Summer 7-2015

3D Manipulation of Objects in Photographs

Natasha Kholgade Banerjee
Carnegie Mellon University

Follow this and additional works at: http://repository.cmu.edu/dissertations

Recommended Citation
3D Manipulation of Objects in Photographs

Natasha Kholgade Banerjee

CMU-RI-TR-15-19

Submitted in partial fulfilment of the requirements of the degree of Doctor of Philosophy in Robotics

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

July 2015

Doctoral Committee:

Yaser Sheikh, Carnegie Mellon University (Chair)
Nancy Pollard, Carnegie Mellon University
Kayvon Fatahalian, Carnegie Mellon University
Alexei Efros, University of California, Berkeley
David Forsyth, University of Illinois at Urbana-Champaign

Copyright © 2015 by Natasha Kholgade Banerjee. All rights reserved.
ABSTRACT

This thesis describes a system that allows users to perform full three-dimensional manipulations to objects in photographs. Cameras and photo-editing tools have contributed to the explosion in creative content by democratizing the process of creating visual realizations of users’ imaginations. However, shooting photographs using a camera is constrained by real-world physics, while existing photo-editing software is largely restricted to the 2D plane of the image. 3D object edits, intuitive to humans, are simply not possible in photo-editing software. The fundamental challenge in providing 3D object manipulation is that estimating the 3D structure of the object, including the geometry and appearance of object parts hidden from the viewpoint of the camera is ill-posed. 3D object manipulations reveal hidden parts of objects that were not previously seen from the viewpoint of the camera.

The key contributions of this thesis are algorithms that leverage 3D models from public repositories to obtain a three-dimensional representation of objects in photographs for 3D manipulation with seamless transition in appearance of the object from the original photograph. 3D models of objects in online repositories cannot be directly used to manipulate photographed objects, as they show mismatches in geometry and appearance, and do not contain three-dimensional illumination representing the scene where the photograph was captured. The work in this thesis provides a system that aligns the 3D model geometry, estimates three-dimensional illumination, and completes the appearance over the object in three dimensions to provide full 3D manipulation. To correct the mismatch between the geometry of the 3D model and the photographed object, the thesis presents an automatic model alignment technique that performs an exhaustive search in the space of viewpoint, object location, scale, and non-rigid deformation. We also provide a manual geometry adjustment tool that allows users to perform final corrections while imposing smoothness and symmetry constraints. Given the matched geometry, we present an illumination estimation approach that uses the visible pixels to obtain three-dimensional environment illumination that produces plausible effects such as cast shadows and smooth surface shading. Our appearance completion approach relates visible parts of the object to hidden parts using symmetries over the publicly available 3D model.

Our interactive system for editing photographs re-imagines typical photo-editing operations such as rotation, translation, copy-paste, scaling, and deformation as 3D manipulations to objects. Using our system, users have created a variety of manipulations to photographs, such as flipping cars, making dynamic compositions of multiple objects suspended in the air, performing animations, and altering the stories of historical images and personal photographs.
This thesis is dedicated to my best friend, Puppy.
I have learnt much from you,
and am always in awe of how easily
you manage to make every minute of life
on this—in the opinion of Douglas Adams—mostly harmless planet
worth living over and over.
ACKNOWLEDGEMENTS

I would like to acknowledge my advisor, Yaser, in providing me an immense amount of guidance throughout every step of my Ph.D. I owe to you my obsession of readjusting figures and text in my presentations and papers. Your dedication in motivating me and your other students toward achieving perfection has been instrumental in shaping the way I see mentorship.

I would also like to thank my other committee members, Alyosha, Kayvon, Nancy, and David, for taking the time to evaluate my work. I would especially like to thank Alyosha for giving me the opportunity to be a teaching assistant in Computational Photography. Special thanks to Kayvon for your Graphics Lab talk on how not to give a talk. Quite a few of my views on teaching stem from your talk.

I would like to thank Zhe, David, Tomas, Taylor, Jean-François, Iain, Moshe, Rachel, and Spencer for their invaluable guidance and help in my Ph.D. work. In particular, I owe much to Spencer for making 3D models of a variety of non-human characters, and most importantly, for animating the origami crane. Without your help, there would have been no ‘magic’ in my work. I would also like to thank TurboSquid for providing the 3D models used in this work.

Of course, none of my being here would have been possible without the environment provided by my family—my parents and my brother Nitish. Also, my life in the US would have been impossible without my ‘US parents’ Anup and Anita, and my dear little sweethearts, Anish and Anjani. I owe much to you all for your love and guidance. Also, many thanks to my new family: Ma, Baba, Bobby, Kelli, and Brady for welcoming me into their lives.

A special shout out to Marynel, Adam, Stephanie, Karli, Nate, Xinrou, Lavanya, Kaushik, Anusha, Radhika, Ram, Sh Shank, Karan, Arvind, and Teja. Those wonderful years will not be forgotten. Thanks to the computer vision and graphics folks for lunches, dinners, and discussions: Eakta, Hyun Soo, Varun, Yair, Hanbyul, Ed, Abhinav, Dey, Shaurya, and Allie. Eakta, your guidance about professorship was especially invaluable. You have always been a great role model, and if I am even a fourth as good as you, I shall consider myself a star.

If there is anyone in the whole universe, and all the bubble universes in the multiverse, that I am indebted to the most, it is my husband, Sean. You took my topsy-turvy life, and made it, well, about 2,079,460,347 times more topsy-turvy. By vows to be an annoying pest no more than sixteen inches away from me, you have made me the happiest person, and I have never found so great a desire to take the next step in life as I do today. Thanks for [life, the universe, and] everything.
# CONTENTS

<p>| ABSTRACT | iii |
| DEDICATION | v |
| ACKNOWLEDGEMENTS | vii |
| 1 INTRODUCTION | 1 |
| 1.1 Core Contribution of This Thesis | 3 |
| 1.2 Scientific Challenges | 5 |
| 1.3 Thesis Overview | 6 |
| 1.4 Broad Impact | 9 |
| 2 BACKGROUND | 13 |
| 2.1 Theories of 3D Shape Perception | 14 |
| 2.2 Computational Approaches to Estimate 3D Shape | 18 |
| 2.3 Image Manipulation with Estimated 3D Structure | 24 |
| 2.4 Relation to This Thesis | 27 |
| 3 ALIGNMENT OF 3D MODEL GEOMETRY | 29 |
| 3.1 Background | 30 |
| 3.2 Overview of Automatic Alignment Approach | 33 |
| 3.3 Multi-Level Laplacians of Gaussian | 35 |
| 3.4 Multi-Level Tree for Patch Localization | 35 |
| 3.5 Coarse-to-Fine Alignment | 40 |
| 3.6 Manual Geometry Adjustment | 41 |
| 3.7 Results | 44 |
| 3.8 Summary | 53 |
| 4 ESTIMATION OF ILLUMINATION AND VISIBLE APPEARANCE | 61 |
| 4.1 Background | 61 |
| 4.2 Illumination and Appearance Estimation Overview | 66 |
| 4.3 Optimization Approach for Estimation of 3D Illumination and Surface Reflectance of Visible Pixels | 68 |
| 4.4 Fine-scale Texture Difference | 71 |</p>
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5</td>
<td>Results and Evaluation</td>
<td>72</td>
</tr>
<tr>
<td>4.6</td>
<td>Summary</td>
<td>76</td>
</tr>
<tr>
<td>5</td>
<td>Completion of Appearance in Three Dimensions</td>
<td>77</td>
</tr>
<tr>
<td>5.1</td>
<td>Background</td>
<td>78</td>
</tr>
<tr>
<td>5.2</td>
<td>Setting up Graph Structure over 3D Model</td>
<td>80</td>
</tr>
<tr>
<td>5.3</td>
<td>Relating Hidden and Visible Parts via Symmetries</td>
<td>81</td>
</tr>
<tr>
<td>5.4</td>
<td>Appearance Completion Using Markov Random Field</td>
<td>83</td>
</tr>
<tr>
<td>5.5</td>
<td>Summary</td>
<td>85</td>
</tr>
<tr>
<td>6</td>
<td>Interactive System for 3D Manipulation of Objects in Photographs</td>
<td>87</td>
</tr>
<tr>
<td>6.1</td>
<td>Overview of Interaction System</td>
<td>87</td>
</tr>
<tr>
<td>6.2</td>
<td>Re-imagining 2D Edits as 3D Manipulations</td>
<td>89</td>
</tr>
<tr>
<td>6.3</td>
<td>Vintage Photographs and Non-Photorealistic Media</td>
<td>92</td>
</tr>
<tr>
<td>6.4</td>
<td>Tying 3D Modeling Software to Photographs</td>
<td>93</td>
</tr>
<tr>
<td>6.5</td>
<td>User Study</td>
<td>95</td>
</tr>
<tr>
<td>6.6</td>
<td>Discussion</td>
<td>98</td>
</tr>
<tr>
<td>7</td>
<td>Future Work: Unifying 2D Repositories with 3D Repositories</td>
<td>101</td>
</tr>
<tr>
<td>7.1</td>
<td>Enriching 3D Models with Physical Properties using Large Image and Video Databases</td>
<td>104</td>
</tr>
<tr>
<td>7.2</td>
<td>Applications</td>
<td>109</td>
</tr>
<tr>
<td>7.3</td>
<td>Proof of Concept</td>
<td>111</td>
</tr>
<tr>
<td>7.4</td>
<td>Related Work</td>
<td>114</td>
</tr>
<tr>
<td>7.5</td>
<td>Technical Challenges</td>
<td>116</td>
</tr>
<tr>
<td>7.6</td>
<td>A Word on Statistical Approaches for Using Large 2D Repositories to Learn Object Interactions</td>
<td>119</td>
</tr>
<tr>
<td>8</td>
<td>Discussion</td>
<td>121</td>
</tr>
<tr>
<td>8.1</td>
<td>Toward a Seamless User Experience in Interacting with Objects in Photographs</td>
<td>121</td>
</tr>
<tr>
<td>8.2</td>
<td>Toward a Unified Representation of Objects</td>
<td>124</td>
</tr>
<tr>
<td></td>
<td>Appendix</td>
<td>129</td>
</tr>
<tr>
<td>A</td>
<td>Likelihood for Children of Patch from Renders for 3D Model Alignment</td>
<td>129</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td>131</td>
</tr>
</tbody>
</table>
Chapter 1

INTRODUCTION

Humans have always used visual media to communicate their imagination. In visually realizing their imagination, people have been constrained by the limitation of the medium. For instance, realizing the rich detail of the visual world using a paintbrush and canvas is often a cumbersome task, requiring skills that the average person may not possess. Experts have navigated around the constraints of the medium to create works of art such as the Mona Lisa. However, it is only through the advent of technological advances such as the camera and photo-editing software such as Photoshop that have allowed everyday users to express their creative imagination. These technological advances have led to an explosion in the quantity of creative visual content available today, in users’ personal collections and on the Internet.

While they have expanded the creative capabilities available to the average user,
CHAPTER 1. INTRODUCTION

the camera and photo-editing software have introduced restrictions of their own. In shooting with a camera, users are constrained by real-world physics. For instance, to create *Dalí Atomicus*, photographer Phillipe Halsman required twenty six takes to have Salvador Dalí, three cats, a chair, an easel, and water simultaneously suspended in the air in his desired configuration. While photo-editing software such as Photoshop allows users to “defy physics”, current photo-editing tools are largely restricted to two-dimensional edits of the pixels in the image. Three-dimensional manipulations of objects in photographs are impossible in current photo-editing tools, as such tools only have access to 2D pixels, and not the 3D scene behind the photograph.

This thesis presents a system that allows users to perform full-range 3D manipulations to objects in photographs. Our system re-imagines typical 2D edits in photo-editing software as 3D manipulations to objects, such as rotations, translations, deformations, and 3D copy-paste. The goal of this thesis is to provide perceptually plausible end results of 3D manipulation of the object with seamless transition of the object from the original photograph. Using our system, users have flipped cars as in Figure 1.1, created dynamic compositions of suspended fruit that appear to defy physics, animated objects, and manipulated historical photographs containing airplanes that do not exist today. By providing 3D control over photographs intuitive to everyday users, the work in this thesis takes the next step in democratizing the creation of visual content.

The principal challenge in this thesis is that estimation of 3D structure from a single photograph is highly under-constrained. The camera projection discards depth information. More importantly, the photograph lacks information about the geometry and appearance of the hidden parts of the object. As shown by the underside of the cab in Figure 1.1, three-dimensional manipulations of objects in photographs are likely to reveal these hidden parts. An important question is, how can we reveal hidden parts of an object in a manner consistent with human interpretation of objects in three-dimensions? As humans, we have a lifetime of experience handling objects in the real-world. We use our experience to infer the geometry and appearance of hidden parts of everyday objects, even without directly viewing them (Palmer, 1999).

One conceivable approach is to re-create the scene with the appropriate geometry, appearance, and illumination using computer-aided design (CAD) software. However, using CAD software to model scenes and render photorealistic images that match a photograph, and that allow plausible seamless manipulations of scene objects, requires teams of experts that only large special effect companies can afford. In addition, it calls for specialized calibration equipment and photographs of objects at several views and exposures. Performing such tasks is not within the capabilities of the average consumer.

The key insight of this thesis is that we can instead use public repositories of 3D models as a proxy for our lifetime of experience. Currently, public repositories of 3D models are expanding at an unprecedented rate, due to the move toward
Figure 1.2: The contribution of this thesis is to leverage 3D models from large repositories to obtain a three-dimensional reconstruction of the scene underlying a photograph sufficient for full 3D object manipulation. We obtain the illumination, appearance and geometry of the object and contact surfaces in three dimensions. As shown by the alternate viewpoint of the scene on the bottom right, we use the 3D model to complete the geometry and appearance of the object hidden in the original photograph. Using the completed geometry and appearance, users can reveal novel parts during 3D manipulation to create the result shown in Figure 1.1.

model standardization, and the rise of 3D printing and scanning technology. In 2012 alone, TurboSquid.com had 30,000 new high-quality 3D models uploaded to their repository\(^1\). At the time of writing this thesis, Google 3D Warehouse cites their 3D model count to be over 2 million\(^2\). The expanse and ever-increasing diversity of public repositories assures that we will find a 3D model that resembles the object in the photograph. This thesis leverages the geometries and textures of 3D models found in large repositories to reveal parts of the object not seen in the original photograph.

1.1 Core Contribution of This Thesis

The work in this thesis takes a step toward unifying 3D model repositories with two-dimensional media such as photographs and videos. The core contribution of this thesis is to leverage 3D model repositories to enrich the range of creative control over photographs. In particular, as shown in Figure 1.2, the work in this thesis uses 3D models to reconstruct a sufficient three-dimensional representation of the scene underlying a photograph so that users can manipulate objects in the photograph in 3D to create perceptually plausible results.

\(^1\)http://blog.turbosquid.com/2013/01/25/squid-stats-2012-the-year-in-review/
\(^2\)http://3dprint.com/65723/materialise-trimble-partner/
CHAPTER 1. INTRODUCTION

We are currently in the age of “Big Visual Data”. Enormous quantities of images and videos are uploaded to the Internet daily. For instance, as of the time of writing this thesis, 300 hours of video are uploaded to YouTube every minute. There exist several approaches to leverage large image and video databases to enrich the range of creative manipulations available in images, by providing, for instance, scene completion (Hays and Efros, 2007), insertion of novel content from images and webcams (Lalonde et al., 2007, 2009), and altering images to match different times of day (Shih et al., 2013). Public repositories of 3D models, such as 3D Warehouse, TurboSquid and Thingiverse, have also begun to expand rapidly. The move toward model standardization, rising interest in leveraging 3D models from fields such as real estate, medicine, and digital forensics, and the emerging pervasiveness of robotic systems, 3D printing, and virtual reality in the consumer market have contributed to the expansion of 3D model repositories. A significant research and development effort is being devoted toward enriching consumer experience on using 3D models by providing consumer oriented 3D model and scene design (3DTin; Ciara; Sculpteo; Chaudhuri et al., 2011; Kalogerakis et al., 2012a), fast 3D model search (Chen et al., 2003; Funkhouser and Shilane, 2006; Fisher and Hanrahan, 2010; Russell et al., 2013), synthesis of plausible 3D model arrangements (Fisher et al., 2012), and fast rendering of 3D scenes.

However, the two types of repositories have been largely independent of each other. While there have been a number of approaches to bridge the gap between 3D model repositories and image repositories by aligning 3D models to images, going as far back as Ayache and Faugeras (1986), Huttenlocher and Ullman (1987), and Lowe (1987a), these approaches do not provide a complete unification of 3D model repositories with image repositories, as they do not leverage one modality to provide novel content in another, i.e., they do not leverage the aligned 3D models to provide novel content in images, or vice versa. While there exist approaches to use 3D models in generating novel content in images, most approaches have focused on inserting a new object into a photograph (Debevec, 1998; Karsch et al., 2011; Karsch et al., 2014), as opposed to using the 3D model to provide novel content about objects in the photograph.

The work in this thesis is the first to leverage 3D model repositories to provide full 3D manipulation of objects in photographs with perceptually convincing results. Our results in Chapter 6 demonstrate that by tying 3D models to objects in images, we can greatly expand the range of creative control users have over their photographs. As shown in Figure 1.2, our work leverages 3D models to build a plausible three-dimensional reconstruction of the geometry, appearance, and illumination of the scene underlying the photograph. As discussed in Section 1.2, our work addresses the challenges of correcting the geometry, appearance, and illumination of the 3D model to match the object in the photograph. By addressing the challenges, we obtain a 3D reconstruction sufficient for perceptually plausible results of object manipulation.

1.2 Scientific Challenges

One solution to provide 3D manipulation over an object in a photograph is to replace the object in the photograph with a publicly available 3D model of the object. However, this insertion-based approach fails to provide perceptually plausible results of 3D manipulation seamless from the original photograph, due to three technical challenges: deviation between the geometries of the 3D model and the object in the photograph, mismatch in appearance of the 3D model and the photographed object, and unknown illumination. The following subsections discuss these challenges.

1.2.1 Geometry Mismatch

For natural objects, such as fruit or animals, the geometry of each instance will be slightly different, and it is inconceivable to model every instance of a mango or labrador in 3D. Even artificial objects manufactured according to standardized specifications may consist of flexible materials as in the case of pillows, hats, and hand bags, or may undergo weather-dependent or accident-dependent shape changes such as car dents. While a perfect match may be found (e.g., a specific brand of chair), many 3D models are created with artistic license and their geometry will likely not be metrically accurate. As shown in Figure 1.3, the mesh of the 3D model of the chair from the online repository does not line up with the chair in the photograph. While 3D scanning might provide metric accuracy and has been extended to capture the interiors of living spaces (Izadi et al., 2011; Zhou et al., 2013), current 3D scanning techniques introduce artifacts for non-Lambertian and fuzzy materials. They cannot be used to model large structures such as the exteriors of trucks and buildings, or non-existent objects such as historical airplanes or old architectures.
1.2.2 Appearance Mismatch

Although both artists and scanning techniques provide detailed descriptions of object appearance in the form of surface reflectance on a texture map, these descriptions may not match the colors and textures of the particular instance of the object in the photograph. An artist may not have the technical capability to capture the accurate BRDF of an object being modeled, and may use artistic license to determine the appropriate choice of colors. In the case of the chair in Figure 1.3, the artist provides a flat shaded appearance, while the original chair has an appearance of woven fiber. 3D scanning techniques may contain illumination-dependent appearance or appearance generated by capturing multiple photographs of textureless objects with removable markers. In addition, photographed instances of objects may have discolorations due to aging, weathering, or changes in the manufacturing process (for instance, upgrade in choice of colors or materials).

1.2.3 Unknown Illumination

To perform realistic manipulations in 3D, we need to generate plausible lighting effects, such as shadows on an object and on contact surfaces. The environment illumination that generates these effects is not known a priori. Naïvely guessing the positions of the light sources can provide unnatural lighting effects as in the case of the rotated chair to the right of Figure 1.3. While there exist techniques to accurately measure the illumination in a scene (Debevec, 1998; Reinhard et al., 2005), these techniques involve the use of calibration objects such as light probes and specialized hardware to capture the dynamic range of the illumination, which are generally out of the reach of the average consumer. Furthermore, for several scenes such as vintage photographs, challenging environments, and non-photorealistic media such as paintings, the user may not have access to the original scene to take illumination measurements.

Addressing these challenges of accurately matching the geometry, estimating the unknown illumination, and completing the appearance from a single photograph is crucial to provide seamless manipulation of objects in photographs.

1.3 Thesis Overview

Our approach provides the following specific contributions to provide a three-dimensional representation of objects by matching the 3D model to the photograph:

1. An automatic approach to align the geometry of rigid and non-rigid 3D models to photographs, with a tool to perform manual geometry adjustments using sparseness and symmetry constraints.

2. A new non-parametric model of image-based lighting for illumination-aware compositing of manipulated objects into the photograph.
3. An automatic method to complete the appearance of hidden parts of objects using symmetries in the 3D model.

In addition, to allow users to perform plausible seamless manipulations, our approach provides

4. An interactive system through which users can manipulate aligned 3D models and create new compositions.

The manual geometry adjustments tool, the image-based lighting approach, the appearance completion method, and the interactive system for manipulating 3D objects appears in the following paper:


This thesis addresses the challenge of mismatch in geometry by providing an automatic approach to align 3D models to photographs with a manual tool for final adjustments. The approach, described in Chapter 3, is distinct from prior work such as Xu et al. (2011), Aubry et al. (2014), and Lim et al. (2014) in that it provides precise alignment of a 3D model to the contours of a photograph, which is crucial for seamless manipulation. Our challenge to automate the precise alignment of the 3D model is that the alignment requires an exhaustive search in the space of all possible viewpoints, scales, 3D illumination environments, and 2D locations of 3D model points in the image. If done naïvely, the exhaustive search is computationally infeasible. To make the search tractable, we leverage an illumination-invariant representation of the 3D model and the image, and we exhaustively span the space of rigid poses by rendering the 3D model from several discrete viewpoints and scales.

To precisely match the point locations on each discrete render to the object in the image, we perform dynamic programming over a multi-level tree of patches to simultaneously localize them in the image. In cases where the automated geometry alignment fails, the thesis presents a manual correction tool for performing final adjustments so that users have a fall-back for seamless 3D manipulation.

Plausible manipulation of objects in photographs requires the creation of illumination effects such as shadows and shading. Cast shadows on objects and surfaces are an important cue for depth perception (Puerta, 1989). In this thesis, we provide an image-based lighting algorithm to estimate the environment illumination from a photograph for plausible lighting effects. We discuss this approach in Chapter 4. To represent illumination, we use an environment map, which is a sphere in 3D that specifies the illumination incident on the object from all directions. Our approach uses a non-parametric model that captures the features of area lighting, such as blurred edges of cast shadows and smooth shading. To estimate area light sources, such as lamps, windows, and ceiling lights, we use von Mises-Fisher kernels.
(Fisher, 1953), which represent Gaussians on the surface of the environment map. Spatial groupings of these kernels represent larger illumination sources. We enforce such spatial groupings while maintaining a sparse set of light sources by imposing priors on the estimation. In addition to environment illumination, our estimation approach provides an estimate of appearance that matches the original photograph in visible parts of the object.

To provide seamless object interaction in photographs, we need to reveal hidden appearance of object parts consistent with their appearance in the original photograph. Given the estimate of the appearance from the illumination estimation approach of Chapter 4, this thesis provides an algorithm that uses symmetries over the 3D model as a guide appearance completion, as described in Chapter 5. Most objects have a principal plane of symmetry, often termed the bilateral plane. Even in cases where objects which may not have strict symmetries, they may have approximate symmetries, for instance, a mango, or partial symmetries. For instance, while the interior of a car is not symmetric, the steering wheel can be symmetric. Our approach uses the 3D model to determine hidden parts that are symmetric to visible parts, and completes appearance while maintaining a seamless boundary with the visible parts. For hidden parts that are visually distinct from visible areas such as the taxi cab in Figure 1.1, we use the appearance of the publicly available 3D model. The approach thus allows users to reveal novel areas while maintaining a seamless transition in appearance from the original photograph.

This thesis presents a system that allows users to interact with and manipulate objects in photographs to create novel compositions. We describe this system in Chapter 6. As input, the user provides a photograph and a 3D model obtained from a public repository using a word search. The system performs automatic model alignment and allows the user to perform manual geometry adjustments using an interactive tool. The system then estimates environment illumination and completes appearance. The user can now interact with our system to perform manipulation operations such as moving and rotating one or more objects or their parts, duplicating objects, performing non-rigid deformations, and making objects interact with each other. Through our system, users can create a variety of manipulation effects such as animations, physical simulations, and compositions that defy physical constraints. Our system also allows users to edit vintage photographs or non-photorealistic media, such as paintings. We thereby re-imagine typical operations in current photo-editing software to allow users to interact with photographed objects in three dimensions while providing perceptually plausible results.

The thesis takes a step toward unifying repositories of 3D models with repositories of images and videos by leveraging repositories of 3D models to reveal novel parts of objects in photographs. There exist interesting opportunities for future work in the opposite direction, i.e., to leverage repositories of two-dimensional content such as images and videos to provide novel information for 3D models. In particular, public repositories of 3D models lack specification of physical properties such as
object mass, coefficients of friction, and elastic stiffness. Such physical properties have the scope to allow everyday users to use 3D models for applications such as physical simulations of objects in photographs, robotic manipulations, and 3D printing. Chapter 7 describes potential methods for future work on estimating physical properties of objects using 3D models aligned to videos of objects. We demonstrate a proof of concept, and we detail the technical challenges that would arise in estimating physical properties. Chapter 8 provides a discussion on the next steps to be taken in extending the work in this thesis to provide a seamless interaction experience in photographs, and provides concluding remarks on the unification of large repositories of information.

1.4 Broad Impact

The work in this thesis provides the potential for contributions toward computer science research, the commercial sector, and in fields outside of computer science.

1.4.1 Research in Computer Science

Our work uses 3D models from large repositories to constrain the ill-posed estimation of geometry, illumination, and appearance of objects for full-range 3D object manipulation. We have opened the scope for research on unifying large 3D model repositories with large repositories of images. The work in this thesis demonstrates that by leveraging 3D model repositories, we can enrich the creative control of images by providing novel content in a perceptually plausible manner. To make 3D manipulation in objects seamless, we have created the need for real-time 3D model search within large repositories, using graphical processing units for real-time alignment and manipulation, and performing object-aware estimation of geometry and appearance using large image collections. Chapter 7 discusses the potential to move in the opposite direction, i.e., to leverage large repositories of images and videos to learn novel content for 3D models. In particular, we discuss the potential to populate 3D model repositories with physical properties of objects for applications such as user-independent physical simulations, robotics, and 3D printing. Learning and synthesizing novel content by bridging large 3D repositories with large 2D repositories and large repositories of text and speech expands the scope for collaborations across multiple disciplines in computer science, such as computer graphics and vision, machine learning, linguistics, and distributed computing.

1.4.2 Commercial Sector

In providing intuitive 3D control over photographs for the average user, our work has significant potential in impacting commercial applications where the impact of visual results is paramount. A particular contribution of our approach, as shown in Chapter 6 is to bridge the gap between 3D modeling software and photo-editing software,
thereby accelerating the flow of creativity between photographers and artists familiar with 2D photo-editing, and 3D modelers and animators. Our work has already generated interest amongst real-estate consumers due to its potential to realistically stage houses by combining manipulation of existing content such as walls, cabinets, and electrical appliances with insertion of new content such as furniture. The thesis also provides the potential to improve virtual shopping experience by allowing the user to manipulate browsed items—for instance, by rotating vehicles, furniture, and appliances or by manipulating clothing and accessories to gauge fit and style. Currently, some Internet companies are beginning to provide three-dimensional views of items by acquiring large photo collections of such items. Tying already available 3D models to a single photograph of a merchandise item will greatly reduce the effort invested by virtual shopping companies providing realistic three-dimensional views of items to potential consumers.

1.4.3 Research in Other Fields

We expect the work in this thesis to be applicable to several fields outside of computer science, such as digital forensics, medicine, and K-12 education.

Digital Forensics. Computer graphics techniques are currently being used to assist forensic anthropologists in identifying and examining human remains (Davy-Jow et al., 2013), and in reconstructing facial features from mutilated remains to assist forensic investigators in person identification (Vanezis et al., 2000). However, there has been almost no work done in providing computational approaches to assist investigators with visually back-tracing events using crime scene photographs. At present, investigators either use their prior experience or subject dummies to hypothesized causes. By tying physical simulations to photographs, the work in this thesis has the potential to assist investigators in realistically visualizing potential scenarios behind a crime scene or an accident, such as back-tracing the impact of a bullet in sending a victim to the ground, or determining the speed at which a car may have hit a truck.

Medicine. 3D models of human body parts are becoming increasingly precise. However, their use in medicine has been limited. By tying 3D models to operation theater or endoscope cameras, the work in this thesis has the potential to provide computer-aided guidance to doctors on parts of organs that may be hidden from view during surgery. Similarly, the work has the potential to guide motion planning in surgical robots such as the da Vinci surgical system.

K-12 Education. By enriching user interaction in photographs, we foresee the potential of this work in allowing K-12 students to gain practical experience about objects in their surroundings. While K-12 students show curiosity about the internal workings of machines, vehicles, furniture, and electrical appliances, providing expo-
sure to such objects in the real world may prove unsafe. Instead, we see enormous potential to allow students to pry open objects by performing 3D manipulations to parts of objects. A particular area of interest is designing virtual reality software where students learn topics in mathematics, physics, engineering, and computer science by answering quiz questions relating to the object being inspected before taking the object apart. We also see potential for using the software to engage students in building novel objects from spare parts. By allowing students to capture photographs of their surroundings, and manipulating objects to probe their internal workings, we believe we can generate significant motivation amongst students to pursue STEM education.
Chapter 2

BACKGROUND

The idea behind providing 3D control over photographic elements is that humans infer photographed content in three dimensions, rendering 3D control more intuitive than 2D manipulation of pixels. In this chapter, we discuss theories that describe human perception of three-dimensional information from single images, and we detail psychophysical studies done in support of each theory. We then discuss the computational approaches to extracting three-dimensional information from single images, and we detail approaches to use the extracted three-dimensional information in computer graphics applications, with an emphasis on image manipulation. We conclude the chapter with the relationship of our work to the theories of the perception of three-dimensional information.

Much work has been done on understanding 3D geometry from images. While there has been substantial work done on inferring 3D geometry using multiple images, such as approaches on photometric stereo (Woodham, 1980) and structure from motion approaches (Hartley and Zisserman, 2004; Snavely et al., 2006), we focus on work done on understanding the 3D scene from a single image. Work on single image 3D geometry estimation may be categorized into two parts: work on obtaining the 3D geometry of visible surfaces in an image, or what David Marr termed the 2.5D sketch (Poggio, 1981; Marr, 1982), and work on interpreting the complete 3D structure or Marr’s 3D model from a single image. This 3D structure includes hidden surfaces. Palmer (1999) and Hoiem (2007) provide a comprehensive overview of perceptual studies and computational approaches for perception of the depth of visible surfaces. Since this thesis focuses on providing full 3D control over objects, we discuss the perception and computational estimation of complete 3D structure of a scene, including hidden surfaces, from a single image.
2.1 Theories of 3D Shape Perception

There are two main categories of theories about how humans perceive the 3D shape of an object (Burgund and Marsolek, 2000):

**Viewpoint Dependence Theories.** The main premise of these theories is that the mental representation of the 3D shape of an object corresponds to several 2D images of the object from multiple views. Given the retinal image of an object in a scene, the brain perceives the 3D shape of the object by comparing the retinal image to the 2D images. Section 2.1.1 discusses viewpoint dependence theories and psychophysical studies supporting these theories.

**Viewpoint Invariance Theories.** These theories propose that the brain maintains three-dimensional representations of objects that are viewpoint invariant. Given the retinal image of the object from a viewpoint, the brain infers the three-dimensional representation of the object by comparing viewpoint invariant features from the image to the viewpoint invariant representation of the object. We discuss viewpoint invariance theories and psychophysical studies supporting these theories in Section 2.1.2.

2.1.1 Viewpoint Dependence Theories

Most viewpoint dependence theories have their roots in empiricism (Pizlo, 2008), according to which knowledge is gained primarily through sensory experiences as opposed to being innate (Psillos and Curd, 2010). Helmholtz (1866) was the first to advance an empirical theory of 3D shape perception in his *Treatise of Physiological Optics*. He equated perception of 3D shape to “unconscious inference”, according to which the brain infers 3D information by combining 2D retinal input with constraints learnt through prior sensory experience. He hypothesized that humans develop associations between 2D images of several views of an object, due to their daily experience with handling objects. Given the retinal image of an object, we associate the image with the views we have seen before, and retrieve the closest view. Irvin Rock (1977) refined Helmholtz’s theory with the proposal that the visual system uses additional constraints such as rules of geometric optics to learn smart associations between images. Subsequently, multiple views theories were proposed by Tarr and Pinker (1989), and Bülthoff and Edelman (1992), according to which an object was recognized by the human visual system by normalizing it to the nearest familiar 2D view stored within our mental representation.

One of the main claims of viewpoint dependence theories is that shape constancy, i.e., our ability to infer two different views of an object as belonging to the same object, cannot be achieved for unfamiliar views of objects. According to these theories, given an unfamiliar view of an object as retinal input, the brain cannot infer the three-dimensional object that correctly matches the retinal input. Evidence for poor shape constancy in humans comes from experiments conducted by Rock and his associates (Rock et al., 1981; Rock and DeVita, 1987; Rock et al., 1989), and by Bülthoff
and Edelman (Bülthoff and Edelman, 1992, 1993; Edelman and Bülthoff, 1992). In his experiments, Rock showed subjects two images of different views of abstract wireframe objects, and asked subjects to report whether they belonged to the same object or to different objects. Bülthoff and Edelman showed images of amorphous clay objects, or “amoebae” to their subjects. In both sets of experiments, subjects showed poor shape constancy for images of the same object. To offset the effect of unfamiliarity of the objects themselves, Tarr (1995) trained subjects to become familiar with a set of viewpoints of asymmetrical objects composed of a series of blocks, and tested shape constancy with novel views of the familiarized objects. Tarr found poor shape constancy for the novel views. The evidence for poor shape constancy for unfamiliar views of objects led the above researchers to conclude that 3D shape perception is viewpoint dependent.

2.1.2 Viewpoint Invariance Theories

The main issue with the claim of poor shape constancy for unfamiliar views as stated by viewpoint dependence theories is that for everyday objects, we tend to have approximate shape constancy (Palmer, 1999). Even if we have not seen the underside of a chair, we are rarely surprised when we see the underside. In fact, when Slater and Morrison (1985) conducted experiments similar to Tarr (1995), where they familiarized newborn infants with a set of views of objects, they found that the infants registered less surprise when shown novel views of familiarized objects than when shown views of novel objects. While the conclusions of these experiments and experiments on mental rotation (Shepard and Metzler, 1971)$^1$ reveal the difficulties in resolving the debate between viewpoint dependent and viewpoint invariant theories, they also highlight an important issue with the experiments conducted by the proponents of viewpoint dependent theories: the objects used in experiments by Rock, Bülthoff, Edelman, and Tarr are abstract, and as noted by Pizlo (2008), they do not embody structural properties such as symmetry, planarity, and rectangularity found in everyday objects.

Viewpoint invariance theories remedy this issue by hypothesizing that the brain imposes constraints such as simplicity, planarity, rectangularity, and symmetry on the 3D structure of everyday objects, and uses these constraints to achieve shape constancy for novel views of objects. Many viewpoint invariance theories derive their organizing principles from the Gestalt law of Präganz, which states that the

$^1$Shepard and Metzler (1971) performed classic experiments on mental rotation, where they showed subjects either two different views of the same object, or a view of an object and a view of its mirror image. They measured the time it took subjects to report whether the object was the same or different, and found that when the object was the same, the quantity of time taken was proportional to the angle of rotation. They concluded that subjects perform mental rotation of the second view through intermediate representations to match it to the first view. As Tarr (1995) points out, their conclusion provides support for viewpoint dependence. Shepard and Metzler however claimed that subjects used viewpoint invariant processes to identify the object before performing mental rotation.
brain performs perceptual groupings of objects and their parts in a manner that
best satisfies the requirements of the groupings being regular, symmetrical, and
simple (Koffka, 1935). Influenced by Gestaltists, most viewpoint invariance theories
hypothesize that shape constancy is innate.

While proponents of Präganz have struggled to formalize measures for simplic-
ity (the principle of continuity under Präganz may be likened to the constraint of
smoothness over surfaces), measures for planarity, rectangularity, and symmetry are
fairly straightforward. Symmetry, in particular, is inherent in nature, as shown by
the bilateral symmetries of humans, vertebrates, and leaves, the radial symmetries
of flowers, sea anemones, and snowflakes, and the approximate symmetries of fruit,
plants, and trees (Blum, 1973; Stewart, 2001). Several man-made objects also show
symmetry. Kanade and Kender (1980) demonstrated that symmetry in an object in
3D generates a skewed symmetry in the 2D projection of the object. Psychophysical
experiments on testing subjects’ performance at detecting symmetry in random dot
displays (Barlow and Reeves, 1979) show that humans are fairly accurate at perceiv-
ing exact and approximate symmetries. Symmetry contributes toward establishing
an object-centered reference frame (Marr and Nishihara, 1978), and is hypothesized
to eliminate redundancy in mental representations of objects (Attnave, 1954; Bar-
low, 1961). As will be seen in later chapters in this thesis, computing planes of
bilateral and approximate symmetries forms the basis of completing hidden parts of
objects based on the content visible in a single image.

Marr provided an approach on viewpoint invariant representation while describ-
ing the object-centered coordinate system for the 3D model of an object (Marr and
Nishihara, 1978). However, in contrast to most proponents of viewpoint invari-
ance theories who hypothesized that 3D shape perception does not require depth
perception, Marr proposed that depth (or the 2.5D sketch) is reconstructed before
the full 3D model. Marr’s aim was to provide a computational approach to recon-
structing 3D shape from image, as opposed to providing a perceptual theory. Bieder-
man (1985) provided a perceptual theory on recognizing shape, in which he proposed
that the brain recognizes 3D shape as a collection of primitives or “geons”, such as
blocks (Roberts, 1965), generalized cylinders (Binford, 1971), and ellipsoids (Pent-
that the Gestalt principles of symmetry, simplicity, and planarity apply at the level
of individual geons (Biederman, 1987). Influenced by Poggio and Edelman (1990),
Pizlo (2008) proposed that shape perception is an inverse problem, whose solution
requires the use of constraints such as planarity, symmetry, and compactness.

Viewpoint invariance theories hypothesize that our brain demonstrates approxi-
mate shape constancy even for unfamiliar views of objects, provided that the simplic-
ity constraint can be applied to the object. Evidence for high shape constancy with
increased simplicity has been provided by Hochberg and his associates, where they
measured 2D simplicity by the number of line intersections in a wireframe drawing
of the projection of a 3D object, as originally proposed by Kopfermann (1930). They
hypothesized that line drawings with lower 2D simplicity (i.e., higher number of line intersections) would be perceived as 3D objects. Their hypothesis was confirmed in experiments performed on projections of a cube (Hochberg and McAllister, 1953) and families of complex polyhedra (Hochberg and Brooks, 1960), in which subjects perceived objects with a lower measure of 2D simplicity as 3D objects. Perkins (1976) conducted experiments to determine the ability of subjects to reconstruct the shape of an unfamiliar polyhedron from a single orthographic view, by asking the subjects to estimate the sizes of two angles of the polyhedron as perceived in three dimensions. His analysis demonstrated that subjects applied symmetry, rectangularity, and planarity constraints to perceive the polyhedron in three dimensions. Additional support for symmetry and planarity constraints was shown in experiments by Pizlo et al. (2005), who found that subjects showed highest accuracy at shape constancy for unfamiliar symmetric polyhedra with planar faces, followed by asymmetric polyhedra with planar faces and symmetric polyhedra with non-planar faces. Subjects showed the worst accuracy at shape constancy for asymmetric polyhedra with non-planar faces. In fact, they showed higher accuracy at shape constancy with unconnected vertices than with asymmetric non-planar objects, suggesting that subjects mentally joined unconnected vertices while applying symmetry and planarity constraints.

In general, the brain may use both viewpoint dependent processes and viewpoint invariant processes to interpret objects in monocular images in three dimensions. In experiments performed by Cooper (1994), subjects were shown two orthographic views of a polyhedral object, and were asked to identify the correct isometric view (A) corresponding to the orthographic views from between a target and a distractor pair, and to identify the correct isometric view from between a target and a distractor pair for a viewpoint which shared two sides (B), one side (C), and no sides (D) with A. She found that the subjects’ accuracy at identifying B was nearly the same as their accuracy at identifying A (which was above 80%). However, their accuracy at C and D was lower (at around 60%). Based on her findings, Cooper concluded that mental representations are neither strictly view-dependent, or viewer-centered, nor strictly view-invariant, or object-centered, but somewhere in between (Cooper, 2013). As a side note, an issue with her experiments was that three of the unseen sides of A (which would all be seen in D), were planar faces, indicating that subjects may have used planarity constraints of view-invariant processes in making their decision even for drastic view changes. Burgund and Marsolek (2000) hypothesize that neural systems underlying viewpoint dependent processes may be dissociated from neural systems underlying viewpoint invariant processes in the brain. In particular, their studies demonstrated that subjects showed view-dependent priming when presented different views of an object to their right brain hemisphere, i.e., to the left part of their visual field, but not when presented different views of the object to their left brain hemisphere, i.e., to the right part of their visual field.
CHAPTER 2. BACKGROUND

2.2 Computational Approaches to Estimate 3D Shape

In this section, we detail the computational approaches to estimating the 3D structure of objects in a single image, including the surfaces that are hidden from the perspective of the camera. Since the estimation of the 3D structure of objects in the scene underlying a 2D image is highly ill-posed, most computational approaches impose constraints on the estimation. We examine the constraints in the light of the viewpoint dependence and viewpoint invariance theories presented in Section 2.1. In general, approaches on estimating 3D pose may be classified into approaches that use collections of primitives to represent the imaged object, detailed in Section 2.2.1, approaches that apply constraints of symmetry, planarity, and curvature under perspective projection to directly lift the 3D model from the image, discussed in Section 2.2.2, and approaches that use a 3D model mesh consisting of vertices and faces to represent the object in the image, discussed in Section 2.2.3.

2.2.1 Using Collections of Primitives to Estimate 3D Structure

Several approaches express objects as collections of primitives, such as blocks (Roberts, 1965), generalized cylinders (Binford, 1971), or superquadric ellipsoids (Pentland, 1986). The primitives are inherently “simple”, as each primitive is described by a small set of parameters. As discussed in Section 2.1.2, constraints such as symmetry and planarity can be enforced at the level of the primitives. Most approaches for estimating 3D structure as collections of primitives thus tend to follow Biederman’s theory of recognition by components, and may be categorized as viewpoint invariant.

Blocks. Roberts first introduced the “blocks world” framework in his thesis on machine perception of three-dimensional objects (Roberts, 1965). As shown in Figure 2.1(a), he reconstructed objects in an image as compositions of three-dimensional polyhedral blocks such as cuboids and prisms. Inherently, these primitives fulfil the criteria of having symmetry and planarity, allowing complex structures to be built by enforcing simple constraints such as contact, collinearity, and coplanarity, while satisfying perspective projection constraints in the sensory input. Despite its simplicity, the blocks world assumption has been quite successful, and forms an integral part of several approaches to estimate 3D scene structure in outdoor and indoor images (Gupta et al., 2010; Hedau et al., 2010; Lee et al., 2010; Zhao and Zhu, 2011; Xiao et al., 2012; Schwing and Urtasun, 2013).

Generalized cylinders. First proposed by Binford (1971), generalized cylinders have found significant success in representing 3D models in computer vision (Marr and Nishihara, 1978; Gross, 1994; O’Donnell et al., 1994, 1998). They underlie Marr’s object-centered hierarchical representation of objects in 3D, and as pointed out by Koenderink (1990), they represent nearly everything about object structure.
As shown in the top row of Figure 2.1(b), generalized cylinders consist of an axis and a cross-section whose shape changes as a function of location along the axis. In the absence of constraints, generalized cylinders have several parameters to solve for such as the parameters of the axial curve, the parameters of the 2D shape of each cross-section, and the angle between the curve and each cross-section. To reduce the number of parameters to solve for, most approaches enforce constraints found in typical manmade and natural objects. For instance, several approaches use straight homogeneous generalized cylinders (SHGCs, Shafer and Kanade, 1983) which have straight axes, constant cross-section shape, and varying cross-section size as shown in Figure 2.1(c). Approaches that use SHGCs to estimate shape include those of Marr (1977), Brady et al. (1985), Ponce et al. (1989), and Gross and Boult (1996). Some approaches apply constraints of symmetry to the SHGCs (Horard and Brady, 1987; Ulupinar and Nevatia, 1988, 1995; Sato and Binford, 1993; Sayd et al., 1996). Zerroug and Nevatia (1996), and Ulupinar and Nevatia (1995) analyze cylinders with planar axial curves, such as a snail shell, while Gross (1994) analyzes cylinders with three-dimensional axial curves and circular cross-sections such as plumbing pipes. Terzopoulos et al. (1988) apply constraints of symmetry while fitting generalized cylinders with curved axes to objects in images. Brooks (1981) used generalized cylinders with straight and circular axes, and simple polygons as cross-sections in his ACRONYM system for estimating 3D structures of objects from

Figure 2.1: (a) Collections of block primitives such as the cuboid, triangular prism, and hexagonal prism shown in the first row have been used to represent objects in images as shown in the second row. Image courtesy: Roberts (1965). (b) A generalized cylinder, shown in the top row, describes a surface parametrized in terms of an open or closed three-dimensional curve representing its axis and several closed three-dimensional curves representing cross-sections along the axis. As shown in the bottom row, a straight homogeneous generalized cylinder has a linear axis in three dimensions, and cross-sections with constant shape and varying scale along the axis. (c) Super-quadric ellipsoids with powers equal to 2, greater than 2, between 1 and 2, and less than 1 represent regular ellipsoids, rounded cuboids, bloated octahedra, and eight-pointed stars respectively.
2D images.

**Superquadric ellipsoids.** To address the issue of the large number of parameters in generalized cylinders, Pentland (1986) provides a simpler representation of the scene in terms of superquadrics. As shown in Figure 2.1(c), superquadrics are generalizations of ellipsoids where the squaring operations are replaced by arbitrary powers. Powers higher than 2 yield rounded cuboids, powers between 1 and 2 yield bloated octahedra, while powers less than 1 yield eight-pointed stars. Inherently, superquadrics are symmetrical, allowing them to be used to represent the shape of objects with symmetries, such as human figures, from single images (Terzopoulos and Metaxas, 1991).

### 2.2.2 Lifting the 3D Structure from 2D Images

Some approaches directly apply the Gestalt constraints of symmetry and planarity to images together with the constraint of perspective projection to recover full 3D structure. Even without any prior information, Koenderink (1984) provided mathematical constraints on the relation between the intrinsic curvature of an object and its apparent curvature under a projection. Prasad et al. (2006) incorporate these constraints together with the constraint of symmetry to inflate 3D models of curved objects with bilateral symmetry, such as vases, genie lamps, bananas, and donuts, from images of such objects. Kanade (1981) reconstructs 3D shape by applying constraints such as symmetry (Kanade and Kender, 1980), planarity, and parallelism constraints in conjunction with junction labelings (Kanade, 1980). Lifting the 3D model from the image while maintaining constraints of smoothness, symmetry, and planarity has been successfully used in creating 3D free-form shapes from user-drawn sketches (Igarashi et al., 1999; Mitani et al., 2002; Schmidt et al., 2005; Karpenko and Hughes, 2006; Kara et al., 2007; Nealen et al., 2007; Owada et al., 2007; Rose et al., 2007; Andre and Saito, 2011).

### 2.2.3 Using 3D Models to Estimate 3D Structure

Despite the appeal of decomposing objects into simple primitives, most primitive-based approaches have been successful only for ideal cases where the objects reconstructed are composed of primitives (Dickinson et al., 1997). For most objects, this is hardly the case\(^2\). Approaches that lift the 3D model from the 2D image produce accurate 3D model reconstructions only when exact symmetry and planarity constraints completely relate hidden parts of objects to visible parts, which is not be generalizable to most objects or their parts. For instance, the underside of a taxi-cab

\(^2\)Actually, all objects can be represented as combinations of generalized cylinders, since any object part can be “sliced” into cross-sections along an axis. However, most primitive-based approaches rarely solve for the parameters of this generic case of a generalized cylinder, as the number of parameters end up being nearly the same as the number of vertices of a 3D model.
COMPUTATIONAL APPROACHES TO ESTIMATE 3D SHAPE

is not symmetric to the top side, and may not be recovered through symmetry and planarity alone. To recover fine-grained 3D structure of objects, several approaches perform alignment of 3D models to objects in monocular images. Here, a 3D model represents a collection of vertices. In some cases, the vertices may be connected via edges and faces to form a mesh. Approaches to align 3D models of objects to images may be divided into feature-based approaches, and exhaustive viewpoint-search approaches.

**Feature-based approaches.** Feature-based approaches extract features from the 3D model and the input image, find matches between image features and model features, and solve for the transformation that best aligns the model features to the image features. Typically, these methods use approaches such as RANSAC (Fischler and Bolles, 1981) to generate several hypotheses for the transformation using various base sets of matched image-model feature pairs. They then determine the best transformation by testing the hypotheses against verification sets. Most feature-based approaches are related to viewpoint invariance theories of 3D shape perception, as they attempt to directly map features from the 3D model to the 2D image without dependence on view.

Early work in feature-based model alignment such as that of Ayache and Faugeras (1986) and Huttenlocher and Ullman (1987) addresses the alignment of 3D models of flat objects to images. To align solid 3D shape to 2D images, Goad (1983) exhaustively searches for matches over combinations of object edges and image edges by using the latest hypothesis of position and orientation to measure the distance between the image edge and the object edge projection. Walter and Tropf (1983) assume orthogonal projection and use the 3D model geometry to determine that on fixing a single 3D point in the image, the projection of the next 3D point lies on a circle in the image, and the projection of the third 3D point lies on an ellipse in the image. They use shape elements such as curves, concentric circles, and parallel lines as features, and an augmented transition network to keep the search for matches (Woods, 1970) tractable. Lowe (1987a) creates perceptual groupings out of edge segments in an image, and solves for the transformation between matched image groupings and groupings of line segments in the 3D model. The perceptual organization approach of Lowe derives from the Gestalt principles underlying viewpoint invariant mental processes of object recognition (Lowe, 1984). Approaches such as those by Gordon and Lowe (2006), Rothganger et al. (2006), Collet et al. (2009), Arie-Nachimson and Basri (2009), Glasner et al. (2011), and Yan et al. (2007) match descriptors affiliated with 3D points to descriptors extracted from interest points in the image. These methods obtain descriptors for 3D points by collecting keypoint descriptors for the 2D point projections of the 3D points in 2D views of the object. Keypoints extracted include Harris points (Harris and Stephens, 1988), SIFT points (Lowe, 2004), and affine invariant interest points (Mikolajczyk and Schmid, 2002), while descriptors include SIFT (Lowe, 2004) and histogram of gradients (Dalal and
Triggs, 2005) or variants thereof (Glasner and Shakhnarovich, 2011). Such approaches of matching 3D point features from a 3D model to 2D point features from an image have been extended to perform image localization within the context of large 3D models, for instance to register an image to the 3D model of a city (Sat-tler et al., 2011), or to determine the worldwide pose of an image with respect to a georeferenced coordinate system (Li et al., 2012). Other features used to match 3D models to images include edges (Drummond and Cipolla, 2002) and global shape (Dambreville et al., 2008).

One issue of feature-based approaches is that they require a large number of “good” matches for the hypothesis verification step to eliminate outliers. The approaches discussed in the previous paragraph work well for 3D models created from scans of highly textured objects, and on images with appearance similar to the original objects. They generally perform poorly on textureless objects, and on 3D models where the appearance is mismatched from the original object, such as user-designed 3D CAD models found in online repositories. To mitigate this issue, several approaches use a data-driven technique to learn feature descriptors which locate a 3D model point in the image with high accuracy. To obtain feature descriptors, Pepik et al. (2012) and Lim et al. (2013) use histograms of oriented gradients (Dalal and Triggs, 2005), and Zia et al. (2013) use a dense variant of the shape context (Andriluka et al., 2009) extracted from wireframe renderings of the 3D model. To obtain high discriminative power for the feature affiliated with a 3D point, such approaches use discriminative classifiers such as support vector machines (Pepik et al., 2012), linear discriminant analysis (Lim et al., 2013) using the method of Hariharan et al. (2012), and random forests (Zia et al., 2013).

**Exhaustive viewpoint-search approaches.** Despite the attempt of approaches in the previous paragraph to learn generic descriptors, the viewpoint may be initialized incorrectly if the 3D-2D match fails, for instance, if points on an object from a certain view appear similar to other points on the object from alternative views. To impose viewpoint consistency (Lowe, 1987b), exhaustive viewpoint-search approaches compare the image to a large number of 2D views of the object, usually obtained by rendering the 3D model of the object from multiple viewpoints and scales. They rank each view according to its matching score in the image, and may perform a final step to align the 3D model to the best matching view. The approach for estimating 3D geometry in this thesis falls under the exhaustive viewpoint-search category.

At first glance, exhaustive viewpoint-search approaches appear to be related to viewpoint dependence theories of 3D shape perception, which suggests that they may preclude prediction of the 3D shape of unseen parts of an image. Indeed approaches that perform object detection and/or object pose estimation using multiple 2D views of the object without maintaining spatial relationships between the views can be categorized as viewpoint dependent. Typically, such approaches proceed by
developing a classifier for each view, and treating object detection or pose estimation as a multi-class problem. Examples of the classifiers used in multi-view object detection or pose estimation include neural networks (Rowley et al., 1998), boosting (Jones and Viola, 2003), naïve Bayes (Ozuysal et al., 2009), and deformable parts models (López-Sastre et al., 2011).

However, approaches that maintain spatial relationships between the 2D views of the object share elements with both viewpoint dependence and viewpoint invariance theories. Methods that obtain the 2D views by rendering a 3D model implicitly maintain spatial relationships between the 2D views through the 3D model. Such methods may be used to infer the geometry of novel viewpoints of an object simply by re-rendering the 3D model from the novel viewpoint. In several cases, such as in the case of user-designed 3D CAD models, the 3D model itself embodies properties such as symmetry, planarity, and simplicity. 3D modelers use blocking to lay out the 3D model to maintain smoothness and simplicity before performing fine touches. They create the structure of one half of a bilaterally symmetric object before reflecting it across the bilateral symmetry plane. As will be discussed in Chapter 3, this thesis enforces constraints of smoothness and symmetry while aligning the 3D model to the image.

Huttenlocher and Ullman (1990) provided one of the first approaches to perform multi-view alignment of 3D models to an image. They extended the feature-based approach of using corners and inflection points from their work on aligning flat 3D models (Huttenlocher and Ullman, 1987) to align multiple views of the 3D model of an object to an image and to select the best matching view. While Huttenlocher and Ullman’s approach was generative, almost all exhaustive viewpoint-search approaches to align 3D models with images since then have used discriminative models to obtain higher accuracy of alignment. Some approaches use discriminative models at the level of the viewpoint. For instance, Poggio and Edelman (1990) train a network of generalized radial basis functions (Poggio and Girosi, 1990) to learn a mapping between a small set of perspective views to a standard view, and use the network to recognize an object from multiple views. Ullman and Basri (1991) learn a linear mapping between a target view and a set of canonical views. Other approaches use discriminative models at the level of detecting object parts. These include the approach of Stark et al. (2010) that uses AdaBoost (Freund and Schapire, 1997) trained on shape context descriptors (Andriluka et al. 2009) from wireframe renderings, the approach of Aubry et al. (2014) that uses linear discriminant analysis (Hariharan et al., 2012) trained on histograms of gradients, and the approach of Lim et al. (2014) that uses the deformable parts-based model (Felzenszwalb et al., 2010) trained on histograms of gradients from rendered and real images.

Discriminative methods usually require additional annotated information to provide negative examples or to augment the positive examples for the classifier. The objective in this thesis is to align a single user-provided 3D model to the image without the inclusion of any additional annotated training data. As will be seen in
Chapter 3, we transform the views to a space within which a fully generative model yields high alignment accuracy.

2.3 Image Manipulation with Estimated 3D Structure

The work in this thesis estimates the full 3D structure of an object, and uses the estimated structure to perform plausible image manipulations while revealing novel parts of objects. In this section, we detail approaches that use the 3D structure estimated from a single image in performing image manipulations. There has been a large body of work in manipulating the 2.5D information, i.e., the three-dimensional information such as geometry and appearance of visible surfaces in an image. We summarize these approaches in Section 2.3.1. Work in image manipulation of the 3D structure of objects while revealing hidden parts has been limited. We detail the approaches that perform full 3D manipulation of objects in Section 2.3.2. There exists a parallel line of work on using estimated 3D structure to insert novel content into images. We discuss this line of work in Section 2.3.3.

2.3.1 Manipulating 3D Information of Visible Surfaces

To approximate the effect of three-dimensional view changes, approaches such as those of Barrett and Cheney (2002) and Igarashi et al. (2005) triangulate the object contour in the two-dimensional image, and apply affine transformations to the triangles. While these methods do not explicitly estimate the 3D structure, their affine approximation can provide a limited range of view changes. Approaches that estimate the depth of visible surfaces in an image using, for instance, shape from shading (Horn, 1970), have been successful at providing edits to material and lighting properties of the image. For instance, Fang and Hart (2004) retexture objects in images by extracting plausible user-editable source and normal vectors on a surface using pixel intensities, segmenting the surface into patches based on normal directions, texturing the patches, and aligning and distorting the patches to match undulations on the surface. Khan et al. (2006) extract shape from shading to provide editing of material properties of objects, such as transparency, translucency, gloss, and texture, in a single image. A version of the 2.5D structure is provided by intrinsic images, first introduced by Barrow and Tenenbaum (1978), where a single image is decomposed into a reflectance image which represents the appearance of objects and a shading image which represents shading due to the interaction of lighting and geometry. Such approaches may be used to recolor images by altering the reflectance image while maintaining consistent shading information (Bousseau et al., 2009; Carroll et al., 2011). While these techniques produce realistic results to images in many cases, the three-dimensional information estimated by these methods cannot be used to provide view changes.

Several approaches provide decompositions of the image pixels in terms of depth
images to provide view changes as, for instance, in the case of the approaches of Chen and Williams (1993) and Oh et al. (2001) who provide depth-based segmentation of the image into layers, and Chen et al. (2011), who extend this approach to videos. The approaches of Horry et al. (1997) and Criminisi et al. (2000) provide view changes by imposing the Manhattan world constraints on room interiors, and reconstructing the 3D structure of the room using vanishing lines and points. Hoiem et al. (2005) and Saxena et al. (2009) use statistical approaches to assign depth to pixels in the image for re-rendering the image from novel viewpoints. The approaches of Blanz and Vetter (1999), Guan et al. (2009), and Zhou et al. (2010) also use data-driven techniques to provide view changes, however, they learn novel appearance from novel viewpoints for a small subset of objects, namely faces (Blanz and Vetter, 1999) and bodies (Guan et al., 2009; Zhou et al., 2010).

2.3.2 Manipulating 3D Structure of Objects in Images to Reveal Hidden Parts

The approaches presented in this section come closest to the approach presented in this thesis in providing 3D manipulation of objects while revealing novel parts of the objects in this section. Most of the approaches on full 3D manipulation rely on the constraints of symmetry, planarity, and regularity discussed in Section 2.1.2. The approach of Debevec et al. (1996) uses symmetries to complete the 3D structure of architectural structures such as building from a single photograph, and to reveal them from novel viewpoints. However, they use multiple photographs to reveal novel non-symmetric views. The interactive images approach of Zheng et al. (2012) and the 3-Sweep approach of Chen et al. (2013) use the primitives discussed in Section 2.2.1 as proxies: cuboids in the case of Zheng et al. and generalized cylinders in the case of Chen et al. However, as discussed in Section 2.2.3, these approaches are successful only when the geometries of the object are closely represented by collections of the primitives used the approaches. In addition, they cannot correctly reveal hidden areas that are not symmetric to visible areas.

To our knowledge, the only approach that uses a full 3D model for showing novel viewpoints is the Deep Photo approach of Kopf et al. (2008). However, their approach is different from ours in the following ways. First, they address the problem of image registration to 3D models of large-scale outdoor structures such as cities and terrain, similar to the approaches of Sattler et al. (2011) and Li et al., (2012), where 3D model deformation is not required. In contrast, our approach targets objects deforms the 3D model to account for differences between the appearance and geometry of the 3D model and the objects in the image. Second, while their approach performs texture transfer to fill holes in small view changes, they blend into the 3D model structure for drastic view changes, instead of transferring appearance from the image to novel views. As will be discussed in Chapter 5, our approach uses the symmetries and smoothness to complete the appearance in novel parts with appearance similar to the visible areas to maintain coherent appearance over the 3D
model.

A point to note is that most of the approaches discussed in this section (Debevec et al., 1996; Kopf et al., 2008; Chen et al., 2013) do not employ lighting models, and fill illumination-dependent appearance to novel areas. Zheng et al. (2012) are the only approach to provide illumination changes while manipulating the object, however, they use a single point light source, which does not work in most indoor illumination environments. The objective of this thesis is to provide perceptually convincing results of 3D manipulation of objects in photographs, especially in indoor environments. As such, as discussed in Chapter 4, we estimate an environment map that represents illumination in typical indoor scenes, and as discussed in Chapter 5, we transfer illumination-free appearance to novel parts of the objects.

2.3.3 Inserting Synthetic Objects into Images using Estimated 3D Structure

A parallel line of work focuses on leveraging the estimated 3D structure to insert novel content into images. While these methods do not estimate geometry and appearance of hidden parts of objects, they do estimate three-dimensional illumination, including light sources not seen in the image. As such, they may be classified as approaches that leverage 3D structure as opposed to 2.5D structure. Lalonde et al. (2007) insert 2D cut outs form a large “photo clip art” library by matching the illumination of the scene to the shading on the 2D cut outs. Work on inserting 3D models may be traced back as far as approaches of Fournier et al. (1992) and Drettakis et al. (1997) who addressed the issue of providing plausible illumination for computer augmented reality. The approach of Debevec (1998) renders synthetic 3D objects into the photograph of a real scene by using illumination captured using a mirrored sphere. Karsch et al. (2011, 2014) remove the requirement of physical access to the scene by estimating the scene geometry, appearance, and illumination from the photograph.

An alternative to the method of directly manipulating an object in a photograph discussed in this thesis could have been to inpaint the photographed object and replace it with an inserted object using one of the 3D model insertion approaches discussed in the previous paragraph. However, such an insertion-based approach discards the useful information about the geometry, illumination, and appearance contained in object pixels in the original photograph. Furthermore, when creating animations, object insertion methods are unlikely to produce a seamless break from the original photograph, as peculiarities of the particular instance that was photographed (e.g., smudges, defects, or a naturally unique shape) will not exist in a stock 3D model.
2.4 Relation to This Thesis

To provide full user-intuitive 3D manipulation of objects in a single image, this thesis shares elements with both view dependence and view invariance theories of 3D shape perception in humans. Our approach of aligning the 3D model geometry to the image, discussed in Chapter 3 falls within the category of the exhaustive viewpoint-search approaches discussed toward the end of Section 2.2.3. However, we differ from those approaches in that we provide non-rigid deformation of the 3D model to fit the contours of the object in the image. To provide non-rigid deformation consistent with human interpretation of scenes, we appeal to viewpoint invariance theories of shape perception, and apply the Gestalt constraints of symmetry and simplicity in the form of smoothness to the 3D model. Our approach for appearance completion similarly uses symmetry and smoothness to complete the appearance of hidden parts of objects that are similar in geometry and appearance to visible parts. In using view-specific approaches for object viewpoint identification and view-independent approaches for object deformation and appearance completion, our approach draws inspiration from the hypothesis of Burgund and Marsolek (2000), who attribute specific-exemplar processing to the right brain hemisphere, and abstract-category processing to the left brain hemisphere.
Chapter 3

ALIGNMENT OF 3D MODEL GEOMETRY

In this chapter, we describe our approach to precisely align the 3D model of an object to its photograph. The main challenge is that there exist differences between the 3D model from a public repository and the object in the photograph. The differences arise due to (1) intrinsic mismatches between the geometry and appearance of the 3D model and the object in the photograph, and (2) large variations in extrinsic factors such as viewpoint, distance from the camera, location in the image, and illumination conditions.

In this chapter, we provide an automated approach to precisely align the geometry of the 3D model of an object to a single RGB image of the object. Our challenge in automating the precise alignment is that the alignment requires an exhaustive search in the space of all possible viewpoints, scales, 3D illumination environments, and 2D locations of 3D model points in the image. If done naively, the exhaustive search is computationally infeasible. To make the search tractable, we leverage an illumination-invariant representation of the 3D model and the image, and we exhaustively span the space of rigid poses by rendering the 3D model from several discrete viewpoints and scales. To precisely match the point locations on each discrete render to the object in the image, we perform dynamic programming over a multi-level tree of patches to simultaneously localize them in the image. Our automatic alignment approach works on textured and textureless objects. For the latter case, feature-based methods such as extracting SIFT points (Lowe, 2004) fail to provide a precise match. Sections 3.2 through 3.5 describe our automated alignment approach.

In cases where the automated method does not give an exact match, we provide an interactive geometry correction tool through which the user can make final adjustments to the 3D model by providing correction solely in 2D. Our manual geometry correction tool is described in Section 3.6.
CHAPTER 3. ALIGNMENT OF 3D MODEL GEOMETRY

3.1 Background

Our work on estimating the alignment of 3D models to photographs of objects is related to the approaches on computationally estimating the three-dimensional structure of objects from single images, discussed in Chapter 2. The methods discussed in Section 2.2 of Chapter 2 are automated approaches. We recapitulate these approaches in Section 3.1.1. Several approaches use depth images and temporal information from videos to constrain the alignment of 3D models. These approaches are detailed in Section 3.1.2. There exist several semi-automatic or manual approaches to edit the 3D geometry of meshes. We detail these approaches in Section 3.1.3.

3.1.1 Automatic Approaches on Aligning 3D Model Geometry to RGB images

Several methods view the 3D model of an object as a composite of primitives, similar to the view advanced by Biederman (1985). Such approaches estimate parameters of transformation, composition, and deformation of primitives such as polyhedral blocks (Roberts, 1965; Brooks, 1981), generalized cylinders (Marr, 1977; Brooks, 1981; Brady et al., 1985; Ponce et al., 1989; Ulupinar and Nevatia, 1988; Terzopoulos et al., 1988; Sato and Binford, 1993), and superquadric ellipsoids (Pentland, 1986; Terzopoulos and Metaxas, 1988). Approaches such as those of Kanade et al. (1981) and Prasad et al. (2006) directly apply constraints such as smoothness, symmetry, and planarity to images to lift a 3D model out of the image.

As discussed in Chapter 2, approaches that align 3D models of objects composed of vertices may be divided into feature-based approaches and approaches that exhaustively search the space of alignment by rendering the 3D model. Feature-based approaches identify matching features between the 3D model and the image, and solve for a transformation that best aligns the model features to the image. Lowe (1987a) created perceptual groupings of edge segments in the image and the 3D model, and determined likely matches between the groupings. Approaches such as those of Gordon and Lowe (2006), Rothganger et al. (2006), Collet et al. (2009), Arie-Nachimson and Basri (2009), Glasner et al. (2011), and Yan et al. (2007) match descriptors of interest points such as SIFT points (Lowe, 2004) across the image and the 3D model. Such interest point based approaches work well in cases where objects have high textural information. However, they do not provide accurate matches for textureless objects, or for objects whose appearance in the image is significantly different from the 3D model. To handle this issue, approaches such as those of Pepik et al. (2012), Zia et al. (2013), and Lim et al. (2013) use a discriminative approach to learn per-point classifiers that generalize across multiple views and appearances of a particular point.

One issue with feature-based approaches is that they depend upon a good viewpoint initialization. If the individual matches between points in the image and points on the 3D model fail to provide sufficient accurate matches, the viewpoint will be...
initialized incorrectly. Exhaustive viewpoint search approaches span the space of viewpoints, scale and location by rendering the 3D model over a large set of views and scales, and comparing each render to the image. They leverage the global similarity of the image with the render from the correct viewpoint. Most viewpoint search approaches leverage discriminative classifiers to obtain high accuracy of rigid viewpoint estimation. As discussed in Section 2.2.3 of Chapter 2, classifiers used include networks of generalized radial basis functions (Poggio and Edelman, 1990), linear mapping (Ullman and Basri, 1991), Ada-Boost trained on shape context descriptors (Stark et al., 2010), linear discriminant analysis (Aubry et al., 2014), and discriminatively trained deformable-parts model (Lim et al., 2014). Most of these approaches address the issue of viewpoint alignment, and use features such as shape context (Andriluka et al., 2009) and histograms of gradients (Dalal and Triggs, 2005) that maintain invariance to low-level information in images. To perform seamless manipulation of objects, our work requires precise alignment of the 3D model to the contours of the object in the photograph, and as such, addresses both rigid alignment of viewpoint and non-rigid deformation to match the photograph. The approach of Xu et al. (2011) provide a semi-automatic exhaustive viewpoint-search approach to obtain viewpoint alignment and deformation, however, (1) they perform alignment and deformation in two separate steps, and (2) they provide photo-inspired deformations to 3D models as opposed to precise matches.

3.1.2 Approaches on Aligning 3D Model Geometry to Depth Images and Videos

Depth and video provide powerful cues to constrain the estimation of 3D structure of objects from images. Several approaches address the issue of recognizing objects from depth or RGB-D images (Hetzel et al., 2001; Golovinskiy et al., 2009; Bo et al., 2011; Xiong et al., 2011; Sochler et al., 2012). Saenko et al. (2011) use depth information from the Kinect sensor to constraint the segmentation of objects, and to impose size priors on objects for pruning false positives. Kim et al. (2013) address the issue of localizing an object in 3D by generating segmentation hypotheses for the object from the depth map of the scene, and combining the hypotheses using a structural SVM to provide the best location in 3D. The approaches of Guo et al. (2013), Zhang et al. (2013), and Lin et al. (2013) predict the dimensions of planar surfaces such as walls and tables in depth images for the purpose of indoor scene understanding. Hinterstoisser et al. (2011) perform realtime 3D object alignment of untextured objects by combining complimentary information from images and dense depth maps. Song et al. (2014) employ exhaustive viewpoint search to render depth maps from the 3D model of an object. They train an Exemplar SVM classifier per rendered depth map, and run the classifier as a sliding window across an input depth image to detect the object in the input depth image.

Several approaches use the constraint of smoothness imposed by videos to align and track 3D models of objects across video frames. One of the earliest systems for
3D model based tracking of object motion has been provided by Gennery (1982), who extrapolated the position and orientation from previous tracking data to hypothesize the next orientation of the object. Gennery used Sobel edges to identify features in the frames, and included filtering in the tracking process to smooth the alignment over frames. Verghese et al. (1990) perform tracking using the assumption that objects move less than one pixel over the video frames. Bray (1990) uses optic flow disparities (Barnard and Thompson, 1979) to estimate the new location of the object using an inversion of the perspective transform (Lowe, 1987a). Lowe (1992) uses an estimation of the standard deviation for tracked features to constrain the search for matching the 3D model to the image. Daucher et al. (1993) use Kalman filtering to predict the pose of the object over new, and re-render the 3D model from the new viewpoint to measure error between edges. Marchand et al. (1999) use a robust approach to estimate object motion over frames with a 2D affine model, and fit the 3D pose of the object to the estimated displacement by iteratively optimizing a non-linear function. Drummond and Cipolla (2002) match edges from renders of the CAD model to frames of the video, and use a Lie group formalism to reformulate tracking as an optimization solved using iterative reweighted least squares. Choi and Christensen (2012) combine detection and tracking within a particle filtering framework on the $SE(3)$ group. Pauwels et al. (2013) combine dense motion and stereo cues with sparse keypoints on graphics processing units to provide real-time object pose estimation and tracking.

3.1.3 Manual Editing of 3D Model Geometry

There exist several approaches to deform the geometry of 3D model meshes for modeling and animation. Most techniques impose constraints on the deformation to preserve properties such as smoothness and articulation. Kobbelt et al. (1998) obtain a hierarchy of fine to coarse representations of a mesh. The user edits the mesh at the coarsest level, and the edits are propagated to finer levels of the mesh. Joshi et al. (2007) represent the mesh of a 3D model in terms of harmonic coordinates of an enclosing cage, which allows the user to manipulate an articulated 3D model using a limited set of controls. Sumner et al. (2005) provide an inverse kinematics-based approach to deform articulated structures as a user edits the end effectors. Yu et al. (2004) provide smooth deformation by manipulating the gradient field over the 3D mesh. Sorkine et al. (2004) maintain smoothness over the mesh by minimizing the Laplacian over the mesh. Sorkine and Alexa (2007) extend the work of Sorkine et al. (2004) by requiring their model to be locally rigid to accurately represent bending of elongated structures. Alternatives to point-to-point editing of meshes include sketch- and contour-based approaches, such as those of Nealen et al. (2005) and Kraevoy et al. (2009).
Figure 3.1: We present an approach to precisely align the 3D model of an object to a single RGB image of the object. We exhaustively span the space of poses of the 3D model by rendering it from several viewpoints and scales. Our approach uses an illumination-invariant multi-level patch-based approach to estimate the best matching viewpoint, scale, and the precise match of patch locations in the best match at various levels. We perform a non-rigid 3D alignment of the 3D model to the matched locations.

3.2 Overview of Automatic Alignment Approach

Early approaches to estimate the 3D alignment of object models to images, such as methods based on aligning low-level primitives (Roberts, 1965; Marr, 1977; Brooks, 1981; Terzopoulos and Metaxas, 1991), or methods based on aligning 3D model meshes (Lowe, 1987a; Huttenlocher and Ullman, 1987) showed success on images that were ‘clean’, i.e., images that had strong gradients, and to objects that were composed of low-level primitives, or whose underlying 3D shapes precisely matched the 3D models. Methods that use representations invariant to low-level features—e.g., histograms of gradients (Dalal and Triggs, 2005—have achieved tremendous success at performing detection or recognition of objects and categories in 2D images (Felzenszwalb and Huttenlocher, 2005; Felzenswalb et al., 2010), obtaining planar 3D representations (Gupta et al., 2010; Hejrati et al., 2012; Fouhey et al., 2014), performing 3D object detection (Aubry et al., 2014), and estimating 3D pose (Lim et al., 2014) from single images in the wild. However, the invariance provided by histogram-based methods to low-level features has, in general, precluded precise alignment.

In this chapter, we present an automated approach to perform precise alignment of 3D models to objects in single images of the objects. Our approach works on both textured objects and textureless objects such as the cup in Figure 3.1. Our main challenge is that the objective function modeling the precise alignment of the 3D model to match the object in the image is highly non-linear (Lowe, 1987a). To estimate the optimally matching 3D model, we need to perform an exhaustive search in the vast space of all possible viewpoints, scales, 3D illumination environments, and 2D locations of all possible points on the 3D model in an image, which turns out to be computationally feasible. As an estimate, to evaluate the goodness of match of 100 points on the 3D model in an image of size $320 \times 240$ by using 324 viewpoints, 16 scales, and 1,000 light sources, with light intensities discretized over 10 values, we would need to evaluate nearly $10^{1492}$ hypotheses.
To tractably perform the non-convex optimization, we make three main contributions. First, we exhaustively span the space of rigid pose alignments of the 3D model to the image by generating discrete synthetic renders of the 3D model at a variety of viewpoints and scales as shown in Figure 3.2. Second, we eliminate the search in the space of illuminations by leveraging an illumination-invariant representation of the renders and the image generated by convolving them with Laplacian of Gaussian kernels. The resulting Laplacian of Gaussian images for the render and the input image are shown in Figures 3.2(b) and 3.2(d). Third, since the viewpoint and geometry of the 3D model in the discrete render may not match that in the image, we perform fast exhaustive search for the precise matches of points in the render to all locations of the image by leveraging a multi-level tree of patches extracted at various resolutions from the illumination-invariant representation of each render. The multi-level tree shown in Figure 3.2(c) allows us to use dynamic programming to perform the simultaneous matching of patches. We match each patch to the image by performing fast normalized cross-correlation of the patch with the illumination-invariant representation of the image.
3.3 Multi-Level Laplacians of Gaussian

As input, our method takes the RGB image $I \in \mathbb{R}^{W \times H \times 3}$ of an object, shown at the top left of Figure 3.2(d) and the 3D model of the object, shown in the center of Figure 3.2(a). Here, $W$ and $H$ represent the width and height of the image, and $3$ represents the number of channels for RGB color space. As shown in Figure 3.2(a), we exhaustively sample the space of rigid poses by rendering the 3D model from a variety of viewpoints $\mathcal{O} = \{\Omega_1, \Omega_2, \cdots, \Omega_{N_\Omega}\}$ and a variety of scales $\mathcal{S} = \{s_1, s_2, \cdots, s_{N_s}\}$. Here, $N_\Omega$ is the number of viewpoints, and $N_s$ is the number of scales. We generate square renders of width $U$, where $U \propto s$, and $s \in \mathcal{S}$. Let $R \in \mathbb{R}^{U \times U \times 3}$ represent the render from viewpoint $\Omega \in \mathcal{O}$ and scale $s \in \mathcal{S}$. For each render $R$, we create a multi-level representation as shown in Figure 3.2(b), by converting $R$ to grayscale and convolving the resulting grayscale image with LoG kernels of decreasing standard deviations $\sigma_1, \sigma_2, \cdots, \sigma_L$ to yield a set of LoG images $\mathcal{R} = \{R_1, R_2, \cdots, R_L\}$. Here $L$ is the number of levels in the multi-level representation. In results shown in Section 3.7, we use $L = 5$, however, for simplicity of explanation we describe our approach using $L = 3$. At the topmost level, i.e., at $R_1$, the LoG contains coarse blobs representing various consistent regions in an image, and provides distinctiveness. At the bottommost level, i.e., at $R_L$, the LoG represents fine-scale details of corners and texture, and offers precision but not distinctiveness. For instance, in Figure 3.2(b), $R_L$ contains several precise edge features as shown by the inset. Section 3.4 describes the multi-level tree that allows us to combine the distinctiveness of higher levels with the precision of lower levels in searching for the best matching render and patch locations. As shown in Figure 3.2(d), we develop a similar multi-level representation for the image $I$, by converting $I$ to grayscale and convolving the resulting grayscale image with LoG kernels to provide a set of LoG images $\mathcal{I} = \{I_1, I_2, \cdots, I_L\}$. We set the standard deviations to decrease by powers of 2, i.e., $\sigma_l = \frac{\sigma_1}{2^{l-1}}$, where $l \in \{1, 2, \cdots, L\}$. We set $\sigma_1 = \frac{U}{16}$.

3.4 Multi-Level Tree for Patch Localization

For each level $l$, where $l \in \{1, 2, \cdots, L\}$, we extract $N_l$ patches $R_i(y_{il}) \in \mathbb{R}^{U_l \times U_l}$ from locations $y_{il} \in \mathbb{R}^2$ in the LoG $R_i$ of the render at level $l$. Here $i \in 1, 2, \cdots, N_l$, and $U_l = 2[3\sigma_l] + 1$. The top row of Figure 3.3 shows patch $R_3(y_{31})$ for location $y_{31}$ at level 3, patch $R_3(y_{32})$ for location $y_{32}$ at level 3, and patches $R_2(y_{21})$ and $R_3(y_{21})$ for location $y_{21}$ at levels 2 and 3. We extract the locations $y_{il}$ by running the interest point extraction algorithm underlying SIFT (Lowe, 2004) on our multi-level Laplacian of Gaussian representation. To extract a large number of interest point locations per level, we set the cornerness threshold for interest point extraction to a low value.

Our objective is to simultaneously estimate the locations of all patches in the multi-level representation of $I$, i.e., in the set $\mathcal{I}$. We represent the location of a patch
\( \mathbf{R}_l(y_u) \) in the LoG of the image \( \mathbf{I}_l \) at level \( l \) as the random variable \( x_{il} \). The sample space of each \( x_{il} \) is the discrete two-dimensional grid \( \mathbb{R}^{WH} \) representing the domain of the image \( \mathbf{I} \). The objective corresponds to computing the maximum of the joint probability \( p(x_{11}, x_{12}, \cdots, x_{N_{ll}}|\mathcal{I}) \) of the locations \( x_{11}, x_{12}, \cdots, x_{N_{ll}} \) given the set \( \mathcal{I} \) of LoGs for \( \mathbf{I} \) i.e.,

\[
\max_{x_{11}, x_{12}, \cdots, x_{N_{ll}}} p(x_{11}, x_{12}, \cdots, x_{N_{ll}}|\mathcal{I}).
\] (3.1)

The optimization of the joint probability in Equation (3.1) is computationally intractable. To determine the best localizations of all patches in the image, we would need to evaluate \((WH)^\sum_{l=1}^{L} N_l\) hypotheses. To tractably search the space of possible patch locations, we construct a multi-level tree by linking each high-level patch to several low-level patches as shown in Figure 3.2(c). Using the tree structure, we use dynamic programming to perform exact inference over the belief network represented by the tree. Consider the tree structure shown in Figure 3.2(c). Using Bayes’ rule and the rules of conditional independence, we express the joint probability \( p(x_{11}, x_{21}, x_{22}, x_{31}, x_{32}, x_{33}, x_{34}|\mathcal{I}) \) as products of conditional probabilities of the locations of lower-level patches given higher-level patches, i.e., as

\[
p(x_{11}, x_{21}, x_{22}, x_{31}, x_{32}, x_{33}, x_{34}|\mathcal{I}) = \\
p(x_{11}|\mathcal{I}) p(x_{21}|x_{11}, \mathcal{I}) p(x_{31}|x_{21}, \mathcal{I}) p(x_{32}|x_{21}, \mathcal{I}) \\
p(x_{22}|x_{11}, \mathcal{I}) p(x_{33}|x_{22}, \mathcal{I}) p(x_{34}|x_{22}, \mathcal{I}).
\] (3.2)

We use the max-product rule to break down the maximization of the probability in Equation (3.2) as

\[
\max_{x_{11}} \left( \max_{x_{21}} p(x_{11}|\mathcal{I}) \max_{x_{31}} p(x_{21}|x_{11}, \mathcal{I}) \max_{x_{32}} p(x_{31}|x_{21}, \mathcal{I}) \right) \\
\max_{x_{22}} p(x_{22}|x_{11}, \mathcal{I}) \max_{x_{33}} p(x_{32}|x_{21}, \mathcal{I}) \max_{x_{34}} p(x_{33}|x_{22}, \mathcal{I}).
\] (3.3)

Equation (3.3) allows us to leverage dynamic programming to perform inference over the tree structure shown in Figure 3.2(c). It should be noted that while the number of hypotheses evaluated is reduced to \((WH)^\sum_{l=1}^{L} N_l\), the inference is exact due to the tree structure.

The remainder of this section describes the conditional probability \( p(x_{21}|x_{11}, \mathcal{I}) \) of the child patch location \( x_{21} \) given its parent \( x_{11} \), however, the description applies to all patch locations. We use Bayes’ rule to express the \( p(x_{21}|x_{11}, \mathcal{I}) \) as

\[
p(x_{21}|x_{11}, \mathcal{I}) = p(\mathcal{I}|x_{21}, x_{11}) p(x_{21}|x_{11})/Z.
\] (3.4)
In Equation (3.4), \( p(x_{21}|x_{11}) \) represents the prior probability of knowing the location \( x_{21} \) of the child patch given the location \( x_{11} \) of its parent, and \( p(I|x_{21}, x_{11}) \) represents the data likelihood. We describe these terms in detail in Sections 3.4.1 and 3.4.2. The term \( Z \) in Equation (3.4) represents the evidence that ordinarily forces \( \sum p(x_{21}|x_{11}, I) \) to equal 1. We obtain numerically stable results by setting \( Z \) to 1, as otherwise it introduces division by small values and yields spurious responses in \( p(x_{21}|x_{11}, I) \).

### 3.4.1 Prior Probability on Location

The term \( p(x_{21}|x_{11}) \) in Equation (3.4) represents the prior probability of knowing the location \( x_{21} \) of the child patch in the image given the location \( x_{11} \) of its parent in the image. We use the 3D model render to enforce the prior that the displacement between \( x_{21} \) and \( x_{11} \) should be similar to the displacement between \( y_{21} \) and \( y_{11} \), where \( y_{21} \) and \( y_{22} \) represent the locations of the child and parent patches in the render. We model the displacement similarity as a Gaussian prior \( p(x_{21}|x_{11}) \), where

\[
p(x_{21}|x_{11}) = T^{-1} \exp \left( -\lambda_{2}^{-2} \|x_{11} + \Delta y_{21}\|^2 \right),
\]

\[
T = \sum_{x_{21}} \exp(-\lambda_{2}^{-2} \|x_{11} + \Delta y_{21}\|^2), \quad \text{and} \quad \Delta y_{21} = y_{21} - y_{11}. \]

The Gaussian prior in Equation (3.5) is similar to the one used by Felzenszwalb and Huttenlocher (2005). In Equation (3.5), the standard deviation \( \lambda_{2} \) (or in general \( \lambda_{l} \)) represents the extent to which the patch can jitter within the LoG \( I_{3} \) (or, in general, in \( I_{l} \)). At the bottommost level, i.e., at level \( L \), we set it to the standard deviation \( \sigma_{L} \) of the LoG kernel at \( L \), to allow the patch to jitter within the region corresponding to the size of its parent patch. For all other levels, \( \lambda_{l} \) is set to 2 to reflect the decrease in patch size by powers of 2.

### 3.4.2 Data Likelihood

In Equation (3.4), \( p(I|x_{21}, x_{11}) \) represents the data likelihood of finding the location \( x_{21} \) of the child patch in the image. We assume that the LoG images are generated by the patches independently, yielding

\[
p(I|x_{21}, x_{11}) = p(I_{1}|x_{21}, x_{11}) \times p(I_{2}|x_{21}, x_{11}) \times p(I_{3}|x_{21}, x_{11}).
\]

We further assume that the likelihood of generating LoGs at the level of child \( x_{21} \) is independent of its parent \( x_{11} \), allowing us to discard the dependence on \( x_{11} \) in Equation (3.6). We also assume that each non-leaf patch location (i.e., \( x_{21}, x_{22}, \) and \( x_{11} \)) depends only on itself and on its children, allowing us to discard the first multiplicative term in Equation (3.6). This assumption simplifies the data likelihood.
CHAPTER 3. ALIGNMENT OF 3D MODEL GEOMETRY

Figure 3.3: Developing the data likelihood $p(I|\mathbf{x}_{21}, \mathbf{x}_{11})$ for the non-leaf patch location $\mathbf{x}_{21}$: We use convolution (represented by $\otimes$) to marginalize out the patch likelihoods $p(I|\mathbf{x}_{31})$ and $p(I|\mathbf{x}_{32})$ of the children $R_3(y_{31})$ and $R_3(y_{32})$ of patch $R_2(y_{21})$ corresponding to $\mathbf{x}_{21}$ with their respective appearance priors $p(\mathbf{x}_{31}|\mathbf{x}_{21})$ and $p(\mathbf{x}_{32}|\mathbf{x}_{21})$. We multiply the resulting marginalizations with the likelihood $p(I|\mathbf{x}_{21})$ of finding $R_2(y_{21})$ within its level $I_2$ to obtain the final likelihood $p(I|\mathbf{x}_{21}, \mathbf{x}_{11})$ for $\mathbf{x}_{21}$ over all levels. Patch likelihoods and appearance priors are obtained by exponentiated normalized cross-correlation (represented by $\odot$).

\[
p(I|\mathbf{x}_{21}, \mathbf{x}_{11}) = p(I_2|\mathbf{x}_{21})p(I_3|\mathbf{x}_{21}). \tag{3.7}
\]

**Likelihood at Level of Patch.** Suppose $I_2(\mathbf{x}_{21})$ represents a patch of size $U_2 \times U_2$ extracted from the LoG of the image $I_2$ at location $\mathbf{x}_{21}$, and $\mathbf{a}$ represents the patch vectorized as a column vector. Let $R_2(y_{21})$ similarly represent the patch of size $U_2 \times U_2$ extracted from the LoG $R_2$ of the render at location $y_{21}$, and let $\mathbf{b}$ represent the patch vectorized as a column vector. We model $p(I_2|\mathbf{x}_{21})$, i.e., the contribution of the patch $R_2(y_{21})$ in generating $I_2(\mathbf{x}_{21})$ as

\[
p(I_2|\mathbf{x}_{21}) = \exp \left( \beta \left| \frac{\mathbf{a} - \mu_a}{\sigma_a} \right| - \frac{\mathbf{b} - \mu_b}{\sigma_b} \right). \tag{3.8}
\]

The map at the end of the center row in Figure 3.3 represents $p(I_2|\mathbf{x}_{21})$. The term in the absolute value in Equation (3.8) is the normalized cross-correlation of the
patch $R_2(y_{21})$ with the patch $I_2(x_{21})$ from the LoG of the image. In Equation (3.8), $\mu_a$ and $\mu_b$ are the mean intensities in the patch $I_2(x_{21})$ and the patch $R_2(y_{21})$ respectively, while $\sigma_a$ and $\sigma_b$ are the standard deviations in their intensities. The normalized cross-correlation provides illumination-invariance between features in the patches that represent similar features in the image. To speed up the matching, we perform fast normalized cross-correlation of $R_2(y_{21})$ with the entire LoG $I_2$ using Fourier transforms and integral sums (Lewis, 1995). We take the absolute value of the normalized cross-correlation to account for contrast reversal. The parameter $\beta$ represents the peakiness of the probability distribution, and is set to 4 for our experiments.

Likelihood for Lower Level. In Theorem 1 in Appendix A, we show that the term $p(I_3|x_{21})$, i.e., the contribution of the patch at position $x_{21}$ in generating the LoG $I_3$ at a level $l = 3$ lower than the level of $x_{21}$ can be modeled by marginalizing the data likelihoods of the children of the patch corresponding $x_{21}$, i.e., we show that

$$p(I_3|x_{21}) = \frac{\sum_{x_{31}} p(I_3|x_{31})p(x_{31}|x_{21})}{\sum_{x_{32}} p(I_3|x_{32})p(x_{32}|x_{21})}^{1/2}.$$  (3.9)

The first two maps in the bottommost row of Figure 3.3 represent the marginalizations in Equation (3.9). Replacing Equation (3.9) into Equation (3.7), we write data likelihood $p(I|x_{21}, x_{11})$ as

$$p(I|x_{21}, x_{11}) = p(I_2|x_{21})\left(\frac{\sum_{x_{31}} p(I_3|x_{31})p(x_{31}|x_{21})}{\sum_{x_{32}} p(I_3|x_{32})p(x_{32}|x_{21})}\right)^{1/2}.$$  (3.10)

The bottom right map in Figure 3.3 represents $p(I|x_{21}, x_{11})$. Equation (3.10) allows our approach to model the data likelihood of a patch as having contributions from itself and from its children, thus strengthening the response of the patch. In this respect, our approach is distinct from Felzenszwalb and Huttenlocher (2005), where the data likelihood of parent parts is modeled distinctly from the children parts. The dependence on child patches in Equation (3.10) is similar to the Hough transform (Ballard, 1987), however, unlike Ballard (1987), we model their contribution in generating the LoG of $I$. To emphasize leaf patches in contributing toward the joint probability in Equation (3.1), we use the square root of the right hand side of Equation (3.10) for non-leaf patches such as $x_{21}$ to represent the data likelihood $p(I|x_{21}, x_{11})$.

The term $p(x_{31}|x_{21})$ in Equation (3.10), shown as the second map in the center row in Figure 3.3 is distinct from the similar term $p(x_{21}|x_{11})$ in Equation (3.4). Here, we use it to model a prior over the expected appearance of the patch at location $x_{21}$ in the image due to its children. The appearance prior represents the notion that on average the response of the image due to a patch, i.e., the response
CHAPTER 3. ALIGNMENT OF 3D MODEL GEOMETRY

represented by \( p(I_3|x_{31}) \), should be similar to the response of the render due to the patch, as a render from the correct viewpoint provides a reasonable initializer for the illumination-invariant appearance of the object. To model \( p(x_{31}|x_{21}) \), we extract patch \( R_3(y_{21}) \in \mathbb{R}^{U_2 \times U_2} \) at the location of the parent \( y_{21} \) from the LoG \( R_3 \). \( R_3(y_{21}) \) is shown as the third and sixth maps in the first row of Figure 3.3. Let \( R_3(y_{31}, y_{21}) \) represent a patch of size \( \mathbb{R}^{U_3 \times U_3} \) extracted from \( R_3(y_{21}) \) at the location of each child patch \( y_{31} \). Also, let \( R_3(y_{31}) \) represent the child patch extracted from \( R_3 \) at location \( y_{31} \). If \( a \) and \( b \) respectively represent \( R_3(y_{31}, y_{21}) \) and \( R_3(y_{31}) \) vectorized, we can write \( p(x_{31}|x_{21}) \) similar to Equation (3.8), i.e., as

\[
p(x_{31}|x_{21}) = Q^{-1} \exp \left( \beta \left| \frac{(a-\mu_a)^T(b-\mu_a)}{U_3^2 \sigma_a \sigma_b} \right| - \beta \right), \tag{3.11}
\]

where \( Q = \sum_{x_{31}} \exp \left( \beta \left| \frac{(a-\mu_a)^T(b-\mu_a)}{U_3^2 \sigma_a \sigma_b} \right| - \beta \right) \). As in Equation (3.8), the term in the absolute value in Equation (3.11) represents a single value in the normalized cross-correlation between the patch \( R_3(y_{31}) \) and the patch \( R_3(y_{21}) \). Since Equation (3.11) does not depend upon the image, we build and store \( p(x_{31}|x_{21}) \) prior to the test phase. We similarly obtain the probability \( p(x_{32}|x_{21}) \), shown by the fourth map in the second row of Figure 3.3, by performing the normalized cross-correlation between \( R_3(y_{32}) \) and \( R_3(y_{21}) \). In practice, we perform the marginalization \( \sum_{x_{32}} p(I_3|x_{32})p(x_{32}|x_{21}) \) by convolving the map of the likelihood \( p(I_3|x_{32}) \) over the entire image with the map of the prior probability in appearance \( p(x_{32}|x_{21}) \) in the Fourier domain.

### 3.5 Coarse-to-Fine Alignment

To provide precise alignment with fast exhaustive search, we provide a two-step coarse-to-fine alignment approach. In the first step, we render the 3D model over 324 orientations (36 samples in azimuth spaced 10° apart, 3 samples in elevation spaced 20° apart, and 3 samples in in-plane rotations spaced 20° apart), and over 16 scales to yield 5184 square template renders of width \( U = 180 \). The \( s \)th scale is set as \((.75 + .05(s - 1))^{-1}\). We use a one-level tree \((L = 1)\) with \( \sigma_1 = 3 \) to select a set of 20 candidates from these 5184 templates. In the second step, we obtain 5 fine samples in viewpoint around the space of each of the 20 candidates, with elevation offsets 5°, 0°, 0°, 0°, and −5°, azimuth offsets 0°, 5°, 0°, −5°, and 0°, and no in-plane offsets. We obtain 5 samples in scale, yielding 500 templates. We use our multi-level alignment approach with \( L = 5 \) to obtain the top five best matching poses together with the precise alignments of the patches from each best matching pose. We use RANSAC to align the rotation \( R \) and its translation \( t \) of the 3D model to the matched image locations. Within each RANSAC iteration, we use the efficient PnP algorithm (Lepetit et al., 2009) with \( n = 4 \) correspondences. We assume focal length is provided by EXIF tags or through camera calibration. To account for
differences between the 3D model geometry from the original image, we provide a final non-rigid deformation of the 3D model to precisely match the point locations by performing the following optimization:

\[
\min_{\Delta X} \sum_{l=1}^{L} \sum_{i=1}^{N_l} \left\| (x_{il}K_3 - K_{1:2}) (RX_{il} + \Delta X_{il} + t) \right\|^2 + \\
\lambda \sum_{j=1}^{N_{\text{model}}} \left\| \Delta X_j - \sum_{k \in N_j} \frac{\Delta X_k}{d_j} \right\|^2 + \gamma \sum_{j=1}^{N_{\text{model}}} \left\| \Delta X_j \right\|^2,
\]  

(3.12)

where \(\Delta X\) is the non-rigid deformation of the 3D model. In Equation (3.12), the first squared term represents the DLT-linearized reconstruction error (Hartley and Zisserman, 2004) of the projection of the 3D model \(X\) oriented using \(R\) and \(t\) and augmented with deformation \(\Delta X\) from the estimated patch locations \(x_{il}\). The second term, similar to the one used in Laplacian surface editing (Sorkine et al., 2004), constrains the 3D model mesh to be smooth by ensuring that all \(k\) vertices in the \(1\)-ring \(N_j\) of the \(j\)th vertex have similar deformation as the \(j\)th vertex. The third term constrains the deformation to be small. \(N_{\text{model}}\) represents the number of points on the 3D model, \(K_{1:2} \in \mathbb{R}^{2 \times 3}\) represents the matrix of the first two rows of the intrinsic parameters matrix, and \(K_3\) is a row-vector corresponding to the third row of the intrinsic parameters matrix. We get the final aligned 3D model \(Y\) by applying the transformation \(R\) and \(t\), and the deformation \(\Delta X\) to the original 3D model \(X\). The \(j\)th vertex of \(Y\) is given by

\[
Y_j = RX_j + t + \Delta X_j.
\]  

(3.13)

### 3.6 Manual Geometry Adjustment

We provide a user-guided approach to provide final corrections to the aligned geometry \(Y\) of the 3D model to match the photographed object, for cases where the automatic method does not provide an exact match. The geometry alignment approach also allows the user to bypass the automatic alignment approach if desired, and manually provide rotation \(R\) and translation \(t\), in which we set the \(j\)th vertex in \(Y\) as \(Y_j = RX_j + t\). As shown in Figure 3.4(a), after the camera is estimated, the user provides a set \(B\) of start points \(y_k \in \mathbb{R}^2, k \in B\) on the projection of the stock model, and a corresponding set of end points \(z_k \in \mathbb{R}^2\) on the photographed object for the purpose of geometry correction. We use a point-to-point correction approach, as opposed to sketch or contour-based approaches (Nealen et al., 2005; Kraevoy et al., 2009), as reliably tracing soft edges can be challenging compared to providing point correspondences. The user only provides the point corrections in 2D. We use them to correct \(Y\) to \(Z\) in 3D by optimizing an objective in \(Z\) consisting
CHAPTER 3. ALIGNMENT OF 3D MODEL GEOMETRY

of a correction term $E_1$, a smoothness prior $E_2$, and a symmetry prior $E_3$:

$$E(Z) = E_1(Z) + E_2(Z) + E_3(Z).$$  \hspace{1cm} (3.14)

The correction term $E_1$ forces projections of the points $Z_k$ to match the user-provided 2D corrections $z_k$, $k \in B$. As shown in Figure 3.4(b), we compute the ray $v_k = K^{-1}[z_k^T 1]^T$ back-projected through each $z_k$. $E_1$ minimizes the sum of distances between each $Z_k$ and the projection $\frac{v_k v_k^T}{\|v_k\|^2} Z_k$ of $Z_k$ onto the ray $v_k$:

$$E_1(Z) = \sum_{k \in B} \left\| Z_k - \frac{v_k v_k^T}{\|v_k\|^2} Z_k \right\|^2_2.$$  \hspace{1cm} (3.15)

Unlike traditional rotoscoping, the correction term encourages the vertex coordinates to match the photograph only after geometric projection into the camera. The corrected vertices $Z_k$ are otherwise free to move along the lines of projection such that the overall deformation energy $E(Z)$ is minimized.

The smoothness prior $E_2$ preserves local smoothness over the corrected model. As shown in Figure 3.4(c), the term ensures that points in the neighborhood of the corrected points $Z_k$ move smoothly. In our work, this term refers to the surface deformation energy from the as-rigid-as-possible framework of Sorkine and Alexa (2007). The framework requires that local deformations within the 1-ring neighborhood $D_i$ of the $i^{th}$ point in the corrected model should have nearly the same rotations $R_i$ as on the original model:

$$E_2(Z) = \sum_{i=1}^{N} \sum_{j \in D_i} \left\| (Z_i - Z_j) - R_i (Y_i - Y_j) \right\|^2_2.$$  \hspace{1cm} (3.16)

The local rotation $\tilde{R}_i$ in the neighborhood of a vertex $Z_i$ is distinct from the global rigid rotation $R$.

The symmetry prior $E_3$ preserves the principal symmetry (or bilateral symmetry) of the model, as shown in Figure 3.4(c). If a point on the original model $X_i$ has a symmetric counterpart $X_{sym(i)}$, $E_3$ ensures that the point $Z_i$ on the deformed 3D model remains symmetric to its symmetric counterpart $Z_{sym(i)}$, i.e., that they are related through a symmetric transform,

$$S = \begin{bmatrix} I_3 - 2n_Z n_Z^T & 2n_Z d_Z \end{bmatrix},$$  \hspace{1cm} (3.17)

where $I_3$ is the $3 \times 3$ identity matrix, and $n_Z$ and $d_Z$ are the normal and distance of the principal plane of symmetry in the corrected model geometry $Z$. $E_3$ is thus
Figure 3.4: Manual geometry adjustment. (a) The user makes a 2D correction by marking a start-end pair, \((y_k, z_k)\) in the photograph. (b) Correction term: The back-projected ray \(v_k\) corresponding to \(z_k\) is shown in black, and the back-projected ray corresponding to \(x_k\) is shown in red. The top inset shows the 3D point \(Y_k\) for \(x_k\) on the original model, and the bottom inset shows its symmetric pair \(Y_{\text{sym}}(k)\). We deform the original model geometry (light grey) to the user-specified correction (dark grey) subject to smoothness and symmetry-preserving priors.

Given as

\[
E_3(Z) = w \sum_{i=1}^{N} \| S[Z^T]_i 1^T - Z_{\text{sym}(i)} \|_2^2 .
\]  

(3.18)

Here, \(w\) is a user-defined weight, that the user sets to 1 if the object has bilateral symmetry, and 0 otherwise. To determine \(X_{\text{sym}(i)}\) for every stock model point \(X_i\), we compute the principal plane \(\pi = [n^T X - d]^T\) on the stock model using RANSAC. At each RANSAC iteration, we randomly choose two points \(X_{i_r}\) and \(X_{j_r}\) on the stock 3D model, and compute their bisector plane \(\pi_r\) with normal \(n_r = \frac{X_{i_r} - X_{j_r}}{||X_{i_r} - X_{j_r}||^2}\), and distance from origin \(\frac{1}{2}n_{r}^T(X_{i_r} + X_{j_r})\). We maintain a score \(n_r\) of the number of points that when reflected across \(\pi_r\) have a symmetric neighbor within a small threshold \(\mu\). After \(R\) iterations, we retain the plane with maximum score as \(\pi\). We then reflect every \(X_i\) across \(\pi\), and obtain \(X_{\text{sym}(i)}\) as the nearest neighbor to the reflection of \(X_i\) across \(\pi\).

The objective function in Equation (3.14) is non-convex. However, note that given the symmetry \(S\) and local rotations \(R_i\), the objective is convex in the geometry \(Z\), and vice versa. We initialize \(R_i = I_3\), where \(I_3\) is the \(3 \times 3\) identity matrix, and \(S\) with the symmetry transform of the original model, \([I_3 - 2ny^Ty^T \quad 2nydy^T]\). We alternately solve for the geometry, and the symmetry and local rotations till convergence to a local minimum. Given \(S\) and \(R_i\), we solve for \(Z\) by setting up
CHAPTER 3. ALIGNMENT OF 3D MODEL GEOMETRY

a system of linear equations. Given $Z$, we solve for $\bar{R}$ through SVD as described by Sorkine and Alexa (2007). To solve for $S$, we assume that the bilateral plane of symmetry passes through the center of mass of the object (which we can assume without loss of generality to be at the origin), so that $dZ = 0$. To obtain $nZ$, we note that the first three columns of $S$ (which we refer to as $S_3$) form an orthogonal matrix, which we extract using SVD, as follows: We create matrices $A = [Z_1, \ldots, Z_N]$ and $B = [Z_{sym(1)} \cdots Z_{sym(N)}]$, perform the SVD decomposition of $AB^T$ as $U\Sigma V^T$, and extract $S_3 = VU^T$. Then, we extract $nZ$ as the principal eigenvector of the matrix $\frac{I_3 - S_3}{2}$. We substitute $nZ$ and $dZ = 0$ in Equation (3.17) to get $S$.

3.7 Results

We present results and evaluations of the automated alignment approach discussed in Sections 3.2 to 3.5, as well as user-provided alignments and comparisons of the manual geometry adjustment approach discussed in Section 3.6.

3.7.1 Results of Automated Geometry Alignment

For the automated alignment approach, we show qualitative results and quantitative analysis on a dataset of small objects such as a wooden bridge, a cup, an IKEA mug, a wooden car, a toy pumpkin, a teapot, a coffee creamer can, and an angry bird stuffed toy. We used the RGB camera of a Microsoft Kinect v1 sensor to capture several images of size $640 \times 480$ for each object from a variety of viewpoints. We downsampled these images to $320 \times 240$. We used Autodesk 123D Catch to build the 3D model for all objects except the IKEA mug by capturing 30 to 50 photographs of resolution $3648 \times 2048$ for each object using a Canon PowerShot SD1200 IS camera. The bridge, car, pumpkin, and bird objects had sufficient fine grained texture to find correspondences within the high resolution photographs in Autodesk 123D Catch. For smooth objects such as the cup and the teapot, we applied markers over sticky tape on the objects, and removed the tape after building the 3D model. For the IKEA mug, we used a 3D model obtained from 3D Warehouse. Figure 3.9(a) shows wireframes of the 3D models. We also show qualitative results of alignments of the alignment of publicly downloaded 3D models of a life-size taxi cab, an apple, and a mango to photographs of these objects. We obtained the 3D models for these objects from TurboSquid.

Figure 3.5 shows the top 3D alignment of the objects to the input images captured using the Kinect camera. The figure shows the original input image, the patches localized in the image at levels 1, 3, and 5 of the 5-level tree, a render of the estimated pose in place of the original object pixels, and an edge representation of the estimated pose render superimposed over the image. The patch at level 1 represents the best matching render. Figure 3.6 shows the top five alignments per input image. Additional alignment results are provided in Figures 3.14 to 3.20, at the
end of this chapter. Tables 3.1 and 3.2 show mean squared rotation and translation errors per object. The second row represents the error of the most accurate match from the top five results. Providing the top 5 matches reduces the error by about 26% on average. We compare our rotation accuracy to the automated approach of Aubry et al. (2014). Their approach provides the nearest matching render, which is used to infer the nearest rotation, however, they do not provide 3D translation. We show lower mean squared errors than Aubry et al. (2014) for all objects except the car and the pumpkin. As shown in Figure 3.9(b), since the car is rectilinear, front and back views of the car share several edge orientations with non-car objects or with side views of the car. We show alignments of the 3D models of the taxi-cab, the apple, and the mango in Figure 3.10. For the apple and the mango, we enforced high weight on non-rigid deformation, while for the taxi cab, we introduced a single scale factor for stretching along the length of the taxi-cab. The top row shows the original 3D models for these objects.

Figure 3.7 shows histograms of the mean squared rotation and translation errors plotted versus azimuth, elevation, and in-plane rotations for each object. With the exception of the car, all objects have low mean translation error in azimuth. The teapot, bridge, and cup show low azimuth error on average. The bridge shows high error for the azimuth that corresponds to a view of a single plane, due to errors in PnP alignment for full planar surfaces. Translation errors versus elevation and azimuth are higher for the creamer as the lack of a distinction between the lid and the jar in the 3D model causes smaller scale renders to align with the lower part of the creamer. The teapot shows high error for lower elevations where the top edge of the knob resembles the lower edge. For most objects, our approach shows low error for in-plane rotations.
Figure 3.5: Alignments of 3D models for images of objects such as a wooden bridge, a cup, an IKEA mug, a teapot, a pumpkin, and a wooden car. The columns correspond to the original image of the object, the patch locations at levels 1, 3, and 5, the estimated pose rendered in place of the original object, and the edges of the rendered estimated pose placed on the original image.
Figure 3.6: Original images with top five matched poses rendered in place of the imaged objects.
CHAPTER 3. ALIGNMENT OF 3D MODEL GEOMETRY

Figure 3.7: Histogram of mean squared error (MSE) in rotation and translation per object versus azimuth, elevation, and in-plane rotations.

Figure 3.8: Alignments of 3D models of mug and angry bird stuff toy.
3.7.2 Evaluation of Automated Alignment Precision

To evaluate the precision of alignment for varying numbers of levels, we selected a set of 324 template images of the angry bird 3D model, rendered using 36 samples in azimuth spaced $10^\circ$ apart, 3 samples in elevation spaced $20^\circ$ apart, and 3 samples in in-plane rotations spaced $20^\circ$ apart, 1 sample in scale. We rendered 3240 ground truth images of the 3D model of the angry bird stuffed toy, using the same translation and same samples in scale, elevation, and in-plane rotations as the template images, and using 360 samples in azimuth spaced $1^\circ$ apart. To test invariance to illumination, we used a different illumination environment to render the ground truth images from the one used to render the template images. We assigned each ground truth rendered image to the nearest two templates from the set of 324 templates, and we applied our multi-level alignment approach with varying numbers of levels (i.e., varying $L$) to obtain the best match from the two templates. We then obtained the errors in the estimated rotation and translation from the ground truth rotation and translation for each choice of $L$. We categorized the ground truth renders according to their deviation in azimuth angle from the best matching template, and we obtained the mean squared errors over each value of deviation.

Figure 3.11 shows plots of the mean squared rotation and translation errors versus deviation in azimuth angle. With the exception of the 1-level tree, increasing $L$ reduces the error at the deviation of $0^\circ$, i.e., when the ground truth render precisely corresponds to a template render. For the 1-level tree, we do a simple assignment of the best matching template rotation and translation to the ground truth render, which introduces 0 rotation error for the ground truth render at $0^\circ$, and zero translation errors for all renders. For $L = 2$ and upwards, we observe small patch misalignments of up to 1 pixel due to similar local edge gradients that introduce small deviations in PnP alignment at the $0^\circ$ deviation mark. For higher deviations from $0^\circ$, increasing $L$ reduces the rotation error significantly. At $L = 5$, the slope
Figure 3.10: 3D model alignments to photographs of a taxi-cab, an apple, and a mango. The top row shows the original publicly available 3D models of these objects. The remaining rows are ranked according to the best matches of viewpoint and scale for the objects.
of the rotation error graph is small. As such, for the second fine matching step in Section 3.5, we use $L = 5$.

### 3.7.3 Evaluation of Manual Geometry Adjustment

We examine the usability of the manual geometry adjustment tool by evaluating the time taken for four users to perform geometry correction of 3D models using the geometry adjustment tool. Figure 3.12 shows examples of the models aligned by four different users to photographs of objects represented by the 3D models. Users aligned 3D models of objects such as a chair, a crane, a taxi, a top hat, a banana, and a mango. For this evaluation, the users directly aligned the stock 3D models as opposed to 3D models automatically corrected by our approach. Table 3.3 shows

![Diagram showing mean squared rotation and translation error versus deviation in azimuth from the best matching render.](image)

**Table 3.3: Times taken (minutes) to align 3D models to photographs.**

<table>
<thead>
<tr>
<th>User</th>
<th>Banana</th>
<th>Mango</th>
<th>Top hat</th>
<th>Taxi</th>
<th>Chair</th>
<th>Crane</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12.43</td>
<td>7.10</td>
<td>8.72</td>
<td>16.09</td>
<td>45.17</td>
<td>32.22</td>
</tr>
<tr>
<td>2</td>
<td>6.33</td>
<td>3.08</td>
<td>10.08</td>
<td>7.75</td>
<td>20.32</td>
<td>14.92</td>
</tr>
<tr>
<td>3</td>
<td>2.23</td>
<td>2.42</td>
<td>4.93</td>
<td>2.57</td>
<td>6.12</td>
<td>22.07</td>
</tr>
<tr>
<td>4</td>
<td>5.63</td>
<td>5.13</td>
<td>6.17</td>
<td>7.52</td>
<td>8.53</td>
<td>19.03</td>
</tr>
</tbody>
</table>
the times taken (in minutes) to align various 3D models to photographs used in this paper. The amount of time spent on aligning is directly related to the complexity of the model in terms of the number of connected components. Users spent the least amount of time on aligning the mango, banana, and top hat models, as these models consist of a single connected component each. Though the taxi consists of twenty-four connected components, its times are comparable to the simpler models, as the stock model geometry is close to the photographed taxi. The long alignment times of the chair and origami crane are due to the number of connected components (three and eight respectively), and the large disparity between their stock models and their photographs. We show examples of user alignments in Figure 3.12.

Figure 3.13 shows results of geometry correction through our user-guided approach compared to the semi-automated approach of Xu et al. (2011) for the objects used in the timing evaluation from Table 3.3. In the system of Xu et al., the user input involves seeding a graph-cut segmentation algorithm with a bounding box, and rigidly aligning the model. Their approach automatically segments the photographed object based on connected components in the model, and deforms the 3D model to resemble the photograph. Their approach approximates the form of most of the objects. However, it fails to exactly match the model to the photograph. Through our manual geometry adjustment approach, shown in the bottom row of Figure 3.13, users accurately correct the geometry to match the photographed objects.
3.8 Summary

In this chapter, we have presented an automated approach to precisely align the 3D model of an object to an image, and a manual geometry correction approach to adjust the 3D model geometry if needed. By providing geometry alignment, we ensure that the contours of the 3D model projected into the image line up with the contours of the object, so that during 3D object manipulations, the contours transition seamlessly from the original photograph. In the next chapter, we use the aligned 3D model to estimate three-dimensional illumination that replicates the shadows and shading found in the original photograph.
CHAPTER 3. ALIGNMENT OF 3D MODEL GEOMETRY

Figure 3.14: Results of 3D model alignment for the bridge.
Figure 3.15: Results of 3D model alignment for the car.
CHAPTER 3. ALIGNMENT OF 3D MODEL GEOMETRY

Figure 3.16: Results of 3D model alignment for the cup.

Figure 3.17: Results of 3D model alignment for the mug.
Figure 3.18: Results of 3D model alignment for the toy.
Figure 3.19: Results of 3D model alignment for the teapot.
Figure 3.20: Results of 3D model alignment for the bird.
Chapter 4

Estimation of Illumination and Visible Appearance

To provide plausible results of three-dimensional object manipulation in a photograph, we need to apply perceptually convincing shadows and shading to the object as it is manipulated. In this chapter, we describe the estimation of three-dimensional illumination given a single photograph of an object and the 3D model of the object aligned to the contours of the object in the photograph. We assume that the 3D model has been aligned to the object in the photograph by a geometry alignment approach such as the automatic and/or manual adjustment approaches discussed in Chapter 3. Section 4.1 provides a background on work done in the area of estimating illumination from images. In Section 4.2, we provide an overview of our approach to simultaneously estimate three-dimensional illumination and reflectance for visible parts of objects in a single photograph, where the 3D models of the objects have been aligned to the photograph. In Section 4.3, we provide the optimization underlying the simultaneous estimation of illumination and reflectance from the image. Section 4.5 provides results of estimated illumination, and compares our illumination estimation approach to existing approaches on determining illumination for a photograph.

4.1 Background

Our work on estimating illumination in three dimensions falls in the category of image-based inverse lighting techniques. The work done on obtaining illumination for the scene underlying an image can be divided into approaches that decompose an image into two intrinsic images—a reflectance image and a shading image—approaches that acquire the three-dimensional illumination directly in the scene using a light probe, and approaches that estimate the three-dimensional illumination from the image. These approaches are detailed in the following subsections.
CHAPTER 4. ESTIMATION OF ILLUMINATION AND VISIBLE APPEARANCE

4.1.1 Intrinsic Images

Barrow and Tenenbaum (1978) introduced the concept of intrinsic images to explain the role of early visual processing in describing a scene. They provided a decomposition of an input photograph into four intrinsic images representing distance (or depth to various pixels in the photograph), reflectance, orientation (or surface normal directions), and illumination (or shadows). Approaches to estimate intrinsic images typically provide two images: one that represents reflectance, and one that represents the shading, i.e., the combination of the distance, orientation, and illumination images of Barrow and Tenenbaum. Without any constraints, the decomposition of a single image into two shading and reflectance images is ill-posed. One group of approaches places constraints over the reflectance image. For instance, the Retinex approach (Land and McCann, 1971) assumes that large changes in gradient arise from changes in reflectance. Bousseau et al. (2009) and Carroll et al. (2011) ask users to mark local regions with similar reflectance, and obtain the reflectance and shading images by incorporating the user-provided constraints within a least-squares system. Other constraints placed on the reflectance image include sparse basis colors (Omer and Williams, 2004; Gehler et al., 2011), similarity of reflectance over distant pixels in repeating textures (Shen et al., 2008; Zhao et al., 2012), and similarity of reflectance in crowd-sourced non-local patches in the image (Bell et al., 2014).

Another group of approaches learns the relationship between local areas in a photograph and corresponding local regions in the shading and reflectance image. Sinha and Adelson (1993) use local gray-level junction analysis to classify observed edges as being due to illumination or due to reflectance changes, and reason about the global consistency of the local hypotheses in conjunction with the 3D object structure and hypothesized illuminant direction. The approaches of Freeman et al. (2000) and Bell and Freeman (2001) learn the relationship between the photograph and its intrinsic images using rendered images as training data. Freeman et al. use a Markov network over patches from the rendered images and the corresponding synthesized shading and reflectance images, while Bell and Freeman label the coefficients of a steerable pyramid representation of the rendered images as being due to shading or due to reflectance. They apply the learnt relationship in test images to estimate shading and reflectance. Tappen et al. (2005) train discriminative classifiers to recognize that color changes indicate reflectance changes, and that shading patterns have a unique appearance that can be discriminated from common reflectance patterns.

4.1.2 Using a Light Probe to Acquire Illumination

While the intrinsic images approach has proven successful for applications such as recoloring and producing consistent interreflections, their 2.5D representation limits their usability to provide lighting under novel views of objects generated by full three-dimensional object manipulation. The category of approaches discussed here and in Section 4.1.3 obtain a three-dimensional representation of the illumination,
usually termed an environment map. The environment map, first introduced by Blinn and Newell (1976), represents the three-dimensional illumination in a scene as a texture map surrounding the object. Environment maps in rendering applications are usually spherical (Blinn and Newell, 1976) or cubical (Greene, 1986), however, there exist approaches to represent environment maps as arbitrary polyhedral meshes (Szeliski and Shum, 1997; Sato et al., 1999).

One method of obtaining the environment illumination is to directly acquire the illumination from the real-world scene underlying the image using “light probes”—devices or techniques to probe the lighting in the scene. Miller and Hoffman (1984) discuss approaches to acquire cubical environment maps by photographing the environment in six directions using a flat field lens with a 90° field of view, and to acquire spherical environment maps by capturing a single photograph of a mirrored sphere. To acquire the full range of illumination intensities, they recommend taking multiple low dynamic range (LDR) images of the scene or sphere using exposure bracketing, registering the LDR images, and combining them into a single high dynamic range (HDR) image.

Approaches to acquire environment maps using multiple photographs include the panorama stitching methods of Chen (1995) and Szeliski and Shum (1997). Similar to Miller and Hoffman (1984), Debevec (1998) captures HDR environment maps by combining several LDR images of a mirrored sphere using the approach of Debevec and Malik (1997). Cameras fitted with fish-eye lenses have been used to capture environment maps of the sky (Greene, 1984; Stumpf et al., 2004) and of surrounding environments (Sato et al., 1999; Knecht et al., 2010). To avoid the use of HDR images in outdoor sunlit scenes, Reinhard et al. (2005) recommend simultaneously photographing a mirrored sphere to capture the ground, sky, and clouds, and a diffuse sphere to indirectly measure sun intensity. Graham et al. (2012) combine the diffuse and mirrored sphere into a single light probe. Calian et al. (2013) bypass the estimation of illumination and the computation of illumination effects by using a 3D printed shading probe to directly capture the shading in the scene. Since these approaches capture illumination at a single point, they work well for distant light sources, however, they may not capture the effect of illuminants closer to the object. To address this, several approaches provide solutions for the light field near a scene using large arrays of mirrored spheres (Unger et al., 2003), HDR video camera systems (Unger et al., 2008), or two mirrored spherical light probes (Corsini et al., 2008).

The approaches discussed in this subsection have been tremendously successful in generating environment maps to the point that several websites provide environment maps for a variety of indoor and outdoor scenes\(^1,2,3\). Environment maps have been instrumental in pushing the boundary of virtual reality applications. However, most

---

\(^1\)http://www.hdrlabs.com/sibl/archive.html  
\(^2\)https://www.hdri-hub.com/free-hdri-environments-for-download  
\(^3\)http://hdrmaps.com
average users taking a photograph with a camera may not have access to equipment such as mirrored spheres, high resolution DSLR cameras with exposure bracketing, or fish eye lenses needed to obtain environment maps for an image. In many cases, for instance, in the case of the historical photograph shown in Figure 4.1, it may be impossible to have access to the original real-world scene. To address this situation, the approaches in the next subsection address the estimation of illumination from the two-dimensional image.

### 4.1.3 Image-Based Estimation of 3D Illumination

When three-dimensional illumination cannot be directly acquired from the 3D scene underlying a 2D image such as the World War II photograph in Figure 4.1, several approaches estimate the 3D illumination from the image. Even when the 3D geometry of the scene has been provided, the estimation of illumination is ill-posed: for instance, a red chair under a white light may appear the same as a white chair under a red light. The ill-posed nature of the problem is similar to the one faced by the intrinsic images approaches discussed in Section 4.1.1. To constrain the problem, most approaches to estimate 3D illumination apply priors to the illumination, and simultaneously solve for the three-dimensional illumination in the scene and the surface reflectance of objects in the scene.

The estimation of illumination from a real image may be traced back to the common illumination approaches of Fournier et al. (1992) and Drettakis et al. (1997), who were attempting to obtain a unified illumination environment to composite computer generated imagery with real images for the purpose of computer augmented reality (hence the term “common illumination”). Assuming that the 3D model geometry has been obtained for the real-world scene, these methods either manually add unseen light sources for the real image or treat objects in the real image as emitters. They update the shading of real objects in the scene with the illumination contribution due to synthetic objects and light sources (for instance, cast shadows of synthetic objects on real surfaces), and vice versa. Karsch et al. provide automatic approaches to estimate the light intensity (Karsch et al., 2011) and three-dimensional light source positions (Karsch et al., 2014) for the purpose of rendering synthetic objects into real images.

Several approaches use an illumination basis to provide constraints on the en-
vironment map. The spherical harmonics basis, an orthogonal basis for the unit sphere (MacRobert, 1948), has been used to represent illumination for fast forward rendering applications due to its ability to express low-frequency environment illumination in terms of a few basis components (Cabral et al., 1987; Sillion et al., 1991; Ramamoorthi and Hanrahan, 2001b; Kautz et al., 2002). As Ramamoorthi and Hanrahan (2001a) demonstrated, the first nine components of the spherical harmonics basis accurately represent the diffuse shading over a convex object such as a sphere. Sato et al. (2003) provided an analytical approach to obtain the number of spherical harmonics required to represent the bidirectional reflectance distribution function of an object. Several approaches have used the first few spherical harmonics to accurately represent diffuse shading over objects with smooth surfaces (e.g., faces) for tasks such as illumination-invariant face recognition (Basri and Jacobs, 2003; Zhang and Samaras, 2003), illumination-invariant object recognition (Osadchy et al., 2003; Shirdhonkar and Jacobs, 2005), dense shape reconstruction of moving objects under arbitrary lighting (Simakov et al., 2003), exposing digital forgeries (Johnson and Farid, 2007; Kee and Farid, 2010), automatic face swaps (Bitouk et al., 2008), and shape, albedo, and illumination from shading (Barron and Malik, 2012a; Barron and Malik, 2012b).

While the spherical harmonics basis accurately produces low-frequency illumination effects such as diffuse shading an object, Ng et al. (2003) demonstrate that the spherical harmonics basis requires on the order of 10,000 components to represent high-frequency illumination effects such as cast shadows and specularities. The number of spherical harmonics needed to estimate high-frequency illumination from an image may exceed the number of pixel-based constraints, thereby potentially underconstraining the estimation of illumination. Ng et al. (2003) demonstrate that a Haar wavelet basis represents high-frequency illumination with fewer (around 200) components due to their ability to provide locality in angular and frequency space. Approaches such as those of Okabe et al. (2004), Haber et al. (2009), and Romeiro et al. (2010) use Haar wavelets to estimate high-frequency illumination. Mei et al. (2009) use a banded approach to represent illumination, where they represent low-frequency illumination using the first nine spherical harmonics coefficients, and high-frequency illumination effects such as cast shadows by constraining the $L_1$ norm of the illumination to be small so as to estimate sparsely distributed point light sources. Several approaches use scene-specific approaches to estimate illumination. The approaches of Sun et al. (2009), Lalonde et al. (2012), and Shan et al. (2013) employ a sun and/or sky model to estimate illumination in outdoor images. Data-driven approaches use a dataset of illumination environments to constrain the estimation of illumination from images of indoor (Karsch et al., 2014) and outdoor (Karsch et al., 2014; Lalonde et al., 2014) scenes.

In this work, we focus on using a generic basis to estimate illumination that plausibly recreates cast shadows and object shading due to area light sources such as fluorescent lights, lamps, and windows typically found in indoor scenes. As shown
in our evaluation in Section 4.5, approaches that use Haar wavelets as a basis are prone to artifacts of negative light, while the $L_1$ prior of Mei et al. (2009) may produce unnaturally sharp cast shadows. To represent high-frequency cast shadows while maintaining local smoothness around a light source, we propose the use of a basis composed of von Mises-Fisher kernels. Approaches such as those of Hara et al. (2008) and Panagopoulos et al. (2009) use von Mises-Fisher kernels to estimate illumination as a mixture of kernels in an expectation-maximization formulation. These methods require the pre-specification of the number of mixtures either manually (Panagopoulos et al., 2009), or by iteratively adding light sources till Williams’ statistical test is satisfied (Hara, et al., 2008). We circumvent the requirement to prespecify the number of von Mises-Fisher kernels by using the von Mises-Fisher basis. In addition, the von Mises-Fisher basis allows us to apply constraints of sparseness and grouping so as to obtain cast shadows generated by area light sources.

### 4.2 Illumination and Appearance Estimation Overview

The input to our approach to estimate illumination and reflectance for visible parts of the object consists of a photograph $I \in \mathbb{R}^{W \times H}$ shown at the top left of Figure 4.2(a), a filled background image, shown at the bottom left Figure 4.2(a), and the 3D model geometry $Z$ aligned to the photograph, as shown at the top right of Figure 4.2(a). Here, $W$ and $H$ are the width and height of the photograph. For the estimation of illumination and reflectance, we assume that the 3D model $Z$ has been pre-aligned to the contours of the object in the photograph using, for instance, the approaches discussed in Chapter 3. In addition, our approach takes a mask image as input that labels the ground and the shadow pixels. Given the aligned 3D model geometry, our approach computes a mask for the object pixels, and augments the user-provided ground and shadow mask with the object mask. The augmented mask is shown at the bottom right of Figure 4.2(a).

Using the provided inputs, our approach estimates 3D illumination $L$ in the form of a spherical environment map shown in Figure 4.2(b), and the appearance $\bar{T}$ for visible object and ground pixels in the form of surface reflectance $\bar{P}$ shown in Figure 4.2(c) and a difference $\delta$ representing fine-scale texture, shown in Figure 4.2(d). Given an arbitrary 3D manipulation of the object, composed of rotations and translations as shown in Figure 4.2(e), our approach recreates plausible shadows and shading shown in Figure 4.2(f) by applying the 3D illumination to the manipulated geometry, renders the object with surface reflectance for originally visible object pixels as shown in Figure 4.2(g), and composites the rendered result with the difference to produce final appearance in originally visible parts of the object that is consistent with the input photograph. The 3D manipulation with the final appearance for the ground and visible parts of the object is shown in Figure 4.2(h).

We represent the photograph $I$ as the sum of a rendered image $H$ and the fine-
scale texture difference $\delta$, i.e., as:

$$ I = H + \delta. \quad (4.1) $$

The rendered image $H$ is obtained by applying a rendering function $f$ to the 3D model geometry $Z$, the surface reflectance $\bar{P}$, and the 3D environment illumination $L$, i.e., as

$$ H = f(Z, \bar{P}, L). \quad (4.2) $$

As discussed in Section 4.3, our approach performs an optimization to estimate 3D illumination $L$ and surface reflectance $P$ which when applied to $Z$ using the rendering function $f$ in Equation (4.2) produces $H$ that best approximates $I$ under a sum-squared difference metric. As mentioned in Section 4.1.3, even with the aligned 3D geometry $Z$, the estimation of illumination and reflectance is ill-posed. We constrain the illumination to produce shadows and shading typical of indoor area lighting by expressing the illumination as a linear combination of a basis of von Mises-Fisher kernels, as shown in Figure 4.3, and by leveraging priors of sparseness and grouping of the basis coefficients. To constrain the estimation of reflectance, we apply a prior
CHAPTER 4. ESTIMATION OF ILLUMINATION AND VISIBLE APPEARANCE

on the reflectance similar to Retinex (Land and McCann, 1971) by enforcing sparsity on the deviation of the estimated reflectance \( \bar{P} \) from the reflectance \( P \) of the original 3D model. Our optimization approach in Section 4.3 provides piecewise-smooth surface reflectance that lacks fine-scale texture detail. As discussed in Section 4.4, we obtain a difference \( \delta \) that represents fine-scale texture detail by subtracting \( H \) from \( I \).

4.3 Optimization Approach for Estimation of 3D Illumination and Surface Reflectance of Visible Pixels

We represent the rendering function \( f \) using the Lambertian reflection model. Under this model, the \( i \)th pixel \( H_i \) of the rendered image \( H \) is generated as

\[
H_i = f_i(Z, \bar{T}, L) = \bar{P}_i \circ \int_\Omega g_i(\omega) L(\omega) d\omega, \tag{4.3}
\]

where \( \bar{P}_i \in \mathbb{R}^3 \) represents the diffuse surface reflectance of the \( i \)th pixel in RGB color space and \( L(\omega) \in \mathbb{R}^3 \) represents the intensity of the light source in RGB color space along the direction of the solid angle \( \omega \). We assume that the light sources lie on a sphere, i.e., that \( L(\omega) \) is a spherical environment map. The symbol \( \circ \) represents the Hadamard (element-wise) product. The quantity \( g_i(\omega) \in \mathbb{R} \) is known a priori, as it depends solely on the 3D geometry of the scene and the 3D point \( Z_i \) back-projected from the \( i \)th pixel location to the scene geometry. Specifically, \( g_i(\omega) \) is given by

\[
g_i(\omega) = n_i \cdot s_i(\omega) v_i(\omega), \tag{4.4}
\]

where \( v_i(\omega) \) is the visibility of the point \( Z_i \) from the light source along direction \( \omega \), \( n_i \) is the surface normal at the point \( Z_i \), and \( s_i(\omega) \) is normalized vector from \( Z_i \) to the light source along solid angle \( \omega \).

To estimate the unknown quantities \( \bar{P} \) and \( L \), we optimize an objective function in \( \bar{P} \) and \( L \), composed of a data term \( F_{data} \), a term \( F_{illum}(L) \) that places priors on the illumination, and a term \( F_{ref} \) that places priors on the surface reflectance.

\[
F(\bar{P}, L) = F_{data}(\bar{P}, L) + F_{illum}(L) + F_{ref}(\bar{P}), \tag{4.5}
\]

The following subsections detail the terms in the objective function \( F \).

4.3.1 Data Term

We use the term \( F_{data} \) in Equation (4.5) to represent the generation of pixels in a single photograph. \( F_{data} \) penalizes the distance between the pixels from the original photograph \( I_i \) and the pixels in the rendered image \( H_i \) under the sum-squared
Environment Map = $\alpha_1 + \alpha_2 + \ldots + \alpha_K$

Figure 4.3: We represent the environment map as a linear combination of the von Mises-Fisher (vMF) basis. We enforce constraints of sparseness and grouping of basis coefficients to mimic area lighting and produce soft cast shadows.

The per-pixel weights $\tau_i$ can be pre-specified to emphasize certain areas of the photograph such as cast shadows. Our approach obtains per-pixel weights $\tau_i$ from the ground and shadow map shown in Figure 4.2(d). Substituting the expression for $H_i$ from Equation (4.3) into Equation (4.6), we obtain

$$F_{\text{data}}(\bar{P}, L) = \sum_{i=1}^{N_{\text{pixels}}} \tau_i \left\| I_i - H_i \right\|_2^2.$$  \hspace{1cm} (4.6)

We represent the illumination $L(\omega)$ as a linear combination of von Mises-Fisher (vMF) kernels $\Gamma(\omega)$, i.e.,

$$L(\omega) = \Gamma(\omega)\alpha.$$  \hspace{1cm} (4.8)

$\Gamma \in \mathbb{R}^{3 \times K}$ is a functional basis, shown in Figure 4.3, and $\alpha \in \mathbb{R}^K$ is a vector of basis coefficients. The vMF kernels provide the advantage of representing high frequency illumination effects over the classical spherical harmonics representation used in Ramamoorthi and Hanrahan (2001a), while avoiding unnaturally sharp cast shadows that arise due to the point light representation used in Mei et al. (2009). In addition, non-negativity constraints on the basis coefficients ensure non-negativity of illumination, in contrast to the Haar wavelet basis used in Ng et al. (2003), Okabe et al. (2004), Haber et al. (2009) and Romeiro and Zickler (2010). The $k$-th component of $\Gamma$ corresponds to the $k$-th vMF kernel, given by

$$h(u(\omega); \mu_k(\omega), \kappa) = \frac{\exp(\kappa \mu_k(\omega)^T u(\omega))}{4\pi \sinh \kappa},$$  \hspace{1cm} (4.9)
CHAPTER 4. ESTIMATION OF ILLUMINATION AND VISIBLE APPEARANCE

where \( \mu_k \) is the \( k \)th mean direction vector, \( u(\omega) \) is a unit vector along direction \( \omega \), and the concentration parameter \( \kappa \) describes the peakiness of the distribution (Fisher, 1953). Replacing the expression for \( L(\omega) \) from Equation (4.8) into Equation (4.7), we obtain

\[
F_{\text{data}}(\overline{P}, L) = \frac{N_{\text{pixels}}}{N_{\text{pixels}}} \sum_{i=1}^{N_{\text{pixels}}} \tau_i \left \| I_i - \overline{P}_i \circ \left( \int_{\Omega} g_i(\omega) \Gamma(\omega) d\omega \right) \alpha \right \|^2_2, (4.10)
\]

\[
= \sum_{i=1}^{N_{\text{pixels}}} \tau_i \left \| I_i - \overline{P}_i \circ \left( \int_{\Omega} g_i(\omega) \Gamma(\omega) d\omega \right) \alpha \right \|^2_2, (4.11)
\]

\[
= \sum_{i=1}^{N_{\text{pixels}}} \tau_i \left \| I_i - \overline{P}_i \circ (D_i \alpha) \right \|^2_2. (4.12)
\]

The matrix \( D_i \in \mathbb{R}^{3 \times K} \) is a function of the aligned 3D model geometry and the vMF basis \( \Gamma \) and can be computed prior to the illumination and appearance estimation. The use of \( \alpha \) in Equation (4.12) converts the optimization in \( \overline{P} \) and \( L \) to one in \( \overline{P} \) and \( \alpha \).

4.3.2 Prior on Illumination Coefficients

We use the term \( F_{\text{illum}} \) to force the algorithm to find a sparse set of light sources using an \( L_1 \) prior on the illumination coefficients \( \alpha \) from Equation (4.12). In addition, according to the elastic net framework (Zhou and Hastie, 2005), we place an \( L_2 \) prior to force groups of correlated coefficients to be turned on. The \( L_2 \) prior forces spatially adjacent light sources to be switched on simultaneously to represent illumination sources such as area lights or windows. We thus obtain the following form for \( F_{\text{illum}} \):

\[
F_{\text{illum}}(L) = \lambda_1 \left \| \alpha \right \|_1 + \lambda_2 \left \| \alpha \right \|^2_2. (4.13)
\]

4.3.3 Prior on Reflectance

The prior \( F_{\text{illum}} \) on the illumination does not resolve the ambiguity between illumination and reflectance. To address this ambiguity, we use the reflectance \( \overline{P} \) from the texture map of the stock 3D model to constrain the estimation of the reflectance from the photograph. We specifically constrain the deviation of the reflectance \( \overline{P} \) from the original stock model reflectance \( P \) using a piecewise constancy prior, \( F_{\text{ref}} \), given as:

\[
F_{\text{ref}}(\overline{P}) = \lambda_{\text{ref}} \sum_{i=1}^{N^s} \sum_{j \in \mathcal{N}_i} \left \| (\overline{P}_i - P_i) - (\overline{P}_j - P_j) \right \|_1. (4.14)
\]
The prior $F_{\text{ref}}$ is related to color constancy assumptions made in approaches to perform retinex (Land and McCann, 1971). $N_i$ represents the 4-neighborhood of the $i^{th}$ pixel in image space.

4.3.4 Optimization

We optimize the objective $F$ in Equation (4.5) subject to positivity constraints on $\alpha$:

$$\{\hat{P}^*, \alpha^*\} = \arg \min_{\hat{P}, L} F(\hat{P}, \alpha), \text{ s.t. } \alpha \geq 0, L^* = H \alpha^*.$$

(4.15)

The above optimization is non-convex due to the bilinear interaction of the surface reflectance $\hat{P}$ with the illumination $L$. If we know the reflectance, we can solve a convex optimization for the illumination, and vice versa. We bootstrap the estimation using the appearance of the stock model. To do so, we initialize the reflectances with the stock model reflectance $P$ for the object. We alternately solve for illumination and reflectance until convergence to a local minimum. With the positivity constraints on the coefficients, the optimization of $L$ turns out to be a quadratic programming problem which is solved efficiently using the MOSEK optimization toolkit. We use the SeDuMi package to optimize the reflectances. To represent the vMF kernels and $L$, we discretize the unit sphere into $K$ directions, and compute $K$ kernels, one per direction. Our approach simultaneously solves for a single RGB reflectance for every contact surface such as the ground plane. We initialize the reflectance of contact surfaces as the median value of ground plane pixels in the photograph, specified using the user-defined mask shown in Figure 4.2(d).

4.4 Fine-scale Texture Difference

The constraints on the optimization, in particular, the reflectance prior $F_{\text{ref}}$, induce the optimization to yield smooth reflectance that does not contain details of fine-scale texture such as upholstery patterns, graininess, and discolorations due to dirt and wear-and-tear. Even if we were to eliminate the reflectance prior, the Lambertian reflectance model itself does not capture the true BRDF of the object. Without any constraining model, the optimization of the true BRDF at every point on an object is highly unconstrained. We therefore use the Lambertian reflection model to approximate the BRDF of the object, and we model details of fine-scale texture in a difference $\delta$.

The use of the fine-scale texture difference in our approach is inspired by approaches that perform differential rendering for inserting synthetic objects into real photographs (Debevec, 1998; Karsch et al., 2011; Karsch et al., 2014). To render illumination effects of inserted synthetic objects such as cast shadows on real objects in a photograph, such approaches require a description of the actual 3D scene under-
CHAPTER 4. ESTIMATION OF ILLUMINATION AND VISIBLE APPEARANCE

Figure 4.4: Original photograph, illumination for original pose of object, and illumination for 3D manipulation of object for a chair, a car on a cliff, a crane, a taxi-cab, and a top hat.

lying the photograph. Since a precise description of the scene geometry and BRDF of every possible surface in the photograph can be time-consuming and often infeasible, these approaches provide a good approximation to the 3D model geometry, appearance, and illumination of the scene, and maintain a difference between the original photograph and a render of the 3D scene from the perspective of the image to represent fine-scale details of the image. These approaches render illumination effects for synthetic objects by inserting them into the 3D scene approximation, and add on the difference of fine-scale details to obtain a result that appears perceptually convincing. Our approach focuses on obtaining perceptually convincing illumination effects of real objects already present in the photograph on other objects or surfaces in the photograph. Hence, we represent the difference not only for the background in the original photograph, but also for the manipulated object.

For the $i$th pixel, we compute the texture difference $\delta_i \in \mathbb{R}^3$ as the residual between the RGB pixel value in the original photograph $I_i$ and the RGB pixel value $H^*_i$ in the render $H$ synthesized by inserting the estimated values of illumination $L^*$ and reflectance $P^*$ into the Lambertian reflection model in Equation (4.3), i.e.,

$$\delta_i^* = I_i - H_i^* = I_i - \bar{P}_i^* \int_\Omega g_i(\omega) \circ L^*(\omega)d\omega.$$  

(4.16)

4.5 Results and Evaluation

Figure 4.4 shows results of the estimated illumination in photographs of five objects: the chair from Figure 4.2, a car standing on a cliff, an origami crane, the taxi-cab from
RESULTS AND EVALUATION

Figure 4.5: Plots of mean-squared reconstruction error (MSE) versus number of basis components for the vMF basis in green (method used in this paper) compared to the Haar basis (Haber et al., 2009), spherical harmonics with $L_1$ prior (Sph+L1, Mei et al., 2009), background image projected (Bkgnd, Khan et al., 2006), and a light probe, on the object and the ground, ground only, and object only.

Chapter 1, and a top hat. The top row shows the original photograph, the center row shows the illumination over the object and contact surfaces for the original pose of the object, and the bottom row shows the illumination for a 3D manipulation of the object, such as rotation, translation and/or non-rigid deformation. For all objects in Figure 4.4, 3D geometry alignment was performed manually. We estimated gray illumination for all objects except the crane. The illumination on the crane tends toward blue due to the blue tinge on the illumination-dependent appearance of the crane in the original photograph. As shown by results and user study in Chapter 6, the blue illumination has limited influence on perception of the end result once the appearance and texture difference is applied. As shown in Figure 4.4, we accurately represent soft illumination and cast shadows. We obtain plausible cast shadows for outdoor scenes such as those of the car on the cliff and the taxi-cab, by forcing the sparseness weight $\lambda_1$ to take on a high value.

To evaluate the illumination, we captured fifteen ground truth photographs of a red IKEA chair in various orientations and locations using a Canon EOS 5D Mark II Digital SLR camera mounted on a tripod and fitted with an aspheric lens. We also captured a photograph of the scene without the object to provide a ground-truth background image. We aligned the 3D model of the chair to each of the fifteen photographs using our manual geometry adjustment approach from Chapter 3. We evaluated our illumination and reflectance estimation approach against a ground truth light probe, and three approaches: (1) the approach of Haber et al. (2009) that models illumination using Haar wavelets, with positivity constraints on the Haar coefficients and regularization of the Laplacian of the illumination (2) the approach of Mei et al. (2009) that models illumination as a sum of low frequency illumination represented using the first 9 spherical harmonic components, and $L_1$-sparse high frequency illumination, and (3) the approach of Khan et al. (2006) that projects out the background of the photograph as an environment map. We fill the parts of the Khan et al. environment map not seen in the image using the PatchMatch algorithm (Barnes et al., 2009).

For all experiments, we estimated the illumination using Photograph 1 shown
CHAPTER 4. ESTIMATION OF ILLUMINATION AND VISIBLE APPEARANCE

Figure 4.6: [Figure is best viewed on a monitor] Comparison of von Mises-Fisher (vMF) basis for illumination estimation versus the Haar basis (Haber et al., 2009), spherical harmonics with $L_1$-prior on high frequency illumination (SpH+L1, Mei et al., 2009), background image backprojected (Bkgnd, Khan et al., 2006), and a light probe. Top row: original photograph and environment maps for $K = 4096$. Bottom two rows: ground truth and synthesized images for a 3D manipulation on the chair. Our approach produces soft cast shadows, avoids highlights, and captures the effect of lights not seen in the original photograph.

at the left of Figure 4.6(a), and we synthesized images corresponding to the poses of the chair in Photographs 2 to 15 by applying the estimated illumination to the geometry of each pose. The synthesized images contain all steps of our approach except the application of the fine-scale texture difference. We computed mean-squared reconstruction errors between Photographs 2 to 15 and their corresponding synthesized images for three types of subregions in the photograph: object and
Figure 4.7: Environment maps obtained using Haber et al. (2009), Mei et al. (2009, SpH+L1), and our approach (vMF) for various values of the number of components, $K$. As $K$ increases, the environment map using vMFs approximates the light sources in the scene underlying the photograph.

ground, ground only, and object only. As a baseline, we obtained the ground truth illumination by capturing an HDR image of a 4-inch chrome sphere as a light probe. Figure 4.5 shows the mean-squared reconstruction error (MSE) for the von Mises-Fisher basis compared against Haber et al. (2009), Mei et al. (2009), Khan et al. (2006), and the light probe, for increasing numbers of basis components (i.e. $K$). We used the same discretization for the sphere as the number of components. The number of basis components is a power of two to ensure that the Haar wavelet basis is orthonormal. We used $\lambda_1 = .01$ and $\lambda_2 = 1$ for our method, a regularization
weight of 1 for the approach of Haber et al. (2009), and a regularization weight of .01 for the method of Mei et al. (2009). The reconstruction error for Haar wavelets increases with increasing number of basis components, since in attempting to capture high frequency information (such as the edges of shadows), they introduce artifacts of negative illumination such as highlights that contribute to the error according to the $L_2$ metric.

Figure 4.6(a) shows environment maps for the light probe, the background image projected out according to Khan et al. (2006), and the three approaches to estimate illumination. Figure 4.6(b) shows images of the chair in Photograph 2 synthesized using the light probe, the background image, and the three illumination estimation approaches with $K = 4096$. The ground truth for Photograph 2 is shown at the top-left of Figure 4.6(b). As shown in the figure, the approach of projecting the background using Khan et al. (2006) does not capture the influence of ceiling lights which are not observed in the original photograph. The Haar wavelets’ approach introduces highlights due to negative light, while the approach of Mei et al. (2009) generates sharp cast shadows. Our approach produces smooth cast shadows and captures the effect of lights not seen in the original photograph. Figure 4.7 shows environment maps for varying values of $K$ for the approaches of Haber et al. (2009), Mei et al. (2009), and for our approach.

4.6 Summary

This chapter presents our approach to estimate three-dimensional illumination from a single photograph of an object, assuming that the 3D model of the object has been aligned to the photograph. Our approach employs priors of sparseness and grouping on coefficients of the von Mises-Fisher basis to obtain illumination that represents area lighting found in typical indoor scenes. As shown by the evaluation in Figure 4.6, our approach provides cast shadows with smooth edges, and smooth shading on the object, consistent with human interpretation of indoor scenes. As shown by the 3D manipulation of the taxi cab photograph in Chapter 1, and the photograph of the car on the cliff in Chapter 6, the approach can provide plausible illumination for outdoor scenes. The illumination estimation process simultaneously provides an estimate of appearance in terms of the surface reflectance and the fine-scale texture difference for visible parts of the object. The next chapter discusses the completion of the appearance over novel parts of the object exposed during 3D object manipulation.
In Chapters 3 and 4, we provided algorithms that aligned the geometry of the 3D model to the object in the photograph, and that estimated three-dimensional illumination to create plausible shadows and shading. These approaches provide light sources and geometry of object parts that may not be seen in the original photograph. However, while Chapter 4 provides the illumination-free appearance of the object in visible parts, we still need to handle the appearance of objects in parts of the object hidden from the viewpoint of the camera. In this chapter, we discuss the completion of appearance to novel parts of the object for full 3D manipulation of the object.

To complete the appearance, we hearken back to the Gestalt principles of symmetry, smoothness, and planarity discussed in Chapter 2. Most objects show several types of symmetries, such as bilateral symmetries of humans, vertebrates, chairs, vehicle bodies, and leaves, radial symmetries of flowers, sea anemones, and snowflakes, and approximate symmetries of fruit, plants, and trees (Blum, 1973; Stewart, 2001). Humans are fairly good at detecting exact and approximate symmetries (Barlow and Reeves, 1979) and using detected symmetries to infer hidden parts of objects (Pizlo, 2008). Given the left side of the chair in Figure 5.1, we recognize that the right side of the chair, though not seen in the viewpoint of the camera, must be symmetric in geometry and appearance to the left side of the chair. While the back side of the chair is not precisely symmetric to the chair parts visible in the image, it is approximately symmetric to the front of the chair on account of its near planarity.

In this chapter, we provide an approach to

![Figure 5.1: The left visible side of the chair is exactly symmetric to its right side hidden in the first photograph, while the front of the chair is approximately symmetric to the back.](image-url)
complete the appearance over an object in three dimensions by using exact and approximate symmetries obtained using the publicly available 3D model of the object. Our approach uses inference over a probabilistic graphical model—in particular, a Markov random field (MRF)—to complete the appearance. Section 5.2 discusses using points on the 3D model to set up the structure of the graph. In Section 5.3, we provide a method to relate visible parts of the object to hidden parts using multiple planes of symmetry estimated over the 3D model by a RANSAC-based approach. The symmetry detection approach simultaneously covers the object with multiple candidates for appearance per 3D model point by reflecting appearance from visible parts of the object to hidden parts. In Section 5.4, we provide a method that infers the best assignment of appearance candidate per 3D model point by enforcing seam smoothness and texture consistency over the Markov random field. Our inference approach ensures that object vertices visible from the viewpoint of the camera retain the appearance from the original photograph. For hidden parts of the object that are not symmetric in geometry and appearance to visible parts of the object, our appearance completion approach automatically defaults to the appearance in the publicly available 3D model to provide full completion of appearance over the object.

5.1 Background

Our work on completing appearance in novel parts of the object is related to approaches that synthesize textures in images. We detail these approaches in Section 5.1.1. Our approach to relate hidden and visible parts using symmetries is related to symmetry estimation approaches detailed in Section 5.1.2.

5.1.1 Texture Synthesis and Mapping

To plausibly reveal the hidden parts of objects during their manipulation, we need to complete the appearance of hidden areas, which can be achieved by completing a texture map. There exists much work on synthesizing textures in two-dimensional images. (Heeger and Bergen, 1995) did early work in texture synthesis by forcing the histograms of a noise image to match those of a texture image. These histograms were derived from steerable pyramid representations of the images. Non-parametric sampling methods have been more commonly used in texture synthesis (Efros and Leung, 1999; Wei and Levoy, 2000; Efros and Freeman, 2001). Efros and Freeman (2001) use dynamic programming to synthesize texture using blocks that match existing texture within a boundary. Kwatra et al. (2003) and Agarwala et al. (2004) use graph-cuts to grow texture along arbitrary outlines. Barnes et al. (2009) accelerate texture synthesis through a randomized algorithm to compute the approximate nearest neighbor field. Herzmann et al. (2001) direct the transfer of texture using analogies across matching parts of a guiding image. Such approaches cannot directly
be used to synthesize texture on a texture map of a 3D model, as they do not incorporate neighborhood constraints arising due to the geometry in three dimensions. The approach of Huang et al. (2014) uses a three-dimensional constraint in the form of planar projection to inpaint novel appearance in images. Their planar constraint works for a variety of images such as those with building façades and train tracks, however, it does not work for objects with more complex geometries.

To represent texture in three dimensions, there exist approaches that directly map the texture from images onto the 3D model through its texture map (Levy et al., 2001; Kraevoy et al. 2003; Tzur and Tal, 2009). These approaches map image pixels without removing the effect of appearance. In addition, these methods do not provide completion of appearance for object parts hidden from the viewpoint of the camera. Several approaches directly synthesize the texture on the 3D mesh. Turk (2001) uses a hierarchy of meshes together with a user-defined vector field to guide texture growth. Other approaches that grow texture on surfaces include those of Tong et al. (2002) and Liu et al. (2004). Kopf et al. (2007) extend 2D texture synthesis to synthesize volumetric textures from 2D exemplars. These methods grow texture globally, either on the surface or within a volume. Due to their lack of knowledge of the distinction between local parts of the object, they cannot be used to transfer texture from object parts visible in the viewpoint of the camera to hidden parts of the object.

5.1.2 Computation of Symmetry

We leverage symmetries to complete appearance from visible areas of an object in a photograph to symmetric hidden areas. There are several approaches to extract symmetries from images and 3D models. Kanade and Kender (1980) demonstrated that symmetry in 3D models corresponds to skew symmetry in images. Since perfect symmetries have limited practical use for natural objects, such as fruit or animals, and flexible or damaged objects, most approaches focus on computing approximate symmetries. Gal and Cohen-Or (2006) determine salient geometric features based on partial symmetries in an object for shape matching. Mitra et al. (2006) detect partial and approximate symmetries by regarding them as invariances under rotations, translations, reflections, and uniform scaling. Lee and Liu (2012) estimate curved glide-reflection symmetry from 2D images of objects such as leaves and insects that have a curved intrinsic symmetry axis. Xu et al. (2009) intrinsic reflectional symmetries for objects whose parts have non-planar symmetry manifolds.

Several of these approaches extract and use symmetries for the purpose of geometry estimation. Many of the approaches on using generalized cylinders as primitives discussed in Chapter 2 use symmetries as a prior to fit generalized cylinders to objects (Horaud and Brady, 1987; Ulupinar and Nevatia, 1988, 1995; Sato and Binford, 1993; Sayd et al., 1996; Terzopoulos et al., 1988). Pauly et al. (2005) use symmetries together with a large collection of 3D models to complete 3D scans of objects. Mitra and Pauly (2008) use detected symmetries to determine regular structures, and to
extend or modify geometry in architectural models. Bokeloh et al. (2011) use symmetries to provide structure-aware deformation of objects. Work has also been done in exploiting symmetry to infer missing appearance in images (Kim et al., 2012).

In general, these approaches are mutually exclusive: approaches focused on symmetries from geometry do not respect appearance constraints, and vice versa. In this thesis, we extract symmetric relationships using an intersection of the two criteria of geometric symmetry and appearance similarity. This prevents hallucination of appearance between geometrically similar but visually distinct parts, such as the planar underside and top of a taxi-cab, or between visually similar but geometrically distinct parts such as the curved surface of a top-hat and its flat brim.

5.2 Setting up Graph Structure over 3D Model

For the rest of the discussion in this chapter, we will assume that we have the 3D model of the object $Z$ aligned to its pixels in the photograph, and the illumination-independent appearance for visible pixels of the object in the input photograph. The 3D model may be aligned using the approach discussed in Chapter 3. The illumination-independent appearance may be obtained from an input photograph using the simultaneous illumination and appearance discussed in Chapter 4. In this case, the appearance consists of the estimated surface reflectance $\bar{P}^*$ from Equation (4.15) in Section 4.3 and fine-scale texture difference $\delta^*$ from Equation (4.16) in Section 4.4.

The surface reflectance $\bar{P}^*$ and texture difference $\delta^*$ are provided for the pixels of the object in the photograph. Our aim is to populate the texture map associated with the stock 3D model $X$ by leveraging symmetries computed using the 3D geometry and appearance of the stock 3D model. We first compute the 3D model point location $X_s$ for every texel location $u_s$ in the texture map by determining barycentric coordinates for $u_s$ within the texture map mesh and applying the barycentric coordinates to the stock 3D model mesh. Each $u_s \in \mathbb{R}^2$ represents the $u$- and $v$-coordinates into the texture map. The index $s$ belongs to an index set $\mathcal{I}$, where $\mathcal{I} = \{1, 2, \cdots, N_u N_v\}$ represents indices of all texel locations in the texture map, and $N_u$ and $N_v$ represent the width and height of the texture map. We also maintain a set of points $Z_s$ on the aligned 3D model $Z$ for every texel location $u_s$ and stock 3D model point $X_s$ by applying its barycentric coordinates to the aligned 3D model mesh.

We use a Markov random field (MRF) graphical model to perform appearance completion. The MRF graph consists of $|\mathcal{I}|$ vertices. The $s^{th}$ vertex on the graph corresponds to the $s^{th}$ texel location and the point $X_s$ on the stock 3D model, where $s \in \mathcal{I}$. Edges in the graph are obtained by connecting each $X_s$ to the $K_s$ of the $k$ nearest neighbors of $X_s$. 

80
5.3 Relating Hidden and Visible Parts via Symmetries

For every vertex on the MRF graph, we assign multiple candidates of appearance by reflecting the appearance from the parts of the object visible in the photograph to hidden parts across multiple planes of symmetries. Each reflection of appearance across a plane of symmetry creates “layers” of appearance candidates. The $s^{th}$ vertex on the MRF receives $L_s$ layers of appearance candidates, $\tilde{P}_{is}$ and $\tilde{\delta}_{is}$, where $i_s \in \{1, 2, \cdots, L\}$ and $s \in I$. We leverage regularities such as symmetry and planarity in the geometry and appearance of the stock 3D model to estimate planes of symmetry that relate visible parts of the object to hidden parts. The first layer of appearance candidates consists solely of the appearance visible in the original photograph assigned to the set of texels $I_v$ corresponding to visible parts of the object.

We use the aligned 3D model $Z$ to determine the set of aligned 3D points $Z_s, s \in I_v$ that are visible from the viewpoint of the camera. Using the correspondence across $Z_s, u_s,$ and $X_s$, we now have a set of points $X_s, s \in I_v$ on the original stock 3D model representing points visible from the viewpoint of the camera, and $X_s, s \in I_h$ representing points hidden from the viewpoint of the camera. Here $I_h = I \setminus I_v$. We then set the first layer of appearance candidates $\tilde{P}_{is}$ and $\tilde{\delta}_{is}$ for $i_s = 1$ and $s \in I_v$ to the visible appearance $\bar{P}_s$ and $\bar{\delta}_s$ obtained by backprojecting the appearance of visible pixels to the texture map.

While it is possible to pre-compute planes of symmetry on an object, our objective is to compute symmetries relating visible parts of an object to hidden parts. Many of these symmetries turn out to be approximate. For instance, different parts of a banana with approximately similar curvature are identified as symmetries using our approach. Pre-computing all such possible symmetries is computationally prohibitive. To feasibly cover the object, we use a greedy approach to compute symmetries, detailed in Algorithm 1. At the $m^{th}$ iteration, Algorithm 1 performs three steps. Step 1 uses RANSAC to estimate the optimal plane of symmetry $\pi_m$ separating the visible set $I_v$ from the hidden set $I_h$. Step 2 reflects all prior layers of appearance across $\pi_m$. Step 3 updates the visible set $I_v$ to include the vertices related by plane $\pi_m$. The $m^{th}$ iteration generates $2^{m-1}$ new layers by reflecting all prior $2^{m-1}$ layers across $\pi_m$. In all, the algorithm performs $M$ iterations, and generates $L = 2^M$ layers of appearance candidates. The total number of iterations $M$ is specified by the user. The algorithm uses the criteria of geometric symmetry captured by the distance $\|X_s - 2n_m n_m^T X_t + 2n_m d_m\|$ in vertex space and appearance similarity represented by the distance $\|P_s - P_t\|$ in the original stock model surface reflectance to compute $\pi_m$ and to transfer appearance candidates. For vertices for which no appearance candidates can be generated using geometric symmetry and appearance similarity, as in the case of the underside of the taxi cab shown in Chapter 1, the algorithm defaults to the stock model appearance $P$. For such vertices, the fine-scale texture difference is set to 0.

Figure 5.2 shows an example of the algorithm in action. Planes $\pi_1$ and $\pi_2$...
Algorithm 1 Associating Appearance Candidates Through Layers.

Set user-defined values for $\mu$, $\nu$, and $M$.
$\forall i_s \in \{1, 2, \cdots, 2^M\}$, $\forall s \in \mathcal{I}$, set $\tilde{P}_{i_s} \leftarrow \infty$ & $\tilde{\delta}_{i_s} \leftarrow \infty$.
$\forall s \in \mathcal{I}_v$, if $i_s = 1$, set $\tilde{P}_{i_s} \leftarrow \infty$ and $\tilde{\delta}_{i_s} \leftarrow \infty$.

for $m = 1$ to $M$ do
  Step 1: Perform RANSAC for optimal plane of symmetry $\pi_m$:
  Initialize $n_m \leftarrow 0$, $\mathcal{I}_m \leftarrow \emptyset$, $n_m \leftarrow [0 0 0]^T$, and $d_m \leftarrow 0$.
  for $r = 1$ to $R$ do
    Randomly select $s_r \in \mathcal{I}$ and $t_r \in \mathcal{I}_h$.
    Compute bisector plane $\pi_r = [n_r^T - d_r]^T$, where
    $n_r \leftarrow \frac{X_{s_r} - X_{t_r}}{\|X_{s_r} - X_{t_r}\|}$ and $d_r \leftarrow \frac{1}{2} n_r^T (X_{s_r} + X_{t_r})$.
    Initialize $\mathcal{I}_r = \emptyset$.
    for $s \in \mathcal{I}_h$ do
      Obtain $t^* = \arg \min_{t \in \mathcal{I}_v} \|X_s - 2n_r n_r^T X_t + 2n_r d_r\|$.
      if $\|P_s - P_{t^*}\| < \nu$ and $\|X_s - 2n_r n_r^T X_{t^*} + 2n_r d_r\| < \mu$ then
        Add $s$ to $\mathcal{I}_r$.
      end if
    end for
    if $|\mathcal{I}_r| > n_m$ then
      $n_m \leftarrow |\mathcal{I}_r|$, $\mathcal{I}_m \leftarrow \mathcal{I}_r$, $n_m \leftarrow n_r$, and $d_m \leftarrow d_r$.
    end if
  end for
  Optimal plane of symmetry $\pi_m = [n_m^T - d_m]^T$.
  Step 2: Reflect appearance candidates across $\pi_m$:
  for $s \in \mathcal{I}$ do
    for $i_s = 1$ to $2^{m-1}$ do
      Set $j_s = i_s + 2^{m-1}$.
      Obtain $t^* = \arg \min_{t \in \mathcal{I}_v} \|X_s - 2n_m n_m^T X_t + 2n_m d_m\|$.
      Set $\tilde{P}_{j_s} \leftarrow \tilde{P}_{i_s}$ and $\tilde{P}_{j_s} \leftarrow \tilde{P}_{i_s}$, where
      $\|P_s - P_{t^*}\| < \nu$, $\|X_s - 2n_m n_m^T X_{t^*} + 2n_m d_m\| < \mu$.
    end for
  end for
  Step 3: Update $\mathcal{I}_v \leftarrow \mathcal{I}_v \cup \mathcal{I}_m$ and $\mathcal{I}_h \leftarrow \mathcal{I}_h \setminus \mathcal{I}_m$.
  end for
  if $|\mathcal{I}_h| > 0$ then
    $\forall s \in \mathcal{I}_h$, set $\tilde{P}_{i_s} \leftarrow P_s$ and $\tilde{\delta}_{i_s} \leftarrow 0$.
  end if

82
Figure 5.2: We build an MRF over the object model to complete appearance. (a) Due to the camera viewpoint, the vertices are partitioned into a visible set $I_v$ shown with the visible appearance, and a hidden set $I_h$ shown in green. Initially, the graph has a single layer of appearance candidates, labeled Layer 1, corresponding to visible parts. At the first iteration, we use the bilateral plane of symmetry $\pi_1$ to transfer appearance candidates from Layer 1 to Layer 2. At the second iteration, we use an alternate plane of symmetry $\pi_2$ to transfer appearance candidates (c) from Layer 1 to Layer 3, and (d) from Layer 2 to Layer 4. We perform inference over an MRF to find the best assignment of appearance candidates from several layers to each vertex. This result was obtained after six iterations.

computed for iterations $m = 1$ and $m = 2$ are shown in Figures 5.2(a) and 5.2(b). Layer 1 in Figure 5.2(a) is created by using the illumination-independent appearance of the visible pixels from the original photograph. At iteration $m = 1$, Layer 2 in Figure 5.2(b) is generated by transferring appearance candidates across $\pi_1$ from Layer 1 to Layer 2. Using the plane $\pi_2$ obtained in iteration $m = 2$, the algorithm generates Layers 3 and 4 in Figures 5.2(c) and 5.2(d) from Layers 1 and 2. The inference approach discussed in Section 5.4 chooses the best assignment of candidates to provide the completed appearance in 3D shown in Figure 5.2(e).

5.4 Appearance Completion Using Markov Random Field

To obtain the completed appearance for the entire object from the appearance candidates, we need to select a set of nodes such that (1) each vertex on the 3D model is assigned exactly one node, (2) the selected nodes satisfy smoothness and consistency constraints, and (3) visible vertices retain their original appearance. To do this, we perform inference over the MRF $\mathcal{G}$ using tree-reweighted message passing (TRW-S, Kolmogorov 2006). While graph-based inference has been used to complete texture in images (Kwatra et al., 2003), our approach uses the layers obtained by geometric and appearance-based symmetries from Section 5.3.

For every vertex on the 3D model $X_s$, $s \in \mathcal{I}$, we find an assignment of reflectance
values that optimizes the following objective:

$$\left\{ i_1^*, \ldots, i_{|I|}^* \right\} = \arg \min_{i_1, \ldots, i_{|I|}} \sum_{s=1}^{|I|} \Phi(\tilde{P}_{i_s}) + \sum_{s=1}^{|I|} \sum_{t \in K_s} \Phi(\tilde{P}_{i_s}, \tilde{P}_{i_t}).$$  \hspace{1cm} (5.1)

The pairwise term in the objective function, $\Phi(\cdot, \cdot)$, enforces neighborhood smoothness via the Euclidean distance. We bias the algorithm to select candidates from the same layer using a weighting factor of $0 < \beta < 1$. This provides consistency of texture. We obtain the following form for the pairwise terms:

$$\Phi(\tilde{P}_{i_s}, \tilde{P}_{i_t}) = \begin{cases} \beta \left\| \tilde{P}_{i_s} - \tilde{P}_{i_t} \right\|^2_2 & \text{if } i_s = i_t, \\ \left\| \tilde{P}_{i_s} - \tilde{P}_{i_t} \right\|^2_2 & \text{otherwise}. \end{cases}$$ \hspace{1cm} (5.2)

The unary term $\Phi(\cdot)$ forces visible vertices to be assigned the reflectance estimated for the visible parts of the object using illumination and appearance estimation. To do so, we set the unary term at the first layer for visible vertices (i.e., vertices indexed by $\mathcal{I}_{\text{vis}}$) to $\zeta$, where $0 < \zeta < 1$, else we set it to 1:

$$\Phi(\tilde{P}_{i_s}) = \begin{cases} \zeta & \text{if } s \in \mathcal{I}_{\text{vis}}, i_s = 1, \\ 1 & \text{otherwise}. \end{cases}$$ \hspace{1cm} (5.3)
We use the tree-reweighted message passing algorithm to perform the optimization in Equation (5.1). We use the computed assignment to obtain the reflectance values $\bar{P}^\star$ for all vertices in the set $I$. We build an analogous graph for the fine-scale texture difference, and solve an analogous optimization to find its assignment.

The assignment step covers the entire object with completed appearance. For areas of the object that do not satisfy the criteria of geometric symmetry and appearance similarity, such as the underside of the taxi cab in Chapter 1, the algorithm defaults to the stock model appearance as shown in Figure 5.3. The algorithm also defaults to the stock model appearance when after several iterations, the remaining parts of the object are partitioned into several small areas where the object lacks structural symmetries relative to the visible areas. In this case, we allow the user to fill the appearance in these areas on the texture map of the 3D model using the PatchMatch algorithm (Barnes et al., 2009).

5.5 Summary

In this chapter, we provide an approach that leverages symmetries over the 3D model of an object to complete illumination-free appearance from visible parts of an object in a photograph to its hidden parts. Our approach maintains smoothness and consistency of texture over the entire 3D model by performing inference over a Markov random field. Repeated reflection of the surface appearance using symmetries provides appearance candidates for the Markov random field. To maintain a seamless transition from the original photograph, the Markov random field ensures that points on the 3D model corresponding to visible pixels in the photograph receive the illumination-free appearance estimated from the photograph in Chapter 4. Our approach also ensures that novel non-symmetric parts of an object are completed directly from the stock 3D model. The next chapter composites the appearance estimated in this chapter with the illumination and geometry obtained in the previous two chapters to provide edited results when a user performs 3D manipulation to an object in a photograph.
Chapter 6

Interactive System for 3D Manipulation of Objects in Photographs

This thesis presents an interactive system for users to perform 3D manipulations on objects in photographs. We present an overview of our system in Section 6.1. The system allows the user to provide a photograph and a 3D model as input, and applies the 3D model alignment, illumination estimation, and appearance completion approaches from Chapters 3 through 5 to recover the complete geometry, illumination, and appearance of the object in three dimensions. As covered in Sections 6.2 through 6.4, our approach re-imagines edits in traditional two-dimensional photo-editing software such as rotations, translations, copy-paste, scaling, and deformations as manipulations to objects in three dimensions. As shown by the user study in Section 6.5, our approach provides perceptually convincing results of manipulation such as shading and shadows on the object and contact surfaces. The end result is seamless from the original photograph.

6.1 Overview of Interaction System

Our interaction system is shown in Figure 6.1. As shown in Figure 6.1(a), the user loads an input photograph on the left, and the 3D model $X$ of the object to be manipulated on the right. The user also specifies the location to a filled background image with the object in-painted out. If the input photograph contains EXIF information, we estimate the focal length using the EXIF information. In the absence of EXIF information, if a rectangle can be identified in the image, the user marks two sets of parallel lines, with one set perpendicular to the other. We compute the vanishing points of the two lines, and use them to extract the focal length through the absolute conic as described by Hartley and Zisserman (2004). In the absence of a rectangle, we use DLT to solve for a projection matrix between correspondences on the 3D model and the image, and we use QR factorization to
CHAPTER 6. INTERACTIVE SYSTEM FOR 3D MANIPULATION OF OBJECTS IN PHOTOGRAPHS

Figure 6.1: Overview: (a) The user loads the photograph and the 3D model into the system. (b) The system performs an automatic alignment of the 3D model geometry to the photograph. The user (c) makes fine adjustments to the geometry alignment if needed, and (d) marks the ground plane. The system (e) estimates 3D illumination and (f) completes the appearance over the object. (g) The user performs 3D manipulations such as rotations, translations, and copy-paste to the object. (h) The system applies the estimated illumination and appearance to the aligned 3D model geometry to produce the end result.

retrieve the intrinsic parameters for the image, including focal length. We use the estimated focal length in the automatic estimation of 3D model alignment discussed in Chapter 3. Figure 6.1(b) shows results of automatic alignment using our system. As shown in Figure 6.1(c), the user may make manual adjustments to the 3D model if necessary to obtain the final model \( Z \).

The user assists the system to estimate contact surfaces such as the ground plane using one of two methods. If a rectangle can be identified in the photograph, the user marks a set of parallel lines perpendicular to each other. We estimate vanishing points as described in the previous paragraph, and we use these points to obtain the horizon line \( v_l \). We calculate the ground plane normal as the vector perpendicular to the horizon, \( n_g = \frac{K - T v_l}{\|K - T v_l\|} \). In the absence of the rectangle, the user marks three points \( X_0, X_1, \) and \( X_2 \) on the base of the 3D model in counter-clockwise order as shown in Figure 6.1(d). We determine the corresponding points on the aligned 3D model \( Z_0, Z_1, \) and \( Z_2, \) and we compute the normal of the ground plane as \( n_g = R^T \left( \frac{Z_1 - Z_0 \times (Z_2 - Z_0)}{\|Z_1 - Z_0 \times (Z_2 - Z_0)\|} - t \right) \), where \( R \) and \( t \) are the rigid rotation and translation of the 3D model estimated in Chapter 3. For both methods, we compute the distance \( d_g \) of the plane from the origin as \( d_g = \min_i n_g^T (RZ_i + t) \), \( i \in \{1, ..., N\} \), where \( N \) is the number of points on the 3D model.

Our approach uses the estimated ground plane and the aligned 3D model to automatically estimate illumination and complete appearance as shown in Figures 6.1(e) and 6.1(f) using the approaches discussed in Chapter 4 and 5. As shown in Figure 6.1(g), the user performs edits to the objects in the photograph, such as rotations, translations, and copy-paste, in three dimensions. The system applies the
6.2 Re-imagining 2D Edits as 3D Manipulations

The goal of our system is to tie into the existing mode of operations in photo-editing software. As such, we re-interpret 2D edits to pixels in photo-editing software, such as rotation, translation, scaling, deformation, and copy-paste as three-dimensional edits to objects. We have used the 3D model aligned using the automated geometry alignment approach with manual geometry correction where needed to create results for the apple and wooden bridge in Figures 6.5 and 6.8, the toy pumpkin in Figures 6.1 and 6.11, and the taxi-cab in Figure 6.9. For all other results, 3D model alignment has been done manually. We use a separately captured background photograph for the photograph of the chair. For all other images, we use in-painting techniques such as PatchMatch (Barnes et al., 2009) in photo-editing software.

estimated illumination and completed appearance to the aligned 3D model to produce the final result of 3D manipulation, shown in Figure 6.1(h). We have provided the executables and source code for our 3D object manipulation system at http://www.cs.cmu.edu/~om3d/.
CHAPTER 6. INTERACTIVE SYSTEM FOR 3D MANIPULATION OF OBJECTS IN PHOTOGRAPHS

Rigid Transformations. Figure 6.2 shows rigid transformations such as rotations and translations performed to photographs of objects such as the chair and a pen. We show the original image, the object composited in the original view, the results of 3D manipulation, the original and corrected 3D model meshes, and the estimated illumination as an environment map. As shown by the second column, our approach replicates the original photograph with high accuracy, which allows us to achieve a seamless transition from the original photograph. The user uses the system to re-orient the chair into a more plausible seating configuration by placing it next to the sofa in three dimensions as shown in the third column of the top row. The ability to move the chair and other furniture in the scene in an intuitive three-dimensional space opens up the scope for applications such as furniture re-arrangement for virtual home staging. As shown by the end result of the pen, our approach integrates 3D object manipulation with traditional 2D editing to produce perceptually realistic effects of the shadow on newly inserted pen strokes.

Figure 6.3 shows the result of our approach on the laptop from Zheng et al. (2012). The approach of Zheng et al. represents objects in a photograph using cuboidal proxies. While their method produces realistic results, they are unable to show hidden parts of the object. Our approach reveals the hidden logo using the stock 3D model while maintaining plausible illumination over the object and the contact surface.

The main objective of providing three-dimensional manipulation to photographs in this thesis is to provide users with greater creative control to change the story of their photographs. The input photograph in Figure 6.4 shows a car parked on a cliff next to two on-lookers. The user combines a 3D rotation and translation of the car with a 2D crop to introduce a dramatic nuance into the end result. The perceptual plausibility of the end result contributes in evoking the intended emotion without causing the viewer to be distracted by artifacts that may be introduced using traditional 2D photo-editing software.

3D Copy-Paste. In Figure 6.5, we show results of 3D copy-paste on photographs of fruit, a watch, and the wooden bridge. Chapter 1 shows results of duplicating the taxi in Figure 1.1 to create a taxi jam. We obtain the result in Figure 6.1(h) by copy-pasting the toy pumpkin in 3D. Performing the duplication effects shown in the figures by rescaling and re-orienting pixel segments using two-dimensional photo-editing software can prove tedious, or even impossible for effects such as showing novel non-symmetric parts of the watch and the taxi-cab. By providing copy-paste of objects in three dimensions, our approach automatically handles the scale and
orientation of individual objects based on the rotations and distances into the scene specified by the user. Since our approach maintains the three-dimensional structure of objects, it correctly handles the geometry, illumination, and appearance resulting from occlusion, contact, and collision of multiple objects. The dynamic composition results of suspended fruit would require several re-takes using a camera, however, the user creates the effect using our object manipulation system with minimal effort.

**Deformations.** In Figure 6.6, the user performs a non-rigid deformation of the top-hat to convert it into a magician’s hat. Based on the manual geometry adjustments approach presented in Chapter 3, the object deformation approach enforces symmetry constraints and smoothness using the as-rigid-as-possible framework of Sorkine and Alexa (2007) as the user performs deformation. Our system can integrate any of the mesh editing approaches discussed in Section 3.1.3 of Chapter 3 to provide object deformation.

In addition to non-rigid deformation, our system provides local rigid transformations of parts of the object, as shown in the case of the watches in Figure 6.6. The user selects a portion of the object such as the handle of the watch, either as an entire connected component or as a set of vertices, and applies a rotation or transla-
CHAPTER 6. INTERACTIVE SYSTEM FOR 3D MANIPULATION OF OBJECTS IN PHOTOGRAPHS

Figure 6.7: Our approach provides perceptually plausible results on the painting and the historical World War II photograph of Avenger airplanes.

...tion to the selected portion. The user may choose to enforce smoothness constraints at the juncture of the part and the rest of the 3D model so as to avoid artifacts or disconnections of parts.

6.3 Vintage Photographs and Non-Photorealistic Media

Our approach approximates the generation of an image using a diffuse reflection model. However, the diffuse reflection model combined with the fine-scale texture difference produces perceptually plausible results for a wide variety of images, as shown by the user study in Section 6.5. In particular, we generate perceptually plausible results for vintage photographs and non-photorealistic media such as paintings by using the completed difference of fine-scale texture to represent film graininess, canvas weave, brush strokes, painting style, and artifacts of digitization. The top row of Figure 6.7 shows results of 3D manipulation of the lemon in the painting of vegetables. We show the original image, the image replicated using our approach, 3D manipulations to the object, the original and aligned 3D models, and the estimated 3D illumination. The illumination estimation recovers illumination for the lemon that generates purple shadows similar to the color of the shadows in the original photograph.

The bottom row of Figure 6.7 shows results of 3D manipulation on a historical photograph of TBF Avenger airplanes during World War II. The input photograph shows the airplanes flying in formation, and is alleged to have been taken by the pilot of the sixth airplane in formation. The user re-aligns the airplanes to appear as though they are flying toward the camera. We constrain the illumination estimation to provide grayscale illumination so as to remain consistent with the black-and-white photograph. Re-orienting the airplanes to achieve the end result in the final image would be impossible to stage in the real world today, as the airplanes have been
retired from flight.

An interesting area for future work is to use 3D models to characterize the dependence of painting style on geometry, illumination, and appearance. Currently our approach does not distinguish between differences in brush strokes or hatching that represent variations in shading across an object. While there do exist approaches to replicate hashing and painting styles using 3D objects (Kalogerakis et al., 2012b, Bassett et al., 2013), these methods do not disambiguate the influences of geometry, illumination, and appearance on the end result of hashing. Decomposing the influence of various scene elements on brush strokes would be crucial to maintain seamless transition from the original image while ensuring artist-directed effect of lighting and geometry on the end result.

### 6.4 Tying 3D Modeling Software to Photographs

A direct impact of the work in this thesis is to bridge the gap between 3D modeling software and photo-editing software by representing photographed objects in 3D space. We show results of using 3D modeling software to perform physical simulations, insert synthetic objects and light sources into images, and create animations.

**Physical simulations.** Our approach allows users to leverage physical simulations to create physically plausible object motions in photographs. The three-dimensional representation of the object and contact surfaces such as the ground plane can be directly fed into a physics engine. As shown in Figure 6.8, the user uses the setup for rigid body dynamics in the 3D modeling software Autodesk Maya to provide simulation quantities such as object mass, coefficients of friction, bounciness, and gravity direction for the 3D representation of the wooden bridge and ground plane in the photograph at the bottom left of the figure. The user provides an initial spin.
CHAPTER 6. INTERACTIVE SYSTEM FOR 3D MANIPULATION OF OBJECTS IN PHOTOGRAPHS

Figure 6.9: (a) Given the original photograph, the user (b) duplicates the top taxi cab using 3D copy-paste after moving it slightly upward, and (c) simulates a non-rigid deformation of the two cabs by performing a physical simulation of the duplicated taxi cab crashing into the original cab.

Figure 6.10: Our approach allows users to combine object insertion with 3D photo-manipulation.

to the object, and uses the Bullet physics engine in Maya to simulate the dynamics of the wooden bridge as it falls to the ground under the influence of gravity. Our approach applies the illumination and completed appearance to the ensuing motion of the bridge to create a perceptually and physically plausible end result of the bridge falling to the ground. In Figure 6.9, the user duplicates the top taxi cab as shown by Figure 6.9(b), and creates a car-crash simulation using nCloth in Maya. The simulation provides a non-rigid deformation of the taxi cabs. Our approach creates perceptually plausible results of the resulting simulation as shown in Figure 6.9(c).

Computer Augmented Reality. By providing a three-dimensional representation of illumination and geometry, our approach can plausibly relight new objects and incorporate new sources of illumination inserted into the photograph. In Figure 6.10(b), the user removes the red chair and inserts an armchair near the sofas. In providing insertion of novel objects and light sources, our approach shares elements with the computer augmented reality approaches of Fournier et al. (1992), Drettakis et al. (1997), Debevec (1998) and Karsch et al. (2011, 2014). However, as shown in Figure 6.10(c), our approach further bridges the gap between photo-editing and
USER STUDY

3D model insertion by allowing users to simultaneously insert new objects such as the armchair and the banana and perform manipulations on existing objects, while ensuring plausible interactions such as cast shadows, contact, and collisions between existing and novel objects. In Figure 6.11, the user duplicates the toy pumpkin, dims the environment illumination, and inserts new lights inside the toy pumpkins to convert them to jack o’lanterns.

Animations. In bridging photo-editing software with 3D modeling software, our aim has been to allow users to combine creativity in both domains in achieving their imagination, such as performing animations from single photographs. The top row of Figure 6.12 shows the original photograph of an origami crane, the aligned 3D geometry, illumination applied to a 3D manipulation, reveal of missing parts, and filling of missing parts to provide the final output of 3D manipulation. The bottom row of Figure 6.12 shows frames from a sequence where the artist animates the origami crane. The artist attached a rig to the aligned 3D model of the crane, and used keyframe animation in Maya to create the motion of the crane flying away from the camera. We used the three-dimensional representation of the crane to apply depth-of-field effects with increasing distance of the crane from the camera. By tying animation techniques from 3D modeling software, the work in this thesis takes a step toward allowing users to bring their photographs to life.

6.5 User Study

We evaluated the perceived realism of 3D object manipulation in photographs through a two-alternative forced choice user study. This user study specifically tests the contribution of the completed appearance with the texture difference in producing perceptually plausible manipulations. We asked participants to compare and choose the more realistic image between the original photograph and an edited result produced using our approach. We recruited 39 participants mostly from a pool of graduate students in computer science, and we conducted the study using a webpage-based
CHAPTER 6. INTERACTIVE SYSTEM FOR 3D MANIPULATION OF OBJECTS IN PHOTOGRAPHS

Figure 6.12: Top row: 3D manipulation of an origami crane. We show the corrected geometry for the crane in the original photograph, and the estimated illumination, missing parts, and final output for a manipulation. Bottom row: Our approach uses standard animation software to create realistic animations such as the flying origami crane.

survey. Each participant viewed four image pairs. Each image pair presented a choice between an original photograph and one of two “modified” conditions: either an intermediate result produced as part of our approach (Condition 1), or the final result of our approach (Condition 2). Specifically, the Condition 1 image was generated using the corrected geometry, illumination, and surface reflectance, but without using the fine-scale texture difference. The presentation order was randomized, and each user saw each of the original photographs once. Across all participants, we obtained a total of 40 responses for the original photo/final result image pairs, and 38 responses for the original photo/intermediate output image pairs.

In the event that participants cannot distinguish between the original photograph and a modified image (either Condition 1 or Condition 2), the probability of selecting the original photograph should be 0.5. 71.05% of the time (i.e., on 54 of 76 image pairs), participants chose the original photograph as being more realistic than the Condition 1 image. We applied a one-sided binomial test on the following null hypothesis $H_0$ and alternate hypothesis $H_{alt}$:

$H_0$: Probability of selecting the original photograph over the Condition 1 image is equal to 0.5.

$H_{alt}$: Probability of selecting the original photograph over the Condition 1 image is greater than 0.5.

We rejected the null hypothesis at the 2.5% level\(^1\) ($p = 0.0001565$), indicating that

\(^1\)For a single set of hypotheses, the test for rejection of the null hypothesis would be performed at the 5% level. However, since we simultaneously tested two sets of hypotheses, one for Condition 1 and one for Condition 2, we used the Bonferroni (1936) correction to determine the appropriate rejection level.
the probability of selecting the original photograph over the Condition 1 image is greater than 0.5. We concluded that users are statistically more likely to select the original photograph when presented with the choice between the original and the Condition 1 image, i.e., the intermediate result without the fine-scale texture difference.

47.5% of the time (i.e., on 38 of 80 image pairs), participants chose the original photograph as being more realistic than the Condition 2 image. Since the experimentally observed percentage of pairs in which the original was chosen was close to 50%, we applied a two-sided binomial test on the following hypotheses:

\[ H_0: \text{Probability of selecting the original photograph over the Condition 2 image is equal to 0.5.} \]

\[ H_{alt}: \text{Probability of selecting the original photograph over the Condition 2 image is not equal to 0.5.} \]

We failed to reject the null hypothesis at the 2.5% level \((p = 0.7376)\) indicating that participants are equally likely to select both the original and the Condition 2 image, i.e., the final result of our approach. In some cases, participants reported choosing the more realistic image based on plausible configurations of the object, e.g., an upright well-placed chair as opposed to one fallen on the ground. Figure 6.13 shows the images used in the user study.
6.6 Discussion

This chapter demonstrates the results of performing 3D manipulations to objects in photographs. By providing a three-dimensional representation of the geometry, illumination, and appearance of objects to be manipulated, our approach re-imagines typical photo-editing operations such as rotations, translations, scaling, deformation, and copy-paste in three dimensions. Our system can be tied into existing mode of operation within photo-editing tools, and is integral for bridging the gap between 3D modeling software and 2D photo-editing software.

The fundamental limitation of our approach is related to sampling. We may lack pixel samples for a photographed object if it is small in image-space, in which case, manipulating it to move it closer to the camera can produce a blurred result. However, as cameras today exceed tens of megapixels in resolution, this problem is much less of an issue, and may be addressed via final touch-ups in 2D. Failures can occur if an object is photographed with the camera’s look-at vector perpendicular to the normal of the bilateral plane of symmetry, e.g., if a wine-bottle is photographed from the top. In these cases, symmetry constraints are difficult to exploit, and the completion has to rely heavily on the texture provided with the 3D model. Since our illumination model represents diffuse reflection, we do not model properties such as refraction, strong specularity, or inter-reflection. Such high frequency effects are
modeled in our approach by the difference of fine-scale texture. In several cases, such as in the case of the taxi-cab in Chapter 1, inaccuracies in light effects are not perceptually evident. However, in cases where such illumination inaccuracies would be noticeable, as in the case of a crystal vase, accurately representing the bidirectional reflectance distribution function will be crucial to provide perceptually convincing results for such objects. Chapter 8 expounds on the issue of perceptual plausibility in edited content.

We find that the end result depends upon the closeness of the 3D model in representing the object in the original photograph. Figure 6.14 shows the result of using increasingly deviating 3D models to represent the taxi-cab from Chapter 1. The 3D model of the Ford Crown Victoria Cab accurately represents the main dimensions of the taxi-cab in the image. The 3D model of the Lincoln town car is close, however, the hood of the car shows slight alignment artifacts due to its deviation from the front of the taxi-cab in the photograph. The 3D model of the Toyota Corolla is significantly different from the cab in the photograph: its length is smaller, and it is more curved than the taxi-cab. As such, deforming the Corolla introduces several artifacts in the aligned 3D model. For mid- to low-frequency effects such as cast shadows, the similarity in the overall shape of all 3D models is sufficient to use them as a light probe. As such all three 3D models provide a similar environment map and illumination effects.

While at this juncture, there may be several photographs where closely matching 3D models may not be found in which case substitutions such as the Toyota Corolla may need to be made, the exponential trend in the availability of 3D models suggests that models not found today will be available online in the near future. We expect that rising ubiquity in 3D scanning and printing technologies will contribute to the increase in model availability. The expansion in 3D model repositories will necessitate searching large databases of several million 3D models for the object in the photograph. Approaches such as those of Aubry et al. (2014) are a much needed step in the direction of image-based 3D model search. An important area of future research on 3D model retrieval is providing 3D model hypotheses sufficient to perform perceptually plausible 3D object manipulations with seamless transition from the original photograph.
Chapter 7

Future Work: Unifying 2D Repositories with 3D Repositories

This thesis takes a step in the direction of unifying 3D model repositories with images by leveraging 3D models to perform full three-dimensional manipulations of objects in photographs. However, the unification of 3D model repositories with repositories of two-dimensional media is by no means complete. In particular, there exist interesting opportunities in stepping in the opposite direction, i.e., in leveraging

![Figure 7.1: Unifying 3D model repositories with repositories of 2D content such as images and videos to provide novel content in 3D and 2D domains. (a) The work in this thesis leverages 3D models to provide reveal of novel parts of objects in images, and has the scope to be extended to videos. (b) An interesting area of future research is to use repositories of images and videos to learn physical properties such as material densities, elastic stiffnesses, and coefficients of friction for objects represented by 3D models.](image)

Repositories of 3D Models

Repositories of Images and Videos
CHAPTER 7. FUTURE WORK: UNIFYING 2D REPOSITORIES WITH 3D REPOSITORIES

![Figure 7.2: A flying cricket bat hitting the ground, a child sliding down a slide, children throwing stuffed toys and letting them fall to the ground, and a car swerving off a race track amidst other cars are just a few of the diverse videos available on the Internet.](image)

large repositories of images and videos to enrich 3D models. One of the immediate areas for impactful research is augmenting 3D models with physical properties of objects. As of now, while 3D models in online repositories such as TurboSquid, 3D Warehouse, and Thingiverse contain detailed geometry and appearance descriptions of the surface of an object, they contain no information about physical properties such as mass densities of the materials of object components, coefficients of friction of object surfaces, and stiffness.

Repositories of images and videos provide rich sources of information on object interactions. Videos on Internet websites such as YouTube or Vimeo particularly show a diverse range of object motions such as, for instance, a tennis racket hitting a ball, ball bouncing, bat flailing in the air, child pushing chair on carpet, child sliding down a slide, child flinging stuffed toy onto floor, car racing on a track, car hitting concrete wall, car toppling over, street performer juggling batons, violinist running a bow across a violin, rockets launching, flowers moving in the summer breeze, leaves falling gently on an autumn morning—the list is endless. Video frames from a few of these interactions are shown in Figure 7.2. Many of the motions of objects and their interactions with other objects depend on the physical properties of the objects themselves. For instance, a toddler might push her plastic chair across the tile floor with minimal effort, however, she would struggle when pushing a dining chair across the carpet.

While repositories of two-dimensional content have been used to create textured
3D models\textsuperscript{1}, to texturize existing 3D models (Levy et al., 2001; Kraevoy et al., 2003; Tzur and Tal, 2009), and to learn surface material properties of objects such as gloss (Fleming et al., 2003), roughness (Ho et al., 2006), and reflectance (Sharan et al., 2008), work on estimating physical properties of objects from two-dimensional content such as video has been limited. Bhat et al. (2002) use the video of a rigid body to estimate external physical properties such as the direction of the gravity vector and the camera orientation, however, they assume that inertial parameters of the object, i.e., the mass and the moment of inertia, are known. Masutani et al. (1994) estimate the relative inertial parameters of an unknown rigid body from video simulations by estimating the angular velocity using optical flow over multiple frames, however, they cannot obtain absolute properties such as object mass as they estimate motion under unknown external forces. Matsuno and Sawada (1998) obtain absolute parameters from video simulations by assuming that a known object has been targeted at the unknown rigid body, and modeling object dynamics before and after collision. The approaches of Masutani et al. (1994) and Matsuno and Sawada (1998) are directed toward robotic operations on space missions where gravitational forces on objects are minimal. They are not suited to the estimation of physical properties from typical videos of object interactions, where forces from sources such as gravity, air drag, frictional contact, and human interactions are prevalent. Bhat et al. (2003), Bouman et al. (2013), and Davis et al. (2015) estimate material properties such as mass density, elastic stiffness, and damping of cloth from video. While their approaches model collision due to cloth deformation, they cannot be directly extended to bulk three-dimensional objects where large parts of the object may be hidden from the viewpoint of the camera.

This chapter discusses the potential of using large image and video databases to learn physical properties for 3D models found in online repositories. As discussed in Section 7.1, we recommend aligning 3D models to objects in images and videos to obtain kinematic quantities such as velocity and acceleration, and using dynamic models to infer physical properties such as material densities, coefficients of friction, and stiffness. As discussed in Section 7.2, enriching 3D models with physical properties opens up the scope for interesting applications in the area of tying physical simulations to photographs, 3D printing and robotic manipulation of real-world objects. We demonstrate how one may use 3D models aligned to videos to learn physical properties such as the coefficient of friction between two surfaces in Section 7.3. While work on estimating physical properties of objects from videos has been limited, there has been a significant body of work on using statistical approaches to measure the physical properties of objects in structural engineering, automotive engineering, and computer graphics. We discuss the related work in these areas in Section 7.4. Section 7.5 discusses the technical challenges inherent in using images and videos “in the wild” to learn physical properties of objects. Section 7.6 discusses the potential to use statistical approaches for modeling object interactions.

\textsuperscript{1}http://www.123dapp.com/catch
CHAPTER 7. FUTURE WORK: UNIFYING 2D REPOSITORIES WITH 3D REPOSITORIES

7.1 Enriching 3D Models with Physical Properties using Large Image and Video Databases

As of now, 3D models in online repositories lack descriptions of physical properties for objects. Large image and video databases such as Google Image Search, YouTube, Vimeo, and Flickr provide the scope to learn physical properties, since they contain diverse interactions between objects in scenes. However, their main issue is that they are two-dimensional representations of the actual three-dimensional interactions between objects in the underlying 3D scene, and cannot be used to directly infer physical properties of the actual objects.

A solution to obtaining a three-dimensional representation of the interactions of objects in a video is to align the 3D model to the object over several frames in the video, using, for instance, the approaches discussed in Chapter 3. Let us assume that the 3D model consists of \( N \) particles \( x_i \in \mathbb{R}^3 \), where \( i \in \{1, 2, \ldots, N\} \). The time \( t \) for every frame may be determined \textit{a priori} by extracting the video frame rate from metadata. By aligning the 3D model to the video frames, we obtain the real-world positions \( z_i(t) \in \mathbb{R}^3 \) and angular displacements \( \theta_i \) of the particles in the object in three dimensions at each time \( t \).

To estimate physical properties of objects such as material densities, coefficients of friction, and stiffness using the aligned positions \( z_i(t) \), we may use equations describing object dynamics, i.e., the actions of forces and torques on objects. For the sake of clarity, we drop the dependence on \( t \) in the notation. The general form of the dynamics between multiple bodies, for instance, between particles in the object and its surroundings is given by

\[
M(q) \ddot{q} + d + \lambda^T C q = f(q, \dot{q}), \tag{7.1}
\]

\[
c(q, \dot{q}) = 0. \tag{7.2}
\]

Here \( q \in \mathbb{R}^{6N} \) represents the state vector and consists of the positions \( z_i \) of the particles and their angular displacements \( \theta_i \). \( M(q) \in \mathbb{R}^{6N \times 6N} \) represents the mass matrix of the particles, and models quantities such as masses and moments of inertia of particles. The vector \( d \in \mathbb{R}^{6N} \) represents the quadratic contribution of angular velocity associated with centrifugal effects. \( C_q \in \mathbb{R}^{P \times 6N} \) represents the Jacobian of the constraints vector \( c \in \mathbb{R}^P \), while \( \lambda \) represents the Lagrangian multiplier. The vector \( f \in \mathbb{R}^{6N} \) represents forces and torques acting on the particles.

In Equation (7.1), the mass matrix and physical properties that the force vector \( f \) depends on are not known \textit{a priori}. The general idea proposed in this chapter is to obtain kinematic quantities such as velocity and acceleration by computing the

\(^2\)Typically, the 3D model consists of a mesh describing surfaces of the object, however, to describe quantities such as mass and elasticity, we require a representation of the 3D model in terms of three-dimensional primitives, which may be obtained by tetrahedralizing (Shewchuk, 1998) or voxelizing (Nooruddin and Turk, 2003) the mesh. We refer to these primitives here as particles.
first and second derivatives of the state vector \( \mathbf{q} \) using the aligned 3D model. The translational velocities \( \mathbf{v}_i \) and acceleration \( \mathbf{a}_i \) are obtained as the first and second derivatives of the displacements of the particles in time, i.e., as

\[
\mathbf{v}_i = \dot{\mathbf{z}}_i, \quad \mathbf{a}_i = \ddot{\mathbf{z}}_i,
\]

while their rotational velocities \( \mathbf{\omega}_i \) and accelerations \( \mathbf{\alpha}_i \) may be obtained as the first and second derivatives of the angular displacements of the particles, i.e., as

\[
\mathbf{\omega}_i = \dot{\mathbf{\theta}}_i, \quad \mathbf{\alpha}_i = \ddot{\mathbf{\theta}}_i.
\]

We can then estimate the unknown physical properties by substituting the kinematic quantities into Equation (7.1), and solving the equation in the unknown physical properties. Since the alignment provides discrete estimates of the positions and angular displacements of the particles, Equations (7.3) and (7.4) may be discretized using finite differencing. The following subsections describe estimating physical properties using instantiations of Equation (7.1) under specific situations.

### 7.1.1 Using the Motion of a Falling Object to Estimate its Material Density

We consider the estimation of the mass density \( \rho \) of a rigid mono-material object such as the tumbler in Figure 7.3(a) falling downwards through a fluid such as air under the influence of gravity. In this case, Equations (7.3) and (7.4) collapse to estimating a single translational velocity \( \mathbf{v} \in \mathbb{R}^3 \) and acceleration \( \mathbf{a} \in \mathbb{R}^3 \) for the object using the position \( \mathbf{z} \in \mathbb{R}^3 \) of the center of mass of the 3D model aligned over several frames of the video, i.e.,

\[
\mathbf{v} = \dot{\mathbf{z}}, \quad \mathbf{a} = \ddot{\mathbf{z}}.
\]

Equation (7.1) collapses to

\[
m\mathbf{a} = \mathbf{f},
\]

where \( m \in \mathbb{R} \) is the mass of the object. The net force \( \mathbf{f} \in \mathbb{R}^3 \) on the object along the direction of motion can be expressed as the combination of the force of gravity \( mg \) acting downward, and a drag force \( \mathbf{f}_D \) acting upwards on the object, i.e., as

\[
\mathbf{f} = mg + \mathbf{f}_D,
\]

Here, \( g \) is the acceleration due to gravity. The drag force \( \mathbf{f}_D \) may be approximated by the drag equation, attributed to Rayleigh:

\[
\mathbf{f}_D = -\frac{1}{2} \rho_f A C_d \mathbf{v} \cdot \mathbf{v}.
\]
Figure 7.3: (a) The forces acting on an object falling vertically down through a fluid such as air include the downward force of gravity, and the upward drag force which depends on the density of the fluid, $\rho$, the instantaneous velocity $v$, the area $A$ of the cross-section perpendicular to the direction of fall, shown in the projection on the right, and the drag coefficient $C_d$ which depends only upon the object shape. (b) An object falling under the influence of gravity along an inclined surface experiences a component of the force of gravity downward along the surface, and a force of friction upward along the surface whose magnitude is given by the normal force from the surface on the object scaled by the coefficient of friction $\mu$.

Here $\rho_f$ is the density of the fluid. The quantity $A \in \mathbb{R}$ is the projected area and can be measured by projecting the aligned 3D model along the direction of fall. The drag coefficient $C_d \in \mathbb{R}$ of an object depends solely on the shape of an object, and not on its mass. For the 3D model of an object, $C_d$ can be determined independent of the input video by conducting physical simulations of drag where the model is subjected to a known external force, and allowed to fall under the influence of the known external force and the drag force till the object reaches its terminal velocity. At the terminal velocity, the acceleration is nearly zero and any dependence on mass in the physical simulation is removed.

Substituting the expressions for $f$ and $f_D$ from Equations (7.6) and (7.8) into Equation (7.7) and re-arranging, we have

$$m (g - a) = \frac{1}{2} \rho_f AC_d \|v\| v. \quad (7.9)$$

With several estimates of $a$ and $v$, we can estimate $m$ by solving an over-constrained system of linear equations in $m$. The mass density $\rho$ may then be obtained as $\rho = \frac{m}{V}$, where $V$ is the volume obtained by voxelizing or tetrahedralizing the 3D model.
7.1.2 Using an Object Sliding Down an Incline to Estimate Coefficient of Friction

Here, we consider the estimation of the coefficient of friction between a rigid object and an incline given a video of the object sliding down the incline with no lateral motion or rotation. Similar to Section 7.1.1, the velocity, acceleration, and object dynamics are given by Equations (7.5) and (7.6). The magnitude of the net force \( \| \mathbf{f} \| \) is the difference between the component of the force of gravity acting along the incline, i.e., \( mg \sin \theta \) and the frictional force acting against the direction of motion, given by the magnitude of the normal force \( mg \cos \theta \) scaled by the coefficient of friction \( \mu \) between the surface and the object, i.e.,

\[
\| \mathbf{f} \| = mg \sin \theta - \mu mg \cos \theta. \tag{7.10}
\]

The quantity \( g \) is the magnitude of the acceleration due to gravity, i.e., 980 cm/s\(^2\). The forces are shown acting on the block in Figure 7.3(b). Substituting Equation (7.6) in Equation (7.10), we have

\[
m \| \mathbf{a} \| = mg \sin \theta - \mu mg \cos \theta. \tag{7.11}
\]

\[
\mu = \tan \theta - \frac{\| \mathbf{a} \|}{g} \sec \theta. \tag{7.12}
\]

Given multiple estimates of the acceleration \( \mathbf{a} \), we may solve a linear system of equations to obtain \( \mu \). The angle of incline \( \theta \) may be determined by aligning a plane to the surface below the incline and determining the angle between the base of the 3D model and the plane. In Section 7.3, we provide results of estimating the coefficient of friction from the video of the wooden bridge from Chapter 3 sliding on a surface. Instead of finite differencing, we assumed constant acceleration and integrated Equation (7.3) with initial velocity and position constraints to obtain the following parametric model of position

\[
z(t_f) = z(t_1) + v(t_1) \Delta t_{f \rightarrow 1} + a \Delta t_{f \rightarrow 1}^2. \tag{7.13}
\]

Here, \( t_f \) is the time at frame number \( f \), \( t_1 \) is the time at the first frame, \( \Delta t_{f \rightarrow 1} = t_f - t_1 \), \( v(t_1) \) is the initial velocity of the object at the first frame, \( z(t_1) \) is the initial position of the object, and \( z(t_f) \) is the position of the object in frame \( f \).

7.1.3 Estimating the Elastic Stiffness of a Deformable Object

The stiffness of an elastic object provides the resistance of the object to deformation by an external force. An object with low stiffness, such as a plush toy, undergoes high deformation when subjected to an external force, for instance, when let fall to the ground under the influence of gravity. An object with high elastic stiffness, such as a steel girder, undergoes negligible deformation even at high forces.
Deformable objects such as plush toys and rubber balls change their intrinsic shape under an external force. To estimate changes in length, area, or volume, we need to perform a non-rigid alignment of the 3D model to the frames of the object in the video. The automated and manual approaches for geometry alignment discussed in Chapter 3 both perform non-rigid alignment using smoothness and symmetry constraints, and may be leveraged to obtain the deformation in 3D shape of the object over the frames of the video.

To describe the deformation of an elastic object, we consider the case where an object is being deformed under an external force such as the force of gravity. We can eliminate rotational dynamics from Equation (7.1), and we can express the net force \( f(q, \dot{q}) \) as the combination of an external force \( f_{ext} \), a damping force \( D\dot{q} \) that depends on the velocity vector \( \dot{q} \), and the net internal force \( Kq \) that depends on the point positions \( q \). On rearranging terms, this yields the form:

\[
M\ddot{q} + D\dot{q} + Kq = f_{ext},
\]  

(7.14)

where matrices \( M, D, \) and \( K \) are of size \( 3N \times 3N \), where \( N \) is the number of particles comprising the 3D model. Given the alignment of the 3D model to multiple frames of a deformable object in a video, quantities \( \ddot{q} \) and \( \dot{q} \) may be determined using finite differencing. The matrix \( K \) is a positive semi-definite matrix that represents the elastic stiffness of the object, i.e., the physical quantity of interest, and models the resistance to deformation. Equation (7.14) assumes that the particles within the object are connected by elastic springs whose spring constants are represented by the elements of the matrix \( K \). The damping matrix \( D \) is typically represented as a linear function of the mass matrix \( M \) and the stiffness matrix \( K \), i.e., as \( \alpha K + \beta M \) for unknown constants \( \alpha \in \mathbb{R} \) and \( \beta \in \mathbb{R} \).

For an object with uniform mass density such as a squash ball, the mass matrix may be represented as \( M = \rho I_{3N} \), were \( \rho \) is the mass density of the object and \( I_{3N} \) is the \( 3N \times 3N \) identity matrix. Substituting for the damping matrix and mass density, we have

\[
\rho I_{3N}(\ddot{q} + \alpha \dot{q}) + K(q + \beta \dot{q}) = f_{ext},
\]  

(7.15)

In general, \( K \) is sparse as particles are assumed to be connected to a few neighboring particles. Assuming we know the external force \( f_{ext} \), such as for the case where an object falls under the influence of gravity without an initial push, we need to simultaneously estimate the symmetric matrix \( K \), \( \rho \), \( \alpha \), and \( \beta \). The number of unknowns turn out to be \( N_{unk} = 3 + 2(\text{NNZ}(K) - 3N) \), where \( \text{NNZ}(K) \) represents the number of non-zero elements in \( K \). Without any priors, we need at least \( N_{unk} \) deformed 3D model alignments to constrain the estimation of the unknowns using Equation (7.15).
7.2 Applications

Enriching 3D models with physical properties of objects provides a more comprehensive description of a real-world object, compared to the current specification of a 3D model in terms of surface meshes and texture maps. We foresee several applications shown in Figure 7.4 for 3D models enriched with physical properties of objects, such as tying physical simulations to 3D manipulations of objects in photographs, 3D printing objects with desired functionality, and informing robotic manipulator arms about object properties.

7.2.1 Tying Physical Simulations to Photo-Manipulations in 3D

The approach presented in this thesis creates a perceptually plausible result of 3D object manipulation from user-provided manipulations of the object in 3D, either as an edit consisting of a single manipulation or as an animation consisting of multiple transformations or deformations of the object. As shown in Chapter 6, our approach allows users to tie physical simulations of objects to the manipulation, however, it depends upon simulation parameters such as object mass and coefficients of friction of contact surfaces to be provided by the user.

Most average consumers of our 3D manipulation software would not have access to the physical properties of objects. Measuring the properties directly may not be an option, as the user may not have access to the object or a device with which to measure the relevant properties. Using an indirect method such as providing the mass densities or coefficients of frictions using a web look-up may also not be feasible. While it may be possible for a user to look up the material density of a particular component on an object such as the steel legs of a chair, providing details such as the material densities of the internal frame, the foam, and the rubber feet, and the coefficients of friction of the feet and the cloth may be cumbersome for the everyday user. By tying a simulation engine to our approach and by using 3D models with built-in physical properties, the 3D object manipulation discussed in this thesis can be extended to allow users to produce perceptually and physically plausible three-dimensional motions of objects in photographs with minimal effort.

7.2.2 3D Printing

An interesting application of 3D models enriched with physical properties of objects is in guiding 3D printers to produce objects with desired functionality. 3D printers have begun to gain a foothold in the consumer market, and are used to print objects such as toys, holders, containers, spare parts in building projects, and even jewelry. While the range of materials is limited to plastic, metal, and ceramic, it is conceivable that the range and compositions of materials available in 3D printers will expand in the near future.

3D models enriched with physical properties such as mass densities, coefficients
CHAPTER 7. FUTURE WORK: UNIFYING 2D REPOSITORIES WITH 3D REPOSITORIES

(a) Tying Physical Simulations to 3D Object Manipulations in Photographs
(b) Robotic Manipulation
(c) 3D Printing

Figure 7.4: Enriching 3D models with physical properties of objects will have applications toward (a) tying physical simulations to 3D manipulation of objects in photographs, (b) informing robotic manipulator arms about the physical properties of objects, and (c) guiding 3D printers toward printing user-friendly products.

of friction, and object stiffness have the potential to inform the 3D printing algorithm on the appropriate choice of materials and the appropriate 3D printing method in order to maintain the functionality of the object represented by the 3D model. For instance, by enriching the 3D model of a cup using video information on how cups are handled, we may inform a 3D printer to lay down plastic along the direction of typical sliding motion of a cup along a table, i.e., parallel to the handle of the cup. The most plausible placement of print material can reduce friction along the direction of typical motion, and minimize accidental spillage of the contents of the cup. 3D model repositories enriched with physical properties of objects also have the potential to guide research in determining appropriate materials and compositions for 3D printers by providing rapid prototyping for a diverse range of objects.

7.2.3 Informing Robotic Manipulator Arms

Several robotic applications leverage 3D models aligned to input from a camera to guide the movement of a robotic manipulator arm around an object. Providing additional information to the manipulator arm about the properties of the manipulated object has the potential to improve object-dependent grasping in robots. For instance, informing a robotic manipulation algorithm about the stiffness of a glass bottle versus the flexibility of a plush toy will ensure that the robot arm grasps the glass bottle with a lighter grasp as compared to the plush toy to prevent fracture of the bottle. Informing the manipulation algorithm about the coefficient of friction between the surface of the apple in Figure 7.4 and materials such as steel or aluminium will ensure that the digits at the end of the arm apply sufficient force perpendicular to the surface so as generate a frictional force that will prevent sliding. By enriching 3D models in online repositories with physical properties, robotic manipulation algorithms will be able to leverage algorithms independent of the physical properties of objects by tapping into large repositories to access these properties. 3D model repositories enriched with physical properties thus have the potential to generalize robotic manipulation to a far more diverse range of objects the ones handled by
7.3 Proof of Concept

To demonstrate the estimation of physical parameters of objects from videos, we have conducted a simple experiment to estimate coefficient of friction using the approach discussed in Section 7.1.2. We used the wooden bridge shown in Chapter 3 as our test object, and we used a finished wooden surface from a bar stool as the test surface of contact. We captured four videos of the bridge sliding down the bar stool surface with the surface inclined at an unknown angle. We used the Kinect for Windows v2 to capture the video at a spatial resolution of $1080 \times 1920$, and a temporal resolution of 30 frames per second. The videos consisted of the investigator placing the wooden bridge on the bar stool surface and releasing it with minimal push. For the rest of the discussion, we assume that the camera and ground plane are level, i.e., the direction of gravity vector is along the $y$-axis of the camera image plane. The top row of Figure 7.5 shows frames from one capture. Figure 7.7 shows frames from the remaining three videos.

We used a semi-automatic approach to align the 3D model of the wooden bridge to the video frames. The manual steps consisted of pre-scaling the bridge to represent the real-world dimensions of the bridge, rotating the image to orient the bridge vertically, providing a rendered template closest in orientation to the bridge, and marking an approximate bounding box around each instance of the bridge to provide an approximate scale. We ran the automated algorithm discussed in Chapter 3 using three renders in azimuth around the closest approximation, and three scales about the scale provided by the bounding box. We used the automated algorithm to obtain the best matches of patches to the image.

Given the nature of the experiment, we assumed that the object moves along a straight line down the bar stool surface, without any rotation and lateral motion.
CHAPTER 7. FUTURE WORK: UNIFYING 2D REPOSITORIES WITH 3D REPOSITORIES

We used this assumption as a constraint to obtain a consistent alignment of the 3D model to all frames of the video. Given the alignments provided by the automated approach, we manually picked the two frames with the best qualitative alignments, used the rotation of the best alignment as the rotation of the bridge over all frames, and used RANSAC to estimate a translation for each frame that constrained the bridge to lie along the line joining the centers of the transformations of the 3D model to the best two frames. The resulting alignments are shown in the bottom row of Figure 7.5.

To obtain the angle $\theta$ for the right hand side of Equation (7.12), we assumed that the bar stool was parallel, so that $\theta$ could be approximated by the angle between the base of the wooden bridge and the ground surface, or equivalently, by the normal

Table 7.1: Estimated angle of incline, magnitude of net downward acceleration, and coefficient of friction using four videos.

<table>
<thead>
<tr>
<th>Video ID</th>
<th>$\theta$ (in degrees)</th>
<th>$|a|$ (in cm/s$^2$)</th>
<th>$\mu$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>28.04</td>
<td>105.81</td>
<td>.41</td>
</tr>
<tr>
<td>2</td>
<td>24.10</td>
<td>125.34</td>
<td>.31</td>
</tr>
<tr>
<td>3</td>
<td>42.41</td>
<td>335.89</td>
<td>.45</td>
</tr>
<tr>
<td>4</td>
<td>22.79</td>
<td>99.97</td>
<td>.31</td>
</tr>
</tbody>
</table>
n to the base of the wooden bridge, and the normal $n_g$ to the ground surface. To obtain $n$, we manually marked a set of points on the base of the wooden bridge. Given the alignments of the 3D model over all frames of the video, we determined $n$ by fitting a plane to the base points of all transformations of the bridge. This plane approximated the surface of the bar stool. To obtain $n_g$, we marked two pairs of parallel lines on the ground surface (i.e., the white foam board) with the one pair being perpendicular to the other pair. We then obtained $n_g$ using the vanishing points method similar to the approach for estimating the ground plane in photographs described in Chapter 3. Figure 7.6(a) shows the transformations of the bridge in 3D across all frames, the plane representing the surface of the bar stool, and the plane representing the ground surface. We obtained an angle of $28.04^\circ$ using the frames of the video shown in Figure 7.5. Table 7.1 shows the angles estimated for the video in Figure 7.5 and the three videos shown in Figure 7.7.

To obtain the acceleration $a$ for the object, we computed the centroids $z(t_1)$ to $z(t_F)$ of the bridge where $F$ was the number of frames, and we fit the parametric model from Equation (7.13) to the centroids to estimate the acceleration and an initial velocity that might have occurred despite the effort to hold the bridge still. We performed a least squares estimation of $v_1$ and $a$ using the above model, which amounted to solving the following system of equations:

$$
\begin{bmatrix}
I_3 \otimes \Delta t_{2\rightarrow 1} & I_3 \otimes \frac{1}{2} \Delta t_{2\rightarrow 1}^2 \\
I_3 \otimes \Delta t_{3\rightarrow 1} & I_3 \otimes \frac{1}{2} \Delta t_{3\rightarrow 1}^2 \\
\vdots & \vdots \\
I_3 \otimes \Delta t_{F\rightarrow 1} & I_3 \otimes \frac{1}{2} \Delta t_{F\rightarrow 1}^2
\end{bmatrix}
\begin{bmatrix}
v_0 \\ a
\end{bmatrix}
=
\begin{bmatrix}
z(t_2) - z(t_1) \\
z(t_3) - z(t_1) \\
\vdots \\
z(t_F) - z(t_1)
\end{bmatrix}
$$

(7.16)

In Equation (7.16), $I_3$ represents the $3 \times 3$ identity matrix, while $\otimes$ represents the Kronecker product. Figure 7.6(b) shows the centroids of the 3D model transformations across all frames of the video shown in Figure 7.5. The initial spacing of the cen-
troids is small, indicating a small initial velocity, while the final spacing is large due to the acceleration \( \mathbf{a} \). By solving the system of equations in Equation (7.16) for the video in Figure 7.5, we obtained an acceleration of magnitude \( \| \mathbf{a} \| = 105.81 \text{ cm/s}^2 \). Table 7.1 shows the magnitudes of net acceleration estimated for the video in Figure 7.5 and the three videos shown in Figure 7.7. We computed the coefficient of friction \( \mu \) by substituting the values of \( \theta \) and \( \mathbf{a} \) into Equation (7.12) and using the value of 980 cm/s\(^2\) for the magnitude of the acceleration due to gravity. We obtained a value of .41 for \( \mu \) using the frames of the video shown in Figure 7.5. Table 7.1 shows the final value of \( \mu \) for all four videos obtained using Equation (7.12).

We used an independent setup to estimate the coefficient of friction between the bar stool surface and the wooden bridge. We computed the coefficient of friction by measuring the minimum angle of incline at which the bridge starts sliding on the bar stool surface, and obtaining the coefficient of friction as the arctangent of the incline. We performed 30 trials to measure the minimum angle of incline, and obtained the coefficient of friction as the average over all trials. To determine the angle of incline in each trial, we used LEGO blocks to build a vertical pillar. We added blocks till the wooden bridge just began to slide on the surface. We then measured the height of the pillar, and obtained the minimum angle of incline as the arcsine of the ratio of the height of the pillar to the length of the bar stool surface from the start of the pillar to the bottom. The coefficient of friction obtained using this method turned out to be .2413 ± .0225. Issues contributing to the difference between the coefficient of friction estimated from the videos and the coefficient of friction estimated as the arctangent of the minimum angle of incline include slight pixel-level misalignments leading to deviation in depth, imprecise manual marking of the ground plane, lack of rotation in modeling object dynamics, and non-level ground plane and camera leading to unknown down direction for the gravity vector. Section 7.5 provides detailed discussion of the challenges involved in performing estimation of physical properties.

### 7.4 Related Work

The marriage of large repositories of two-dimensional information with large 3D model repositories to estimate the physical properties of objects such as material density, elasticity, and coefficients of friction is a novel domain of research. The approaches of Masutani et al. (1994) and Matsuno and Sawada (1998) estimate inertial parameters in the absence of external forces. Bhat et al. (2002) assume the inertial parameters are known, and estimate external parameters such as gravity direction and camera orientation. Bhat et al. (2003), Bouman et al. (2013), and Davis et al. (2015) estimate material properties such as mass density, stiffness, and damping of largely two-dimensional objects such as cloth from video, however, the approaches cannot be directly extended to objects where several surfaces may be hidden from the camera viewpoint.
Work on estimating physical parameters for objects represented by 3D models draws inspiration from the estimation of properties such as coefficients of friction, mass, and elastic moduli in fields such as structural engineering and automobile engineering as detailed in Section 7.4.1. The estimation of physical properties also shares elements with the estimation of simulation parameters from user-provided animation frames for artist-directed control of physical simulations. Section 7.4.2 summarizes approaches in computer graphics to perform physical simulations, and to estimate simulation parameters for artist-directed control.

7.4.1 Domain-Specific Estimation of Physical Properties
Domains such as structural engineering and automobile engineering routinely employ statistical approaches to estimate physical properties. Structural engineers use dynamic models to determine optimal masses and shapes for support structures such as beams and trusses in buildings (Hibbeler, 2011). A significant body of work has been devoted to the analysis of object dynamics under seismic excitations. For instance, to determine how fast objects dissipate energy during seismic vibrations, several approaches perform modal analyses of damping models for linear structures (Villaverde and Newmark, 1980; Igusa et al., 1984; Yang et al., 1987; Zhou et al., 2004; Song et al., 2008; Chu et al., 2009). Shake tables equipped with high resolution infrared cameras and accelerometers have been used to measure the coefficients of friction between materials such as steel and carpet to determine the resistance of non-structural components in buildings to sliding or falling during earthquakes (Kafali, 2007; Kafali et al., 2007).

In the automotive domain, several approaches have addressed the estimation of the maximum coefficient of friction between car tires and road surfaces. Breuer et al. (1992) and Eichhorn and Roth (1992) measure the causes of reduced friction such as wearing down of surfaces or use of lubricants. Eichhorn and Roth (1992) relate tire acoustics to friction, while Bachmann (1995) measures tire-tread deformation. Several approaches measure tire slip due to traction, braking and turning (Dieckmann, 1992; Gustafsson, 1997; Müller et al., 2001; Ray, 1997).

While approaches in these domains provide precise results, they usually apply known forces and directly measure changes in kinematic quantities such as distance or acceleration in three dimensions to estimate the unknown parameters. As will be discussed in Section 7.5, videos in the wild may contain several unknown parameters such as forces and camera direction, in addition to unknown three-dimensional depth.

7.4.2 Animator Control of Physical Simulations
Approaches that provide artist-directed control incorporate artist-provided keyframes as constraints in the physical simulation. To maintain coherence with the simulation, spatiotemporal constraints are employed (Witkin and Kass, 1988; Cohen, 1992; Ngo and Marks, 1993; Liu et al., 1994; Popović and Witkin, 1999). Popović et al. (2000)
allow animators to re-orient objects in simulation frames, and provide animator-intuitive control over parameters such as shapes, masses, surface normals, elasticity coefficients, and moments of inertia. Chenney and Forsyth (2000) augment simulation dynamics with uncertainty models, and sample multiple simulations using a Markov chain Monte Carlo algorithm to provide plausible examples for animator control of simulations. Several approaches have provided animator control of smoke simulations (Treuille et al., 2003, McNamara et al., 2004; Fattal and Lischinski, 2004) and cloth simulations (Wojtan et al., 2006; Bergou et al., 2007). Control of dynamic simulations of character articulation has received considerable interest in order to provide artist-intuitive kinematic control while maintaining physical plausibility through simulations (Zordan and Hodgins, 2002; Yin et al., 2003; Abe and Popović, 2006; Allen et al., 2007; Nguyen et al., 2010; Sok et al., 2010; Lockwood et al., 2011; Kry et al., 2012). The objective of nearly all approaches is to achieve the desired object motion from the perspective of the animator. As such physical properties of objects may be specified \textit{a priori}, or may be altered using animator-intuitive controls.

7.5 Technical Challenges

The example approaches to estimate physical properties of objects discussed in Section 7.1 and the affiliated proof of concept discussed in Section 7.3 pertain to simplistic ideal cases of object motion in videos. In general, object motions as seen in videos and images in the wild are rarely “clean”. We expect a number of technical challenges arising in leveraging large databases of videos and images to learn physical properties for 3D models. We anticipate that the technical challenges introduced by videos and images in the wild may increase the number of parameters to be estimated for the 3D model to accurately describe the motions of objects in videos and images. Sections 7.5.1 to 7.5.6 discuss the key technical challenges in leveraging databases of two-dimensional content to estimate physical properties for 3D models.

7.5.1 Complexity of Dynamic Models

The dynamic models used in Sections 7.1.1 through 7.1.3 are approximations to real-world dynamics. For instance, the physical model discussed in Section 7.1.2 does not account for the contribution of air drag which can introduce added deceleration of the object along the direction of motion, or induce a lateral shift in the movement of the object. Similarly, lateral motion of the fluid in Section 7.1.1 may induce a lateral swinging motion on the object as it falls through the fluid, thereby affecting the acceleration and yielding inaccurate estimates of mass. The damping and elastic forces in the mass/spring model in Equation (7.14) are obtained as a linear approximation of the stress-strain relationship

\[
\sigma = E\epsilon, 
\]  

(7.17)
Figure 7.8: To use large repositories of videos and images in enriching 3D model repositories with physical properties of objects, we need to address the challenges of (a) accounting for object complexity, (b) estimating extra unknown quantities such as forces exerted by humans, (c) disambiguating camera motion from object motion, (d) aligning 3D models to video frames with motion blur, and (e) aligning 3D models of human bodies to people in images and videos.

used in explicit finite element models (O’Brien and Hodgins, 1999; Nealen et al., 2006), where $\sigma$ represents the stress tensor, $E$ represents the stiffness matrix, and $\epsilon$ represents the strain tensor. To accurately estimate physical properties, we need to use dynamic models that incorporate the various degrees of freedom shown by objects in videos.

### 7.5.2 Complexity of Objects

Most everyday objects are rarely composed of a single material. In general, objects such as the car in Figure 7.8(a) consist of several components with a range of mass densities, coefficients of friction, and material stiffnesses. For an object with $K$ components, each composed of a single material, the mass $m$ is given as

$$m = \sum_{k=1}^{K} \rho_k V_k,$$  \hfill (7.18)

where $\rho_k$ is the material density of the $k^{th}$ component, and $V_k$ is its volume. We now have $K$ unknowns in estimating material density. Even across a surface made of a single material, the coefficient of friction varies due to scratches, dirt, dents, and bumps caused by wear and tear, and may induce rotational spin in addition to translational motion. For anisotropic surfaces and objects, properties such as coefficient of friction and stiffness depend upon direction. To leverage a 3D model for applications such as user-independent physical simulations or informing robotic manipulations on handling various parts of an object, an approach to estimate physical properties of an object needs to individually account for each component on the 3D model.
CHAPTER 7. FUTURE WORK: UNIFYING 2D REPOSITORIES WITH 3D REPOSITORIES

7.5.3 Extra Unknown Physical Quantities

In most videos and images in the wild, quantities such as the external forces applied on objects, the direction of gravity, and the angles of inclinations of surfaces may not be known \textit{a priori}. External forces on an object may arise due to natural effects such as wind or water, the use of automation such as the engine of a car, or the interaction of animals and people such as the child in Figure 7.8(b) with objects. It may not be possible to directly model such forces. An approach to estimate physical properties must simultaneously estimate unknown forces and other quantities such as gravity direction or inclination angles.

7.5.4 Moving Camera

The analysis of object motion discussed in Section 7.1 does not account for possible simultaneous motion of the camera. In the event of camera movement as in the case of Figure 7.8(c), the acceleration and velocity estimated using 3D model alignment will contain components from the motion of the camera and the object. To correctly estimate physical properties of the object independent of the camera, the motion of the camera needs to be disambiguated from the object motion.

7.5.5 Motion Blur in Videos

The frame rate of nearly 30 frames per second on most consumer cameras today is too low to prevent motion blur for the motions of objects moving at moderate to high speeds, such as, for instance, the dropped plush toy in Figure 7.8(d). While high speed cameras such as the GoPro are beginning to penetrate the consumer market, motion blur is inevitable when the projected displacement of the object in the image plane of the camera within a single exposure exceeds the spatial resolution of the camera. The automated 3D model alignment approach discussed in Chapter 3 fails to provide an alignment in the event of motion blur, as motion blur smooths out details of edges and corners along the direction of motion. Approaches to leverage 3D alignment of models to videos for estimating physical properties must account for motion blur in the alignment process.

7.5.6 Alignment of 3D Models of Human Bodies

Most videos on the Internet contain people. To leverage large repositories of videos and images in estimating the physical properties of objects that people manipulate in everyday life, it is imperative to accurately model the interactions of people with the objects in three dimensions. To remain consistent with the use of 3D models to describe full three-dimensional interactions, we recommend aligning 3D models of human bodies to people in videos. The alignment of 3D models of people is a significantly challenging task, due to the wide variation in clothing geometry, appearance, intrinsic body shape, and body pose articulation that people exhibit in
the real world and in videos. While there has been a significant amount of work in estimating 3D pose of people in images in terms of a skeleton (Taylor, 2000; Ramakrishna and Sheikh, 2012; Simo-Serra et al., 2012; Simo-Serra et al., 2013) or in terms of collections of primitives (Sidenbladh et al., 2000; Agarwal et al., 2004; Grest et al., 2005; Sigal and Black, 2006), there has been limited work in simultaneously recovering three-dimensional body pose and body shape (Guan et al., 2009; Zhou et al., 2010), and almost no work in generalizing the alignment of 3D models of humans to photographs and videos of people while accommodating clothing diversity. Accurate alignment of 3D models to photographs and videos is within itself a task worthy of a Ph.D. thesis.

7.6 A Word on Statistical Approaches for Using Large 2D Repositories to Learn Object Interactions

While the estimation of physical properties as suggested in this chapter would allow us to leverage physics-based models of object dynamics, it is dependent to a certain extent on accurately modeling object dynamics in real-world scenarios. With the increasing number of unknowns due to the challenges discussed in Section 7.5, there may be interactions where accurately representing the full range of object dynamics may be highly underconstrained. In addition, approaches to provide forward simulation models of object dynamics have, in general, struggled to provide perceptually plausible object motion for the complex interactions seen in the real-world.

In the light of these issues, it may be desirable to consider statistical approaches that bypass the physical interaction model. Generative approaches may span the space of object motions and allow synthesis of novel interactions by sampling the space, while discriminative approaches may directly model the relationship between physical parameters and object motions. We suggest that the key contribution of a statistical approach for it to be preferable over a physics-based approach should be its generalizability. The physics-based approach developed in this chapter directly provides generalizability by modeling object physics. For instance, a robot that knows the mass of an apple, and the coefficient of friction between the apple and its metal fingers implicitly knows the force that needs to be applied on to the apple to lift it off the ground and also whether the apple will roll or slide on a metal table. Providing this generalizability would be important to represent interactions that are intuitive to the manner in which humans interact with their surroundings.

In the next chapter, we elaborate on the possibility of unifying various modalities of big data to provide a singular representation of the world. Integrating information from modalities such as speech and text may help provide generalizability within statistical approaches by incorporating how humans interpret real-world objects.
Chapter 8

Discussion

In this thesis, we have addressed the task of providing full-range 3D manipulation to objects in single photographs by estimating a 3D reconstruction of the scene sufficient for perceptually plausible manipulation. Since the estimation of a 3D reconstruction from a single image is highly ill-posed, we have provided the insight that 3D models from large repositories can be leveraged to bootstrap the estimation. The 3D models provide a rough initialization for the geometry and appearance of objects in images. We have provided algorithms that correct the geometry and appearance of the 3D models to match the photograph and that estimate scene illumination and hidden geometry and appearance so that users can perform object manipulations shown in Chapter 6. While this thesis has used large 3D repositories to enrich images with novel content, we see interesting opportunities in using large 2D repositories to enrich 3D models with novel content such as the physical interactions of objects as discussed in Chapter 7.

While our work in this thesis has opened the doors to full range 3D manipulations, there is, as of yet, a large scope for research in providing a seamless experience to providing object interactions in photographs for the everyday user. For researchers interested in addressing the task of providing 3D manipulations in photographs, we detail potential areas for impactful work in Section 8.1. Our work in this thesis is part of the recent initiative toward bridging the gaps between large repositories of data such as 3D models, RGB images, videos, speech, and text. We elaborate how one may perform continued research within this domain in Section 8.2.

8.1 Toward a Seamless User Experience in Interacting with Objects in Photographs

Let us imagine the ideal scenario for providing a seamless photo-manipulation experience. The user should input their photograph to the photo-manipulation system.
CHAPTER 8. DISCUSSION

Figure 8.1: *Dalí Atomicus* by Phillippe Halsman, taken from the website of Library of Congress Prints and Photographs Division, at http://hdl.loc.gov/loc.pnp/ppmsca.09633. To allow the everyday user to seamlessly manipulate a photograph to emulate the water splashing, the cats flying, and the subject jumping, we need to model the geometry of water flow, cat fur, and subject articulation, and illumination effects such as glitter on the water surface and shine of the cat fur.

The system should automatically (a) search for 3D models for all objects in the scene, (b) estimate the 3D scene information such as object geometry, scene illumination, object appearance, object deformation characteristics, inter-object contacts, gravity direction, and external forces from the photograph. The user should provide an object interaction, either spoken, for instance “flip the chair on the left”, or by using a rotational hand gesture, similar to interactions in the real world. The system should then react to the input interaction and provide a perceptually convincing end result in terms of the object geometry and appearance, and in terms of real-world physics.

To get to the point of providing a seamless user experience, the first question to ask is how far are we in terms of representing real world objects? The thesis assumes that 3D models from public repositories are accurate, if not precise, descriptions of real world objects. Indeed for the object manipulations shown in this thesis, they do turn out to be accurate. However, if we wanted an average user to replicate the photographic composition in the *Dalí Atomicus*, shown in Figure 8.1, while the work
in the thesis would allow us to suspend the chair, we would still need to provide a 3D reconstruction that (a) represented the geometry of fluids, the fine fur of cats, and the articulations of the subject and the animals, and (b) modeled their response to physical interactions in nature such as the force of gravity or the force of the water. The future work discussed in Chapter 7 has the potential to represent the latter, however, representing the diverse range of object geometries and deformations in itself is a major open area for future work.

The second question is how far are we in terms of searching the space of millions of 3D models from public repositories to find the 3D model representing the object, and estimating its rigid pose and non-rigid deformation? The work in Chapter 3 demonstrates that while we may eliminate the exhaustive search in illumination by providing invariance to illumination, we obtain precise alignment of geometry only if we exhaustively span the space of viewpoints, scales, locations, and geometric deformations. We anticipate that our exhaustive search approach may be extended to perform search amongst millions of 3D models in public repositories. However, at this point, providing fast search on the order of seconds, if not in real time, will become paramount for seamless user interaction. To provide fast search, the ideal line of attack is to determine thresholds at which objects are invariant from each other at various levels of abstraction: categories, instances, parts, and viewpoints. Within their thresholds, objects may be clustered together at various levels. Outside of those thresholds, we would need to perform an exhaustive search. To provide high speed-up for exhaustive search, we recommend investigating techniques that leverage clusters of graphics processing units or distributed computing on the cloud to span the search space. At this juncture, research would need to focus on incorporating memory, throughput, and processing constraints on determining the optimal choice between using GPUs and cloud computing.

The final question is, how far are we in terms of providing a perceptually convincing result in itself? The work in this thesis has been focused on providing effects of illumination such as cast shadows and surface shading that are necessary for depth perception. However, we still have a long way to go in modeling the full diversity of illumination effects such as refractions, translucency, specularity, subsurface scattering, inter-reflections, which would be important for representing and manipulating for instance the glitter of the water or the shine of the cat fur in Figure 8.1. In addition, the lack of information in representing illumination information can affect perceptual plausibility in the end result. For instance, in the result of suspended apples in Figure 6.5, the pixels on the apple in the original photograph lack illumination contribution from sunlight reflected off the snow on the ground, preventing an estimation of this illumination. In an actual instance of apples close to the ground, the reflection off the snow would have attenuated the strong shadow. While aligning 3D models to the subject and the trees may provide one solution to recover the ground contribution to illumination, data-driven techniques that leverage image-environment relationships, such as the approaches of Karsch et al. (2014)
and Lalonde et al. (2014) may be necessary to obtain an accurate estimate of illumination. Finally, while generic Gestalt constraints such as those of smoothness and symmetry have allowed us to represent hidden parts of objects in this thesis, we foresee that object-dependent representation of shape and appearance would be important for to convincingly represent random patterns over natural objects or to model wear and tear.

However, the issue of providing perceptually seamless results is far more challenging that it may seem at face value. Already, the advances in digital rendering of computer generated imagery have begun to challenge our belief on whether, for instance, visual effects in movies are real or computer generated. The work in the domain of photo-editing, object insertion such as Karsch et al., and object manipulation as shown in this thesis are further adding to the challenge of how we interpret visual imagery, both in real and virtual worlds. We are beginning to see static and dynamic content that we have never seen before, and as such, we are getting to the point where we need to re-visit human interpretation of visual content. While some inconsistencies are obvious (such as the lack of cast shadows), others may not be immediately clear, or may contribute to a distraction in perception in a manner we may not have fully comprehended. In some cases, illumination effects that are accurate may themselves seem perceptually inconvincing either due to physical implausibility or due to the lack of a mental picture of the phenomenon. While the field of vision has moved away from psychophysics toward computation, we believe that as we move toward altering user content in domains such as computer augmented reality, we need to re-visit perceptual experiments in the psychophysics domain to understand how humans interpret the altered content.

8.2 Toward a Unified Representation of Objects

The various modalities of big data that we see today—“big textual data”, or textual descriptions of objects and events in online shopping catalogs, weblogs, and social networks, “big visual data”, or the enormous quantities of images and videos that are uploaded to the Internet daily, and “big 3D data”, or expanding public repositories of user-uploaded and scanned 3D models—are all different views to describe the same objects that we see in our daily lives. While they have been largely distinct, the work in this thesis and the recommended future work demonstrate that unifying repositories of various modalities has the potential to generate large quantities novel content to provide added benefit to applications that leverage big data.

The unifying factor behind repositories of large data is the human element. Repositories of big data are largely biased toward the way humans interpret the world. They reflect the preferences we have in communicating our thoughts and feelings about our surroundings, in capturing and creating images, and in designing and building objects. We believe that any approach that aims to bridge the gap between large repositories of data should be focused on either deducing or leveraging
TOWARD A UNIFIED REPRESENTATION OF OBJECTS

this common element of human interpretation of the world through the unification.

3D model alignment methods such as those discussed in Chapter 2 have already begun the process of unifying 3D model repositories with images. Approaches that leverage the unification to provide novel content in images, such as the object insertion method of Karsch et al. (2011, 2014), and the method of providing 3D object manipulation while revealing hidden parts of objects presented in this thesis have demonstrated the scope to push consumer-oriented augmented reality in areas such as online shopping, real estate, digital forensics, medicine, and K-12 education. By enriching 3D models with physical properties we can contribute to the improvement of human-machine interaction in applications such as 3D printing and robotic manipulation of everyday objects for assisted living. Unifying textual information with images has already shown enormous potential to generate novel summarizations for images (Farhadi et al., 2010; Ordonez et al., 2011; Kuznetsova et al., 2012; Kuznetsova et al., 2013). We foresee that unifying textual and spoken descriptions with 3D model and image repositories will provide applications that leverage big data with a coherent body of information on how humans interact with objects.

The unification of multiple modalities of objects calls for a collaboration across multiple disciplines (and the generation of several Ph.D. theses). Already, the work in this thesis has leveraged techniques in rendering from computer graphics, techniques in feature scale selection from computer vision, and techniques in performing regression and inference over probabilistic graphical models from machine learning. We foresee the need for familiarity with speeding up forward rendering and physical simulations via graphics processing units to efficiently probe the search space for aligning 3D models to videos and text. We also foresee significant research opportunities in machine learning to investigate scalable learning algorithms that efficiently direct the probing of the search space. We expect new research in the natural language processing domain to leverage attributes from “big 3D data” in describing objects. We also expect the ensuing research to greatly impact domains such as robotics, human-machine interaction, forensics, medicine, augmented reality, flight simulation, K-12 education, and perceptual science. We hope the insights provided by the work in this thesis inspire research toward developing a unified computational representation of objects consistent with the way humans see the world.
Appendix
Appendix A

Likelihood for Children of Patch from Renders for 3D Model Alignment

In Theorem 1, we show that the term \( p(I_3|x_{21}) \), i.e., the contribution of the patch at position \( x_{21} \) in generating the LoG \( I_3 \) at a level \( l = 3 \) lower than the level of \( x_{21} \) can be modeled by marginalizing the data likelihoods of the children of the patch corresponding \( x_{21} \), i.e., we show that

\[
p(I_3|x_{21}) = \left( \sum_{x_{31}} p(I_3|x_{31}) p(x_{31}|x_{21}) \right)^{1/2}.
\]  (A.1)

Equation (A.1) here corresponds to Equation (3.9) in Chapter 3.

**Theorem 1.** Suppose we have a parent patch at location \( x_{lp} \) in the image, and the locations of its child patches are represented by \( x_{mq} \) where \( m = l + 1, q \in Q_p, Q_p \) represents the set of indices to the child patches, and \( |Q_p| \) is the size of \( Q_p \). Then

\[
p(I_m|x_{lp}) = \left( \prod_{q \in Q_p} \sum_{x_{mq}} p(I_m|x_{mq}) p(x_{mq}|x_{lp}) \right)^{1/|Q_p|}.
\]  (A.2)

**Proof.** Without loss of generalization, let us assume that \( Q_p = \{1, 2, \ldots, |Q_p|\} \). We start with the observation that

\[
\sum_{x_{m1}} \cdots \sum_{x_{m|Q_p|}} p(x_{m1}, x_{m2}, \ldots, x_{m|Q_p|}|x_{lp}, I_m) = 1.
\]

Under the rules of conditional independence, we express the joint probability in the equation above as the product of the individual probabilities of the child patches locations conditioned on the parent patch location and \( I_m \):

\[
\prod_{q \in Q_p} \sum_{x_{mq}} p(x_{mq}|x_{lp}, I_m) = 1.
\]  (A.3)
Using Bayes’ rule to write the conditional probability of the child patch location given the parent patch in terms of a data likelihood term and a prior term, we obtain

\[
\prod_{q \in Q_p} \sum_{x_{mq}} \frac{p(I_m|x_{mq},x_{lp})p(x_{mq}|x_{lp})}{p(I_m|x_{lp})} = 1, \text{ or}
\]

\[
\prod_{q \in Q_p} \sum_{x_{mq}} p(I_m|x_{mq},x_{lp})p(x_{mq}|x_{lp}) \frac{p(I_m|x_{lp})}{(p(I_m|x_{lp}))^{|Q_p|}} = 1. \tag{A.4}
\]

Using the assumption that the LoG \(I_m\) does not depend on \(x_{lp}\) given \(x_{mq}\), and rearranging terms, we obtain the form for \(p(I_m|x_{lp})\) in Equation (A.2). For \(p(I_3|x_{21})\), we set \(m \leftarrow 3, l \leftarrow 2, p \leftarrow 1, \) and \(Q_p \leftarrow \{1, 2\}\) to get the form in Equation (A.1). \(\square\)


BIBLIOGRAPHY


T. Bachmann. The importance of the integration of road, tyre, and vehicle technologies. In FISITA World Congress, 1995. 115


BIBLIOGRAPHY


T. Dieckmann. Assessment of road grip by way of measured wheel variables. In *FISITA Congress*, volume 2, pages 75–81, 1992. 115


C. Goad. Special purpose automatic programming for 3D model-based vision. In Proceedings of ARPA Image Understanding Workshop, 1983. 21


BIBLIOGRAPHY


144


146


BIBLIOGRAPHY


BIBLIOGRAPHY


A. Slater and V. Morrison. Shape constancy and slant perception at birth. Perception, 14:337–344, 1985. 15


153
BIBLIOGRAPHY


I. Stewart. *What Shape is a Snowflake?* W. H. Freeman, 2001. 16, 77


X. Xiong, D. Munoz, J. Bagnell, and M. Hebert. 3-D scene analysis via sequenced predictions over points and regions. In *IEEE International Conference on Robotics and Automation*, 2011. 31


