene itself.

Image Layering Our work was inspired in part by [16], who first proposed using the motion of people (and other objects) to infer scene occlusions relationships. As the technology in the 1990's was more limited, their approach required manual intervention and made a number of simplifying assumptions. Less related to our work, but also worth mentioning is the use of layered representation for view synthesis, e.g. [146, 30, 155, 194, 149]. Like [16], our approach infers occlusion order purely from the movement of objects in the scene, but is entirely automated and leverages modern techniques for object detection and tracking.

Object removal. Prior work on object removal falls mainly into two groups: (1) image inpainting methods and (2) methods for detecting and removing object shadows.

Recently, deep learning and GAN-based approaches have emerged as a leading paradigm for image inpainting. Liu *et al*. [98] inpaint irregular holes with partial convolutions that are masked and re-normalized to be conditioned on valid pixels. Gated convolutions [183] generalize such partial convolutions by providing a learnable dynamic feature selection mechanism for each channel and at each spatial location. Contextual attention [182, 180] allows for long-range spatial dependencies, allowing pixels to be borrowed from distant locations to fill missing regions. Shadows have different forms, e.g., hard shadows, soft shadows, partially occluded shadows, etc. Hole-filling based methods have trouble determining what pixels to inpaint in different shadow scenarios.

Deep learning methods have also been applied to shadow removal. Qu *et al*. [123] extract features from multiple views and aggregate them to recover shadow-free images. Wang *et al*. [163] and Hu *et al*. [63] use GANs for shadow removal, while recently Le *et al*. [83] proposed a two-network model to learn shadow model parameters and shadow mattes. However, these methods only inpaint shadow regions. Realistic object removal involves removing both the object and its shadow, as handled by our method.

Other work has sought to remove pedestrians from street scenes by leveraging multiple views [38]. In our case, we operate on just a single view, and can also recompose new people into scenes. Finally, face replacement [9] has been considered for realism-preserving privacy enhancement tool [8]. Our work considers whole people, and not just faces.

Object insertion. Early methods for object insertion include Poisson blending [117], which can produce seamless object boundaries, but can also result in illumination and color mismatches between the object and the target scene. Lalonde *et al*. proposed *Photo Clip Art*, which inserts new objects into existing photographs by first querying a large dataset of cutouts for compatible objects [81]. Other methods match the color, brightness, and styles of inserted objects to harmoniously embed them into background images [92, 2, 184]. However, a realistic insertion should also consider an object's effect on the background (including shadows).

Some methods insert a 3D object by rendering it into in an image. Karsch *et al*. demonstrate convincing object insertions via inverse rendering models derived from geometric inference [71] or via single-image depth reconstruction [72]. Other work renders inserted objects with estimated HDR environment lighting maps [58, 57]. Chuang *et al*. synthesize shadows for inserted objects via a shadow displacement map [22]. However, these approaches essentially require a full 3D model of either the inserted object or the scene. Liu *et al*. [97] focus on single light source scenes containing hard shadows, whereas our method can handle scenes with soft shadows and spatially varying lighting. Recently, Wang *et al*. [167] proposed a data-driven method that takes a long video of a scene and learns to synthesize shadows for inserted 2D cutout objects. Our method learns from short bursts of images, and can synthesize shadows for 2D cutouts given a single image of a new scene at test time.

Lighting estimation. To capture illumination Debevec [27] captures HDR environment maps via bracketed exposures of a chrome ball. Subsequent methods [18, 41, 58, 80, 87, 143] use machine learning to predict HDR environment maps from single indoor or outdoor images. However, a single environment map is insufficient for compositing cut-out objects into a large captured scene, because different lighting effects will apply depending on, for instance, whether the object is placed in a sunlit area or a shadowed one. In our work, we do not explicitly estimate lighting for each scene, but instead use a rendering network that implicitly learns to generate shadows appropriate for the object location.

Outdoor illumination is primarily determined by the sun position and the weather conditions. Recent works [101, 58, 57, 96] use data-driven methods to estimate the sun azimuth

angle from a single outdoor image. Our work follows this trend and estimates a full 2D sun angle. We find that estimating the sun position aids in synthesizing plausible shadows in different weather and lighting scenarios.

3.2 People as Scene Probes

We start with a video of a scene, taken by a stationary camera. As objects – people, cars, bicycles – pass through the scene, they occlude and are occluded by scene elements, pass into and out of shadowed regions, cast shadows into the scene, and, due to perspective, appear larger or smaller in an image depending on their position in the scene. From the video sequence, we seek to extract occlusion layering, shadowing, and position-dependent scale to enable realistically compositing new objects (of similar classes) into the scene.

We design a fully automatic pipeline to tackle this problem. Our key idea is that the occurrence and motion of existing objects (aka *scene probes*) through the video is the primary cue for inferring properties of the scene. These properties include depth, occlusion ordering, lighting, and shadows. Unlike some related methods, our pipeline does not require active scanning for shadow matting [22], or manual annotation for layering [16]. Furthermore, our pipeline does not require the camera to be calibrated.

3.2.1 People as Occlusion Probes

Occlusion is key for realistic image composition. An inserted object should be occluded by the foreground and occlude the background properly. Cars driving on the street are occluded by the trees on the sidewalk nearer to the viewer, and road signs occlude people walking behind them. We propose a method to estimate the occlusion order by *analyzing the occlusion relationships between people, other moving objects in the video, and static scene structures and objects*. Our method records the occlusion relationship between object and the scene to yield an occlusion map $\tilde{z}(x, y)$, similar to a depth map, for determining which pixels of an object occlude or are occluded by the scene, depending on the location of the object. To make the problem tractable, we approximate the scene as a single ground plane, with moving objects and occluders represented as planar sprites (on vertical planes parallel to the image plane) that are in contact with the ground plane. Based on this simplification,

we can assume a monotonic relationship between object location and occlusion order; the closer the object, the lower in the image its ground contact occurs.

Algorithm We first calculate a median image within a local temporal window of one second to serve as a background plate; we have found the one second window to work well for scenes that are not densely crowded, and with objects (especially people) moving at a natural pace. For each frame in this temporal window, we apply Mask-RCNN [50] to estimate segmentation masks for people, cars, bikes, trucks, buses, and related categories. For each individual object O_i, Mask-RCNN returns a binary mask M_i, and we record the lowest point y_i of the mask. We refine this mask to avoid accidental inclusion of background pixels: each pixel in M_i whose color difference with the median image is greater than a threshold is assigned to refined mask M_i'.

Now we construct the occlusion map. We set the image origin $(x, y) = (0, 0)$ at the lower left corner of the image. The key idea is that if an object O_i with bottom pixel y_i occludes a background pixel, then another object O_j with $y_j < y_i$ is likely to be closer to the camera and would then also occlude this pixel. We initialize the occlusion map with $\tilde{z}(x, y) = -1$ at all pixels and then iteratively update the map for each object O_i:

$$\tilde{z}(x,y) = \begin{cases} y_i, & \text{if } (x,y) \in M_i' \text{ and } y_i > \tilde{z}(x,y) \\ \tilde{z}(x,y), & \text{otherwise.} \end{cases} \tag{3.1}$$

To create a new composite, we initialize image I_{comp} with one of the median images. For a new object O_j (e.g., cropped from another photo) with mask M_j and bottom coordinate y_j, we update I_{comp}:

$$I_{\text{comp}}(x,y) = \begin{cases} O_j(x,y), & \text{if } (x,y) \in M_j \text{ and } y_j < \tilde{z}(x,y) \\ I_{\text{comp}}(x,y), & \text{otherwise.} \end{cases} \tag{3.2}$$

where $O_j(x, y)$ is the color of the object at a given pixel (x, y). Note that this composite image lacks shadows cast by O_j. Further, if O_j is inserted into an area that is itself in shadow, then O_j should be darkened before compositing. We discuss these shadowing effects in the next section.

3.2.2 People as Light Probes

People appear brighter in direct sunlight and darker in shadow. Hence, we can potentially use people to *probe* lighting variation in different parts of a scene. Based on this cue, we compute a lighting map that enables automatically adjusting overall brightness of new objects as a function of position in the image. We do not attempt to recover an environment map to relight objects, or to cast complex/partial shadows on objects (areas for future work). Instead, we simply estimate a darkening/lightening factor to apply to each object depending on its location in the scene, approximating the effect of the object being in shadow or in open illumination. We call this factor, stored at each pixel, the lighting map $L(x, y)$. This lighting map is a *spatially varying* illumination map across the image, whereas the prior work [44, 115, 57, 58] generally solves for a single *directionally varying* illumination model for the entire scene. From the input video, we observe that people walking in well-lit areas tend to have higher pixel intensity than people in shadowed areas. We further assume there is no correlation between the color of people's clothing and where they appear in the image; e.g., people wearing red do not walk along different paths than those wearing blue. Given these conditions, we estimate the lighting map from statistics of overall changes in object colors as they move through the scene. Note that this lighting map is a combination of average illumination and reflection from the surface; it does not give absolute brightness of illumination, but gives a measure of relative illumination for different parts of the scene.

Algorithm Starting with the detected objects $\{O_i\}$ and associated masks $\{M_i'\}$ described in Section 3.2.1, we compute the mean color C_i per object across all pixels in its mask. The lighting map is then the average of the C_i that cover a given pixel, i.e.:

$$L(x, y) = \frac{1}{|\{i \mid (x, y) \in M_i'\}|} \sum_{i|(x,y)\in M_i'} C_i \tag{3.3}$$

When compositing a new object, O_j with mask M_j into the background plate, we first compute the average lighting L_j for the pixels covered by M_j:

$$L_j = \frac{1}{|\{(x, y) \in M_j\}|} \sum_{(x,y)\in M_j} L(x, y) \tag{3.4}$$

and apply this color factor component-wise to all of the colors in O_j. As noted above, this lighting factor makes the most sense as a relative measure. Thus, when compositing a new

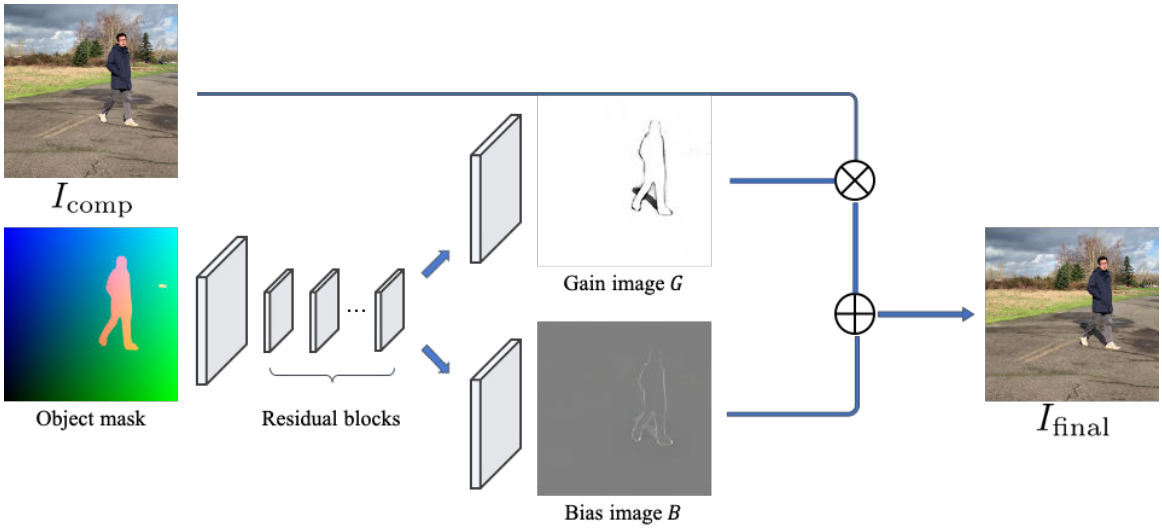


Figure 3.1: The network takes the object mask, x, and y coordinates as input (visualized here in red, green, and blue channels) and outputs a scalar gain image G and color bias image B (mid-gray corresponds to zero bias as shown). Given the shadow-free, composite image I_{comp}, we synthesize the final image $I_{\text{final}} = G \cdot I_{\text{comp}} + B$.

object into the scene in our application scenario, the user would first set the brightness of the object at a given point in the scene (with the lighting multiplied in), and can then move the object to different parts of the scene with plausible changes to the brightness then occurring automatically.

3.2.3 People as Shadow Probes

Shadows are one of the most interesting and complex ways that moving objects interact with a scene. Predicting shadows is challenging, as their shapes and locations depend on the position of the sun in the sky, the weather, and the geometry of both the object casting the shadow and the scene receiving it. Furthermore, unlike other lighting effects, shadows are not additive, as a surface already in shadow does not darken further when a second shadow is cast on it from the same light source. We propose using observations of objects passing through the scene to recover these shadowing effects, using a deep network – a pix2pix [65] GAN with improved losses [166] – trained on the given scene to learn how

objects cast shadows depending on their shapes and locations in scene. Further, since the discriminator encourages generation of realistic images, the network also tends to improve jagged silhouettes.

Algorithm A natural choice of generator would take as input a shadow-free, composite image I_{comp} and directly output an image with shadows. In our experience, such a network does not produce high quality shadows, typically blurring them out and sometimes adding unwanted color patterns. Instead, we use the object masks of inserted objects as input, which are stronger indicators of cast shadow shapes. Inspired by [99], we concatenate an image channel comprised of just the per-pixel x-coordinate, and another channel with just the per-pixel y-coordinate; we found that adding these channels was key to learning shadows that varied depending on the placement of the object, e.g., to ensure the shadow warped across surfaces or was correctly occluded when moving the object around. As in Figure 3.1, we feed this x-y augmented object mask through a deep residual convolutional neural network to generate a scalar gain image G and color bias image B, similar to the formulation in [83, 84]. The final image is then $I_{\text{final}} = G \cdot I_{\text{comp}} + B$. We found that having the generator produce I_{final} directly resulted in repetitive pattern artifacts that were alleviated by indirectly generating the result through bias and gain images.

For training, we take each input image I and follow the procedure in Section 3.2.1 to extract objects $\{O_i\}$ and masks $\{M_i'\}$ from an image and then composite the objects directly back onto the local median image to create the shadow-free image I_{comp}. The resulting I_{final}, paired with the ground truth I, can then be used to supervise training of the generator and discriminator, following the method described in [166].

3.2.4 People as Depth Probes

The size of a person (or other object) in an image is inversely proportional to depth. Hence, the presence of people and their motion through a scene provides a strong depth cue. Using this cue, we can infer how composited people should be resized as a function of placement in the scene. We propose a method to estimate how the scale of an object should vary across an image without directly estimating scene depth or camera focal length, but based instead

on the sizes of people at different positions in the scene. Our problem is related to [14] who rectify a planar image by tracking moving objects, although they require constant velocity assumptions, which we avoid. [23] determines the height of a person using a set of parallel planes and a reference direction, which we do not require. We make two assumptions: (1) the ground (on which people walk) can be approximated by a single plane, and (2) all the people in the video are roughly the same height. While the second assumption is not strictly true, it facilitates scale estimation, essentially treating individual height differences among people as Gaussian noise, as in [56], and solving via least squares.

Algorithm According to our first assumption, all ground plane points (X, Y, Z) in the world coordinate should fit a plane equation:

$$aX + bY + cZ = 1 \tag{3.5}$$

Under the second assumption, all people are roughly the same height H in world coordinates. Under perspective projection, we have:

$$x = X \cdot \frac{f}{Z}, y = Y \cdot \frac{f}{Z}, h = H \cdot \frac{f}{Z} \tag{3.6}$$

where f is the focal length of the camera. Multiplying both sides of Equation 3.5 by $H \cdot \frac{f}{Z}$, we arrive at a linear relation between pixel coordinates and height:

$$a'x + b'y + c' = h \tag{3.7}$$

where a', b', c' are constants. Because people in the scene are grounded, Equation 3.7 suggests that any person's bottom middle point (x_i, y_i) and her height h_i follow this linear relationship.

Given the input video, we use the same segmentation network as in Section 3.2.1 to segment out all the people in the video. For each person in the video, we record her height h_i and bottom middle point (x_i, y_i) in camera coordinates. After collecting all the (x_i, y_i) and h_i from the image segmentation network, we use the least squares method to solve for the (a', b', c') in Equation 3.7.

When inserting a new object into the scene at (x_j, y_j), we apply Equation 3.7 to estimate height h_j. The inserted object will then be resized accordingly and translated to (x_j, y_j).

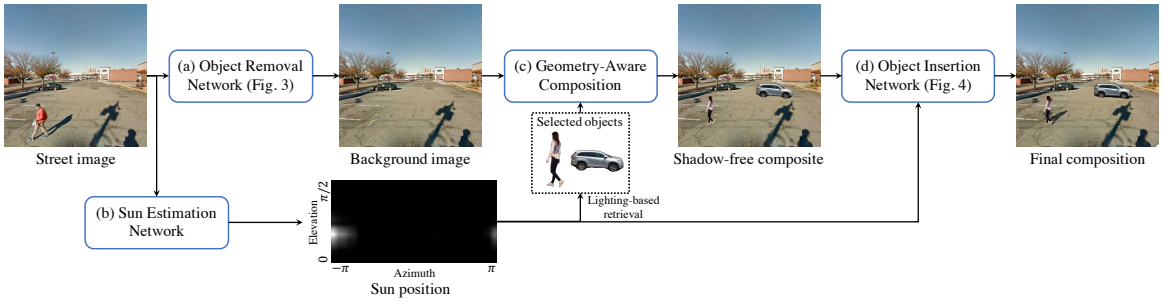


Figure 3.2: Our reconfiguration pipeline has four major components: (1) a removal network that learns to remove existing objects and their shadows, (2) a sun estimation network that learns to predict sun position from an image, (3) a method to scale and insert newly inserted objects with correct occlusion ordering, and (4) an insertion network that learns to cast shadows. Given a street image, our method first removes selected objects, then selects new object that matches the lighting of the scene, composes them with correct scale and occlusion order into the background image, and synthesizes shadows for inserted objects.

In our application, if the user requires a different height for an inserted object, then she can simply place the object and rescale as desired, and the system will then apply this rescaling factor on top of the height factor from Equation 3.7 when moving the object around the scene.

3.2.5 Implementation Details

We use Mask-RCNN [50] as the instance segmentation network. Inspired by [67], our shadow network uses a deep residual generator. The generator has 5 residual blocks, followed by two different transposed convolution layers to output the bias and gain maps. The loss function is the same as in [166]. We use ADAM with an initial learning rate of 1e-4, and decays linearly after 25 epochs to optimize the objectives. More details can be found in supplementary.

3.3 Repopulating Street Scenes

Given an image of a street scene, our goal is to recompose the objects (*e.g.*, cars and pedestrians) in the scene by first removing the existing objects, and then optionally composing one or more new objects into the scene. These stages must respect the illumination in the scene—in particular, shadows (both their removal and insertion) are critical elements that are difficult to handle realistically in prior work.

Our automatic pipeline for addressing this problem has four major components, as shown in Fig. 3.2: (1) a removal network that learns to remove existing objects and their shadows, (2) a lighting estimation network that learns to predict sun position from an image, which helps identify compatible objects for insertion and is used to create better insertion composites, (3) a method to scale the inserted object properly with correct occlusion order based on its placement in the scene, and (4) an insertion network that learns to cast shadows for newly inserted objects. Given a street image, we first use Mask R-CNN [177] to segment existing people and cars, then use that as a mask for the object removal stage. The object removal network completely removes those objects (and their shadows), yielding a background image. The sun estimation network is used to select new objects that match the scene's lighting. Selected objects are then composed into the background image with correct scale and occlusion order to get a shadow-free composite. Finally, our insertion network takes the shadow-free composite, synthesizes shadows, and outputs the final composite.

3.3.1 Data

We train our networks in a novel way by using a dataset of image bursts, i.e., short timelapse image sequences captured over several seconds. As shown in Fig. 3.3, we compute the median of each timelapse image stack to produce a "clean plate" background image free of (moving) objects such as people and their shadows. We also know location and time of day for each timelapse, from which we derive the sun position. (We do not use weather data; the sun position is noted regardless of cloud cover.) Images with and without moving objects, and corresponding sun positions serve as ground truth supervision for our object removal, sun prediction, and object insertion networks. At test time, our pipeline takes in a single

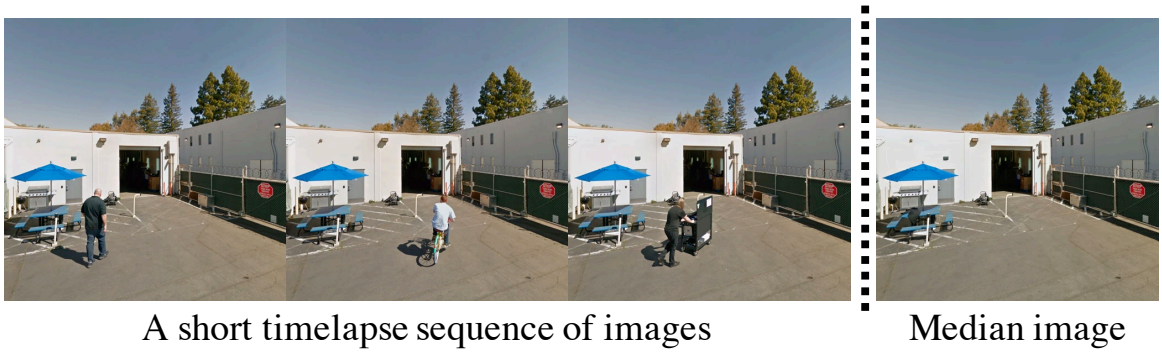


Figure 3.3: Our dataset consists of short image bursts, *i.e.*, short timelapse sequences of images captured over several seconds. We compute the median of each timelapse image stack to produce a "clean plate" background image free of objects and shadows.

street image and can remove and repopulate the objects within. We now describe the four components of our pipeline.

3.3.2 Removal Network

Object removal is a challenging task that involves generating new content in holes left by removed objects, such that the new image is realistic and semantically correct. Given a mask indicating the objects to be removed, standard inpainting methods [98, 182, 183, 180] only fill masked regions, leaving behind shadows. Our goal is to remove both objects and their shadows. We propose a deep network that, given an image and an object mask, constructs a new mask that includes the object and its shadow, then inpaints the region inside this mask. Inspired by PatchMatch-based inpainting [51] and appearance flow [134], our method predicts a flow map that uses nearby patches' features to inpaint the masked region.

Algorithm. Standard inpainting methods operate on an image and a binary mask designating where to inpaint. In our case, the network also automatically detects shadow regions belonging to masked objects. Different objects have different shadow shapes—for example, people can have long, thin, complex shadows, while cars tend to have larger, simpler shadows. Hence, rather than taking a binary mask, our network receives an image and a class

mask produced by Mask R-CNN [177]. This class mask encodes the object category of each pixel with distinct values normalized to $[0, 1]$.

Fig. 3.4 shows the removal network architecture. We feed the input image I and its class mask through four downsampling layers followed by three different branches of residual blocks. The first branch predicts a full inpainting mask M_{inp}, including the object and its shadow; the second predicts a warping flow map F_{warp}; and the third encodes the image as a high-dimensional inpainting feature map F_{inp}. The feature map is then warped by the predicted flow map F_{warp} and fed into four upsampling layers to produce an inpainting image I_{inp}. The final removal image is then computed as

$$I_{\text{remove}} = I_{\text{inp}} \odot M_{\text{inp}} + I \odot (1 - M_{\text{inp}}). \tag{3.8}$$

The feature warping layer uses the high-dimensional features of nearby patches to inpaint the missing area. We found that street scenes often have highly repetitive structures—building facades, fences, road markings, etc.—and the feature warping layer works well in these situations.

3.3.3 Sun Estimation Network

Lighting is key to realistic image composition. An object lit from the left composed into a scene lit from the right will likely look unrealistic. Many traditional lighting estimation methods reconstruct an environment map [18, 41, 58, 80, 87, 143]. However, a single environment map is insufficient for compositing objects into the scene, because different lighting effects will apply depending on object placement, e.g., whether the object is the shade or lit by the sun. In our work, we do not explicitly estimate scene illumination, but instead predict the sun position with a deep network and use the result to help synthesize plausible shadows. Further, we apply the same network to choose objects with similar sun position to be inserted.

Algorithm. Rather than regressing an image to sun azimuth and elevation, we treat this as a classification problem and predict a distribution over discretized sun angles. We divide the range of azimuth angles $[0, 2\pi)$ into 32 bins and elevation angles $[0, \pi/2]$ into 16 bins. We use a network similar to ResNet50 [52], replacing the last fully connected layer with

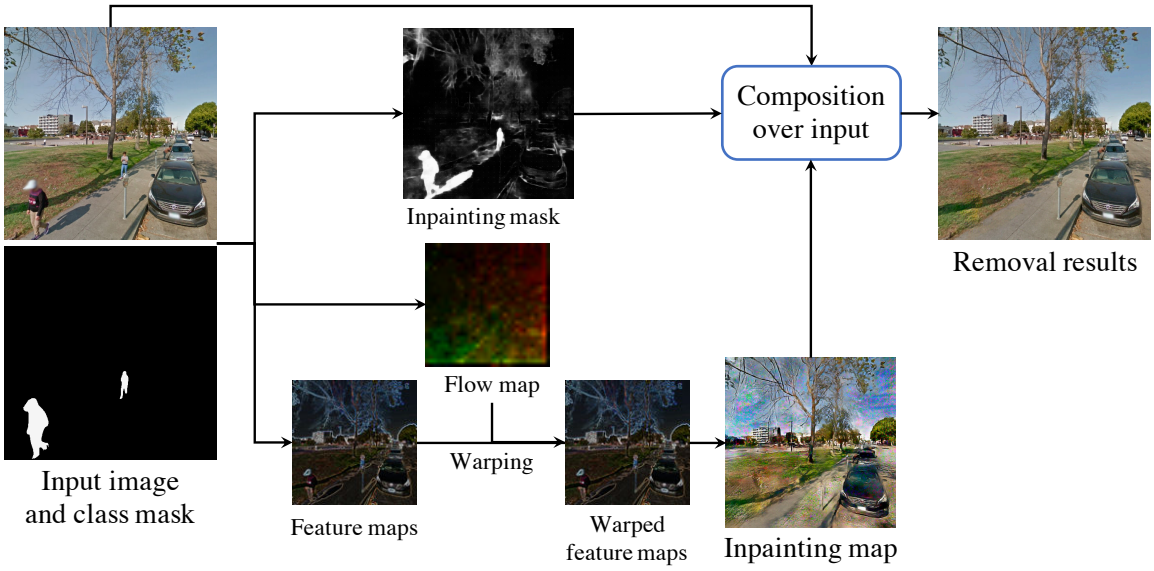


Figure 3.4: The generator of the removal network takes an input image I and a class mask as input, and outputs an inpainting mask M_{inp} and an inpainting map I_{inp}. We synthesize the removal image $I_{\text{remove}} = I_{\text{inp}} \odot M_{\text{inp}} + I \odot (1 - M_{\text{inp}})$.

two, one for azimuth and one for elevation. We train this network using ground truth sun positions as supervision via a cross-entropy loss. In Fig. 3.2 and 3.5, we visualize estimated sun position as a 2D distribution formed by the outer product of azimuth and elevation distribution vectors.

3.3.4 Scene Geometry for Occlusion and Scale

When composed into a scene, a new object should be scaled properly according to its 3D scene position, and should have correct occlusion relationships with other scene structures. To that end, we desire accurate depth estimates for both the target scene and source object, and propose a method to robustly estimate depth for the scene and object jointly. Our method, unlike [198], also reasons about occlusion ordering for inserted objects. Here we take people as an example of inserted objects, but our method also works on other objects, including cars, bikes, and buses. We make three assumptions: (1) the sidewalk and road regions in the image can be well-approximated by a single plane; (2) there is at least one

person present; and (3) people are roughly the same height in 3D. If the second assumption is not met, the user can manually adjust the height scale. The third assumption, while not universally true, facilitates depth estimation by treating individual height difference as Gaussian noise.

Algorithm. As described in [167] (Eq. 7), any object's bottom middle point (x, y) and height h follow a linear relationship:

$$a'x + b'y + c' = h \tag{3.9}$$

Also under perspective projection, the object's height h is up to a scale factor k with its disparity $1/Z$:

$$h = k \cdot \frac{1}{Z} \tag{3.10}$$

Combining Eq. 3.9 and 3.10, we have a linear relation between pixel coordinates and the disparity $1/Z$:

$$a'x + b'y + c' = \frac{1}{Z} \tag{3.11}$$

Given the input image, we first use DeepLab [20] to segment pixels belonging to sidewalk and road. We then use MiDaS [128] to predict a depth map for the scene. MiDaS predicts the disparity map $\hat{D}$ up to a global scale and shift. Therefore, the linear relationship in Eq. 3.11 still holds. After collecting all 2D road/sidewalk pixels (x_i, y_i) and their disparities $\hat{d}_i$, we use least squares to solve for (a', b', c') in Eq. 3.11. Finally, we solve for the scale factor k in Eq. 3.10 using existing objects and their observed 2D heights. If there is no object in the scene, a user can manually set the scale factor.

When inserting a new object into the scene at 2D position (x, y), we apply Eq. 3.11 and Eq. 3.10 to estimate its disparity d and height h, and resize the inserted object accordingly. We then resolve occlusion order by comparing the object's disparity d and the scene's disparity $\hat{D}$ from MiDaS. Pixels with larger disparity than d are foreground and will occlude the object, and pixels with smaller disparity than d will be occluded by the object.

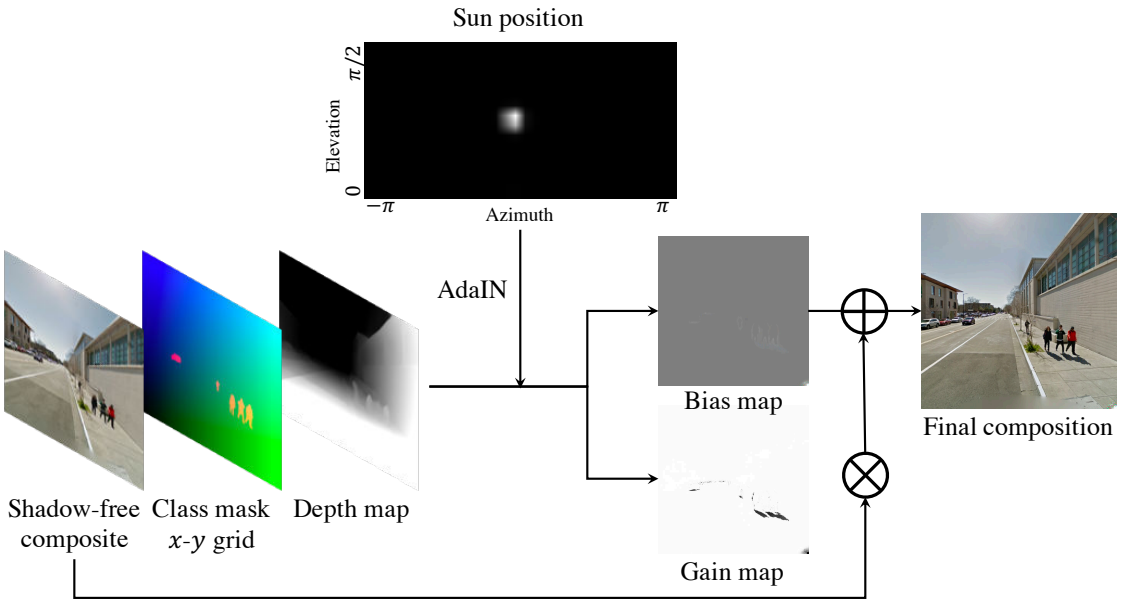


Figure 3.5: The generator of the insertion network takes as input a shadow-free composite image I_{comp}, a class mask, an x-y grid map, a depth map, and the predicted sun position distribution, and outputs a scalar gain image G and color bias image B. Given the shadow-free composite image I_{comp}, we synthesize the final image $I_{\text{final}} = G \odot I_{\text{comp}} + B$.

3.3.5 Insertion Network

Shadows are one of the most interesting and complex ways in which objects interact with a scene. As with the removal network, predicting shadows for inserted objects is challenging, as their shapes and locations depend on sun position, weather, and the shape of both the object casting the shadow and the scene receiving it. Furthermore, unlike other lighting effects, shadows are not always additive, as a surface already in shadow does not darken further when a second shadow is cast on it with respect to the same light source. We propose to use observations of objects in the scene along with the scene's predicted geometry and lighting to recover these shadowing effects, using a deep network to learn how objects cast shadows depending on their shape and scene placement. Unlike the work of Wang *et al.*, which is trained on a long video of a scene and can only insert objects within that same scene [167], our method learns from a database of short image bursts, and can then be

applied to a single, unseen image at test time.

Algorithm. Our insertion network takes as input a shadow-free composite image I_{comp} (where the desired object is simply copy-pasted into the scene). As with the removal network, we consider that shadow effects vary significantly across object categories, and we also provide the class mask introduced in Sec. 3.3.2 as input. In addition, because shadows depend on scene geometry and illumination, we use MiDaS [128] to predict a depth map for the shadow-free image, and feed this to the insertion network, along with the sun position distribution from the sun estimation network. Finally, following [167], we feed a x-y grid map to the network to help stabilize training. As shown in Fig. 3.5, I_{comp}, the class mask, x-y grid map, and depth map are concatenated and passed through four downsampling layers then five residual blocks. The sun azimuth and elevation vectors are concatenated, passed through four MLP layers and fused into five residual blocks via AdaIN [64]. Finally, following [167], two different upsampling layers generate a scalar gain image G and color bias image B. The final image is computed as

$$I_{\text{final}} = G \odot I_{\text{comp}} + B. \tag{3.12}$$

3.4 Evaluation

3.4.1 Occlusion Probing

Following Section 3.2.1, we generated occlusion maps for each scene; two of them are illustrated in Figure 3.6 in yellow to red pseudocolor. The quantization in colors corresponds to how objects moved through the scene; e.g., the two tones in the street correspond to the restricted placement of cars, which are generally centered in one of two lanes. The black regions correspond to pixels that were never observed to be covered by an object; these are treated as never-to-be-occluded during compositing. As a baseline, we also constructed depth maps using the state-of-the-art, depth-from-single-image MiDaS network [82]. MiDaS produces visually pleasing depth maps, but misses details that are crucial for compositing, such as the street signs toward the back of the scene in the second row of Figure 3.6.

Figure 3.7 shows several (shadow-free) composites. For our method, we simply place an object (such as a person, scaled by the method in Section 3.2.4) into the scene, and

Figure 3.6: From left to right: the background image, our estimated occlusion buffer, and the depth predicted by [82]. In our occlusion buffer, black pixels are never occluded. Pixels toward yellow were occluded only by smaller y-value objects, and pixels toward red were occluded by larger y-value objects. Some quantization and noise arises in our occlusion map due, respectively, to common object placements (cars in the road) and object mask errors (arising from object/background color similarity at a given pixel).

it is correctly occluded by foreground elements such as trees, benches, and signs. For the baseline method, the depth of the element must be determined somehow. Analogous to our plane estimation method for height prediction, we fit a plane to the scene points at the bottoms of person detections and then placed new elements at the depths found at the bottom-middle pixel of each inserted element. In a number of cases, elements inserted into the MiDaS depth map were not correctly occluded as shown in the figure, due to erroneous depth estimates and the difficulty of placing the element at a meaningful depth given the reconstruction.

Figure 3.7: Qualitative results on occlusion order estimation. People are reasonably well-occluded by foreground objects using our occlusion buffer, while errors in the depth map approach [82] (e.g., sign not reconstructed, bench reconstructed at ground level) result in incorrect occlusions. (Each of these images is a composite before adding shadows.)

3.4.2 Shadow Probing

We trained our shadow estimation network (Section 3.2.3) on each scene separately, i.e, one network per scene. On average, each scene had $17,000$ images for training with 900 images held out for testing. Figure 3.8 shows example results for shadow estimation using (1) a baseline pix2pix-style method [166] that takes a shadow-free image and directly generates a shadowed image and (2) our method that takes an x-y-mask image and produces bias and gain maps which are applied to the shadow-free image. Both networks had similar capacity (with 5 residual blocks). In this case, we also had ground truth, as we could segment out a person from one of the test images and copy them into the median background image for processing. The conventional pix2pix network tends to produce "blobbier" shadows when compared to the more structured shadows produced by our method, which is generally more similar to ground truth.

Figure 3.8: Shadow synthesis results on the test set. The images synthesized by a conventional pix2pix approach [166] (left) lack details inferred by our network (middle) which more closely resembles ground truth (right). In addition, the pix2pix method injects color patterns above the inserted person in the bottom row. Note that both networks learn not to further darken existing shadows (bottom row, sidewalk near the feet).

3.4.3 Depth (Ground Plane) Probing

For each input video, we predict the plane parameters from the training images using the method described in Section 3.2.4. When inserting a new object into the scene, we apply Equation 3.7 with regressed plane parameters to get its estimated height. We then resize it based on the estimated height, and copy-paste it onto the background frame.

To numerically evaluate the accuracy of the plane estimation as height predictor, we use it to measure relative, rather than absolute, height variation across the image. This measure factors out errors due to, e.g., children not being of average adult height as the absolute model would predict. In particular, we take one image as reference and another as a test

Figure 3.9: Ground plane estimation for height scaling. Given the person's reference height (taken from person in reference image), our algorithm accurately estimates the height in a different location (middle composite in each set). The difference between our estimate and the ground truth is small. (Each of these images is a composite before adding shadows.)

image with the same person at two different positions in the images. Suppose Equation 3.7 predicts height h in the reference image, but the actual height of the object is observed to be $\hat{h}$. The prediction ratio is then $r = \hat{h}/h$. For the same person in the test image, we then predict the new height h' again using Equation 3.7 and rescale the extracted person by $r \cdot h'$ before compositing again. We compared this rescaled height to the actual height of the person in the test image and found that on a small set of selected reference/test image pairs, the estimates were within 3% of ground truth. Figure 3.9 illustrates this accuracy qualitatively.

Note that without relative height prediction, i.e., instead using Equation 3.7 to predict absolute heights, the height prediction error was 13.28%, reasonable enough for inserting an adult of near-average height, though of course more objectionable when inserting, say, a young child. In our demo application, we allow the user to change the initial height of the inserted element.

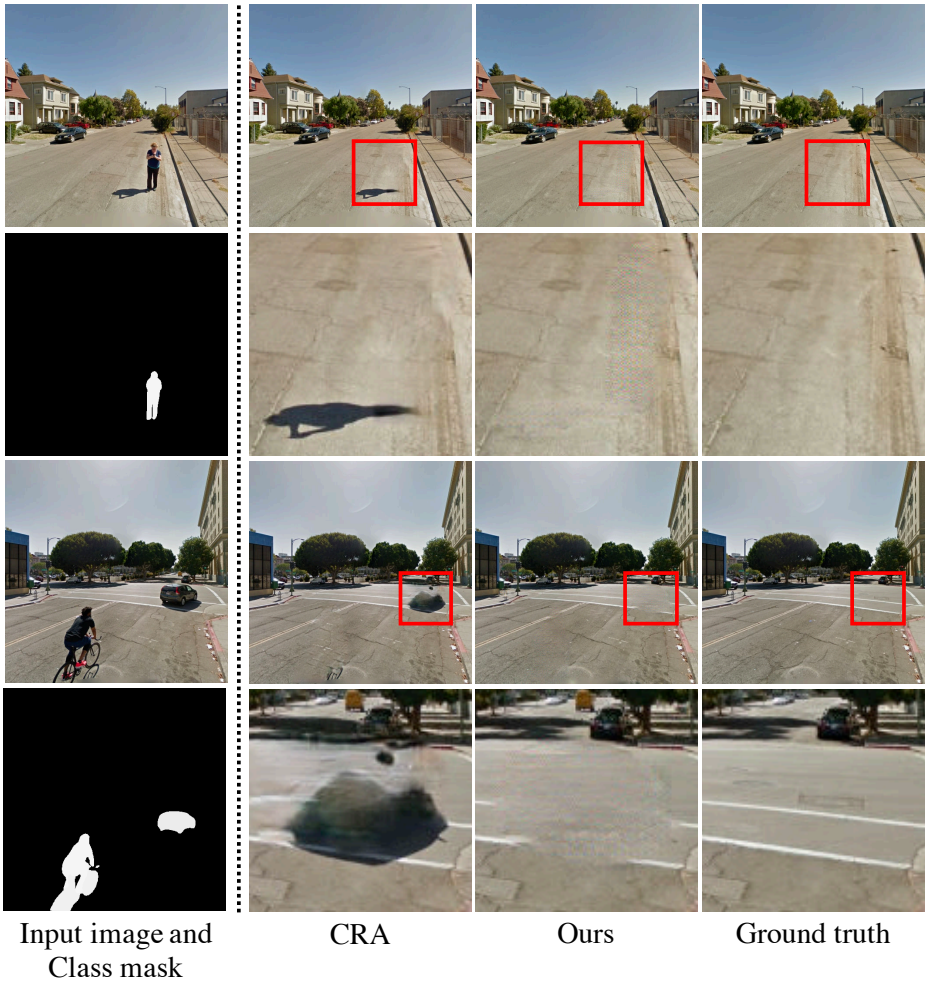


Figure 3.10: Object removal results on the test set. The traditional inpainting method [180] only inpaints the area within the mask and has leftover shadows. Our method removes objects completely along with shadows. In addition, the inpainting method fails to inpaint for large object (car in the second example).

3.4.4 Sun Estimation Network

We train our sun estimation network (Sec. 3.3.3) on the training set with ground truth supervision. Ground truth sun azimuth and elevation angles are calculated from each image's location, orientation, and timestamp using solar equations. Our network takes a street image and outputs two vectors describing distributions of azimuth and elevation angles.

Figure 3.11: Object insertion results on the test set. Our method generates the most realistic shadows with details. The sun position input helps the network to determine the shape of the shadow. The depth map prevents the network from synthesizing broken or detached shadows.

To compute a single pair of angles, we find the highest probability bin from each vector, and use the bin center as the estimated angle. We compare our sun estimation network with [101, 57], adding a fully connected layer to their method to predict the elevation angle. On average over the test set, our azimuth prediction has an angular error of 35.71° vs. 50.17° [57] vs. 52.59° [101], and our elevation prediction has an angular error of 9.79° vs. 13.02° [57] vs. 13.82° [101]. We further convert the sun angles to directions on the unit sphere and compute the angle between predicted and ground truth vectors, yielding an average error of 27.00°. These error rates are reasonably low, and of suitable accuracy for applications like lighting matching and shadow prediction.

Method	All	Sunny	Cloudy
Input	0.113	0.099	0.096
CRA [180]	0.107	0.090	0.083
Ours	**0.104**	**0.079**	**0.080**

Table 3.1: Object removal results on all test images, the sunny subset, and the cloudy subset, measured in LPIPS [189]. Lower is better.

3.4.5 Removal Network

We trained our removal network (Sec. 3.3.2) on the training set with median images as supervision for the object removed images. And the supervision for inpainting masks is computed by thresholding the color difference between median images and original images. Fig. 3.10 shows example object removal results on our test set using (1) the state-of-the-art inpainting method CRA [180] (trained on Places2 [192]), which only inpaints the mask area and (2) our method, which also predicts an enlarged inpainting mask. Both networks realistically inpaint the object region; however, CRA fails to remove object shadows since they are not included in the mask. Our network yields a complete object removal, which overall is more realistic. We show quantitative results using the LPIPS [189] error metric in Tab. 3.1. Both methods achieve a lower error compared to the input image. Ours has much lower LPIPS error than CRA on sunny days (where our method benefits from removing hard shadows), and slightly lower on cloudy days (where the shadows are more subtle). As shown in Fig. 3.10, our method removes objects completely and performs better in the task of object removal. We further tried running CRA [180] using our thresholded inpainting mask. This method gave an LPIPS score of 0.162 on the test set (vs. ours at 0.104). CRA is not trained with our inpainting mask, thus cannot adapt errors in our mask estimation, leading to artifacts in the final output.

Method	All	Sunny	Cloudy
Shadow-free composite	0.073	0.060	0.058
Shadow network	0.069	0.054	**0.056**
Ours (w/o x-y grid)	0.079	0.065	0.069
Ours (w/o sun position)	**0.068**	0.054	**0.056**
Ours (w/o depth map)	0.078	0.060	0.062
Ours	**0.068**	**0.053**	**0.056**

Table 3.2: Object insertion results on all test images, the sunny subset, and the cloudy subset, measured in LPIPS [189]. Lower is better.

3.4.6 Insertion Network

Our insertion network (Sec. 3.3.5) is trained to take shadow-free composite images and render object shadows, using original images as ground truth supervision. Fig. 3.11 shows example results using (1) a baseline pix2pix-style method [167] that takes a shadow-free image and an x-y grid; (2) an ablative method that takes a shadow-free image, x-y grid and depth map; (3) an ablative method that takes a shadow-free image, x-y grid, and predicted sun position; and (4) our method. All methods are trained on our training set. The predicted sun position helps the network produce shadows in the right direction. The depth map and x-y grid stabilize training, preventing the network from overfitting and producing broken or detached shadows. Quantitative results shown in Tab. 3.2 suggest that our method has an advantage over other models. On sunny days, our full model benefits from the depth map, sun position and x-y grid, and outputs realistic, detailed shadows. On cloudy days, our model synthesizes subtle soft shadows, still performing the best overall.

3.5 Applications

In this section, we discuss potential applications of our method to recomposing or repopulating single street images. These applications are enabled by one or more of the components of

our pipeline. We also discuss ethical considerations involved in such applications in Sec. 3.6.

Object lighting matching. When repopulating scenes, selecting objects with similar lighting as the scene is crucial for realistic composition. Hence, we wish to compute the sun position for both the source object and target scene. Hence, we train two sun estimation networks (Sec. 3.3.3), one for scenes and one for objects. The scene sun estimation network takes the image I and predicts sun azimuth and elevation vectors $a_{\text{scene}}, e_{\text{scene}}$, while the object sun estimation network pre-computes sun angle vectors $a_{\text{obj,i}}, e_{\text{obj,i}}$ for each object o_i in the collection. The object o_i that maximizes $a_{\text{scene}} \cdot a_{\text{obj,i}} + e_{\text{scene}} \cdot e_{\text{obj,i}}$ is then selected as the object that best matches the scene's lighting.

Emptying the city. Mask R-CNN [177] can segment out certain set of objects (people, cars, bikes, etc). Our removal network then takes this mask, and synthesizes an image without those objects along with their shadows. This enables applications such as removing all people and cars in NYC or LA. As demonstrated in Fig. 3.12, we use our removal network to remove all the objects—people and cars—in the image, giving users a different visualization of a city. Hence, it can also enhance the privacy of the imagery. Our method successfully removes all objects along with their shadows from the given street image.

Privacy enhancement. While removing all the people in the image enhances privacy, it decreases the liveliness of the street scene as well. To that end, we built a collection of people viewed from the back (or nearly the back) from licensed imagery on Shutterstock. Our pipeline can populate scenes with such people, thus enhancing privacy while retaining a sense of liveliness within the scene.

As above, our method can remove whole categories of objects to yield a background frame I_{back}. Then, we can use our object lighting matching method to find a set of best matching objects, then randomly place each object o_i on sidewalk and road regions in I_{back} via the segmentation map in Sec. 3.2.1. Objects will be automatically resized and occluded using the methods described in Sec. 3.2.1 to get the shadow-free composition I_{comp}. Finally I_{comp} is passed to the insertion network to synthesize the final composition I_{final}. Fig. 3.13 shows results for repopulating street scenes. We substitute the people in the scene with anonymized people, thus enhancing privacy while preserving the realism of street scenes. Note that our work focuses on lighting, and does not attempt to match the

Figure 3.12: Qualitative results for removing all people and cars in a street image. From left to right: the input image, the class mask for objects to be removed, and the removal results generated by the removal network. Our method removes objects completely.

camera viewpoint for inserted objects as in [81] or compensate for differences in camera exposure, white balance, etc. These are left as future work.

Other applications. We have also developed an interactive scene reconfigurator that leverages the elements of our framework. With this tool, a user can take a street image and remove selected existing objects, or conversely, place new objects in the scene. This tool can synthesize street images that are rare in real life, e.g., people walking in the middle of a busy road, or cars driving on the sidewalk. These synthesized scenes could be used for data augmentation for autonomous driving to simulate dangerous situations.

Figure 3.13: Qualitative results for repopulating street scenes. From left to right: the input image, the background image after removing all people, and repopulation results. Our pipeline selects people matching the scene's lighting, places them randomly on sidewalks and roads, and synthesizes realistic shadows.

3.6 Discussion and Ethical Considerations

In this chapter, I introduced People as Scene Probes, a fully automatic pipeline for inferring depth, occlusion, and lighting/shadow information from image sequences of a scene. The central contributions of this work are recognizing that so much information can be extracted just by using people (and other objects such as cars) as scene probes to passively scan the scene. I show that the inferred depth, occlusion ordering, lighting, and shadows are plausible, with the occlusion layering and shadow casting methods outperforming single-image depth estimation and traditional pix2pix shadow synthesis baselines.

I further introduced Repopulating Street Scenes, a fully automatic pipeline for populating, depopulating, or repopulating street scenes. The pipeline consists of four major components: (1) a removal network that can remove selected objects along with their shadows; (2) a sun estimation network that predicts the sun position from an image; (3) a method to scale and occlude inserted objects properly; and (4) an insertion network that synthesizes shadows for inserted objects. These components are trained on short image bursts of street scenes, and can run on a single street image at test time. Further, I show multiple applications of our pipeline for depopulating and repopulating street scenes.

While our work is motivated by goals like improving visualizations of scenes and enhance image privacy, it is important to consider the broader impacts and ethical aspects of computer vision research, particular work related to synthetic imagery. Potential harmful outcomes relating to recomposing street scenes include (1) misuse in creating a false narrative, such as a crowd or protest in a certain location, and (2) misrepresenting a neighborhood by changing the demographics of people therein. In our case, some issues related to synthetic media are mitigated by inherent limitations of our method — for instance, our method can compose separated people into scenes, and synthesize their shadows cast on the ground, but would have trouble generating a dense crowd of people where people would be shadowing each other. That said, any deployment of our methods in a real-world setting would need careful attention to responsible design decisions. Such considerations could include clearly watermarking any user-facing image that has been recomposed, and matching the distribution of anonymized people composed into a scene to the underlying

demographics of that location. At the same time, our work may lead to knowledge useful to counter-abuse teams working on manipulated imagery and synthetic media data methods.

Chapter 4

SUNSTAGE: PORTRAIT RECONSTRUCTION AND RELIGHTING

A light stage [28] acquires the shape and material properties of a face in high detail using a series of images captured under synchronized cameras and lights. This captured information can be used to synthesize novel images of the subject under arbitrary lighting conditions or from arbitrary viewpoints. This process enables a number of visual effects, such as creating digital replicas of actors that can be used in movies [1] or high-quality postproduction relighting [173].

In many cases, however, it is often infeasible to get access to a light stage for capturing a particular subject, because light stages are not easy to find: they are expensive and require significant technical expertise (often teams of people) to build and operate. In these cases, hope is not lost — one can turn to methods that are *trained* on light stage data, with the intention of generalizing to new subjects. These methods do not require the subject to be captured by a light stage but instead use a machine learning model trained on a collection of previously acquired light stage captures to enable the same applications as a light stage, but from only one or several images of a new subject [114, 150, 191, 13, 88, 144, 193]. Unfortunately, these methods have difficulty faithfully reproducing and editing the appearance of new subjects, as they lack much of the signal necessary to resolve the ambiguities of single-view reconstruction, i.e., a single image of a face can be reasonably explained by different combinations of geometry, illumination, and reflectance.

In this chapter, I propose an intermediate solution — one that allows for personalized, high-quality capture of a given subject, but without the need for expensive, calibrated capture equipment. The proposed method, SunStage, uses only a handheld smartphone camera and the sun to simulate a minimalist light stage, enabling the reconstruction of individually-tailored geometry and reflectance without specialized equipment. Our capture setup only requires the user to hold the camera at arm's length and rotate in place, allowing

(a) Reconstruction (b) Edit lighting (c) Soften shadows (d) Swap lighting (e) Edit texture

Figure 4.2: **SunStage.** Given a selfie video rotating under the sun, SunStage reconstructs geometry, material, camera pose, and lighting information. This recovered information can be used to (a) realistically re-render the input images, (b) modify the lighting conditions by adding / removing lights, (c) soften harsh shadows by changing the size of the reconstructed light sources (d) render the person in an entirely new environment, and (e) edit the albedo or material properties to add freckles, makeup, or stickers that realistically interact with scene lighting.

the face to be observed under varying angles of incident sunlight, which causes specular highlights to move and shadows to swing across the face. This provides strong signals for the reconstruction of facial geometry and spatially-varying reflectance properties. The reconstructed face and scene parameters estimated by our system can be used to realistically render the subject in new, unseen lighting conditions — even with complex details like self-occluding cast shadows, which are typically missing in purely image-based relighting techniques, *i.e.*, those that do not explicitly model geometry. In addition to relighting, we also show applications in view synthesis, correcting facial perspective distortion, and editing skin reflectance.

Our contributions include: (1) a novel capture technique for personalized facial scanning without custom equipment, (2) a system for optimization and disentanglement of scene

parameters (geometry, materials, lighting, and camera poses) from an unaligned, handheld video, and (3) multiple portrait editing applications that produce photorealistic results, using as input only a single selfie video.

4.1 Related works

Face modeling. Extensive research has been devoted to the modeling of human faces, leading to various 3D morphable models (3DMMs) [10, 19, 17, 12, 13, 25, 88, 130, 119, 118, 154, 7]. These models are parametric (maybe in the form of neural networks [130]), allowing one to express variations compactly with a vector. They also encode strong priors learned from real scans. The groundbreaking face 3DMM is that of Blanz and Vetter [10] containing models for shape, expression, and appearance (the Phong model). Also influential is the FLAME model [88] that uses vertex-based Linear Blend Skinning (LBS). FLAME is described by a mapping from shape, pose, and expression vectors to a list of vertices. We refer the reader to the survey by Egger *et al.* [36] for different face morphable models.

Such parametric face models provide a low-dimensional space for optimization or learning algorithms. DECA [37] uses the FLAME model to estimate detailed facial geometry (and albedo) from single images, by predicting additional displacement maps and adding them to the estimated FLAME models. More recently, NextFace [32] employs the 3DMM geometry and albedo priors to learn an albedo residual that captures more facial details.

Without modeling 3D face geometry, researchers have also achieved photorealistic synthesis of portrait images using generative models and large-scale high-quality image datasets [69, 70].

Light stage capture. The light stage as described in Debevec *et al.*, achieves impressive portrait reconstruction and relighting by capturing a series of images of the face under varying illumination [28]. Subsequent work made this process faster, more efficient, and explored different types of illuminants [103, 45, 40].

Given that a light stage is not always accessible, a number of methods have been proposed to achieve similar outputs from a single (or few) input portrait images [150, 191, 187, 114,

151, 179, 109, 61, 62]. These method rely on a dataset of light stage captures or synthetic examples as training data.

Our setup can be thought of as a "minimalist light stage" formed by just the sun and a rotating camera, without requiring the high construction and maintenance costs of building a light stage. This parameterization of a sun and skylight model has been shown to be effective in photometric stereo [59, 68] and scene factorization [152, 96]. In a similar spirit, Calian *et al.* [18] focus on lighting estimation using faces as "light probes". Sengupta *et al.* propose to circumvent the need for a complicated light stage by recording the facial appearance responses to varying contents displayed on a desk monitor, and then perform portrait relighting [142]. Sevastopolsky *et al.* also attempt to simplify the capture setup from a light stage to a mobile phone camera with a co-located flash [145]. Unlike our work, which is physically-based, their approaches use neural rendering, and therefore have less direct control over lighting, material, and scene parameters.

4.2 Overview

Our method targets accurate reconstruction of scene lighting, subject geometry, and material properties from a handheld video sequence of a person rotating in place under the sun. Given a selfie video, we take a test-time optimization approach that uses the information from all frames of the video to solve for a physical model of the scene: the geometry and material properties of the face, scene lighting, and camera parameters (Fig. 4.3). This physical model consists of a base face shape parameterized by a low-dimensional deformable model X^b, a displacement map ΔX, a reflectance model with diffuse R^d and specular components R^s, scene lighting L_i, and a perspective camera C. These components are explained in detail in Sec. 4.3.

After this model has been recovered, we can modify the scene and the subject parameters to re-render images. We show several editing applications in Sec. 4.6: editing skin reflectance, relighting with arbitrary environment map, improving harsh lighting conditions (by softening shadows and adding fill lights), and adjusting camera parameters to change viewpoint or manipulate perspective effects.

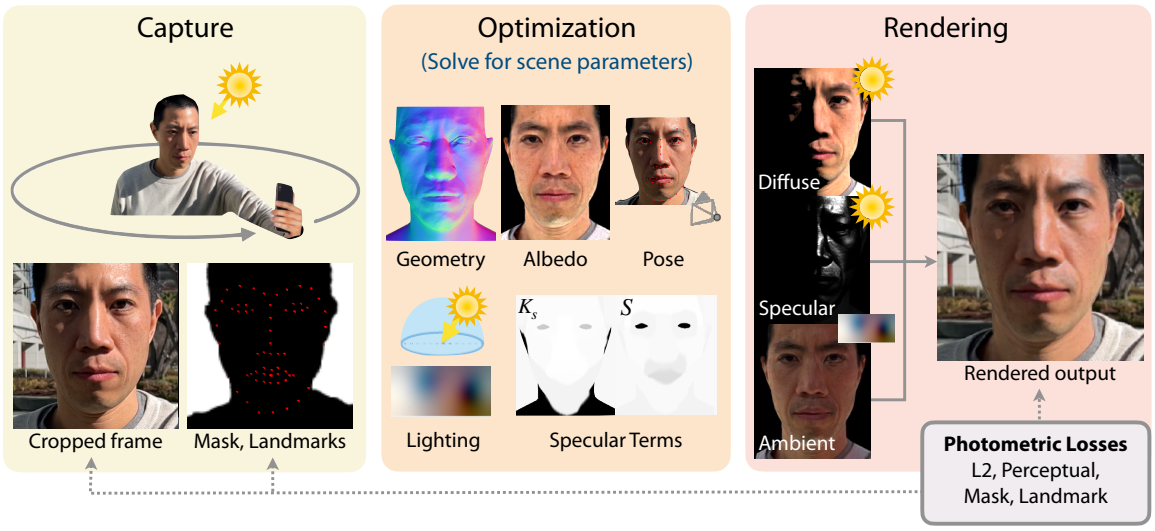


Figure 4.3: **Overview.** Our method jointly reconstructs geometry, skin reflectance, lighting, and camera pose from a selfie video sequence of a person rotating under the sun. Our system begins by extracting supervisory information from the video sequence: facial landmarks, foreground alpha mattes, and camera orientations. These are used to supervise the optimization of a collection of scene parameters (full list in Sec. 4.4.2) used in a physically-based renderer. The rendered output is an image consisting of diffuse, specular, and ambient light contributions. After optimization, the solved scene parameters can be used for a number of editing applications, shown in Sec. 4.6.

4.3 Formulation

Given an input video, our system reconstructs the parameters of a physical model: i.e., geometry and reflectance of the subject, the scene lighting parameters, and the camera parameters. In this section, we detail all of these parameters and describe the rendering process that turns these parameters into an image.

Geometry. We denote X_j as the full mesh of the subject for frame j, composed of a per-frame coarse mesh $X_j^b(\beta, \theta_j, \psi_j)$ and a global displacement map ΔX. The coarse mesh X_j^b is a FLAME deformable face model [88] defined by global shape code β, per-frame pose code θ_j, and per-frame expression code ψ_j. X_j^b also contains per-vertex UV coordinates, which maintain correspondence across variations in θ_j and ψ_j. As such, we model all our global

(per-subject) spatially varying parameters in UV space, and sample values per-fragment when rasterizing.

The displacement map ΔX is used to model fine details like wrinkles that cannot be represented by X_j^b. We displace the coarse geometry by ΔX at rasterization time, by sampling a displacement value per-fragment and displacing each fragment along the surface normal N_j of the coarse mesh X_j^b. After displacement, the updated fragment positions are used to compute a new surface normal N_j'.

X_j^b is optimized per-frame, since it accounts for subtle (and unavoidable) variations in expression and head pose during the capture, which are modeled by θ_j and ψ_j. ΔX, on the other hand, is optimized in UV-space (i.e., globally per-subject), since the deformations it encodes are invariant to the changes in expression or pose. Formally, the final geometry X_j is given by:

$$X_j = X_j^b(\beta, \theta_j, \psi_j) + \Delta X \odot N_j \tag{4.1}$$

where $\odot$ is the Hadamard product.

Reflectance. We model the skin reflectance, denoted as $R(x, \omega_i, \omega_o) \in \mathbb{R}^3$, where x is a 3D point on the face geometry X, ω_i is the incoming light direction, and ω_o is the outgoing direction, using a diffuse and a specular component: $R = R^d + R^s$.

The diffuse component $R^d(x, \omega_i) \in \mathbb{R}^3$ is a Lambertian reflectance model consisting of an albedo map, a, which we optimize as a per-subject UV-space image. For the skin's specular component, we use the Blinn-Phong model [11].

$$R^s(x, \omega_i, \omega_o) = k_s \frac{s+2}{2\pi} \left(h(\omega_i, \omega_o) \cdot n(x)\right)^s \tag{4.2}$$

where $h(\omega_i, \omega_o) = \text{normalize}(\omega_i + \omega_o)$ is the half vector, k_s is the specular intensity, s is the specular exponent, and $(s+2)/(2\pi)$ is the normalization term for the reflection lobe to integrate to 1. Following [174], we segment the UV-space map into 10 segmented specular reflectance clusters. We then optimize for a spatially-varying pair of values (s, k_s) per-cluster, enabling varying shininess across the face.

While Blinn-Phong does not model many complex effects such as subsurface scattering, our experiments with other models for facial reflectance, such as microfacet models [162],

show no significant quality improvements, and often introduce unstable training. More analysis is provided in the supplementary material.

Lighting. We use a sun-sky model to represent lighting as the sum of an "ambient" environment map and the sun: $L_i(x, \omega_i) = L_i^{\text{amb}}(\omega_i) + L_i^{\text{sun}}(\omega_i)$. Note neither $L_i^{\text{amb}}(\omega_i)$ nor $L_i^{\text{sun}}(\omega_i)$ depends on the 3D point x, since we model both as directional lights. Optimization-wise, our lighting parameters consist of a $16 \times 32 \times 3$ environment map for ambient lighting, the sun direction $p^{\text{sun}} \in S^3$, and the scalar sun intensity k^{sun}. We fix the sun color to white $[1, 1, 1]$ in our lighting model to resolve the albedo-illumination ambiguity.

4.3.1 Rendering

We calculate the outgoing radiance L_o at 3D location x as viewed from viewing direction ω_o as:

$$
\begin{aligned}
& L_o(x, \omega_o) \\
&= \int_\Omega V(x, \omega_i) L_i(x, \omega_i) \odot R(x, \omega_i, \omega_o)\, (\omega_i \cdot n(x))\, d\omega_i && (4.3) \\
&= \sum_{\omega_i} V(x, \omega_i) \Big(L_i^{\text{amb}}(\omega_i) \odot R^d(x, \omega_i) && (4.4) \\
&+ L_i^{\text{sun}}(\omega_i) \odot R^d(x, \omega_i) + L_i^{\text{amb}}(\omega_i) \odot R^s(x, \omega_i, \omega_o) \\
&+ L_i^{\text{sun}}(\omega_i) \odot R^s(x, \omega_i, \omega_o) \Big)\, (\omega_i \cdot n(x))\, \triangle\omega_i && (4.5)
\end{aligned}
$$

where $V(x, \omega_i)$ is the light visibility at x from ω_i, and $L_i(x, \omega_i)$ is the incoming radiance reaching x from ω_i. We ignore the specular reflection caused by the ambient lighting, *i.e.*, $L_i^{\text{amb}}(\omega_i) \odot R^s(x, \omega_i, \omega_o)$, since it is much weaker than the specular reflection of the sun. In the next subsections, we will group the terms into a diffuse contribution L_o^d and a specular contribution L_o^s: $L_o = L_o^d + L_o^s$. For the final rendered color value, we apply the Reinhard operator [133] and a gamma correction of $\gamma = 2.2$ to L_o to convert from linear to sRGB space.

Diffuse contribution. The diffuse contribution L_o^d is then given by only the diffuse terms

of Equation 4.5:

$$L_o^d(x) = \sum_{\omega_i} L_i^{\text{amb}}(\omega_i) \odot \frac{a(x)}{\pi}(\omega_i \cdot n(x))\Delta\omega_i$$
$$+ V(x, p^{\text{sun}})k^{\text{sun}}[1,1,1] \odot \frac{a(x)}{\pi}(p^{\text{sun}} \cdot n(x))\Delta p^{\text{sun}} \tag{4.6}$$

where $a(x)$ is the albedo at point x, k^{sun} is the (optimized) sun intensity, and p^{sun} is the (optimized) sun direction. The sun is modeled as a directional light source, so the second summation can be simplified to a single term, only in the direction of p^{sun}. We additionally optimize for a high-dynamic-range (HDR) environment map $E \in \mathbb{R}^{16\times32\times3}$, from which values of L_i^{amb} are sampled.

Specular contribution. The specular contribution L_o^s at each pixel is given by only the specular term due to the sun in Equation 4.5 (recall that we ignore the specular ambient term due to its weak contribution):

$$L_o^s(x, \omega_o) = V(x, p^{\text{sun}})k^{\text{sun}}[1,1,1]k_s\frac{s+2}{2\pi}$$
$$(h(p^{\text{sun}}, \omega_o) \cdot n(x))^s \, (p^{\text{sun}} \cdot n(x)) \, \Delta p^{\text{sun}} \tag{4.7}$$

where we have substituted Equation 4.2 and reduced the summation to just one term at p^{sun} (since L_i^{sun} is 0 elsewhere).

Shadow map. In order to generate a map of self-occluded shadows, we perform two passes of rasterization: first, we render a z-buffer from a virtual orthographic camera aligned with the sun direction, p^{sun}, and then, when rasterizing a given camera viewpoint, compare all fragment positions d_{hit} to the light's z-buffer d_{shadow}. To avoid precision issues and ensure smooth gradients for back-propagation, we implement a soft comparison as follows in generating shadow/visibility maps:

$$V(x, \omega_i) = 1 - \text{sigmoid}\left(k(d_{\text{hit}} - d_{\text{shadow}} \times b)\right) \tag{4.8}$$

where k is the falloff slope, and b the tolerance. We use $k = 800$ and $b = 1.0015$.

4.4 Optimization

The described physical model contains a large number of parameters to be optimized, controlling scene elements like lighting, geometry, pose, and texture. Unfortunately, naïvely

optimizing all these parameters from scratch does not result in an optimal solution, since the final observed appearance of the face can often be explained variously through changes to geometry, material properties, lighting or camera parameters, making optimization severely under-constrained and ambiguous. Therefore, we adopt a two-stage optimization approach, through which parameters are gradually enabled. In this section, we describe this process and the relevant losses that guide optimization.

4.4.1 Coarse alignment

Our system begins by using an off-the-shelf network (DECA [37]) to generate, for each input image, a set of shape parameters β, pose parameters θ_j, and expression parameters ψ_j of a FLAME face model [88], as well as the relative pose parameters of the virtual camera observing the 3D face. Unfortunately, as with many other single-image facial geometry estimators, DECA assumes an orthographic projection model and therefore cannot accurately recover geometry for our selfie capture sequences, which contain heavy perspective effects (Figure 4.7). Without a good initialization for geometry, optimization of lighting and material properties seldom converges to an optimal solution due to the heavily ambiguous nature of our optimization problem.

To circumvent this issue, we employ a first stage of optimization where we only optimize for the parameters of a perspective camera (with a known focal length, extracted from input metadata) and the face geometry parameters $(\beta, \theta_j, \psi_j)$. As initialization for this optimization process, we use the predicted DECA values for each frame's pose θ_j and expression ψ_j, but set all frames to the average predicted shape $\beta_{\text{avg}} = \frac{1}{N}\sum_j \beta_j$, since the identity remains constant across all frames. To convert DECA's orthographic camera to a perspective camera, we additionally optimize for an unknown scale S and translation T_j, which are initialized to empirically chosen values $S = 2.6e4$, $T_j = (0, 0, 1.5e5)$. During optimization, the face shape β and scale S are shared across all frames, while camera pose T_j, expression ψ_j, and pose θ_j are optimized per-frame. Note that DECA controls the relative orientation of the camera and the face by varying the pose code θ instead of the camera rotation. We adopt this formulation and keep the camera orientation fixed relative

to the face. The global orientation of the camera at each frame (and therefore the face) is extracted from the capture video, either through a structure-from-motion system or IMU measurements commonly available on a smartphone.

We use two losses to guide this optimization: a mask loss L_{mask} and a landmark loss L_{lmk}. The FLAME model includes 3D facial landmark points, corresponding to the standard 68-point facial landmarks set [138] used in facial tracking. Our landmark loss minimizes the L1 distance between the 2D projection of these 3D landmarks (into the input camera viewpoint) and 2D landmarks estimated from the input frame by a 2D landmark detector HRNets [164].

The facial landmarks provide a stong constraint on facial feature alignment, but are sparse, and therefore cannot constrain the overall shape or boundary of the mesh. To supplement it, we include a silhouette loss L_{mask}, which penalizes the L2 difference between the rasterized mask of the mesh I_{sil} and the semantic segmentation mask I_{mask} of the input image, using an off-the-shelf semantic segmentation network [93] trained to segment humans in portrait photographs.

The final pose loss is then: $L_{\text{pose}} = L_{\text{mask}} + L_{\text{lmk}}$, optimized using an ADAM optimizer [76]. See supplemental for optimization parameters.

4.4.2 Photometric optimization

Once the 3D model and camera parameters are approximately aligned, we proceed to the second stage of optimization, in which we optimize the precise facial geometry, lighting, and reflectance properties. All the parameters optimized in the first stage (Section 4.4.1) remain as free variables. In total, the parameters optimized during this stage include: **Lighting parameters**: (1) p^{sun}, the global sun direction, (2) E, the global environment map, (3) k^{sun}, the global sun intensity, **Facial geometry parameters**: (4) β, the global FLAME shape code, (5) ψ_j, the per-frame expression code, (6) θ_j, the per-frame pose code, (7) ΔX, the global deformation map, **Material properties**: (8) k_s, the global, spatially-varying specular intensity, (9) s, the global, spatially-varying specular roughness, (10) a, the global, spatially-varying surface albedo, **Camera pose parameters**: (11) T_j, the per-frame

Figure 4.4: **Qualitative: Relighting**. A comparison of our method at rendering a new (unseen) lighting environment (h). Our method is able to realistically synthesize the novel lighting condition, including cast shadows and specularity, and nearly matches the (unseen) target reference image. See supplement for additional details on experimental setup and analysis of results.

perspective camera translation, and (12) S, the global scene scale.

During optimization, we randomly select a frame j, render the face using a differentiable rasterizer [131] and the equations described in Section 4.3 to get the rendered image $\hat{I}$. In addition to the previously defined landmark and mask losses, we include L2 and VGG [67] photometric losses, comparing the original and reconstructed images:

$$L_{\text{photo}} = ||\hat{I}_j \cdot I_{\text{sil}} - I_j \cdot I_{\text{mask}}||_2 \tag{4.9}$$

We also include an L2 regularization L_E and L2-smoothness regularization L_{Es} on the reconstructed environment map, to encourage the majority of the lighting to be explained

(a) Nearest view (b) DECA [37] (c) NextFace[32] (d) NLT [190] (e) Ours (f) GT

Figure 4.5: **Qualitative: view synthesis**. A comparison of our method at the task of generating an image from an unseen viewpoint (f), having only seen a limited collection of input viewpoints. See supplement for more details on experimental setup and analysis of results.

by direct sunlight and to aid in disentanglement of the sun and ambient lighting. The total optimized loss becomes:

$$\begin{aligned} L = \lambda_{\text{mask}} L_{\text{mask}} + \lambda_{\text{lmk}} L_{\text{lmk}} + \lambda_E L_E + \lambda_{E\text{s}} L_{E\text{s}} \\ + \lambda_{\text{VGG}} L_{\text{VGG}} + \lambda_{\text{photo}} L_{\text{photo}} \end{aligned} \tag{4.10}$$

with $\lambda_{\text{mask}}, \lambda_{\text{lmk}} = 0.05, \lambda_{\text{VGG}} = 0.005, \lambda_E = 0.01, \lambda_{E\text{s}}, \lambda_{\text{photo}} = 1$. Additional optimization details are provided in the supplemental materials.

4.5 Evaluation

In this section, we detail quantitative and qualitative experiments comparing our approach with state-of-the-art methods and ablated variants of our method.

Baseline comparisons. We evaluate our method on the tasks of novel-view synthesis and relighting. For novel-view synthesis, we compare our method with DECA [37], Neural Light Transport (NLT) [190], and NextFace [32]. For relighting, we compare with DECA, NLT, NextFace, GCFR [61], image-based methods Deep Single Image Portrait Relighting (DPR)

Figure 4.6: **Decomposition.** We show all the components which comprise our final rendered image to demonstrate that our method not only closely recreates the ground truth image (reproducing realistic highlights and shadows), but also performs a meaningful decomposition of lighting components and facial geometry. Note that our reconstructed surface normals include high frequency details specific to each subject, like wrinkles and birthmarks, which are used in computation of the shadows and specular reflections.

[193], Total Relighting (TR) [114] and NVPR [187]. Additional comparisons and details on the experimental setups are provided in the supplemental materials.

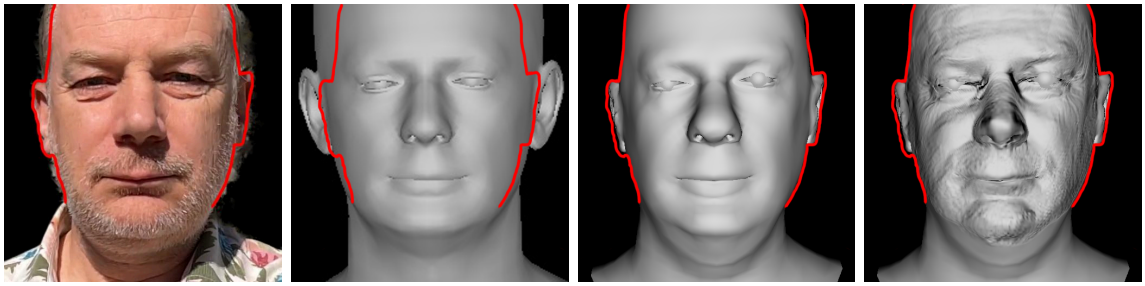
We present qualitative comparisons for relighting in Figure 4.4 and novel view synthesis in Figure 4.5. Quantitative comparisons on these images are provided in Table 4.1. These testing images consist of (1) a multi-view capture of the face, in which the subject remains still and the camera is moved to novel viewpoints in the same environment as the original capture, and (2) front-facing sequences in novel environment lighting and unseen sun positions. All testing images are not seen during training of our method, NLT or NextFace. The results shown in Figures 4.4 and 4.5 as well as Table 4.1 clearly demonstrate that our method outperforms all the baselines at both relighting and view synthesis. Single-image methods (DECA, GCFR, DPR, TR) can generalize to other subjects, but fail to recover

	Relighting			Novel view synthesis		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
DECA [37]	16.41	0.69	0.25	16.64	0.66	0.29
GCFR [61]	16.97	0.70	0.20	-	-	-
DPR [193]	19.03	0.72	0.19	-	-	-
NLT [190]	20.15	0.75	0.18	22.27	0.79	0.15
Total Relighting [114]	20.24	0.79	0.16	-	-	-
NextFace [32]	22.98	0.76	0.15	22.55	0.75	0.15
Ours	23.64	0.83	0.10	25.28	0.84	0.09
Ours w/o coarse	17.83	0.66	0.23	19.65	0.70	0.17
Ours w/o SV k_s, s	21.31	0.77	0.13	21.94	0.77	0.12
Ours w/o L_{mask}, L_{lmk}	16.46	0.61	0.30	18.83	0.68	0.20
Ours w/o L_{mask}	20.13	0.75	0.15	20.54	0.74	0.15
Ours w/o opt. $(\beta, \theta_i, \phi_i)$	18.67	0.69	0.19	18.28	0.66	0.19
Ours w/o soft shadow	21.46	0.77	0.13	22.05	0.77	0.12
Ours w/o ΔX	21.16	0.75	0.15	21.80	0.75	0.14

Table 4.1: **Quantitative comparison**. Comparison of our method on the tasks of novel view synthesis and relighting. See Section 4.5 for a description of the ablated variants.

more faithful and physically accurate facial details. Comparison with multi-image methods (NLT, NextFace) demonstrates that SunStage is a better reconstruction system. Additional analysis of the comparisons is provided in the supplemental materials.

Disentanglement. In Figure 4.6, we demonstrate how SunStage decomposes the appearance of a portrait photograph into different components: specular, diffuse, and ambient. We also visualize the surface normal, albedo, and other intermediate representations to show that our method is able to effectively recover a physically plausible reconstruction of the



(a) Input image (b) DECA [37] (c) Stage 1 (d) Final

Figure 4.7: **Perspective**. DECA's assumption of an orthographic camera is broken by the strong perspective effects in selfies, causing poor alignment (b) with input images (a). Our first stage of optimization (c) (Sec. 4.4.1) improves alignment by solving for the parameters of a perspective camera and refined shape parameters. In the second stage we additionally optimize for a displacement map ΔX to produce our final shape with finer geometric details like wrinkles (d). Red line added to highlight alignment with (a).

real world and disentangle the different components that contribute to the final appearance. We further validate the quality of the reconstructed geometry and materials in the supplementary material.

Ablation studies. In addition to our comparisons with the state-of-the-art baselines, we also compare with ablated variants of our own method. In particular, we include seven such experiments in Table 4.1: our method (1) without the initial first stage of coarse geometric alignment, i.e., directly optimizing both geometric and photometric parameters from the start, (2) without the spatially varying specular parameters, instead using a single global scalar s and k_s, (3) without the geometric alignment losses L_{mask} and L_{lmk}, (4) without just L_{mask}, (5) without shape optimization, i.e., keeping the initial shape code predicted by DECA, (6) without soft shadow computation, i.e., using a hard z-buffer comparison to compute a shadow map instead of our soft comparison operator in Equation 4.8, and (7) without the displacement map ΔX. Visual results for each of these variants are provided in the supplemental material.

4.6 Applications

Relighting. We demonstrate two types of relighting applications: (1) lighting modification and (2) lighting replacement. Practical lighting modification is common in portrait photography when the lighting conditions are not ideal, e.g., when direct sunlight casts undesirable harsh shadows with high contrast. A common practice is to make the light source larger and more diffuse by using a scrim or bounce card. In Fig. 4.8b, we show that by virtually increasing the size and spreading the energy of our reconstructed lighting source (*i.e.*, the sun), we are able to *soften* the shadows and re-render a more visually pleasing face. Another approach to reducing the effects of harsh shadows is adding local fill lights, which reduces the contrast between the lit and shaded regions (Fig. 4.8c). Alternatively, fill lights can also be used for artistic purposes, to create dramatic lighting effects (Fig. 4.2b). Finally, replacing the scene lighting with that of a novel environment (Fig. 4.8d) is a necessary step in realistically inserting a captured subject into a virtual scene, which is useful for visual effects and VR applications.

View Synthesis. In Figure 4.9, we show that our reconstructed 3D model of the face can be used to synthesize new views by manipulating the viewpoint of the camera. We can also change other camera parameters, such as the focal length, to reduce the perspective effects on the face, which is often desirable for selfie images that contain significant facial distortion due to perspective.

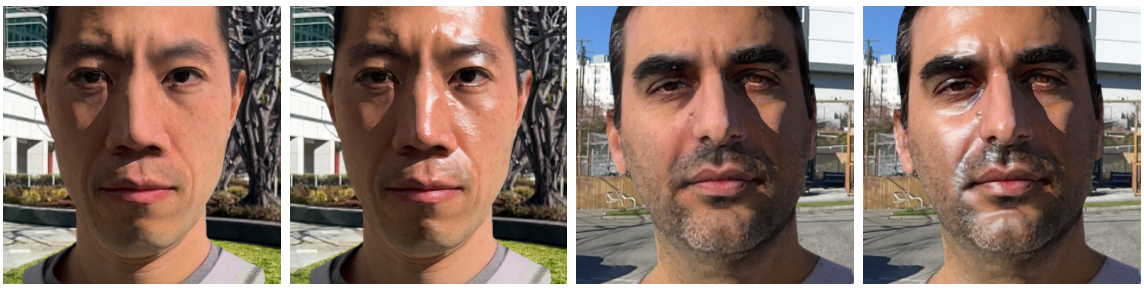
Skin Reflectance Editing. We are also able to edit the reflectance components of the subject. As shown in Figure 4.2e, we can adjust the optimized albedo to add freckles, stickers, or other textures that realistically interact with reflections, shadows, and other elements of scene lighting, or we can adjust the specular properties of the face, making the face more or less shinier, as shown in Figure 4.10.

(a) Input (b) Soft shadows (c) (b) + fill light (d) Swap scene

Figure 4.8: **Adjusting lighting parameters**. We can adjust the recovered scene parameters to improve the lighting conditions in an input image (a) by softening the harsh shadows cast by the nose (b), adding a fill light to brighten the shaded region (b), or replacing the environment altogether (d).

(a) Short focal (b) Long focal (c) Novel view 1 (d) Novel view 2

Figure 4.9: **Changing camera parameters**. We can change the recovered camera parameters to render novel views (c,d) or change the focal length (a,b).



(a) Less shinier (b) More shinier (c) Less shinier (d) More shinier

Figure 4.10: **Adjusting the specularity**. We can change the the specular properties of the face, making the face less shinier (a, c) or more shinier (b, d).

4.7 Conclusion and Discussions

In this paper, we propose SunStage, a lightweight and practical facial capture, rendering, and editing system that can serve as a minimalist light stage. With a video of an individual rotating in-place under the sun, our system reconstructs a physical model of the subject and the scene lighting, which enables us to relight the subject with realistic reflections and cast shadows. Our system allows arbitrary lighting and reflectance control in the reconstructed physical space, which can be rendered to produce photo-realistic results. We demonstrate several applications such as editing skin reflectance, relighting, and view synthesis.

Limitations. Our system inherits the limitations of morphable face models and is unable to model hair, teeth, or clothing geometry, beyond slight deformations. Additionally, certain regions which are seen under constant shadow or specular reflection (and therefore have no cues on reflectance or albedo) are sometimes unable to be decomposed accurately into separate reflectance and lighting components. Visualization and further discussions of the system's limitations are provided in the supplemental material.

Chapter 5

INFINITE TEXTURE: TEXTURE SYNTHESIS AND TEXTURE TRANSFER

This chapter presents Infinite Texture [168], a method for generating arbitrarily large texture images from a text prompt. I also showcase two applications of the generated textures in 3D rendering and texture transfer. A minimal dataset is constructed to ensure consistent shading in the texture transfer results, adding a sense of depth and three-dimensionality.

5.1 Related Work

Texture synthesis. Texture synthesis has been an active research area in computer graphics for several decades. Traditional approaches for texture synthesis can be broadly categorized into two groups: pixel-based synthesis and patch-based synthesis. Efros and Leung [35] proposed to synthesize textures by gradually growing the map from the initial texture, assigning the output pixels one by one in an inside-out, onion-layer fashion. Despite its elegance and simplicity, this method is relative slow and subject to non-uniform pattern distribution. Wei *et al*. [171] proposed a simple pixel-based texture synthesis algorithm based on fixed neighborhood search. The quality and speed of pixel-based approaches can be improved by synthesizing patches rather than pixels. Praun *et al*. [122] repeatedly pastes new patches over existing regions. By using patches with irregular shapes, this method takes advantage of the human visual system and its perception of texutre mapping effects, making texture synthesis work surprisingly well for stochastic textures. Liang *et al*. [90] take a different approach by using a blending algorithm for overlapping regions. Efros and Freeman [34] use dynamic programming to find an optimal path to cut through the overlapped regions, and this idea is improved by [79] via graph cut. Kaspar *et al*. [73] further extends Texture Optimization [79] by making its various parameters and weights to be self-tunable.

Another line of works use Cellular Automata (CA) for texture generation [156, 176].

Recent work by Niklasson *et al*. [111] proposed a learning-based Texture Neural Cellular Automata model where the CA update rule is parameterised with a small neural network. This idea is further investigated in [108] by making the model more compact.

Texture synthesis has seen significant progress with the advent of deep learning-based methods. Gatys *et al*. [42] synthesize new textures by minimizing the Gram loss with the original texture. Followup works [158, 159, 89] speed up the synthesis by adopting a feed-forward generative network. Jetchev *et al*. [66] utilize GANs to generate texture patches from random noise of the same size. Bergmann *et al*. [5] extends this idea by by introducing a periodic function into the input noise, which enables synthesizing high-quality periodic textures. Zhou *et al*. [195] train a generative network to double the spatial extent of texture blocks for synthesizing non-stationary textures. A GAN-based texture synthesis method was used by Verbin and Zickler [161] for estimating surface shape using texture cues, and Verbin *et al*. [160] presented a mathematical formulation for the uniqueness of the solution to the surface recovery problem.

Traditional texture synthesis methods are largely slow and inefficient. Deep learning-based methods are restricted by the receptive field and fail to learn the extreme low- or high-frequncy signal in the texture. We introduce a novel method to leverage image priors from diffusion models and synthesize arbitrarily large, high-quality textures on a single GPU.

Text-to-image synthesis. Generating realistic images from textual descriptions is a challenging task. Early attempts [102, 132, 153, 178, 185, 197] we limited as they employed text-conditional GANs on specific domains [172] and their datasets held closed-world assumptions [94]. However, recent advances in diffusion models [31, 54] and large-scale language encoders [124, 126] have greatly improved text-to-image synthesis, enabling these models to be conditioned on an open-world vocabulary of arbitrary text descriptions. Prominent diffusion models, such as GLIDE [110], DALL-E 2 [127], and Imagen [139], produce photorealistic outputs with the aid of a pretrained language encoder [124, 126]. Although diffusion models have demonstrated unprecedented image synthesis ability, the iterative image sampling processes in the denoising step is often time-consuming. To accelerate the

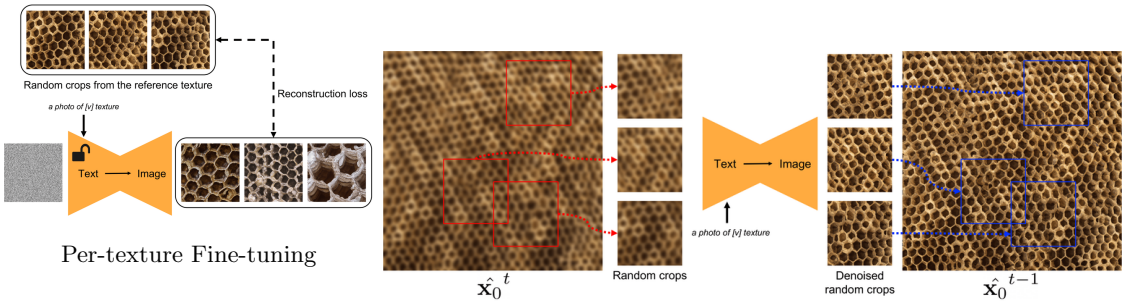


Figure 5.1: **Overview.** Infinite Texture consists of three stages: (1) Generating a reference texture image from the text prompt; (2) Fine-tuning a diffusion model to learn the texture statistics of a particular reference texture image. We train the model using random crops of the reference texture image along with a unique identifier. The text encoder is also trained during this stage; and (3) Using the trained diffusion model to synthesize arbitrarily large textures: at every timestep of diffusion, we denoise small random patches and combine their estimates by taking the average noise estimate in overlapping regions. The resulting noise estimate is used to perform a DDIM sampling step.

sampling, several methods [100, 104, 140, 147] propose solutions for reducing the number of sampling steps. Latent Diffusion Models [135] also greatly speeds up the sampling by sampling in a low-dimensional latent space instead of pixel space.

Since off-the-shelf diffusion models are trained to produce arbitrary images, they are often not particularly well suited for synthesizing stochastically varying periodic textures. We propose a new way to help diffusion models produce textural content by fine-tuning a model to learn texture statistics. Diffusion models are also difficult to use for synthesizing high resolution images due to the memory constraint. For this, we further introduce a strategy for generating spatially-consistent and large textures at inference time.

5.2 Method

Our objective is to generate a diverse collection of high-resolution and high-quality texture samples based on an input text prompt. Infinite Texture achieves this in three stages,

as shown in Fig. 5.1: (1) generating a reference texture image from the text prompt, (2) fine-tuning a diffusion model to learn the statistical distribution of the texture, and (3) combining the output of the diffusion model to synthesize a high-resolution texture. We next provide some background on diffusion models (Sec. 5.2.1), our fine-tuning technique to learn the texture statistics (Sec. 5.2.2), and our novel approach for texture synthesis (Sec. 5.2.3).

5.2.1 Diffusion Models

Diffusion models are probabilistic generative models that are trained to learn a data distribution by gradually denoising a variable sampled from a Gaussian distribution. Specifically, we are interested in a pretrained text-to-image diffusion model $\hat{\mathbf{x}}_\theta$ that, given an initial noise map $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and a conditioning vector $\mathbf{c} = \Gamma(\mathbf{P})$, generates an image $\mathbf{x}_{\text{gen}} = \hat{\mathbf{x}}_\theta(\boldsymbol{\epsilon}, \mathbf{c})$. The conditioning vector is generated using a text encoder Γ and a text prompt $\mathbf{P}$. The pretrained model is trained to minimize a squared error loss to denoise a latent code $\mathbf{z}_t \coloneqq \alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}$ as follows:

$$\mathbb{E}_{\mathbf{x},\mathbf{c},\boldsymbol{\epsilon},t}\left[w_t \|\hat{\mathbf{x}}_\theta(\alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}, \mathbf{c}) - \mathbf{x}\|_2^2\right] \tag{5.1}$$

where $\mathbf{x}$ is the ground-truth image, $\mathbf{c}$ is a conditioning vector (e.g., obtained from a text prompt), and α_t, σ_t, w_t are terms that control the noise scheduler and sample quality, and are functions of the diffusion process time $t \sim \mathcal{U}([0, 1])$. At inference time, the diffusion model is sampled by iteratively denoising $\mathbf{z}_{t_1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ using either the deterministic DDIM [147] or the stochastic ancestral sampler [54]. Intermediate points $\mathbf{z}_{t_1}, \ldots, \mathbf{z}_{t_T}$, where $1 = t_1 > \cdots > t_T = 0$, are generated, with decreasing noise levels. These points, $\hat{\mathbf{x}}_0^t \coloneqq \hat{\mathbf{x}}_\theta(\mathbf{z}_t, \mathbf{c})$, are functions of the $\mathbf{x}$-predictions.

Recent state-of-the-art text-to-image diffusion models use cascaded diffusion models in order to generate high-resolution images from text [139, 127]. Specifically, DALL-E 2 [127] uses a base text-to-image model with 64×64 output resolution, and two unconditional super-resolution (SR) models $64 \times 64 \rightarrow 256 \times 256$ and $256 \times 256 \rightarrow 1024 \times 1024$. We use DALL-E 2 to generate a 1024×1024 reference texture image from the text prompt.

5.2.2 Per-Texture Fine-tuning

The aforementioned diffusion models [127, 135] have shown unprecedented capabilities due to their ability to synthesize high-quality and diverse images based on text prompts. One of the main advantages of such model is strong image priors learned from a large collection of images that the model is initially trained on. While these models are capable of synthesizing interesting and high-quality images, they still lack the ability to reproduce the statistical distribution of a given reference texture image or to synthesize novel variations of the same texture.

Wang *et al*. [165] have demonstrated that a fine-tuned diffusion model outperforms a model trained from scratch for image translation tasks, especially when paired training data is limited. Following [165, 195], we choose to fine-tune a diffusion model for each reference texture image in order to learn the specific statistics associated with that texture. We initialize our model with weights of a pretrained Stable Diffusion v2 [135] checkpoint, leveraging its strong image priors. To ensure consistency in the output texture, we follow [136] and fine-tune the diffusion model with a unique identifier per texture.

The pretrained Stable Diffusion is trained on a large and diverse dataset. However, we only require priors from a small portion of the training data to enable texture synthesis. We fine-tune the pretrained Stable Diffusion checkpoint to purposefully overfit on our reference texture image. Given a reference texture image of 1024×1024, we take random crops of 768×768 and fine-tune the diffusion model on these texture patches. We overfit the model because we only want to inherit image priors, not semantic priors from the pretrained Stable Diffusion. For all patches, we use the text prompt $\mathbf{P}$ of "a photo of [identifier] texture" to prevent any semantic prior. To further disentangle the semantic prior from the image prior, the text encoder Γ is also fine-tuned with the diffusion model $\hat{\mathbf{x}}_\theta$. We use the same training loss as in Eq. 5.1 to fine-tune the diffusion model in an end-to-end manner. Weights for both the text encoder and the UNet are updated together at every iteration.

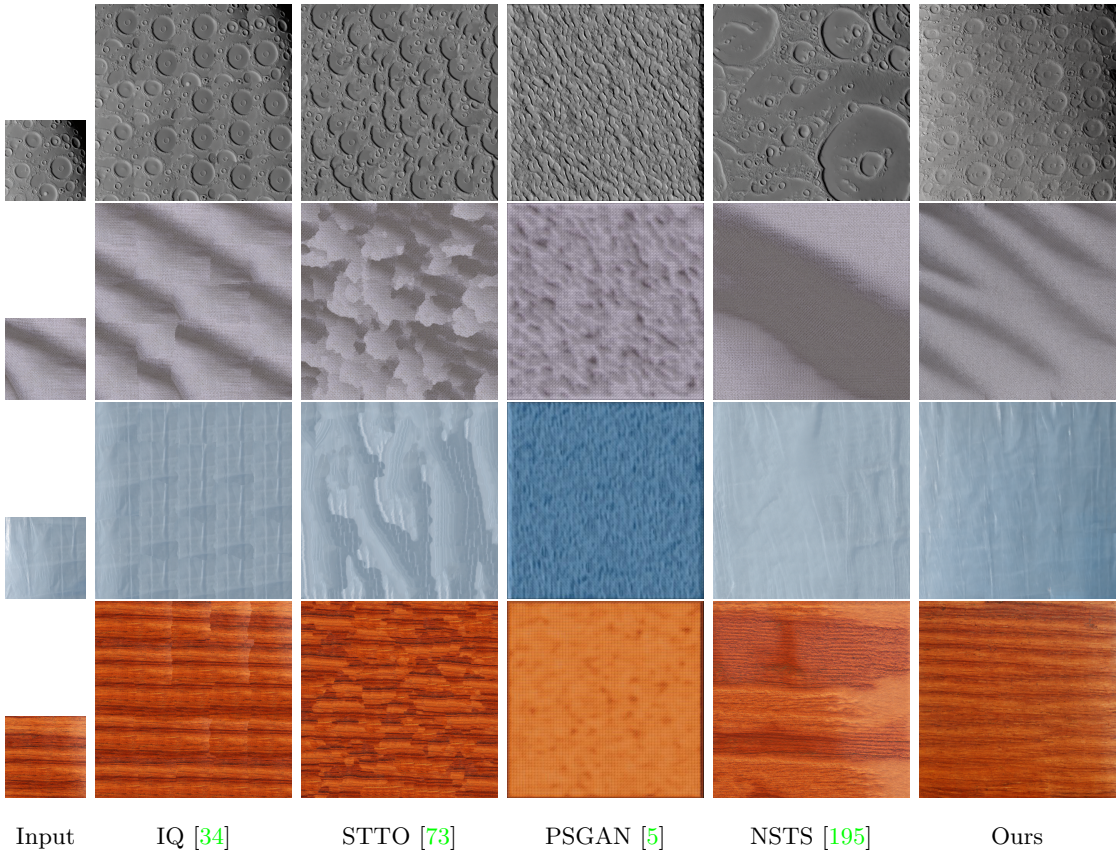


Figure 5.2: **Comparisons with baseline methods in texture synthesis.** We use texture images generated by DALL-E 2 [127] as input exemplar textures. Infinite Texture stands out by generating the most consistent textures with variations. In contrast, image quilting [34] often results in repetitive patterns due to its tiling strategy. STTO [73] tends to converge to solutions where a smooth patch is repeated over and over. PSGAN [5] struggles to capture the high-frequency signal effectively with its periodic signal generator. Non-stationary texture synthesis [195] falls short in producing textures with variations due to having only been trained on small patches.

5.2.3 Texture Synthesis

A naïve approach to generating a high-resolution texture image is to directly denoise a large latent code map $\mathbf{z}_{t_1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ using the fine-tuned diffusion model. However, due to the computational limitations imposed by the model's size, it is not feasible to denoise such

high-resolution latent code, *e.g.* 2048×2048, on a single GPU. To address this constraint, we adopt a progressive denoising strategy inspired by MultiDiffusion [3].

Instead of denoising the entire noisy latent map at once, we take a patch-by-patch decoding approach similar to traditional texture synthesis methods. We extract a set of random crops F_i, that crops the latent map to a constant resolution, specifically 96×96. Since the decoder of Stable Diffusion upsamples the latent map by a factor of 8, F_i would translate to an image of size 768×768. At each denoising time step t, we employ a score aggregation strategy. We begin by denoising the random crops in the latent space:

$$\hat{\mathbf{x}}^t_{0_i} = \hat{\mathbf{x}}_{\theta_{\text{ref}}}(F_i(\mathbf{z}_t), \mathbf{c}) \tag{5.2}$$

where $\mathbf{z}_t$ is the noisy latent map at time step t and $\hat{\mathbf{x}}^t_{0_i}$ is the denoised patch of $F_i(\mathbf{z}_t)$. We then combine the denoised patch $\hat{\mathbf{x}}^t_{0_i}$ to obtain the large $\mathbf{x}$-prediction map $\hat{\mathbf{x}}^t_0$. Following [3], we formulate the combination of $\hat{\mathbf{x}}^t_{0_i}$ as an optimization problem:

$$\hat{\mathbf{x}}^t_0 = \arg\min_{\mathbf{x}} \sum_{i=1}^{n} \|F_i(\mathbf{x}) - \hat{\mathbf{x}}_{\theta_{\text{ref}}}(F_i(\mathbf{z}_t), \mathbf{c})\|_2^2 \tag{5.3}$$

This optimization problem aims to reconcile all of the denoised samples $\hat{\mathbf{x}}_{\theta_{\text{ref}}}(F_i(\mathbf{z}_t), \mathbf{c})$ obtained from different random crops. By minimizing the loss in Eq. 5.3, we ensure that denoised samples from different regions are combined as consistently as possible. Eq. 5.3 is a quadratic Least-Squares and has a closed-form solution: each pixel of the minimizer $\hat{\mathbf{x}}^t_0$ is an average of all denoised sample updates.

Intuitively, we synthesize a high-resolution texture in a score aggregation manner. At every time step t. we first divide the large noisy latent map $\mathbf{z}_t$ into small patches and denoise each patch using the fine-tuned diffusion model. These denoised patches are then aggregated by averaging. We found that using random crops achieves the same image quality as using fixed crops [3], while significantly speeding up the inference time, by a factor of 10. By denoising only small patches, we also overcome memory constraints. Furthermore, our algorithm scales well with resolution since the patch size remains fixed. This allows our method to generate arbitrarily large high-quality textures based on the reference texture image.

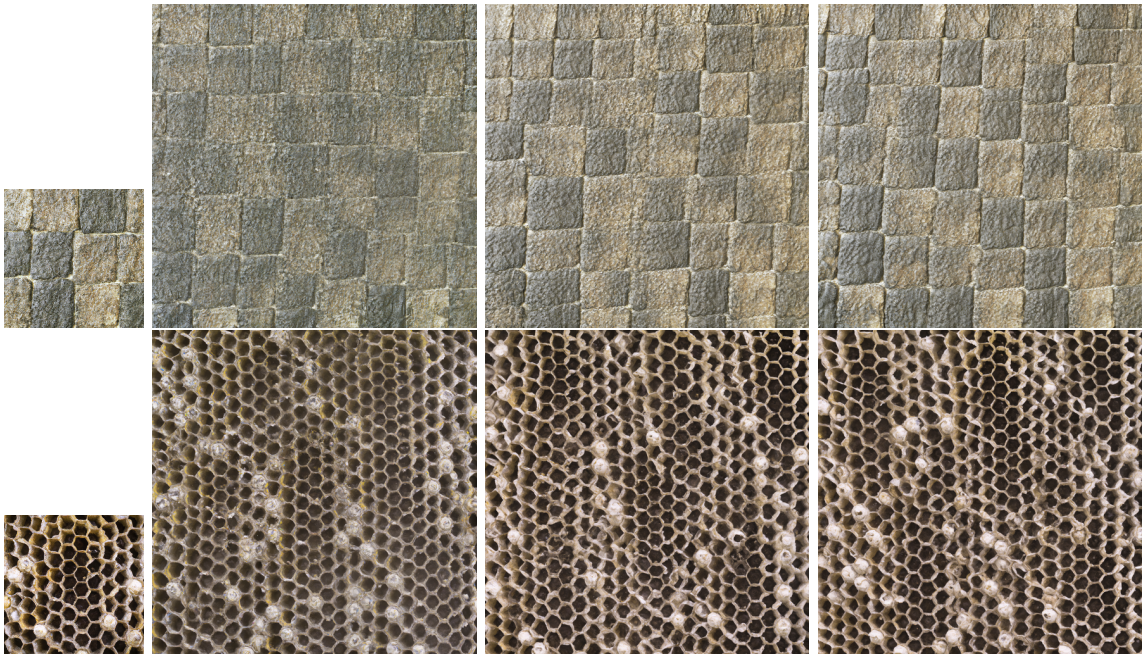


Figure 5.3: **Ablation studies.** Fixing the text encoder introduces color drift due to inherited semantic priors. Conversely, using fixed crops at test time maintains the same image quality but extends runtime from 6 to 50 minutes versus random crops.

5.3 Evaluation

In this section, we perform ablation studies to evaluate the importance of each model component, and compare Infinite Texture against both traditional and deep learning-based state-of-the-art in the tasks of texture synthesis.

5.3.1 Ablation Studies

We perform qualitative ablation on Infinite Texture to evaluate the importance of tuning a text encoder and the impact of using random (vs. fixed) crops.

Fixed vs. trainable text encoder. We conducted a comparison between fine-tuning diffusion models with a fixed text encoder and a trainable text encoder. Fixing the text encoder

helps maintain semantic priors, but it also introduces a false signal to the image synthesis module through the unique identifier. As illustrated in Fig. 5.3, fine-tuning the diffusion model while also training the text encoder preserves the most image priors and results in reduced color drift.

Fixed vs. random crops. At inference time, we employ a set of random crops F_i to denoise the random patches in the large latent map. In contrast, the original MultiDiffusion [3] utilizes a set of fixed crops F_i. As shown in Fig. 5.3, using random crops achieves comparable image quality while significantly improving the runtime by a factor of 10.

5.3.2 Baseline Comparisons

We conducted qualitative comparisons of Infinite Texture with a variety of texture synthesis baselines, covering both traditional [34, 73] and deep learning-based methods [5, 195]. The field of quantitative comparisons in texture synthesis continues to be an active area of research [95]. Standard quantitative metrics, *e.g.*, Gram matrix computation or FID, are not optimal choices for evaluation as they are used as objective functions by certain methods. Instead, we conducted a human study to quantitatively assess the realism of synthesized textures.

Image Quilting. As an early texture synthesis method, Image Quilting [34] is a patch-based approach. It involves tiling an example patch into a grid and utilizing dynamic programming to determine an optimal path for cutting through the overlapping regions, and ideally results in a seamless tile composed of the same patch. However, when applied to high-resolution textures, the complexity of solving the dynamic programming increases, leading to the generation of repetitive patterns. As shown in Fig. 5.2, image quilting tends to produce noticeable repetitive patterns when working with real-world high-resolution texture patches.

Self Tuning Texture Optimization. While non-parametric method [34] struggled with real-world high-resolution textures, Self Tuning Texture Optimization (STTO) [73] is a fully automatic self-tuning texture sytnthesis method that extends Texture Optimization [79, 26] to handle textures with large-scale structures, repetitions, and near-regular

structures. STTO is capable of self-tuning its various parameters, thereby eliminating the need for manual adjustment of parameters on a case-by-case basis. As shown in Fig. 5.2, STTO tends to produce results with small broken structures. It also fails with textures that contain large but unpronounced features (*e.g.* fabric and wood in Fig. 5.2), due to the contour detector's inability to detect reliable edges.

Periodic Spatial GAN. Periodic Spatial GAN (PSGAN) [5] is a texture synthesis method based on Generative Adversarial Networks (GANs). PSGAN extends the DCGAN [125] network structure by incorporating a spatially periodic signal generator to learn the statistical properties of the given texture image. As illustrated in Fig. 5.2, PSGAN fails to capture the frequency of the texture image. This limitation arises from the periodic signal generator's preference for high-frequency textures and the constrained receptive field size.

Non-Stationary Texture Synthesis. Non-stationary texture synthesis (NSTS) [195] is the state-of-the-art method for texture synthesis. It trains a GAN per input texture image to double the spatial extent of texture blocks extracted from the input exemplar texture. Once trained, the fully convolutional generator is capable of expanding the size of the input texture image. The network structure is based on CycleGAN [196]. As shown in Fig. 5.2, NSTS generates visually pleasing textures but fails to accurately follow the distribution of the input texture. This limitation stems from the network being trained solely on small patches (256×256) of the example texture, resulting in overfitting to low-frequency details.

Human study. To quantify the realism of synthesized textures, we conducted a human study via Amazon Mechanical Turk (AMT). This study involved a set of 39 exemplar textures. For each exemplar texture, we presented five synthesized textures from various methods to three human subjects, and asked them to choose the one that most closely resembles the exemplar texture. We also provided instructions to specifically note artifacts such as visible seams, noticeable repetition patterns, color drifts, loss of sharpness, irregular patterns, and inconsistent global structures. Our method was chosen as the best 45% of the time, compared to NSTS at 22%, Image Quilting at 15%, STTO at 13%, and PSGAN at 5%, demonstrating a clear advantage in the task of texture synthesis.

Figure 5.4: **Results of synthesized textures and renderings**. We demonstrate high-resolution textures generated by Infinite Texture, along with photo-realistic renderings of an armchair utilizing the textures (including the wood armrest). Infinite Texture is capable of generating textures with different variations when provided with the same text prompt. These textures can be seamlessly incorporated into 3D rendering pipelines, resulting in nearly infinite material choices for assets in a 3D shape collection. We use the default UV mapping from the CAD model to warp the texture onto the 3D model.

5.4 Applications

In this section, we present more results of Infinite Texture, and its applications in synthetic rendering (Sec. 5.4.1) and retexturing natural images (Sec. 5.4.2).

5.4.1 Texture Generation and Application to 3D Models

We showcase sample textures generated by Infinite Texture in Fig. 5.4. Infinite Texture takes in the text prompt, and leverages DALL-E 2 [127] to synthesize a reference texture image of size 1024×1024. We then fine-tune a Stable Diffusion v2 [135] checkpoint for 1000 iterations, which takes 10 minutes on a single NVIDIA A100 GPU. At inference time, it takes 6 minutes to generate a texture of size 2304×2304 or 1.5 hours for a 85 MP texture in Fig 4.2. As shown in Fig. 5.4, Infinite Texture is able to synthesize textures

of different frequencies, ranging from coarse fabric to dense leather grain. Moreover, it is capable of generating multiple distinct textures from the same text prompt, offering users a range of texture variations. The generated texture images have direct applications in 3D rendering pipelines. As shown in Fig. 5.4, these textures are applied to the same armchair, demonstrating Infinite Texture's ability to provide high-quality, realistic appearance models for large-scale 3D shape collections.

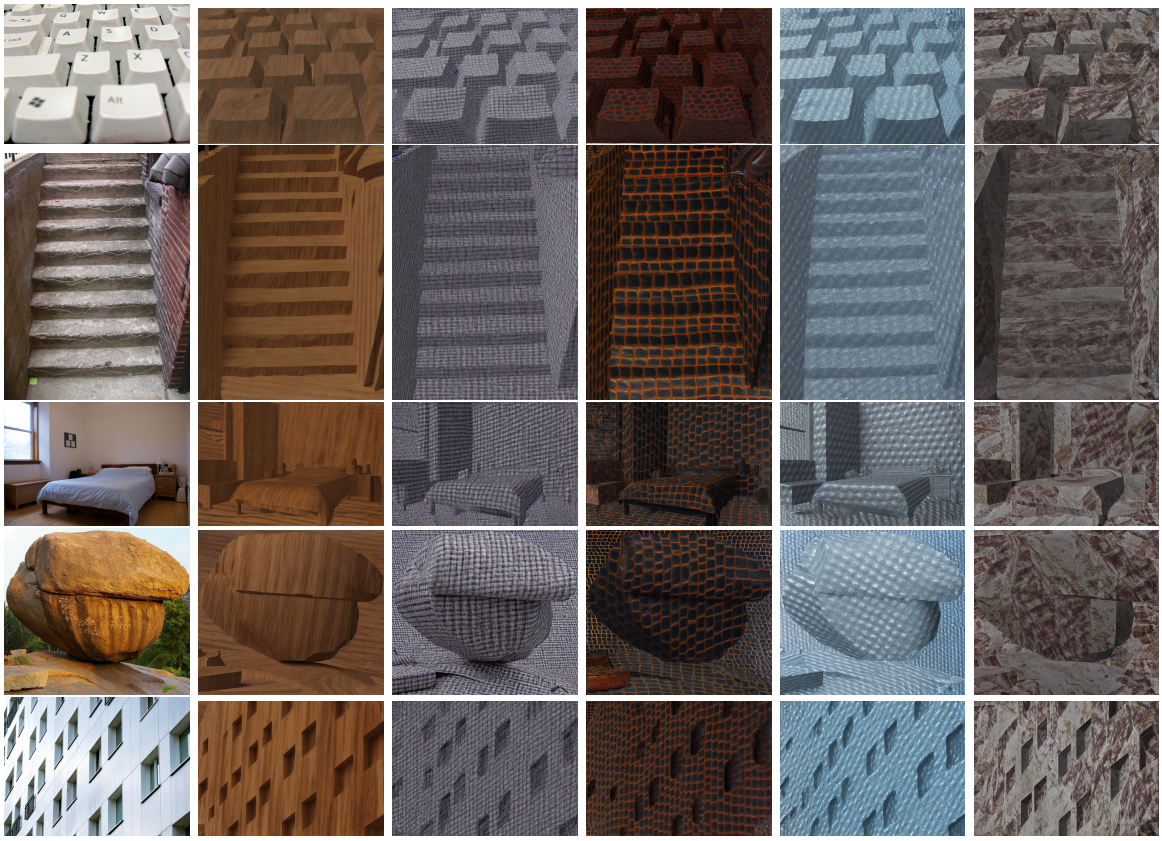
5.4.2 Texture Transfer

Texture transfer is a process where a given texture is applied to another image, guided by various properties of the latter. Early work [34, 53] perform texture transfer based on the brightness of the target image. Deep learning based methods [43, 67, 196] achieved texture transfer by aligning statistics of feature maps in the network via a style loss. In our approach, we formulate texture transfer as an image synthesis task conditioned on scene depth.

We leverage ControlNet [188] to transfer our generated textures to new surfaces. To achieve this, we first employ MiDaS [129] to estimate the depth map from the input image. The depth map is then used as the conditional signal to model the texture's application to the surface. Following Zhang *et al*. [188], we incorporate the conditional depth map into a pretrained diffusion model using ControlNet. ControlNet takes a text prompt and an estimated depth map of the input image as inputs, and transfers the example texture onto the input image.

The ControlNet is fine-tuned on a minimal dataset constructed from texture-mapped primitives, and can generalize to nature images. To construct the minimal dataset, we use the generated texture of size 2304×2304 from Infinite Texture to synthetically render a collection of image/depth pairs. For consistently shaded results, these images are rendered with the same texture applied to surfaces at various orientations, all under globally consistent shading. The ControlNet is built upon a Stable Diffusion checkpoint. We use the same loss as in Eq. 5.1 to train both the ControlNet and the decoder part of the Stable Diffusion.

Fig. 5.5 showcases a collection of our texture transfer results. Our method successfully



(a) Input (b) *Wood* (c) *Fabric* (d) *Alligator skin* (e) *Rubber floor mat* (f) *Granite*

Figure 5.5: **Results of texture transfer.** We present an application of texture transfer facilitated by Infinite Texture. We leverage ControlNet [188] to transfer our generated textures to new surfaces. To achieve this, we first used MiDaS [129] to estimate the depth map from the input image. The depth map then served as the conditioning signal to control the texture's application to the surface observed in the real image. Re-textured results preserve the statistics of the input texture, while sharing consistent shading and shape with the input image.

transfers the appearance of the example texture onto various surfaces. We present results using the same texture applied across the entire scene to demonstrate that synthesized images have consistent shading and shape with the input image. For future work, a segmentation model could be added to apply different textures on different regions of the scene.

5.5 Conclusion and Discussions

In this chapter, I presented Infinite Texture, a method that can generate an infinite number of arbitrarily large, high-quality textures that operates purely from a text prompt. This is accomplished through the fine-tuning of a diffusion model and a smart use of the diffusion model at inference time. Our diverse range of results demonstrates that the method is capable of synthesizing high-resolution, high-quality textures guided by text prompts. While training a diffusion model per prompt may appear cumbersome, it remains the fastest high-resolution texture generator to our knowledge. The trained diffusion model is stable enough for real-world application, enabling photo-realistic renderings for 3D shape collections. We have further showcased an application by utilizing synthesized textures for retexturing any given image. We believe that this work represents significant progress towards the goal of generating high-quality graphics assets from natural language descriptions.

Chapter 6

CONCLUSIONS AND FUTURE WORK

In this thesis, I introduced several methods to enhance photorealistic image synthesis, aiming to create a highly realistic and immersive experience from novel viewpoints, novel scene configurations, and novel illumination. I investigated scene-level methods learning from static video or image collection capture, as well as object-level methods that leverage a novel capture setup. Furthermore, I showcased using a minimal dataset to produce realistic lighting in diffusion models. Below I summarize each of my contributions:

- **Realistic Image Composition from a Static Video:** In Chapter 3.2, I introduced *People as Scene Probes*, a fully automatic pipeline for inferring depth, occlusion, and lighting/shadow information from image sequences of a scene. The central contributions of this work are recognizing that so much information can be extracted just by using people (and other objects such as cars) as scene probes to passively scan the scene. I showed that the inferred depth, occlusion ordering, lighting, and shadows are plausible, with the occlusion layering and shadow casting methods outperforming single-image depth estimation and traditional pix2pix shadow synthesis baselines. I then showcased results using a tool for image compositing based on the synthesis pipeline.

- **Realistic Image Composition from image collections:** People as Scene Probes requires a long video capture to passively scan the scene. The learned scene properties do not generalize to other street scenes. In Chapter 3.3, I introduced *Repopulating Street Scenes*, a fully automatic pipeline for populating, depopulating, or repopulating street scenes. The pipeline consists of four major components: (1) a removal network that can remove selected objects along with their shadows; (2) a sun estimation network that predicts the sun position from an image; (3) a method to scale

and occlude inserted objects properly; and (4) an insertion network that synthesizes shadows for inserted objects. These components are trained on short image bursts of street scenes, and can run on a single street image at test time. Further, I showcased multiple applications of our pipeline for depopulating and repopulating street scenes.

- **Photorealistic Portrait Reconstruction:** I presented *SunStage* in Chapter 4, a lightweight and practical facial capture, rendering, and editing system that can serve as a minimalist light stage. With a video of an individual rotating in-place under the sun, SunStage reconstructs a physical model of the subject and the scene lighting, which enables applications such as relighting the subject with realistic reflections and cast shadows. SunStage allows arbitrary lighting and reflectance control in the reconstructed physical space, which can be rendered to produce photo-realistic results. I demonstrated several applications such as editing skin reflectance, relighting, and view synthesis.

- **High Resolution Texture Synthesis and Transfer:** In Chapter 5, I presented *Infinite Texture*, a method for generating arbitrarily large texture images from a text prompt. This technique supports applications in 3D rendering and texture transfer, ensuring consistent shading and depth through the use of a minimal dataset. I demonstrated the effectiveness of this approach in generating high-resolution, high-quality textures that can be seamlessly integrated into various downstream tasks. This work represents significant progress towards the goal of generating high-quality graphics assets from natural language descriptions.

6.1 Future Work

The ultimate goal is to create a photorealistic world that is indistinguishable from reality, with fine user control and full 3D coverage. Recent advancements, particularly the rise of diffusion models, have shown great potential in achieving high-quality, realistic image synthesis. Diffusion models excel at generating high-quality images and are easy to extend, making them extremely popular among active users who have created an extensive collection

of diffusion models with various styles by fine-tuning base models such as Stable Diffusion. However, these models still face significant limitations, such as high computational costs, the need for large training datasets, and a lack of fine-grained control.

Despite these advancements, several open problems and challenges remain. Addressing these issues is crucial for further enhancing the realism and practicality of photorealistic image synthesis. Below, I outline some key areas for future research and development.

6.1.1 Geometry- and Lighting-Correct Image Synthesis

Both state-of-the-art GANs [70] and diffusion models [127] are renowned for generating images that are strikingly similar to real-world photographs and consistently fool people. However, a closer examination reveals fundamental inconsistencies, such as shadow misalignment and vanishing point inaccuracy [141]. Existing works [21, 6] have shown that generative models implicitly capture complex scene properties, including normals, depth, albedo, and support relations. This suggests that generative models "understand" geometry and lighting, which is the basis of 3D rendering. However, they cannot fully translate this "understanding" into accurate geometry and lighting in output images.

There are two potential ways to overcome these limitations. One approach is to fine-tune the model on a minimal dataset, as demonstrated in Infinite Texture [168], to ensure consistent lighting and geometry. Winter *et al.* [175] have investigated this aspect by collecting a small counterfactual dataset and fine-tuning diffusion models for image composition. The fine-tuned model can realistically model the effects of objects on the scene, *e.g.* occlusions, shadows, and reflections. The other approach is to explicitly model geometry and lighting in diffusion models. One way to do this is by using intermediate results, such as normals, depth, and albedo, as conditional signals, and modeling image synthesis as an image translation problem. Some of these ideas are now being explored, such as in Ding *et al.* [33] and Zhang *et al.* [188].

6.1.2 Personalized Models

Existing image synthesis methods for human faces often rely on zero-shot learning, where neural networks are trained on a large-scale dataset of various identities and then tested on a new identity. These methods typically ignore the fact that generic facial priors often fail to capture the high-frequency facial characteristics specific to the test identity [169]. As a result, the synthesized images may lack the detailed features that make each face unique. Additionally, multiple photos of the same person are often readily available in personal photo albums, which could be utilized to improve the synthesis process.

One future direction for enhancing image synthesis is in developing personalized models. These models would leverage the available photos of an individual to better capture their unique facial features and enhance the realism of the generated images. By using these personal photo collections, personalized models can be fine-tuned to reflect the specific characteristics of the target individual, thus overcoming the limitations of zero-shot learning.

Personalized models would not only improve the accuracy of facial synthesis but also have applications in various fields such as virtual reality, personalized avatars, and custom-fit digital identities. Developing these models involves creating algorithms that can efficiently and effectively use a limited number of personal photos to train the neural networks, ensuring that the synthesized images are both high-quality and true to the individual's appearance. This approach represents a significant step forward in the quest for truly photorealistic image synthesis. Some pioneering work in this direction includes Ding *et al.* [33] and Ruiz *et al.* [137].

6.1.3 3D Content Generation

While remarkable progress has been achieved in the field of 2D image generation due to advanced generative methods and large-scale image-text datasets, generating 3D shapes from a single image or text prompt remains a long-standing problem in computer vision. The success of 2D image generation is hardly transferable to the 3D domain due to the limited availability of 3D training data. Many works introduce sophisticated 3D generative models, but the majority rely solely on 3D shape datasets for training. Given the limited

size of publicly available 3D datasets, these methods often struggle to generalize across unseen categories in open-world scenarios.

Existing works, such as DreamFusion [120] and Magic3D [91], harness the expansive knowledge and robust generative potential of 2D prior models like CLIP and Stable Diffusion. These methods typically optimize a 3D representation (e.g., NeRF or mesh) from scratch for each input text or image. During the optimization process, the 3D representation is rendered into 2D images, and the 2D prior models are employed to calculate gradients for them. While these methods have yielded impressive outcomes, the 3D representations they use do not disentangle material and lighting, which limits the method's application. For example, the reconstructed 3D shapes cannot be easily inserted into other virtual environments, a crucial aspect in creating a photorealistic world. Additionally, the per-shape optimization can be exceedingly time-intensive, requiring tens of minutes or even hours to generate a single 3D shape for each input.

To address these limitations, several potential solutions can be explored. One approach is to develop models that explicitly disentangle material properties and lighting during the 3D reconstruction process. By separating these elements, the resulting 3D shapes could be more easily integrated into various virtual environments, maintaining a consistent appearance under different lighting conditions. Synthetic datasets with ground truth disentanglement, such as Objaverse [29], can be used to bootstrap the training of such models. Additionally, leveraging pre-trained neural networks that can infer material properties and lighting to crowdsource data from internet images can help gather more training data and improve generalization. These advancements could collectively enhance the practicality and applicability of 3D content generation, paving the way towards a truly photorealistic world.

BIBLIOGRAPHY

[1] Oleg Alexander, Mike Rogers, William Lambeth, Jen-Yuan Chiang, Wan-Chun Ma, Chuan-Chang Wang, and Paul Debevec. The Digital Emily Project: Achieving a Photorealistic Digital Actor. *IEEE Computer Graphics and Applications*, 30(4):20–31, 2010. 5, 56

[2] Samaneh Azadi, Deepak Pathak, Sayna Ebrahimi, and Trevor Darrell. Compositional gan: Learning conditional image composition. *arXiv preprint arXiv:1807.07560*, 2018. 26, 28

[3] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 1737–1752. PMLR, 2023. 81, 83

[4] Michael Baxandall. *Shadows and enlightenment*. Yale University Press, 1997. 12

[5] Urs Bergmann, Nikolay Jetchev, and Roland Vollgraf. Learning texture manifolds with the periodic spatial gan. *arXiv preprint arXiv:1705.06566*, 2017. 8, 76, 80, 83, 84

[6] Anand Bhattad, Daniel McKee, Derek Hoiem, and David Forsyth. Stylegan knows normal, depth, albedo, and more. *Advances in Neural Information Processing Systems*, 36, 2024. 91

[7] 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. (Proc. SIGGRAPH)*, 40(4), 2021. 58

[8] Dmitri Bitouk, Neeraj Kumar, Samreen Dhillon, Peter Belhumeur, and Shree K Nayar. Face swapping: automatically replacing faces in photographs. In *ACM SIGGRAPH 2008 papers*, pages 1–8. 2008. 27

[9] Volker Blanz, Kristina Scherbaum, Thomas Vetter, and Hans-Peter Seidel. Exchanging faces in images. In *Computer Graphics Forum*, volume 23, pages 669–676. Wiley Online Library, 2004. 27

[10] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999. 58

[11] James F Blinn. Models of light reflection for computer synthesized pictures. In *Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, pages 192–198, 1977. 61

[12] Timo Bolkart and Stefanie Wuhrer. A groupwise multilinear correspondence optimization for 3d faces. In *Proceedings of the IEEE international conference on computer vision*, pages 3604–3612, 2015. 58

[13] James Booth, Anastasios Roussos, Stefanos Zafeiriou, Allan Ponniah, and David Dunaway. A 3d morphable model learnt from 10,000 faces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5543–5552, 2016. 6, 56, 58

[14] Biswajit Bose and Eric Grimson. Ground plane rectification by tracking moving objects. In *Proceedings of the Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2003. 34

[15] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 25

[16] Gabriel J Brostow and Irfan A Essa. Motion based decompositing of video. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 8–13. IEEE, 1999. 27, 29

[17] Alan Brunton, Timo Bolkart, and Stefanie Wuhrer. Multilinear wavelets: A statistical shape space for human faces. In *European Conference on Computer Vision*, pages 297–312. Springer, 2014. 58

[18] Dan A Calian, Jean-François Lalonde, Paulo Gotardo, Tomas Simon, Iain Matthews, and Kenny Mitchell. From faces to outdoor light probes. In *Computer Graphics Forum*, volume 37, pages 51–61. Wiley Online Library, 2018. 28, 38, 59

[19] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, 2013. 58

[20] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 40

[21] Yida Chen, Fernanda Viégas, and Martin Wattenberg. Beyond surface statistics: Scene representations in a latent diffusion model. *arXiv preprint arXiv:2306.05720*, 2023. 91

[22] Yung-Yu Chuang, Dan B Goldman, Brian Curless, David H. Salesin, and Richard Szeliski. Shadow matting and compositing. *ACM Transactions on Graphics*, 22(3):494–500, July 2003. Sepcial Issue of the SIGGRAPH 2003 Proceedings. 26, 28, 29

[23] Antonio Criminisi, Ian Reid, and Andrew Zisserman. Single view metrology. *International Journal of Computer Vision*, 40(2):123–148, 2000. 34

[24] Leonardo Da Vinci. *The notebooks of Leonardo da Vinci*, volume 1. Lulu. com, 1971. 13

[25] Hang Dai, Nick Pears, William AP Smith, and Christian Duncan. A 3d morphable model of craniofacial shape and texture variation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3085–3093, 2017. 58

[26] Soheil Darabi, Eli Shechtman, Connelly Barnes, Dan B Goldman, and Pradeep Sen. Image melding: Combining inconsistent images using patch-based synthesis. *ACM Transactions on graphics (TOG)*, 31(4):1–10, 2012. 83

[27] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Acm siggraph 2008 classes*, pages 1–10. 2008. 2, 28

[28] 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*, pages 145–156, 2000. 2, 5, 56, 58

[29] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 93

[30] Helisa Dhamo, Keisuke Tateno, Iro Laina, Nassir Navab, and Federico Tombari. Peeking behind objects: Layered depth prediction from a single image. *Pattern Recognition Letters*, 125:333–340, 2019. 27

[31] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 76

[32] Abdallah Dib, Cedric Thebault, Junghyun Ahn, Philippe-Henri Gosselin, Christian Theobalt, and Louis Chevallier. Towards high fidelity monocular face reconstruction with rich reflectance using self-supervised learning and ray tracing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12819–12829, 2021. 58, 66, 67, 69

[33] Zheng Ding, Xuaner Zhang, Zhihao Xia, Lars Jebe, Zhuowen Tu, and Xiuming Zhang. Diffusionrig: Learning personalized priors for facial appearance editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12736–12746, 2023. 91, 92

[34] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346, 2001. 7, 75, 80, 83, 86

[35] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1033–1038. IEEE, 1999. 7, 75

[36] Bernhard Egger, William AP Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, et al. 3d morphable face models—past, present, and future. *ACM Transactions on Graphics (TOG)*, 39(5):1–38, 2020. 58

[37] Yao Feng, Haiwen Feng, Michael J Black, and Timo Bolkart. Learning an animatable detailed 3d face model from in-the-wild images. *ACM Transactions on Graphics (TOG)*, 40(4):1–13, 2021. 58, 64, 66, 67, 69, 70

[38] Arturo Flores and Serge Belongie. Removing pedestrians from Google Street View images. In *Computer Vision and Pattern Recognition Workshops*, pages 53–58, 2010. 4, 27

[39] Andrea Frome, German Cheung, Ahmad Abdulkader, Marco Zennaro, Bo Wu, Alessandro Bissacco, Hartwig Adam, Hartmut Neven, and Luc Vincent. Large-scale privacy protection in google street view. In *2009 IEEE 12th international conference on computer vision*, pages 2373–2380. IEEE, 2009. 4

[40] Graham Fyffe and Paul Debevec. Single-shot reflectance measurement from polarized color gradient illumination. In *2015 IEEE International Conference on Computational Photography (ICCP)*, pages 1–10. IEEE, 2015. 58

[41] Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, and Jean-François Lalonde. Learning to predict indoor illumination from a single image. *arXiv preprint arXiv:1704.00090*, 2017. 28, 38

[42] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. *Advances in neural information processing systems*, 28, 2015. 8, 76

[43] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015. 86

[44] Stamatios Georgoulis, Konstantinos Rematas, Tobias Ritschel, Mario Fritz, Tinne Tuytelaars, and Luc Van Gool. What is around the camera? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5170–5178, 2017. 31

[45] Abhijeet Ghosh, Graham Fyffe, Borom Tunwattanapong, Jay Busch, Xueming Yu, and Paul Debevec. Multiview face capture using polarized spherical gradient illumination. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, pages 1–10, 2011. 58

[46] Ernst Hans Gombrich. *Shadows: the depiction of cast shadows in western art*. Yale University Press, 2014. 18

[47] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 25

[48] Richard L Gregory. *Eye and brain: The psychology of seeing*, volume 38. Princeton university press, 2015. 1, 13

[49] Ruiqi Guo, Qieyun Dai, and Derek Hoiem. Paired regions for shadow detection and removal. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2956–2967, 2012. 26

[50] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 30, 35

[51] Kaiming He and Jian Sun. Statistics of patch offsets for image completion. In *European Conference on Computer Vision*, pages 16–29. Springer, 2012. 37

[52] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 38

[53] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340, 2001. 86

[54] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 76, 78

[55] Burne Hogarth. *Dynamic Light and Shade.* ERIC, 1991. 17

[56] Derek Hoiem, Alexei A Efros, and Martial Hebert. Putting objects in perspective. *International Journal of Computer Vision*, 80:3–15, 2008. 34

[57] Yannick Hold-Geoffroy, Akshaya Athawale, and Jean-François Lalonde. Deep sky modeling for single image outdoor lighting estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6927–6935, 2019. 28, 31, 48

[58] Yannick Hold-Geoffroy, Kalyan Sunkavalli, Sunil Hadap, Emiliano Gambaretto, and Jean-François Lalonde. Deep outdoor illumination estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7312–7321, 2017. 28, 31, 38

[59] Yannick Hold-Geoffroy, Jinsong Zhang, Paulo F U Gotardo, and Jean-François Lalonde. x-hour outdoor photometric stereo. In *International Conference on 3D Vision*, 2015. 59

[60] Seunghoon Hong, Xinchen Yan, Thomas S Huang, and Honglak Lee. Learning hierarchical semantic image manipulation through structured representations. In *Advances in Neural Information Processing Systems*, pages 2708–2718, 2018. 26

[61] Andrew Hou, Michel Sarkis, Ning Bi, Yiying Tong, and Xiaoming Liu. Face relighting with geometrically consistent shadows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4217–4226, 2022. 59, 66, 67, 69

[62] Andrew Hou, Ze Zhang, Michel Sarkis, Ning Bi, Yiying Tong, and Xiaoming Liu. Towards high fidelity face relighting with realistic shadows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14719–14728, 2021. 59

[63] Xiaowei Hu, Yitong Jiang, Chi-Wing Fu, and Pheng-Ann Heng. Mask-shadowgan: Learning to remove shadows from unpaired data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2472–2481, 2019. 27

[64] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017. 42

[65] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 26, 32

[66] Nikolay Jetchev, Urs Bergmann, and Roland Vollgraf. Texture synthesis with spatial generative adversarial networks. *arXiv preprint arXiv:1611.08207*, 2016. 76

[67] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 35, 66, 86

[68] Jiyoung Jung, Joon-Young Lee, and In So Kweon. One-day outdoor photometric stereo via skylight estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4521–4529, 2015. 59

[69] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 58

[70] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 58, 91

[71] Kevin Karsch, Varsha Hedau, David Forsyth, and Derek Hoiem. Rendering synthetic objects into legacy photographs. *ACM Transactions on Graphics (TOG)*, 30(6):1–12, 2011. 28

[72] Kevin Karsch, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, Hailin Jin, Rafael Fonte, Michael Sittig, and David Forsyth. Automatic scene inference for 3d object compositing. *ACM Transactions on Graphics (TOG)*, 33(3):1–15, 2014. 28

[73] Alexandre Kaspar, Boris Neubert, Dani Lischinski, Mark Pauly, and Johannes Kopf. Self tuning texture optimization. In *Computer Graphics Forum*, volume 34, pages 349–359. Wiley Online Library, 2015. 75, 80, 83

[74] Daniel Kersten, David C Knill, Pascal Mamassian, and Isabelle Bülthoff. Illusory motion from shadows. 1996. 16

[75] Daniel Kersten, Pascal Mamassian, and David C Knill. Moving cast shadows induce apparent motion in depth. *Perception*, 26(2):171–192, 1997. 16, 17

[76] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 65

[77] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 25

[78] David C Knill, Pascal Mamassian, and Daniel Kersten. Geometry of shadows. *JOSA A*, 14(12):3216–3232, 1997. 13, 14, 16

[79] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: Image and video synthesis using graph cuts. *Acm transactions on graphics (tog)*, 22(3):277–286, 2003. 75, 83

[80] Jean-François Lalonde, Alexei A Efros, and Srinivasa G Narasimhan. Estimating natural illumination from a single outdoor image. In *2009 IEEE 12th International Conference on Computer Vision*, pages 183–190. IEEE, 2009. 28, 38

[81] Jean-François Lalonde, Derek Hoiem, Alexei A Efros, Carsten Rother, John Winn, and Antonio Criminisi. Photo clip art. *ACM transactions on graphics (TOG)*, 26(3):3–es, 2007. 26, 28, 52

[82] Katrin Lasinger, René Ranftl, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *arXiv preprint arXiv:1907.01341*, 2019. 42, 43, 44

[83] Hieu Le and Dimitris Samaras. Shadow removal via shadow image decomposition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8578–8587, 2019. 26, 27, 33

[84] Hieu Le, Tomas F Yago Vicente, Vu Nguyen, Minh Hoai, and Dimitris Samaras. A+ d net: Training a shadow detector with adversarial shadow attenuation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 662–678, 2018. 33

[85] Donghoon Lee, Sifei Liu, Jinwei Gu, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Context-aware synthesis and placement of object instances. In *Advances in Neural Information Processing Systems*, pages 10393–10403, 2018. 26

[86] Donghoon Lee, Tomas Pfister, and Ming-Hsuan Yang. Inserting videos into videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10061–10070, 2019. 26

[87] Chloe LeGendre, Wan-Chun Ma, Graham Fyffe, John Flynn, Laurent Charbonnel, Jay Busch, and Paul Debevec. Deeplight: Learning illumination for unconstrained mobile mixed reality. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5918–5928, 2019. 28, 38

[88] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph.*, 36(6):194–1, 2017. 6, 56, 58, 60, 64

[89] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Diversified texture synthesis with feed-forward networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3920–3928, 2017. 76

[90] Lin Liang, Ce Liu, Ying-Qing Xu, Baining Guo, and Heung-Yeung Shum. Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics (ToG)*, 20(3):127–150, 2001. 75

[91] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023. 93

[92] Chen-Hsuan Lin, Ersin Yumer, Oliver Wang, Eli Shechtman, and Simon Lucey. St-gan: Spatial transformer generative adversarial networks for image compositing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9455–9464, 2018. 26, 28

[93] Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta. Robust high-resolution video matting with temporal guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 238–247, 2022. 65

[94] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 76

[95] Wen-Chieh Lin, James Hays, Chenyu Wu, Yanxi Liu, and Vivek Kwatra. Quantitative evaluation of near regular texture synthesis algorithms. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 427–434. IEEE, 2006. 83

[96] Andrew Liu, Shiry Ginosar, Tinghui Zhou, Alexei A Efros, and Noah Snavely. Learning to factorize and relight a city. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 544–561. Springer, 2020. 28, 59

[97] Daquan Liu, Chengjiang Long, Hongpan Zhang, Hanning Yu, Xinzhi Dong, and Chunxia Xiao. Arshadowgan: Shadow generative adversarial network for augmented

reality in single light scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 28

[98] Guilin Liu, Kevin J Shih, Ting-Chun Wang, Fitsum A Reda, Karan Sapra, Zhiding Yu, Andrew Tao, and Bryan Catanzaro. Partial convolution based padding. *arXiv preprint arXiv:1811.11718*, 2018. 27, 37

[99] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *Advances in Neural Information Processing Systems*, pages 9605–9616, 2018. 33

[100] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *arXiv preprint arXiv:2206.00927*, 2022. 77

[101] Wei-Chiu Ma, Shenlong Wang, Marcus A Brubaker, Sanja Fidler, and Raquel Urtasun. Find your way by observing the sun and other semantic cues. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6292–6299. IEEE, 2017. 28, 48

[102] Elman Mansimov, Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Generating images from captions with attention. *arXiv preprint arXiv:1511.02793*, 2015. 76

[103] Abhimitra Meka, Christian Haene, Rohit Pandey, Michael Zollhöfer, Sean Fanello, Graham Fyffe, Adarsh Kowdle, Xueming Yu, Jay Busch, Jason Dourgarian, et al. Deep reflectance fields: high-quality facial reflectance field inference from color gradient illumination. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 58

[104] Chenlin Meng, Ruiqi Gao, Diederik P Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. *arXiv preprint arXiv:2210.03142*, 2022. 77

[105] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? *arXiv preprint arXiv:1801.04406*, 2018. 25

[106] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 25

[107] Takeru Miyato and Masanori Koyama. cgans with projection discriminator. *arXiv preprint arXiv:1802.05637*, 2018. 25

[108] Alexander Mordvintsev and Eyvind Niklasson. μnca: Texture generation with ultra-compact neural cellular automata. *arXiv preprint arXiv:2111.13545*, 2021. 76

[109] Thomas Nestmeyer, Jean-François Lalonde, Iain Matthews, and Andreas Lehrmann. Learning physics-guided face relighting under directional light. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5124–5133, 2020. 59

[110] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 76

[111] Eyvind Niklasson, Alexander Mordvintsev, Ettore Randazzo, and Michael Levin. Self-organising textures. *Distill*, 6(2):e00027–003, 2021. 76

[112] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org, 2017. 25

[113] Xi Ouyang, Yu Cheng, Yifan Jiang, Chun-Liang Li, and Pan Zhou. Pedestrian-synthesis-gan: Generating pedestrian data in real scene and beyond. *arXiv preprint arXiv:1804.02047*, 2018. 26

[114] 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. *ACM Transactions on Graphics (TOG)*, 40(4):1–21, 2021. 6, 56, 59, 66, 68, 69

[115] Jeong Joon Park, Aleksander Holynski, and Steve Seitz. Seeing the world in a bag of chips. *arXiv preprint arXiv:2001.04642*, 2020. 31

[116] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019. 26

[117] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *ACM SIGGRAPH 2003 Papers*, pages 313–318. 2003. 28

[118] Frédéric Pighin, Jamie Hecker, Dani Lischinski, Richard Szeliski, and David H Salesin. Synthesizing realistic facial expressions from photographs. In *ACM SIGGRAPH 2006 Courses*, pages 19–es. 2006. 58

[119] Stylianos Ploumpis, Haoyang Wang, Nick Pears, William AP Smith, and Stefanos Zafeiriou. Combining 3d morphable models: A large scale face-and-head model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10934–10943, 2019. 58

[120] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 93

[121] Thomas Porter and Tom Duff. Compositing digital images. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 253–259, 1984. 26

[122] Emil Praun, Adam Finkelstein, and Hugues Hoppe. Lapped textures. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 465–470, 2000. 75

[123] Liangqiong Qu, Jiandong Tian, Shengfeng He, Yandong Tang, and Rynson WH Lau. Deshadownet: A multi-context embedding deep network for shadow removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4067–4075, 2017. 27

[124] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 76

[125] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 84

[126] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 76

[127] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 76, 78, 79, 80, 85, 91

[128] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 40, 42

[129] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022. 86, 87

[130] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. Generating 3d faces using convolutional mesh autoencoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 704–720, 2018. 58

[131] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*, 2020. 66

[132] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International conference on machine learning*, pages 1060–1069. PMLR, 2016. 76

[133] Erik Reinhard. Parameter estimation for photographic tone reproduction. *Journal of graphics tools*, 7(1):45–51, 2002. 62

[134] Yurui Ren, Xiaoming Yu, Ruonan Zhang, Thomas H Li, Shan Liu, and Ge Li. Structureflow: Image inpainting via structure-aware appearance flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 181–190, 2019. 37

[135] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 77, 79, 85

[136] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023. 79

[137] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Wei Wei, Tingbo Hou, Yael Pritch, Neal Wadhwa, Michael Rubinstein, and Kfir Aberman. Hyperdreambooth: Hypernetworks for fast personalization of text-to-image models. *arXiv preprint arXiv:2307.06949*, 2023. 92

[138] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 397–403, 2013. 65

[139] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 76, 78

[140] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 77

[141] Ayush Sarkar, Hanlin Mai, Amitabh Mahapatra, Svetlana Lazebnik, David A Forsyth, and Anand Bhattad. Shadows don't lie and lines can't bend! generative models don't know projective geometry... for now. *arXiv preprint arXiv:2311.17138*, 2023. 91

[142] Soumyadip Sengupta, Brian Curless, Ira Kemelmacher-Shlizerman, and Steven M Seitz. A light stage on every desk. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2420–2429, 2021. 59

[143] Soumyadip Sengupta, Jinwei Gu, Kihwan Kim, Guilin Liu, David W Jacobs, and Jan Kautz. Neural inverse rendering of an indoor scene from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8598–8607, 2019. 28, 38

[144] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D Castillo, and David W Jacobs. Sfsnet: Learning shape, reflectance and illuminance of faces in the wild'. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6296–6305, 2018. 6, 56

[145] Artem Sevastopolsky, Savva Ignatiev, Gonzalo Ferrer, Evgeny Burnaev, and Victor Lempitsky. Relightable 3d head portraits from a smartphone video. *arXiv preprint arXiv:2012.09963*, 2020. 59

[146] Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. Layered depth images. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 231–242, 1998. 27

[147] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 77, 78

[148] Roy Sorensen. *Seeing dark things: The philosophy of shadows*. Oxford University Press, 2008. 10

[149] Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the boundaries of view extrapolation with multiplane images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 175–184, 2019. 27

[150] 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 Transactions on Graphics (TOG)*, 38(4):79–1, 2019. 6, 56, 59

[151] Tiancheng Sun, Zexiang Xu, Xiuming Zhang, Sean Fanello, Christoph Rhemann, Paul Debevec, Yun-Ta Tsai, Jonathan T Barron, and Ravi Ramamoorthi. Light

stage super-resolution: Continuous high-frequency relighting. *ACM Transactions on Graphics (TOG)*, 39(6):1–12, 2020. 59

[152] Kalyan Sunkavalli, Wojciech Matusik, Hanspeter Pfister, and Szymon Rusinkiewicz. Factored time-lapse video. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 26(3), August 2007. 59

[153] Ming Tao, Hao Tang, Fei Wu, Xiao-Yuan Jing, Bing-Kun Bao, and Changsheng Xu. Df-gan: A simple and effective baseline for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16515–16525, 2022. 76

[154] Luan Tran and Xiaoming Liu. Nonlinear 3d face morphable model. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7346–7355, 2018. 58

[155] Shubham Tulsiani, Richard Tucker, and Noah Snavely. Layer-structured 3d scene inference via view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 302–317, 2018. 27

[156] Greg Turk. Generating textures on arbitrary surfaces using reaction-diffusion. *Acm Siggraph Computer Graphics*, 25(4):289–298, 1991. 75

[157] Greg Turk. Texture synthesis on surfaces. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 347–354, 2001. 7

[158] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. *arXiv preprint arXiv:1603.03417*, 2016. 76

[159] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6924–6932, 2017. 76

[160] Dor Verbin, Steven J. Gortler, and Todd Zickler. Unique geometry and texture from corresponding image patches. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4519–4522, 2021. 76

[161] Dor Verbin and Todd Zickler. Toward a universal model for shape from texture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 76

[162] Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 195–206, 2007. 61

[163] Jifeng Wang, Xiang Li, and Jian Yang. Stacked conditional generative adversarial networks for jointly learning shadow detection and shadow removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1788–1797, 2018. 27

[164] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 65

[165] Tengfei Wang, Ting Zhang, Bo Zhang, Hao Ouyang, Dong Chen, Qifeng Chen, and Fang Wen. Pretraining is all you need for image-to-image translation. *arXiv preprint arXiv:2205.12952*, 2022. 79

[166] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018. 26, 32, 33, 35, 44, 45

[167] Yifan Wang, Brian L Curless, and Steven M Seitz. People as scene probes. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 438–454. Springer, 2020. 5, 9, 25, 28, 40, 41, 42, 50

[168] Yifan Wang, Aleksander Holynski, Brian L. Curless, and Steven M. Seitz. Infinite texture: Text-guided high resolution diffusion texture synthesis. *arXiv preprint arXiv:2405.08210*, 2024. 8, 9, 75, 91

[169] Yifan Wang, Aleksander Holynski, Xiuming Zhang, and Xuaner Zhang. Sunstage: Portrait reconstruction and relighting using the sun as a light stage. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20792–20802, 2023. 6, 9, 92

[170] Yifan Wang, Andrew Liu, Richard Tucker, Jiajun Wu, Brian L Curless, Steven M Seitz, and Noah Snavely. Repopulating street scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5110–5119, 2021. 5, 9, 25

[171] Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 479–488, 2000. 7, 75

[172] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. *Caltech-UCSD Birds 200*. Oct 2011. 76

[173] Andreas Wenger, Andrew Gardner, Chris Tchou, Jonas Unger, Tim Hawkins, and Paul Debevec. Performance relighting and reflectance transformation with time-multiplexed illumination. *ACM Transactions on Graphics (TOG)*, 24(3):756–764, 2005. 6, 56

[174] Tim Weyrich, Wojciech Matusik, Hanspeter Pfister, Bernd Bickel, Craig Donner, Chien Tu, Janet McAndless, Jinho Lee, Addy Ngan, Henrik Wann Jensen, et al. Analysis of human faces using a measurement-based skin reflectance model. *ACM Transactions on Graphics (ToG)*, 25(3):1013–1024, 2006. 61

[175] Daniel Winter, Matan Cohen, Shlomi Fruchter, Yael Pritch, Alex Rav-Acha, and Yedid Hoshen. Objectdrop: Bootstrapping counterfactuals for photorealistic object removal and insertion. *arXiv preprint arXiv:2403.18818*, 2024. 91

[176] Andrew Witkin and Michael Kass. Reaction-diffusion textures. In *Proceedings of the 18th annual conference on computer graphics and interactive techniques*, pages 299–308, 1991. 75

[177] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 36, 38, 51

[178] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324, 2018. 76

[179] Yu-Ying Yeh, Koki Nagano, Sameh Khamis, Jan Kautz, Ming-Yu Liu, and Ting-Chun Wang. Learning to relight portrait images via a virtual light stage and synthetic-to-real adaptation. *ACM Transactions on Graphics (TOG)*, 2022. 59

[180] Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, and Zhan Xu. Contextual residual aggregation for ultra high-resolution image inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7508–7517, 2020. 27, 37, 47, 49

[181] Albert Yonas, Lynn T Goldsmith, and Janet L Hallstrom. Development of sensitivity to information provided by cast shadows in pictures. *Perception*, 7(3):333–341, 1978. 16, 17

[182] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018. 27, 37

[183] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4471–4480, 2019. 27, 37

[184] Fangneng Zhan, Hongyuan Zhu, and Shijian Lu. Spatial fusion gan for image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3653–3662, 2019. 26, 28

[185] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915, 2017. 76

[186] Ling Zhang, Qing Zhang, and Chunxia Xiao. Shadow remover: Image shadow removal based on illumination recovering optimization. *IEEE Transactions on Image Processing*, 24(11):4623–4636, 2015. 26

[187] Longwen Zhang, Qixuan Zhang, Minye Wu, Jingyi Yu, and Lan Xu. Neural video portrait relighting in real-time via consistency modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 802–812, 2021. 59, 68

[188] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 86, 87, 91

[189] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 49, 50

[190] Xiuming Zhang, Sean Fanello, Yun-Ta Tsai, Tiancheng Sun, Tianfan Xue, Rohit Pandey, Sergio Orts-Escolano, Philip Davidson, Christoph Rhemann, Paul Debevec, et al. Neural light transport for relighting and view synthesis. *ACM Transactions on Graphics (TOG)*, 40(1):1–17, 2021. 66, 67, 69

[191] Xuaner Zhang, Jonathan T Barron, Yun-Ta Tsai, Rohit Pandey, Xiuming Zhang, Ren Ng, and David E Jacobs. Portrait shadow manipulation. *ACM Transactions on Graphics (TOG)*, 39(4):78–1, 2020. 6, 56, 59

[192] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 49

[193] Hao Zhou, Sunil Hadap, Kalyan Sunkavalli, and David W Jacobs. Deep single-image portrait relighting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7194–7202, 2019. 6, 56, 66, 68, 69

[194] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 27

[195] Yang Zhou, Zhen Zhu, Xiang Bai, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Non-stationary texture synthesis by adversarial expansion. *arXiv preprint arXiv:1805.04487*, 2018. 8, 76, 79, 80, 83, 84

[196] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 84, 86

[197] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5802–5810, 2019. 76

[198] Rui Zhu, Xingyi Yang, Yannick Hold-Geoffroy, Federico Perazzi, Jonathan Eisenmann, Kalyan Sunkavalli, and Manmohan Chandraker. Single view metrology in the wild. *arXiv preprint arXiv:2007.09529*, 2020. 39