Modeling and Rendering the Invisibles and the Impossibles from Single Images: a Human-Computer Interaction Approach

by

YEUNG, Sai Kit

A Thesis Submitted to
The Hong Kong University of Science and Technology
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy
in the Department of Electronic and Computer Engineering

July 2009

Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the Hong Kong University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

_____

YEUNG, Sai Kit

Modeling and Rendering the Invisibles and the Impossibles from Single Images: a Human-Computer Interaction Approach

by

YEUNG, Sai Kit

This is to certify that I have examined the above PhD thesis
and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by
the thesis examination committee have been made.

Professor Limin ZHANG, Thesis Supervision Committee Chairman

Professor Oscar C. AU, Thesis Supervisor

Professor Chi-Keung TANG, Thesis Supervisor

Professor Amine Bermak, Thesis Supervision Committee Member

Professor Roger Shu-Kwan Cheng, Thesis Supervision Committee Member

Professor Helen C. Shen, Thesis Supervision Committee Member

Professor Ross D. Murch, Head of Department

Department of Electronic and Computer Engineering, School of Engineering
July 2009

Acknowledgements

This thesis would not have been possible without the support of many people. Many thanks to my advisor, Prof. Chi-Keung Tang, who has given guidance to me throughout my undergraduate and postgraduate studies. Thanks to Prof. Oscar Au, another advisor who helps me in various issues. His patience and advice are highly appreciated. Thanks to my former advisor, Prof. Pengcheng Shi. His positive attitude benefits many ways in my life.

Thanks to Prof. Sing Bing Kang for his innovative directions and sagacious instructions. I would also like to thank Prof. Michael S. Brown and Prof. Philip Fu for their comments and suggestions. Thanks Prof. Tony F. Chan for the research discussion during my visit in UCLA. Thanks Prof. Max A. Viergever and Prof. Josien Pluim for the guidance and supports during my visit in Utrecht.

Many thanks go to the members of my committee who carefully examine my work and attend my defense in person: Prof. Pheng-Ann Heng, Prof. Amine Bermak, Prof. Roger Shu-Kwan Cheng and Prof. Helen C. Shen. Their advice and patience are highly appreciated.

Special thanks to Tai-Pang Wu, who gave many ideas and implementation on impossible figures especially the 3D approach. Thanks Ruonan Pu, Jia Chen, Yu-Wing Tai and Heung-Sun Ng who give me different supports.

Thanks to my parents and my sister, for their love, support and encouragement ever since I was born. They are the most wonderful people in the world. Many sweet memories go to my pet dog, Siu Don Bo, who stayed with me for 17 years. Thanks for his coming to my home. Without them, this thesis would not have been possible.

And finally, thanks to all my friends. The support they gave me has supplied me endless energy source to continue my work even during the harshest moments.

To my mother and father

Contents

List of Figures

List of Tables

Modeling and Rendering the Invisibles and the Impossibles from Single Images: a Human-Computer Interaction Approach

by YEUNG, Sai Kit

Department of Electronic and Computer Engineering

The Hong Kong University of Science and Technology

ABSTRACT

A human being has a remarkable ability to make 3D connection on seeing 2D images. For example, although transparent objects are invisible to a large extent, we can mentally infer the 3D solid almost instantly and effortlessly. This ability, however, sometimes leads to interesting problems. M. C. Escher was a renowned artist who specialized in embedding impossible figures into architectural drawings. Impossible figure is a special kind of drawing consisting of multiple geometrically possible units connected by linear structures. When the figure is viewed as a whole, structural inconsistencies arise and confuse our visual perception.

The theme of this thesis consists of an appearance-based computational framework for modeling the "invisibles" and the "impossibles", that is transparent objects and impossible figures. We applied our appearance-based model in computer graphics, allowing for the first time the rendering of transparent objects in a new background scene, as well as the rendering of impossible figures at high frame rates, both without any tedious 3D modeling. The key in our approach lies in bringing the user into the modeling loop, i.e., human-computer interaction (HCI).

Human-computer interaction, which can be understood as the supplying of a small amount of hints to help automatic computer algorithm to solve difficult problems, has recently gained a lot of attention in research in computer vision and interactive techniques. The main issue lies in improving computer's performance by user interaction using a

simple interface. This thesis presents a HCI approach to model and render transparent objects (the invisibles) and impossible figures (the impossibles) which are traditionally very difficult in computer vision and graphics.

In this thesis, we exploit our remarkable human visual system to provide prior knowledge, and propose an HCI approach to transfer such prior information in the form of a few simple hints via an easy and intuitive user interface. The computer algorithm then automatically performs the rest of the processing.

We first transform the problem of transparent layer extraction into a soft-segmentation problem, where simple user's hints are available in the form of rough strokes drawn on the image. Then, with the aid of a human user who can easily recognize transparent objects given a single photo, we derive a practical and interactive approach to solve the problem of matting and compositing of transparent and refractive objects. Traditionally, these tasks can only be achieved using multiple images or 3D models. Finally, we present two approaches to model and render impossible figures, which is a long-standing problem in computer vision and computer graphics.

Chapter 1

Introduction

1.1 Background

Human-computer interaction (HCI) lies at the crossroads of many scientific areas including artificial intelligence, computer vision, face recognition, motion tracking, etc. In recent years there has been a growing interest in improving interaction between humans and computers, allowing the human user to interact naturally with the computer, similar in the way human-human interaction takes place [70]. The basic goal of HCI is to improve the interactions between users and computers by making computers more usable and receptive to the user's needs, which will lead to improved results (Figure 1.1). In computer vision and graphics, HCI can be translated into a simple and intuitive user interface to bridge the gap between human and computer, allowing problems which are traditionally very difficult to be solved by a fully automatic computer vision algorithm. HCI also has a high potential in eliminating tedious user intervention typical in many computer graphics problems.

Traditional computer vision approaches focus on deriving automatic algorithm for various tasks. An example is, in environment matting, a technique for modeling the complex light-transport properties of real-world optically active elements such as transparent, refractive and reflective objects. Computer vision approach targets at establishing correspondences across multiple images [93]. Once the correspondences are established, different registration techniques [9, 7, 82, 106, 101, 102, 105] can fuse the information

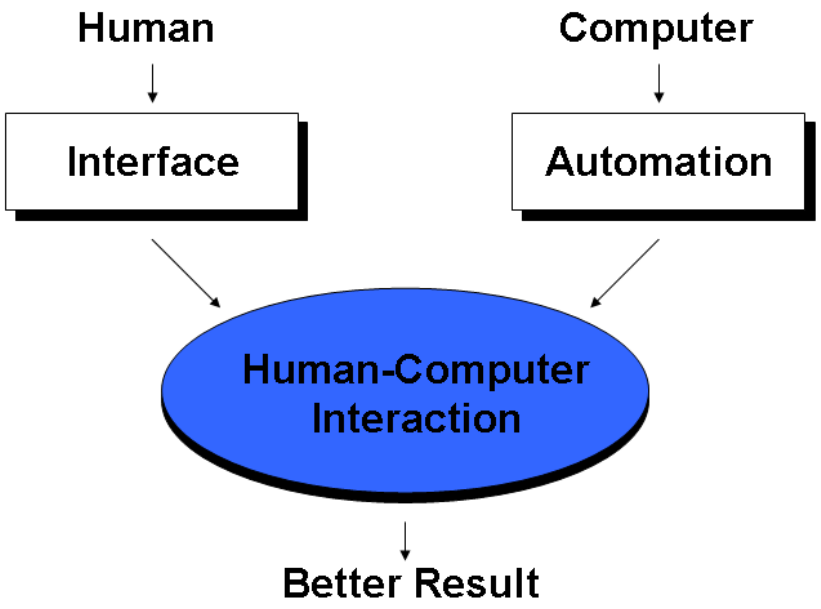


Figure 1.1: The basic goal of HCI is to improve the result that can be achieved by either human or computer alone.

from different images together and solve the environmental matting problem. The whole process is fully automatic.

On the other hand, computer graphics experts propose methods for obtaining environment mattes of real-world objects by using a complicated laboratory setup. These systems illuminate the real objects with carefully calibrated backgrounds, and capture images of the appearance of the object under these backgrounds [107, 15, 61]. Analysis of the images allows the objects' light-transport properties to be computed. Matusik et al. [55] used a turntable, multiple cameras, multiple lights, and monitors (as backgrounds) to capture an environment matte all around a transparent object. These techniques permit the discovery of complex optical behavior of real-world objects without explicit measurement of geometry or transmissivity parameters, and have yielded impressive composite images. However, there are a lot of tedious user involvements and the methods remain limited to situations where the object can be placed in a calibrated laboratory setting.

In addition, some problems can be very difficult or even impossible to be solved by either computer or human users. Consider the problem of environment matting using a *single image*. It is impossible for a computer algorithm to extract the transparent object

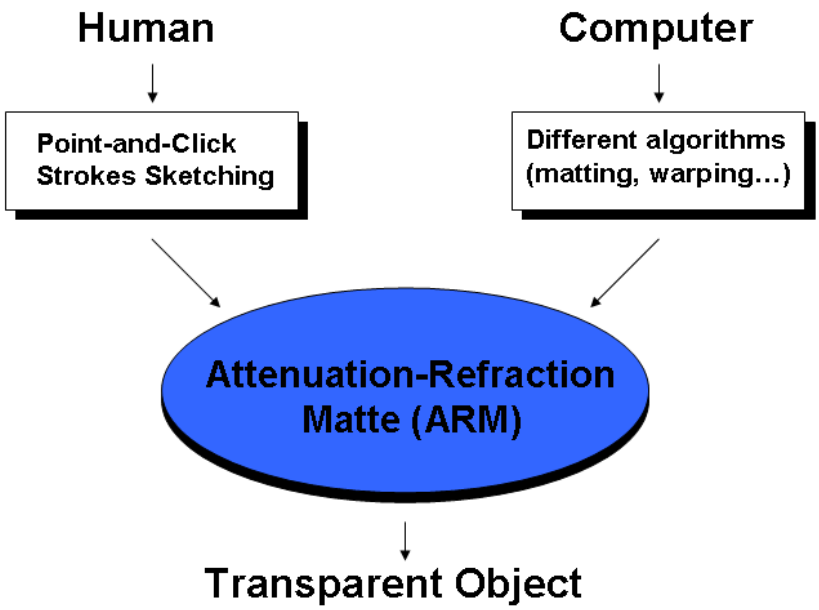


Figure 1.2: By proper integration of user interface and different automatic algorithms, we can perform matting and compositing of transparent objects from a single image which are traditionally very difficult or even impossible by either computer or human.

from a single image since no correspondences can be obtained from a single image without any prior information. On the other hand, the task will also require tremendous efforts for a human user to perform using existing photo-editing tools. By the proper formulation to fuse human interaction and computer automation, this thesis shows that it is possible to solve this difficult problem in a simple and easy way while producing very good results. The integration of human and computer is the key issue in this thesis.

1.2 Contributions

Although transparent objects are ubiquitous, practical approaches in automatic acquisition or modeling of transparent objects remain largely unavailable because transparent objects violate almost all assumptions used in computer vision. This thesis proposes an interactive approach by bringing the user into the modeling loop.

To achieve such human-computer integration, we propose an easy-to-use interface that utilizes the remarkable human recognition ability to solve problems which are traditionally very difficult to be solved by either computer or human only.

Taking transparent matting and compositing as an example (Figure 1.2), by deploying familiar user interfaces such as point-and-click and strokes sketching, we propose a new image matting algorithm which works in tandem with the user interface and automatic algorithm, bridging the gap between human and computer to allow high-quality transparent matting and compositing within minutes and with great ease. The main contribution, or the theme of this thesis, is to propose and to provide solid implementations for the "human" and "computer" components under the auspices of human-computer interaction (HCI) to solve important problems in computer vision and graphics which are relatively unexplored: "invisible" and "impossible" objects:

1.2.1 Single-Image Transparent Layer Extraction [104]

Decomposing or extracting layers from a single image is an ill-posed problem because there are more unknowns than equations. We will transform the problem into a soft-segmentation problem and present an interactive approach where a small amount of color samples are input by the user. Our system will automatically extract coherent transparent layers from a single image. The extracted transparent layers are immediately ready for composition into a new image. The technical contribution is the novel use of the *Expectation-Maximization* (EM) algorithm running over a *Markov Random Field* (MRF) which considers first-order-neighborhood. Our layer results are more homogeneous and less susceptible to the structure caused by the observed background.

1.2.2 Single-Image Matting and Compositing of Transparent and Refractive Objects [103]

We will introduce a new approach for matting and compositing of transparent and refractive objects in photographs. The key to our work is a new image-based matting model, termed the *attenuation-refraction matte* (ARM), that encodes plausible refractive properties of a transparent object along with its observed specularities and transmissive properties. We show that an object's ARM can be extracted directly from a photograph using simple user markup. Once extracted, the ARM is used to paste the object onto a

new background with a variety of effects, including compound compositing, Fresnel effect, scene depth, and even caustic shadows. Our approach allows photo-editing of transparent and refractive objects in a manner that produces realistic effects previously only possible via 3D models or environment matting.

1.2.3 Single-Image Modeling and Rendering of Impossible Figures [94]

We will discuss the difficult problem of modeling and rendering impossible figures from a single image. Impossible figures have long been used in applications such as computer games, non-photorealistic rendering, image synthesis, and photomontage. They are often blended with geometrically-possible scenes to create special effects in computer graphics applications. Our problem is further complicated with only a single image as the input. We will present two practical approaches, one in 2D and the other in 3D, to generate plausible novel views of the impossible figures. In the first approach, we will show how layered segmentation, normal assignment, and simple 2D line fitting followed by image warping can solve the problem. To improve the quality and efficiency, we will present another approach based on view-dependent modeling. Given a set of 3D locally-possible parts of the figure, our algorithm automatically optimizes a view-dependent 3D model, subject to the necessary 3D constraints for rendering the impossible figure at the desired novel viewpoint. A closed-form solution to the optimization problem is derived, thereby allowing an efficient computation and rendering new views of impossible figures at interactive rates. Once the optimized model is available, a variety of compelling rendering effects can be applied to the impossible figure.

1.3 Structure of the Thesis

This thesis is organized as follows:

In Chapter 2, we review related literature on layers extraction, matting of transparent objects and modeling and rendering of impossible figures.

In Chapter 3, we present an Expectation Maximization (EM) framework which incorporates Markov Random Field (MRF) for the single-image layer extraction problem.

Chapter 4 introduces our novel attenuation-refraction matte (ARM) model for matting transparent and refractive objects from single images. We will also present a practical and interactive approach to extract such an image-based model.

In Chapter 5, we discuss the issues involved in the composition of an extracted ARM to a new background. We propose practical algorithms to add a touch of realism despite no 3D model being available.

Chapter 6 describes a 2D approach in modeling and rendering of impossible figures. Chapter 7 presents our 3D modeling approach, which is inspired by the 2D approach and produces significantly better results with significantly less user interaction.

Finally, we conclude the thesis in Chapter 8.

Chapter 2

Literature Survey

In this chapter, we review previous works related in general to layer extraction, environment matting and modeling of impossible figures.

Specific works related to single-image and multiple-image approaches in modeling and rendering of transparent objects and impossible figures will be reviewed in the respective chapters.

2.1 Layer Extraction

Layer extraction is a useful problem in computer vision such that the extracted layers can be analyzed, enhanced or removed, for example, removing unwanted reflection layer from photographs. The problem is under-constrained because the number of layers in general is larger than the number of images. Thus, researchers have proposed different approaches to constrain the layer extraction problems such as depth from focus [69]. They consider situations where the depth at each point in the scene is multi-valued, due to the presence of a virtual image semi-reflected by a transparent surface. This semi-reflected image is linearly superimposed on the image of an object that is behind the transparent surface. By focusing on either of the layers to yield initial separation, and perform mutual blurring from an estimated defocus blur kernels, they separate the superimposed layers from the image.

By using multiple images, [38] and [79] used motion difference between the reflected

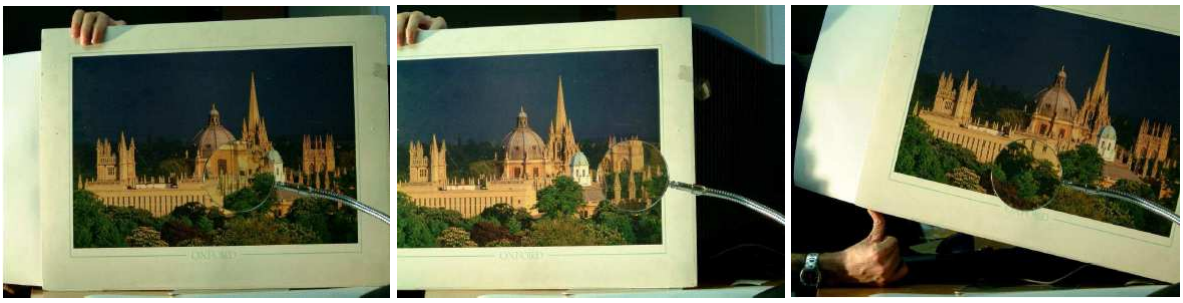


Figure 2.1: Environment matting by multiple images. The input is a set of images of an optically active element the magnifying glass in front of a moving background [93]. Notice that the glass cover different regions of the background so correspondences can be established for further processing.

(a) (b) (c)

Figure 2.2: Environment matting by complicated laboratory setup. To capture the light transport properties of a transparent object, complicated laboratory setup is required. It involves carefully calibrated backgrounds, calibrated multiple cameras, multiple light sources to capture images of the appearance of the object under these specific conditions. Laboratory setup in [107, 15, 55] are depicted in (a)-(c) respectively.

and non-reflected layers to perform separation. One may also assume the motion is repetitive to enforce additional constraint on the separation problem [66]. Another way to constrain the solution is to assume a fixed number of layers presented in the solution, e.g, only two layers [18, 72]. From a single image, in [48], a user-assisted approach that employs a sparsity prior was proposed to separate an image into two layers: the reflection (transparent) layer and background (opaque) layer. This is a similar approach to ours in the sense that they also make use of user interaction to simplify the extraction problem. The difference is that we do not constrain the number of layers being extracted.

2.2 Environment Matting

Environment matting, which captures not just a foreground object and its traditional opacity matte from a real-world scene, but also a description of how that object refracts and reflects light, is known to be a very difficult problem in computer vision and graphics. The extracted foreground object represented in terms of environment matte can be placed in a new environment.

Wexler et al. [93] propose an image-based environment matting algorithm by using multiple images to solve the problem. The multiple images are provided in a way that the transparent object is in different positions among the images so that correspondences can be established, as shown in Figure 2.1.

Apart from an automatic method by accepting multiple images as input, computer graphics researchers have proposed other approaches with complicated laboratory setup to solve the environment matting problem. For example, they put objects in front of known calibrated backgrounds [107, 15, 61]. Matusik et al. [55] used a turntable, multiple cameras, multiple lights, and monitors (as backgrounds) to capture an environment matte all around a transparent object. All these methods involve careful calibration and the results are difficult to reproduce. The environment mattes recovered by their methods, on the other hand, is very accurate due to the accurate laboratory set up adopted in their approaches.

2.3 Modeling of Impossible Figures

Recognizing the importance of operating in the 3D space, 3D geometric approaches have been proposed. Elber [16] created physical models for impossible figures, including those by M.C. Escher, Figure 2.3(a)-(b). Lipson [50] made use of LEGO bricks to build physical models of various impossible figures rendered by Escher. Manual construction of such 3D models can be a tedious process. Moreover, the resulting model only allows rendering the impossible figure at restrictive viewpoints. In contrast, our general approach takes both 2D and 3D into consideration, which can be used to render impossible figures in any one

of the four classes: depth interposition, depth contradiction, disappearing normals, disappearing space. Also, our efficient optimization algorithm models and renders impossible figures at interactive speeds.

Khoh and Kovesi [43] generated novel views of impossible figures by using two complementary halves, which are 3D models related by an inversion transform in the image plane (Figure 2.4). The thickness of the two complementary halves need to be adjusted after inversion transform. Their approach works for a particular subset of impossible figures. A similar approach was adopted by Tsuruno [85], where a 3D model of *Belvedere* is first constructed to create different views of *Belvedere*. In [76], an approach was introduced to generate unfolded surfaces that are geometrically possible, allowing one view of the impossible figure to be projected. Owada and Fujiki [60] proposed a system for modeling impossible figures. They also implemented a constraint solver to seamlessly combine multiple 3D parts in a projected 2D domain, by considering 3D line orientations to render an impossible figure at novel viewpoints.

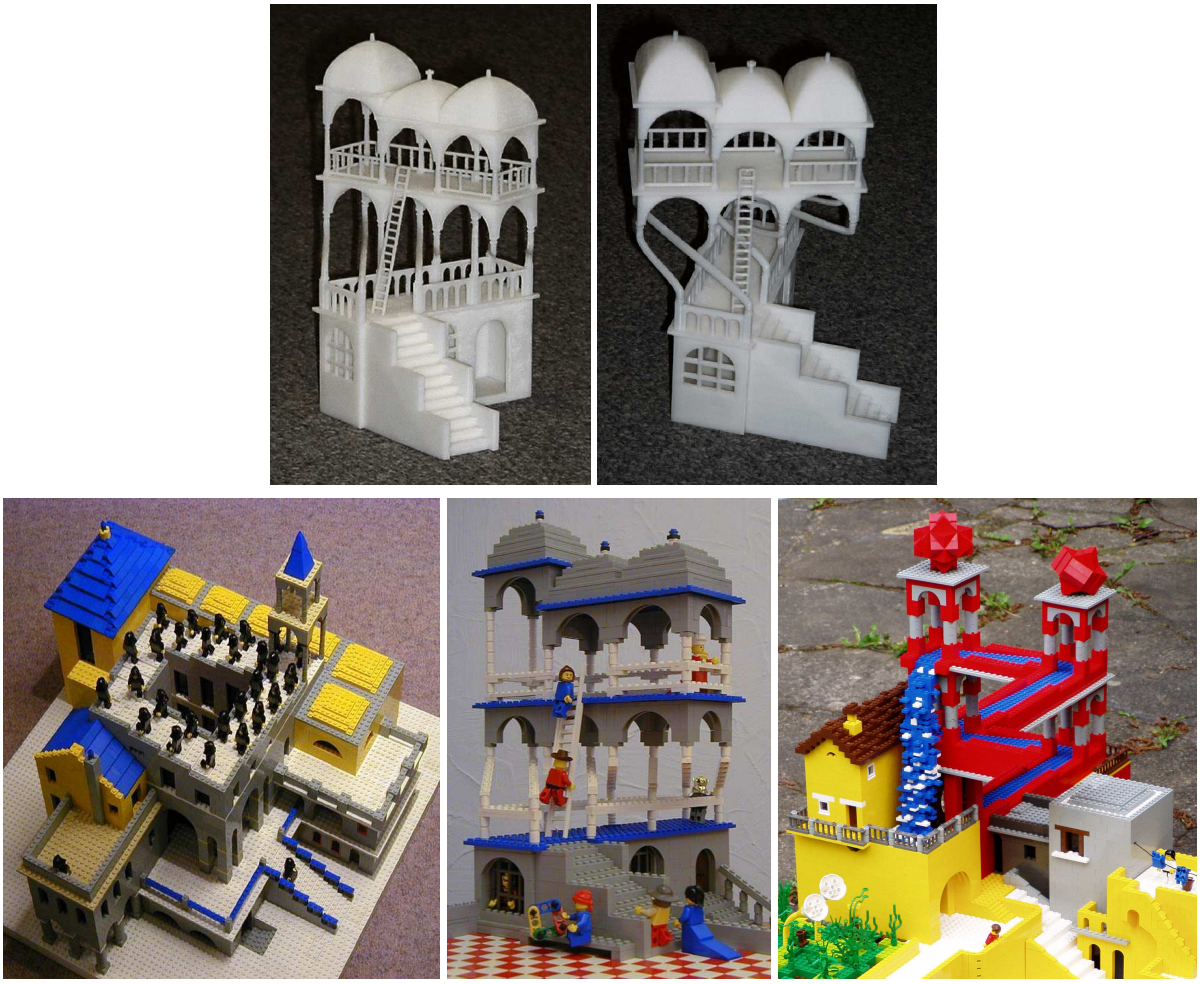
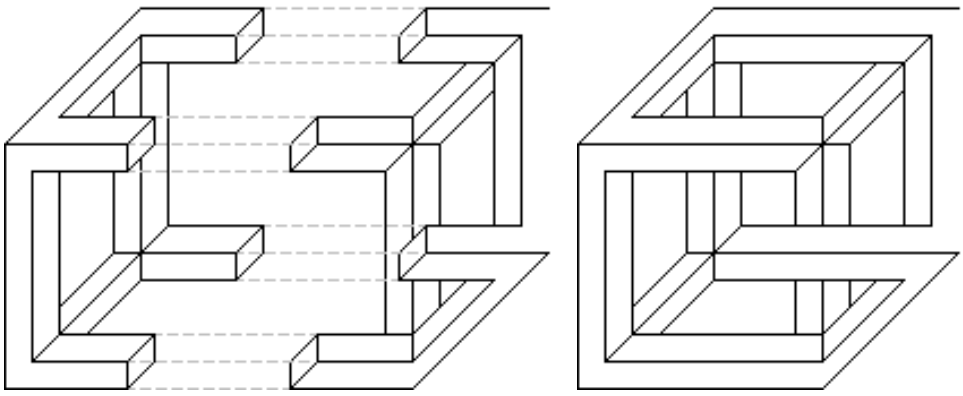


Figure 2.3: Physical models [16] and LEGO bricks models [50] of various impossible figures rendered by M.C. Escher.



(a) 2 complementary halves (b) combined impossible figure

Figure 2.4: Generating novel views of impossible figures by combining two complementary halves which are 3D models related by an inversion transform in the image plane [43].

Chapter 3

Single-Image Transparent Layer Extraction

Layer extraction from a single image is an under-constrained problem, because there are more unknowns than equations. This thesis studies a slightly easier but very useful alternative where only the background layer has substantial image gradients and structures. We propose to solve this useful alternative by means of soft segmentation by an Expectation-Maximization (EM) algorithm defined on a Markov Random Field (MRF). Our layer decomposition algorithm can maintain spatial coherency of smooth and overlapping layers, while preserving image details of the textured background layer. This algorithm can be applied to extract smooth shadows from single images, where the background covered by the shadow can have complex textures. Using the same algorithm, our method can be applied to extract hard or soft shadows of spatially-varying intensity. Unlike previous methods in shadow removal, no simplifying assumption on the camera or the light source is used. Very good results can be obtained both in layer decomposition and shadow extraction with only a small amount of user input.

3.1 Overview

The problem of separating a set of overlapping layers from a single image is a severely under-constrained problem. Previous approaches used depth from focus [69], multiple

images and motion [79, 33], repetitive motion [66], independent component analysis [18], and sparsity priors [48]. This thesis considers a slightly easier version of the problem: given a single image where only the background layer has substantial image gradients and structures, recovering the background layer as well as the overlapping/transparent layers is questionable. That is,

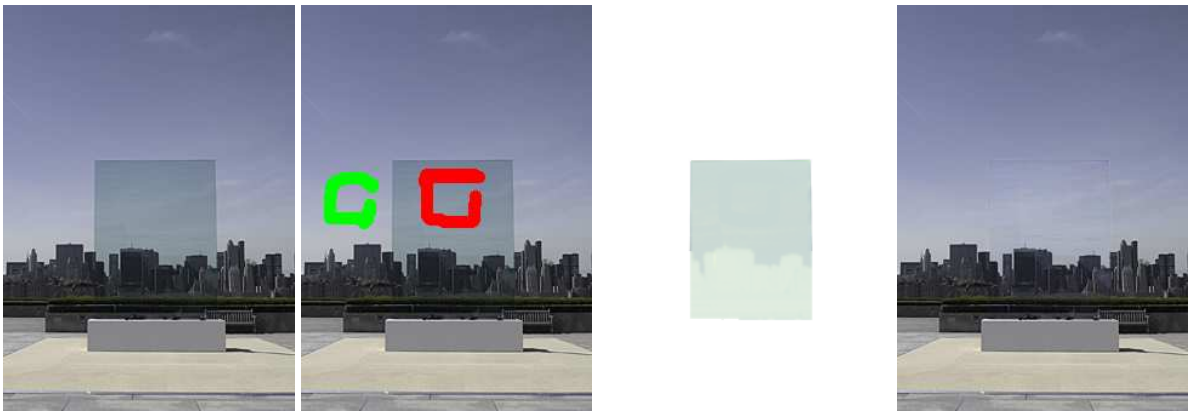
$$I(x, y) = F(x, y) + \beta(x, y)B(x, y) \tag{3.1}$$

where I is the input image. F is a set of overlapping layers possibly with soft and transparent boundaries. B is the background layer, which can be attenuated by a smooth transparent layer β. I, F, B and β are RGB vectors (β is modeled to respond differently in each color channel). This slightly easier problem is still ill-posed, as given a single image I there is still an infinite number of F, smooth β, and B that gives the same I. Suppose we first extract F, and let I' be the resultant image after extracting F, the equation can be reduced to a form equivalent to the intrinsic image representation [5]

$$I'(x, y) = \beta(x, y)B(x, y) \tag{3.2}$$

which was solved using multiple images [91] and a single image [81]. By taking the advantage of the smooth β assumption, this thesis takes an alternative approach to achieve better results by using a small amount of user interaction: Figure 3.1 shows an example of extracting a glass layer with substantial transparency. Note that the background is well separated from the glass layer. Figure 3.2 shows an extracted smooth shadow with a hard shadow boundary, indicating that both high and low frequency components co-exist in the layer. Note that the textures of the image B are preserved after shadow removal.

It turns out that the same algorithm proposed by us, which is formulated using Expectation-Maximization (EM) running over a Markov Random Field, can be used to extract both F and β. If β is not refractive, B can be simply obtained by I'/β. Natural image matting, which has been used to extract, from a single image, overlapping layers with transparent boundaries relates to our work. Figure 3.3 demonstrates that while cur-



(a) (b) (c) (d)

Figure 3.1: Transparent layer extraction. (a) Input image, (b) user scribbles for collecting relevant color cues, (c) transparent layer β, and (d) the background image B extracted by our method.

rent state-of-the-art matting techniques [12, 77, 46, 47] can also be used to extract the martini glass, the F layers (shown here as the glass, highlight, and environment reflection layers combined) produced by our algorithm, which considers first-order spatial neighborhood, is more homogeneous and less susceptible to the structure caused by the observed background. As we shall demonstrate, our method outputs a set of color labels per pixel which serves to reduce the inherent color ambiguities. When properly employed, we believe that such reduction should be very useful to image matting algorithms in general.

We will first highlight some related work on layer decomposition and shadow elimination which is the main application of our EM algorithm. The EM formulation and MAP estimation will then be detailed in section 3.3. Section 3.4 presents some results on employing our EM algorithm in layer extraction which produces soft and overlapping image regions suitable for computational photography applications such as colorization, image matting and composition. Section 3.5 describes the main application of our EM algorithm on shadow extraction, where the motivation of our image-based approach, comparison with state of the art and results are presented.

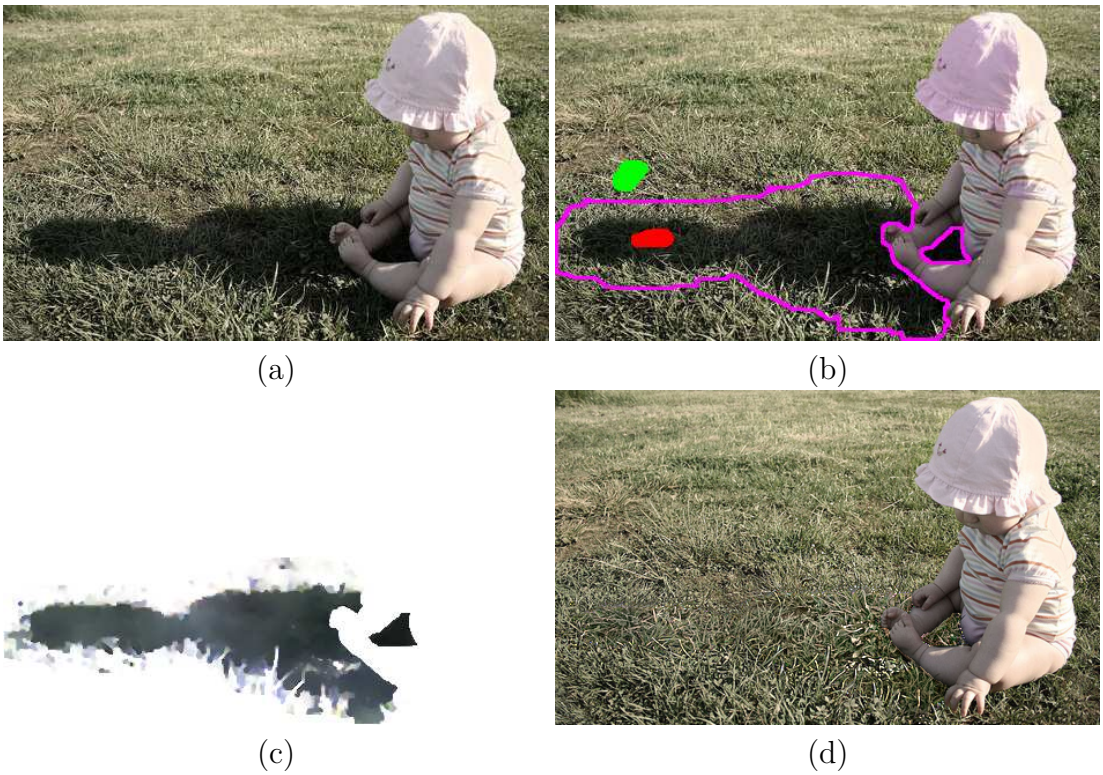


Figure 3.2: Shadow extraction. (a) Input image I. (b) Input strokes (red region denotes definitely attenuated region $\mathcal{S}$, green region denotes definitely unattenuated region $\overline{\mathcal{S}}$, processing region is within the pink border). (c) extracted shadow β, (d) the background B after removing the shadow. Note that the extracted transparent shadow is smooth, of spatially-varying intensity, and free of grass textures. The region delimited by pink loops are processing regions.

3.2 Related Work

We first review related work in layer decomposition, namely, layer separation and image matting. Then, related work in shadow removal and matting will be reviewed.

3.2.1 Layer Decomposition

Intrinsic image separation Shadow and shading removal are often addressed along with intrinsic image estimation, which is one of the most important problems in image layer separation: the separation of the illumination layer and reflectance layer from single or multiple images. Weiss [91] used a sequence of N images to derive one reflectance image and N illumination images. Matsushita et al [53] estimated intrinsic images from image sequences with biased illumination. In [81], Tappen et al, separated intrinsic images from

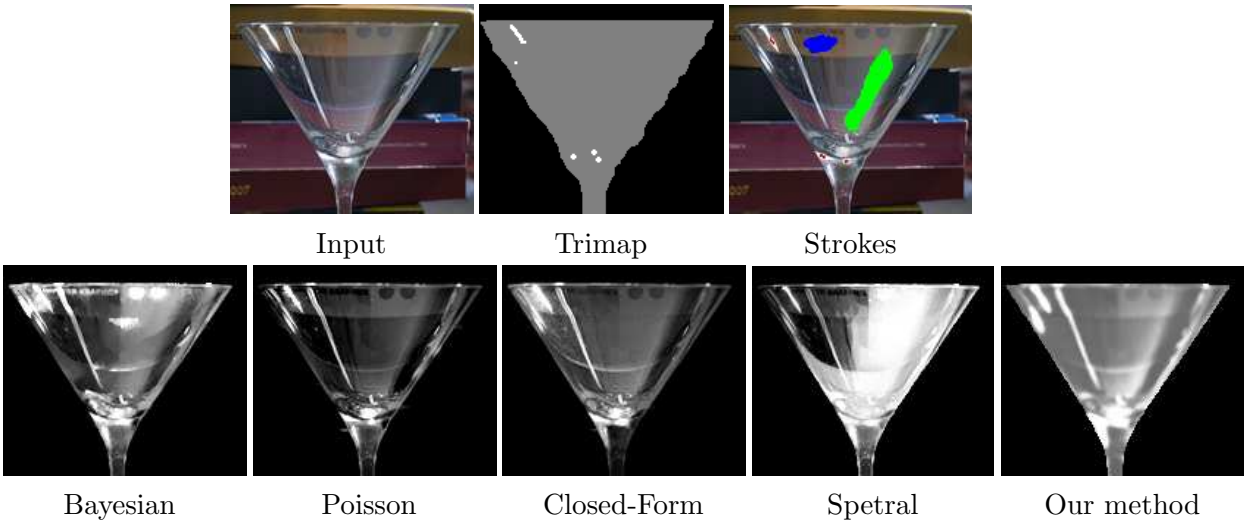


Figure 3.3: Comparison with state-of-the-art matting techniques. The input image, the trimap (used in Bayesian and Poisson matting), the input strokes (used in our method), and the F results of natural image matting using Bayesian matting, Poisson matting, the closed-form solution, and our method. For this example, our result is less susceptible to the background textures or structures.

a single image by classifying image derivatives as changes due to either reflectance or shading, followed by belief propagation to correct ambiguous regions.

Layer separation Deriving compositing layers from a single image or multiple images is also closely related to shadow extraction, because a shadow can be regarded as a semi-transparent layer over the shadowed region. Algorithms on two-layer separation were proposed in [18, 72]. By using multiple images, [38] and [79] used motion difference between the reflected and non-reflected layers to perform separation. From a single image, in [48], a user-assisted approach that employs a sparsity prior was proposed to separate an image into two layers: the reflection (transparent) layer and the background (opaque) layer. Similar to the shadowing effect over the shadowed region, subsurface scattering effect is also semi-transparent over the object of interest. In [99], a model-based approach was proposed which uses an image acquisition system to decompose a set of photometric images into three photometric components. In [100], a model-less approach was presented which separates subsurface scattering into an aggregate term. Layer separation is a massively ill-posed problem in its general form. While some progress has been made, the reported results are still far from usable in real applications. On the other hand, by as-

suming smooth shadows and using simple user interaction, we show that the problem of shadow extraction can be solved to a considerable extent.

Image and shadow matting It is possible to use techniques in natural image matting [12, 77, 90, 46] from a single object foreground layer from the background. In [75], the authors provide a unifying framework to represent the algorithms in [46] and [88] and extend the algorithms to estimate the mattes for multiple image layers. The mattes are extracted under a constrained optimization which enforces that the alpha mattes at each pixel take values in $[0, 1]$ and sum up to 1. The authors also discuss two cases where either of these two constraints can be relaxed and analyze the properties of the resulting solutions. In our formulation, it will be shown that these two constraints will be naturally handled in Eqn. 3.7.

However, these approaches are designed to extract foreground objects which are largely opaque. In layer extraction, however, overlapping layers are largely transparent or semi-transparent over the background. The approach that addressed shadow matting [14] makes several assumptions about the scene lighting and requires a video sequence to solve for the shadow matte. Only hard shadows are extracted in [14]. Natural shadow matting [97] extracted both shadow and shadowless image from a single image. This energy minimization approach achieved reasonably good results within seconds by making use of an ad-hoc shadow probability map and a coarse shadowless image to perform the optimization. The coarse map and image are only approximations.

3.2.2 Shadow Elimination

Shadow removal Using the retinex algorithm proposed in the classical paper [45], an algorithm was proposed in [25] to remove a shadow from a single image, where the goal is to extract meaningful information from an image rather than producing an aesthetic image free of shadows. A shadow removal algorithm using one image was reported in [22], which uses entropy minimization to derive an illumination invariant grayscale image for shadow removal without resorting to any calibration, a significant advancement over an earlier work [24]. This technique, on the other hand, makes several assumptions, such as

Planckian lighting (e.g., sunlight) and narrow-band cameras [24, 22]. In [51], the authors proposed a new approach to shadow removal from a single image that preserves texture characteristics between the shadow and the lit area. While the above approaches are capable of identifying shadows and producing shadow-free images, they are not designed to *extract* the shadow from the image.

Intensity transfer Given a few rough user-supplied hints, it is possible to perform color transfer to remove shadows. For example, in [64, 92], the colors or intensities are assumed to follow a single Gaussian distribution, in which case, it is possible to transfer to the shadowed region the mean and standard deviation of the shadow-free colors. If the single Gaussian assumption is violated, however, the user needs to divide the shadow-free/shadowed regions into smaller patches manually to satisfy the assumption. However, even with additional user interaction, the color transfer approach may still produce unacceptable results because there is no provision for maintaining spatial coherence or handling textures or scene structure, especially along hard shadow boundary.

Color constancy Color constancy algorithms target at estimating the illuminant spectrum for compensating its contribution to apparent image colors, thus allowing shadows to be detected. Conventional color imaging models, however, have unpredictable accuracy. For example, the diagonal transformation model [27, 32] produces good results only when the camera is narrow-band and has non-overlapping sensitivity functions. In [19], the authors proposed the generalized diagonal transform which assumes low-dimensional reflectance and illuminant spectrum. The generalized diagonal transform [20] performs better, but assumes low dimensionality of the reflectance spectrum. In [23], the authors attempt to exploit the relationship between filtered and unfiltered RGBs by forcing the lights to be examined pair-wise without insisting on accuracy of the illuminant estimation but only to discriminate between them. They show promising results by applying their algorithm in a region-based manner. Other approaches require either illumination basis functions or camera sensitivity functions to be known [31, 52, 84]. In [83], a Bayesian color constancy approach was proposed, where shadows are detected by learning a shadow prior.

3.3 The Expectation-Maximization Formulation

Based on the analysis in section 3.1, our algorithm steps are:

1. Extract F (section 3.3.1). We first extract overlapping layers which can be opaque or substantially transparent like the martini glass shown in Figure 3.3. Our EM algorithm produces a set of color labels at each pixel.

2. Extract β (section 3.3.2). Next, we solve Eqn. 3.2. We use the same EM algorithm to determine the amount a pixel is attenuated by the smooth β. Given that β is smooth and B contains most image gradients, we incorporate these considerations as optimization constraints in the Bayesian MAP estimation.

3.3.1 F-extraction

We develop an Expectation-Maximization (EM) algorithm to extract F. Our approach can be regarded as soft segmentation: for each pixel i, compute an optimal set of n soft labels, $\alpha_{ij} \in [0,1]$, $\sum_j^n \alpha_{ij} = 1$ where n is the total number of color segments in the image. In fact, many natural image matting techniques can also be used if $n = 2$ and if F is largely opaque. In our approach, we take advantage of the smoothness assumption, so we can extract $n \geq 2$ overlapping color segments which can be substantially transparent in front of a complex background.

The user-scribbled color samples (e.g. Figures 3.1–3.3) provide the necessary color constraints for our EM algorithm. Consider the two scenarios: 1) Suppose we know the expected color, then we can estimate the corresponding soft region label. 2) Suppose on the other hand the soft region label is known, it will help the color label estimation. In our case, both colors and soft labels are unknown. This becomes a "chicken-and-egg" problem. We propose to optimize the two variables by alternating optimization. EM algorithm, which is one form of alternating optimization guaranteed to converge [8], is a good choice of algorithm.

The following EM derivations except the novel use of MRF are quite standard and should be familiar to readers with knowledge of EM (a gentle tutorial is available [8]).

In the M-step, we employ the MRF assumptions to preserve structural coherency by considering connectivity in each pixel's first-order neighborhood.

We model the collected color statistics, obtained via a stroke-based interface (shown in Figures 3.1–3.3) or k-means clustering (such as Figures 3.7–3.9), using Gaussians to allow uncertainty in the subsequent estimation. In our formulation the three color channels can be processed individually or together. They both produce similar visual results, so for simplicity of notations, the following equations assume a single channel.

Using terminologies typical of EM formulations, let $\mathbf{O} = \{\{\mu_1, \sigma_1\}, \{\mu_2, \sigma_2\}, \cdots, \{\mu_n, \sigma_n\}\}$, $j = 1 \cdots n$, be the set of observations where μ_j and σ_j are respectively the mean and standard variation of the colors sampled inside region j. Notice that σ_j are related to the range whether a particular color sample would fall into the corresponding region j and hence the smoothness. Users could scale up or down this parameter to control the final softness of the results. Let $\mathbf{R} = \{r_i\}$ be the set of hidden variables that describes the classes labels at all pixels. $r_i = j$ if pixel i belongs to region j. The cardinality of $\mathbf{R}$, $|\mathbf{R}|$, is equal to the total number of pixels N to be processed (e.g. the whole image, or the pixels within the pink border as shown in Figure 3.2). The objective function is given by

$$\mathbf{\Theta}^* = \arg\max_{\mathbf{\Theta}} P(\mathbf{O}, \mathbf{R}|\mathbf{\Theta}) \tag{3.3}$$

where $P(\mathbf{O}, \mathbf{R}|\mathbf{\Theta})$ is the complete-data likelihood to be maximized and $\mathbf{\Theta} = \{c_i\}$ is a set of parameters to be estimated, where c_i is the expected color at pixel i. To estimate $\mathbf{\Theta}^*$, the EM algorithm computes the expected value of the complete-data log-likelihood $\log P(\mathbf{O}, \mathbf{R}|\mathbf{\Theta})$ with respect to $\mathbf{R}$ given the observation $\mathbf{O}$ and the current estimated parameter $\mathbf{\Theta}'$:

$$Q(\mathbf{\Theta}, \mathbf{\Theta}') = \sum_{\mathbf{R} \in \varphi} \log P(\mathbf{O}, \mathbf{R}|\mathbf{\Theta}) P(\mathbf{R}|\mathbf{O}, \mathbf{\Theta}') \tag{3.4}$$

where φ is the space containing all possible $\mathbf{R}$ with cardinality equal to N.

Expectation

We first define the marginal probability $p(\mathbf{O}|r_i, \mathbf{\Theta'})$ so that we can maximize the expectation Q defined by Eqn. 3.4 by proceeding to the next iteration given the current parameter estimation. If $r_i = j$, c_i should be similar to μ_j:

$$p(\mathbf{O}|r_i, \mathbf{\Theta'}) \propto \begin{cases} \exp(-\frac{|c_i-\mu_1|^2}{2\sigma_1^2}) & \text{if } r_i = 1 \\ \vdots & \\ \exp(-\frac{|c_i-\mu_n|^2}{2\sigma_n^2}) & \text{if } r_i = n \end{cases} \tag{3.5}$$

Without any prior information, we let $p(r_i = j|\mathbf{\Theta'}) = \frac{1}{n}$ be the mixture probability. Given $\mathbf{\Theta'}$ only, we have:

$$p(\mathbf{O}|\mathbf{\Theta'}) \propto \frac{1}{n}\sum_j \exp(-\frac{|c_i - \mu_j|^2}{2\sigma_j^2}) \tag{3.6}$$

Let α_{ij} be the probability of pixel i belonging to region j, which is the output soft label we need. Then, $\alpha_{ij} = p(r_i = j|\mathbf{O}, \mathbf{\Theta'})$, or

$$\begin{aligned} \alpha_{ij} &= \frac{p(r_i = j, \mathbf{O}|\mathbf{\Theta'})}{p(\mathbf{O}|\mathbf{\Theta'})} = \frac{p(\mathbf{O}|r_i = j, \mathbf{\Theta'})p(r_i = j|\mathbf{\Theta'})}{p(\mathbf{O}|\mathbf{\Theta'})} \\ &= \exp(-\frac{|c_i - \mu_j|^2}{2\sigma_j^2}) / \sum_m \exp(-\frac{|c_i - \mu_m|^2}{2\sigma_m^2}) \end{aligned} \tag{3.7}$$

From Eqn. 3.7, it can be noticed that the soft labels (mattes) α_{ij} at each pixel takes values in $[0, 1]$ and sum up to 1 across the different image layers.

Maximization

Given the marginal distribution α_{ij} estimated in the E-Step, we can maximize the likelihood in Eqn. 3.3 by optimizing the parameters in Eqn. 3.4 using the estimated α_{ij}. We make use of the assumption of smooth layers, and decompose $P(\mathbf{O}, \mathbf{R}|\mathbf{\Theta})$ into a combination of simple elements based on the Markov Random Field (MRF) assumptions:

(1) The hidden variable r_i depends only on the hidden variables of its first-order-neighbors.

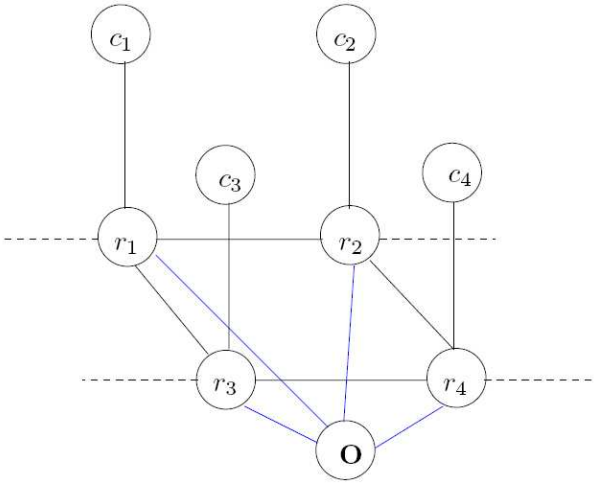


Figure 3.4: The MRF model for estimating the set of soft labels at each pixel.

(2) The observation at i depends only on the hidden variable at i.

These assumptions incorporate the smoothness consideration. See Figure 3.4 for a graphical representation of the MRF assumptions. Therefore:

$$P(\mathbf{O}, \mathbf{R}|\mathbf{\Theta}) = \prod_i \prod_{k \in \mathcal{N}_i} p(r_i|r_k, \mathbf{\Theta}) p(\mathbf{O}|r_i, \mathbf{\Theta}) \tag{3.8}$$

where $\mathcal{N}_i$ is a set of right and bottom neighbors of i and $p(r_i|r_k, \mathbf{\Theta}) = 1$ if k does not exist.

With Eqn. 3.8, $Q(\mathbf{\Theta}, \mathbf{\Theta}')$ in Eqn. 3.4 can be rewritten as:

$$\begin{aligned} & \sum_{\mathbf{R} \in \varphi} \log P(\mathbf{O}, \mathbf{R}|\mathbf{\Theta}) P(\mathbf{R}|\mathbf{O}, \mathbf{\Theta}') && (3.9) \\ = & \sum_{\mathbf{R} \in \varphi} \log \{ \prod_i \prod_{k \in \mathcal{N}_i} p(r_i|r_k, \mathbf{\Theta}) p(\mathbf{O}|r_i, \mathbf{\Theta}) \} P(\mathbf{R}|\mathbf{O}, \mathbf{\Theta}') \\ = & \sum_{\mathbf{R} \in \varphi} \sum_i \sum_{k \in \mathcal{N}_i} \log \{ p(r_i|r_k, \mathbf{\Theta}) \} p(r_i, r_k|\mathbf{O}, \mathbf{\Theta}') \\ & + \sum_{\mathbf{R} \in \varphi} \sum_i \log \{ p(\mathbf{O}|r_i, \mathbf{\Theta}) \} p(r_i|\mathbf{O}, \mathbf{\Theta}') && (3.10) \end{aligned}$$

Here, we have two terms to be defined: $p(r_i|r_k, \mathbf{\Theta})$ and $p(r_i, r_k|\mathbf{O}, \mathbf{\Theta}')$ (Note that $p(\mathbf{O}|r_i, \mathbf{\Theta})$ was defined similarly as Eqn. 3.5 and $p(r_i = j|\mathbf{O}, \mathbf{\Theta}') = \alpha_{ij}$).

For $p(r_i, r_k|\mathbf{O}, \mathbf{\Theta}')$ which models the first-order connection between two adjacent nodes with hidden variables r_i and r_k, regardless of the values of r_i and r_k, it equals to 1 because of the MRF assumptions (Figure 3.4).

For $p(r_i|r_k, \mathbf{\Theta})$, we assume that if the neighborhood pixel pair belongs to the same region, the expected color should be similar, i.e., if $r_i = r_k$, the color c_i should be similar to the color c_k and vice versa.

Suppose the noise model obeys the Gaussian distribution, $p(r_i|r_k, \mathbf{\Theta})$ can be modeled as:

$$p(r_i|r_k, \mathbf{\Theta}) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{|c_i - c_k|^2}{2\sigma^2}) \tag{3.11}$$

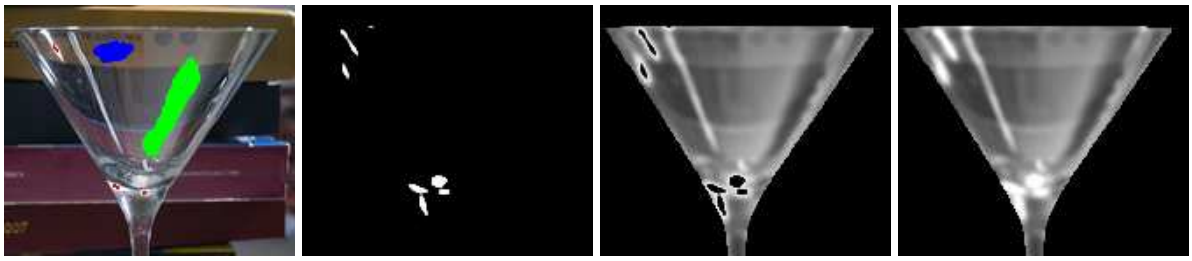
where σ^2 describes the variance of the region color. Since $r_i \in \{0, 1, \cdots, n\}$, we rewrite $Q(\mathbf{\Theta}, \mathbf{\Theta}')$ as

$$\begin{aligned} & Q(\mathbf{\Theta}, \mathbf{\Theta}') \\ = & \sum_{\mathbf{R}\in\varphi} \sum_i \sum_{k\in\mathcal{N}_i} \log\{p(r_i|r_k, \mathbf{\Theta})\} p(r_i, r_k|\mathbf{O}, \mathbf{\Theta}') \\ & + \sum_i \sum_j \log\{p(\mathbf{O}|r_i = j, \mathbf{\Theta})\} \alpha_{ij} \end{aligned} \tag{3.12}$$

$$\begin{aligned} = & \sum_i \sum_{k\in\mathcal{N}_i} \log\left(\frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{|c_i - c_k|^2}{2\sigma^2})\right) \\ + & \sum_i \sum_j \log\left(\frac{1}{\sigma_j\sqrt{2\pi}} \exp(-\frac{|c_i - \mu_j|^2}{2\sigma_j^2})\right) \alpha_{ij} \end{aligned} \tag{3.13}$$

where σ controls the measurement uncertainty (we simply set σ equals to the mean of all σ_j). To maximize Q, we differentiate Q w.r.t. c_i and set the first derivative equal to zero to obtain the parameter updating rule:

$$c_i = \left(\sum_{k\in\mathcal{G}_i} c_k + \sigma^2 \sum_j \frac{\alpha_{ij}}{\sigma_j^2} \mu_j\right) / \left(4 + \sigma^2 \sum_j \frac{\alpha_{ij}}{\sigma_j^2}\right) \tag{3.14}$$



Input strokes α_0 (by red stroke) α_1 (by green stroke) $F = \alpha_0 + \alpha_1$

Figure 3.5: Extracting transparent and overlapping layers from a single image. F for this martini glass will be the combination of the layers α_0 and α_1 which are extracted from the red and green strokes respectively.

where $\mathcal{G}_i$ is first-order neighbors of pixel i. Hence, in the M-Step, the updating rule (compute c_i by Eqn. 3.14) is applied. The E-Step (compute α_{ij} by Eqn. 3.7) and M-Step are iterated alternately until convergence. The initial assignment of c_i is set as the pixel's color I_i. Upon convergence, we obtain the solve labels α_{ij} for each pixel i. Figure 3.5 shows the result of the martini example shown in Figure 3.3. Here, we decompose the original martini glass into 3 layers according to the input strokes. F corresponds to the layers from the red and green strokes.

3.3.2 β-Extraction

After extracting αs which represent the soft segmentation result for F, our problem becomes one of solving Eqn. 3.2, that is, $I' = \beta B$. To extract β, the user marks up on the image to gather color samples in the background outside and inside of the transparent layer, where the two marked-up regions should have similar textures, e.g., Figure 3.6.

We denote the two regions respectively by $\mathcal{S}$ and $\overline{\mathcal{S}}$ (Figure 3.2), and denote the resulting observation of the two marked-up regions by $\mathbf{O}$. Letting $n = 2$, the EM algorithm is used to compute the probability α_{ij}, i.e., $j = 0$ denotes unattenuated background, and $j = 1$ otherwise.

Given the soft label α at each pixel, we can reconstruct its color without attenuation due to β in a Bayesian optimization framework, and remove/extract β from the single input image. The idea is that we first estimate the portion of the background image inside $\mathcal{S}$, and then we scale up the intensity in a permeation manner, starting from the

(a) (b) (c)

Figure 3.6: β-extraction. (a) user scribbles for collecting relevant color cues, (b) transparent layer β, and (c) the background image B extracted by our method.

boundary of $\mathcal{S}$ to the interior of the whole processing region, which is denoted by $\mathcal{U}$.

We use different estimation methods for $\mathcal{S}$ and $\mathcal{U}$ because the two regions exhibit different properties: the attenuation in $\mathcal{S}$ is relatively constant while that in $\mathcal{U}$ can be spatially varying. Moreover, as we shall see, the estimation performed in $\mathcal{U}$ requires a boundary condition, which is derived from the result obtained in region $\mathcal{S}$.

Note from Eqn. 3.2 that we can either optimize β, or the image B to obtain the result. We choose to optimize B, and the smooth β can be simply obtained by I'/B. Note that I' can be obtained after extracting F in the previous step.

Estimating B inside $\mathcal{S}$

The colors of B inside $\mathcal{S}$ are relatively easier to estimate. The β inside $\mathcal{S}$ is roughly constant. Since $\mathcal{S}$ and $\overline{\mathcal{S}}$ have similar textures, by Eqn. 3.2, the mean intensity of the pixels inside $\overline{\mathcal{S}}$ and the mean intensity of B inside $\mathcal{S}$ should be very similar. We let $I_{m\mathcal{S}}$ and $I_{m\overline{\mathcal{S}}}$ be the mean intensities of regions $\mathcal{S}$ and $\overline{\mathcal{S}}$ respectively, thus the β inside $\mathcal{S}$ is given by $\beta = \frac{I_{m\mathcal{S}}}{I_{m\overline{\mathcal{S}}}}$. Therefore, we can estimate the B inside $\mathcal{S}$ simply by pixelwise division I/β.

Estimating B inside $\mathcal{U}$

To extract the attenuation inside $\mathcal{U}$, we employ the Bayesian framework for solving B. We use Λ to denote the information obtained after executing our EM algorithm. We aim to solve the following objective function:

$$\begin{aligned} B^* &= \arg\max_B P(B|I', \Lambda) \\ &= \arg\max_B P(I'|B, \Lambda)P(B|\Lambda) \\ &= \arg\min_B L(I'|B, \Lambda) + L(B|\Lambda) \end{aligned} \tag{3.15}$$

where $L(\cdot) = \log P(\cdot)$. The remaining problem is on how to define the terms in Eqn. 3.15 such that optimization can be performed to estimate the solution.

Definition of $P(I'|B, \Lambda)$ The idea here is that when we scale up the intensity of the pixels in the attenuated region, we want to maintain the textures and structures (gradients) under the attenuation as much as possible. The input I' provides us with the necessary structural information, while at the same time, the shadow present in I' interferes with such information. The optimized Λ, on the other hand, provides us with a confidence measurement to discern whether the image gradient is caused by true image structure, or intensity difference due to attenuation, thus helping us to appropriately weigh the trustfulness of the first-order structure of I:

$$m_{x,y} = \exp(-\frac{||\alpha_x - \alpha_y||^2}{2\sigma_p^2}) \tag{3.16}$$

where $\{x, y\}$ is a first-order neighbor pair, and σ_p^2 controls the certainty of the measurement. $m_{x,y}$ indicates the trustfulness of the structure shared by pixel x and y. Given m, $P(I'|B, \Lambda)$ can be defined as the following:

$$P(I'|B, \Lambda) = \exp(-\frac{\sum_x \sum_{y \in \mathcal{N}_x} ||\nabla B_{x,y} - m_{x,y} \nabla(\frac{I'}{\beta'})_{x,y}||^2}{2\sigma_d^2}) \tag{3.17}$$

where $\nabla B_{x,y}$ is the gradient of B computed at x, and $\nabla(\frac{I'}{\beta'})_{x,y}$ is the gradient of $\frac{I'}{\beta'}$ at x,

where $\{x, y\}$ is a first-order neighbor pair. β' is the rough estimation of β obtained by normalizing the α image into a range image, where each pixel lies within $[\frac{I_{m\mathcal{S}}}{I_{m\overline{\mathcal{S}}}}, 1]$. From the optimization point of view, Eqn. 3.17 minimizes the gradient difference between B and $\frac{I'}{\beta'}$ weighted by the confidence measurement m. Unlike other gradient-based reconstruction methods, we consider $\frac{I'}{\beta'}$ instead of I' due to the use of the image model specified in Eqn. 3.2, where β affects the gradient of both the attenuated and the unattenuated image. Note that although β' is an approximation, our experiments show that it is sufficient for producing decent results. Note that only the pixels in the interior of $\mathcal{U}$ are variables, while the pixels on the boundary of $\mathcal{U}$ are constants. These latter pixels provide the boundary condition for scaling up the intensity of the pixels in the interior of $\mathcal{U}$.

Definition of $P(B|\Lambda)$ To design the prior $P(B|\Lambda)$, we observe that the attenuation is spatially smooth (dominated by low frequency components) except along a boundary (for example, the glass in Figure 3.1 and the shadow in Figure 3.2).

In theory, we want to minimize $||\beta_x - \beta_y|| = ||\frac{I_x}{B_x} - \frac{I_y}{B_y}||$, where $\{x, y\}$ is a first order neighbor pair. However, directly minimizing this will introduce non-convexity to the solution. Instead, we minimize the following $||I_x B_y - I_y B_x||$ based on the assumption that $\frac{I_x}{B_x} = \frac{I_y}{B_y}$ produces a similar effect on the optimization. Hence, the smoothness prior becomes:

$$\exp(-\frac{\sum_x \sum_{y \in \mathcal{N}_x} ||I_x B_y - I_y B_x||^2 m_{x,y}}{2\sigma_1^2}) \tag{3.18}$$

where σ_1 controls the certainty in the smoothness. $m_{x,y}$ is multiplied to weaken the effect of the smoothness prior at high frequency components (e.g., shadow and glass boundary in the previous examples).

In the presence of very hard boundaries and relatively smooth textures, we can impose more smoothness on the texture along a hard shadow boundary. The following expression models this constraint:

$$\exp(-\frac{\sum_x \sum_{y \in \mathcal{N}_x} ||B_x - B_y||^2 (1 - m_{x,y})}{2\sigma_2^2}) \tag{3.19}$$

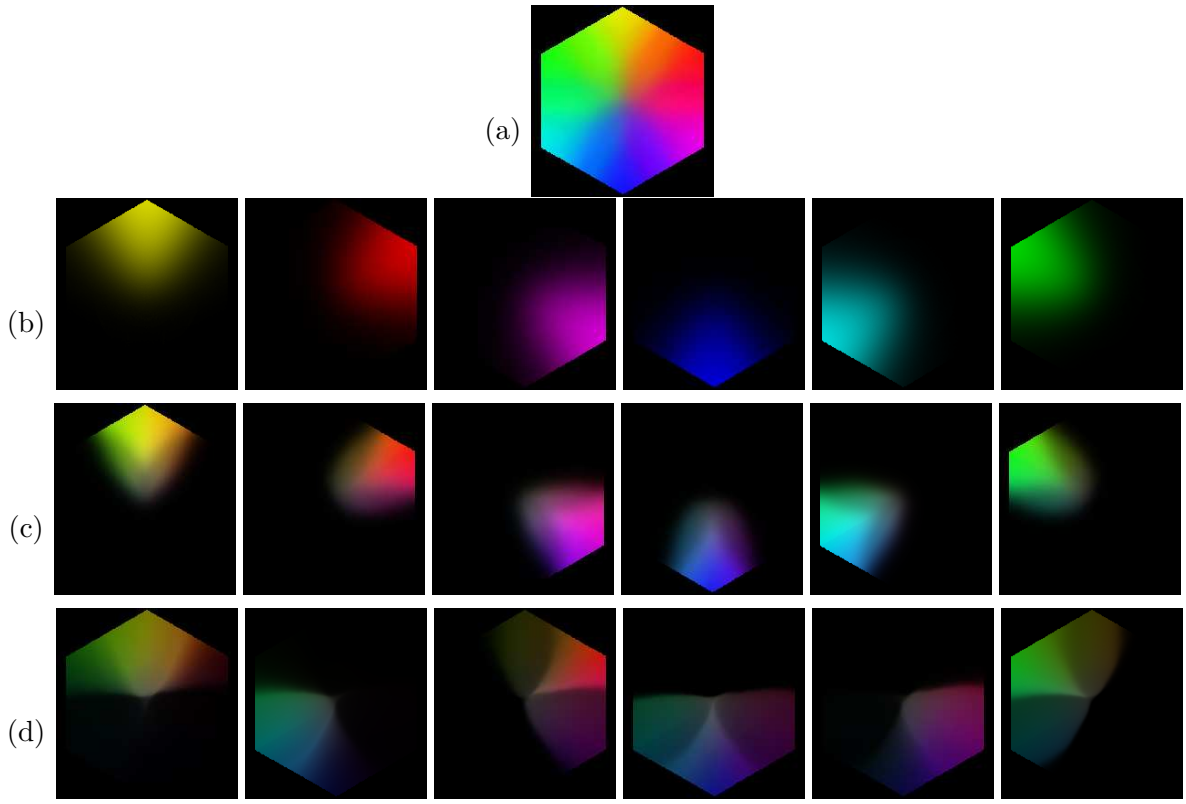


Figure 3.7: Synthetic case. (a) Input synthetic image where the observed color of a pixel may be explained by a mixture of as many as six colors. (b) The ground truth soft segments corresponding to (a). The soft segments produced by using (c) method in [80] and (d) our proposed method. Note that the estimated soft segments are displayed by multiplying the estimated soft labels with the input image.

where σ_2 is the uncertainty in the texture smoothness.

Thus, the complete prior $P(B|\Lambda)$ is the product of Eqn. 3.18 and Eqn. 3.19.

Optimization

In order to estimate B, we set the first derivative of the associated energy function of Eqn. 3.15 to zero and apply Gauss-Seidel method to obtain the solution. There is only one crucial parameter, σ_1, which explicitly controls the smoothness of the shadow. In all examples shown, $\frac{1}{2\sigma_1^2} = 1000$. σ_2 is set to a very large value (infinity) unless in the presence of hard boundary. σ_d^2 and σ_p^2 are always set to 1 and 0.0005.

After computing the background image B, the β image is simply extracted by I'/B due to Eqn. 3.2.

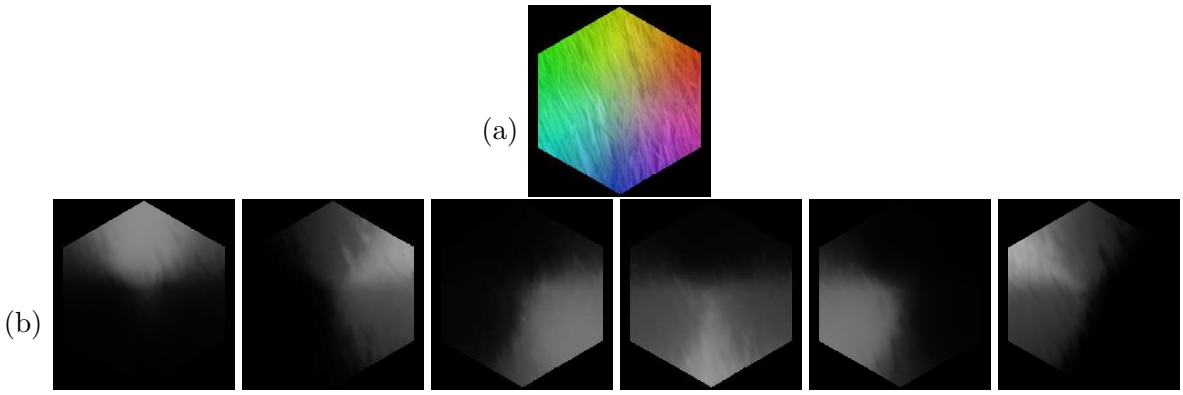


Figure 3.8: (a) Input synthetic image with texture underneath the color mixture. (b) The soft labels produced by using our proposed method. The soft labels are displayed in order to show the smoothness achieved by the MRF consideration in our method.

3.4 Results

For all examples in this section, it takes less than a minute to process each case on a 2.8 GHz PC with 1G RAM.

Synthetic case. Figure 3.7(a) shows a hexagon with spatially-varying colors, produced by compositing six soft color regions as shown. A single pixel's color can be explained by as many as six colors. Figure 3.7(b) shows the ground truth soft segment of the synthetic image. Figure 3.7(c) and (d) show the results produced by [80] and our method respectively. To initialize the algorithm, we used the same set of mean colors estimated by k-means clustering ($k = 6$) for both of the methods. We calculated the difference of the estimated region label with the ground truth region label by $(\sum_i |\alpha_{ij} - \alpha_{ij}^G|)/N$, where α_{ij} is the estimated region label at i, α_{ij}^G is the corresponding ground truth. The mean difference from the six region for [80] is 11.5 and our method is 10.0 (scale 0-255). In Figure 3.8(a), the synthetic image is mixed with texture. With the MRF consideration in our algorithm, one can see that the soft labels obtained with our methods in Figure 3.8(b) are smoothed.

Colorization. We perform soft color segmentation respectively in both the gray image and the example image, the latter of which provides the color statistics. Figure 3.9 shows our result. In this example, we show the effects when the color statistic σ_j in the observation $\mathbf{O}$ is adjusted by the user to achieve different effects. In this example the $\mathbf{O}$'s

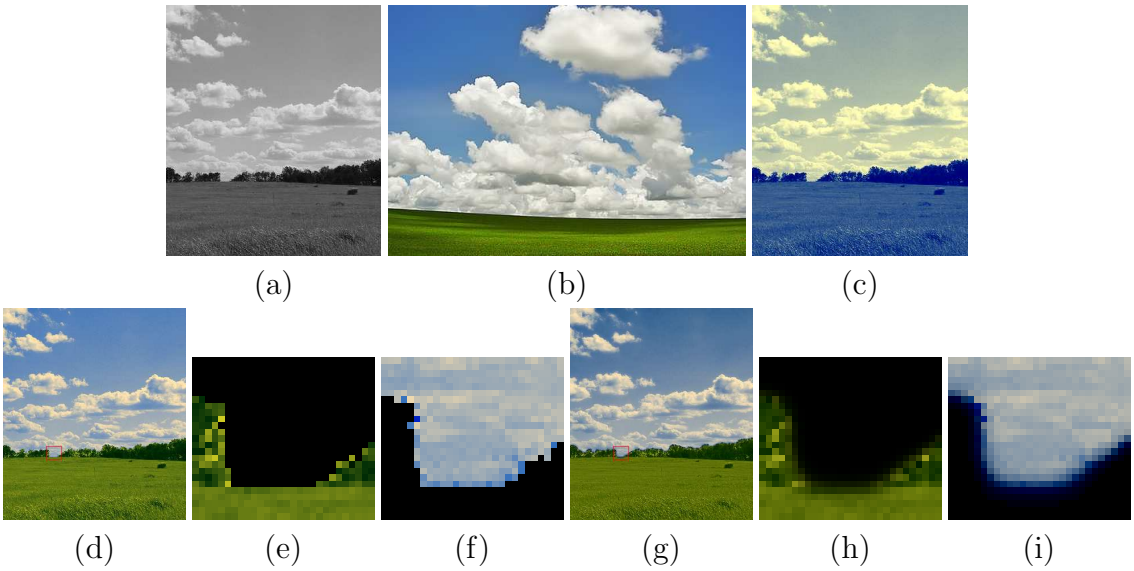


Figure 3.9: Color transfer to a gray scale image. (a) The image to be colorized. (b) The example image, which provides the relevant color statistics for colorizing (a). (c) Result by global color transfer. (d) Result by our method, with small σ_j input for the sky and the grass regions. (e) and (f) Zoom in of respective layers in (d). (g) Result with large σ_j. (h) and (i) Zoom in of respective layers in (g). See the electronic version for color visualization.

in both the gray and example images are obtained automatically by k-means clustering instead of user strokes. Here we input different σ_j as different initialization for the EM algorithm. Figure 3.9(d) is the result with small σ_j, and Figure 3.9(g) shows the result with large σ_j. From the corresponding zoomed images shown in Figure 3.9(e), (f), (h) and (i), we can notice that the boundary of the result with small σ_j is quite unnatural and noticeable. This is because using small σ_j is analogous to a hard segmentation. The boundary of the result with large σ_j, on the other hand, is more natural. But a large σ_j makes the blue color in the sky overlap with the green color of the grass in the color space. So the blue color depicted in Figure 3.9(g) is not as pure as that in Figure 3.9(d). Figure 3.9(c) shows the result when global colorization is performed.

Glass and Shadow removal. Figure 3.1 shows an example of glass removal. The user first samples attenuated and unattenuated color samples, which are served as input to our EM algorithm. Figure 3.2 previews an example of shadow extraction, which will be detailed in the next section. Shadowed and unshadowed color samples are collected, which become the input to the EM algorithm. Figure 3.10 shows an example when both F and β

Input Glass F β markup smooth β $F + \beta$

Figure 3.10: Layer decomposition from a single image (input image from [107]). After extracting the highlight layer, the user marks up on the image attenuated and unattenuated background to gather color and texture statistics, in order for the system to optimize the smooth β free of any textures.

(a) (b) (c)

Figure 3.11: The extracted transparent layers can be used in (a) and (b) image matting and (c) shadow matting.

are extracted. The champagne glass extracted will be represented by the highlight layer (F) and the attenuation layer (β) as show in the figure. In both cases, the extracted smooth layers, which are of spatially-varying intensity and free of any textures, can be used in image and shadow matting, Figure 3.11.

3.5 Application: Shadow Extraction from a Single Image

Shadows are problematic to many vision algorithms because intensity or color constancy are usually assumed. For instance, the computation of optical flows is susceptible to illumination changes caused by cast shadows. In image segmentation, unwanted shadow boundaries may be detected together with true object boundaries.

On the other hand, because shadows provide such strong perceptual cues to an object's depth and its relationship to its background, in image compositing and graphics-related applications, it is often necessary to cut and paste an object and its corresponding shadow as well to create a visually convincing composite.

From an uncalibrated input image, our EM method is capable of extracting a good shadow image β, as well as a shadowless image B where texture gradients under the extracted shadow are preserved. No simplifying assumption on the camera and the light source other than the Lambertian assumption is used. Using Eqn. 3.2, a generic shadow is modeled as per-pixel luminance attenuation. Our EM algorithm optimizes per-pixel shadow probabilities, which are used to indicate how much a pixel is shadowed. The optimized probabilities are directly employed in the Bayesian framework for estimating the shadowless image from which the shadow is extracted. Both hard and soft shadows can be extracted within the same framework.

3.5.1 Motivation: an image-based approach to shadow extraction

Extracting shadows from a single image is a challenging problem. While we human beings have no problem in identifying shadowed regions and predicting their appearance without the shadow, a single, uncalibrated image provides very limited information for a computer algorithm to extract relevant cues for shadow extraction. To gain more insight in computationally dealing with the problem, the work by Finlayson et al., provides some of the most important literatures.

Over the past decade, Finlayson et al., have published extensively (e.g., [20, 21, 19, 24, 25, 22, 29, 30]) on understanding color information from images. Color constancy and intrinsic images are two main areas of focus. By using such information, they proposed several shadow removal methods (e.g., [24, 25, 22, 30]) where decent results have been obtained. Despite recent advancements of their work, their central idea on shadow removal lies in the following:

(1) Shadow boundary detection

(2) Image reconstruction after eliminating the detected shadow boundary

It is quite obvious why step (1) is needed, which is used for locating the shadowed pixels to be processed. Step (2), simply speaking, is used for shadow removal. It assumes constant shadow attenuation, that is, a shadowed region is formed by scaling a constant factor the intensity of the underlying nonshadowed region [22, 29, 30] (see Fig. 3 in [29]). So, after removing the shadow boundary, followed by the Poisson-based image reconstruction on the corresponding gradient image from which the gradient caused by the shadow is subtracted [22, 30], the intensity of the whole shadowed region will be automatically scaled up during the integration/reconstruction process and thus the shadow is removed.

Although their most recent work has demonstrated some seamless results for some cases, and the methods are fully automatic, several issues remain unaddressed and properly addressing them will lead to better results.

The first issue lies on the assumptions used in their work. Planckian lighting (e.g., sunlight) and narrow-band cameras [24, 22] are used in their mathematical derivation. These assumptions are very useful for the automatic detection of shadow boundary. It is not an easy task to automatically determine if a pixel is shadowed or otherwise, but with the use of an appropriate lighting model and certain specialized configuration, it is possible to derive the relevant color information of a pixel and hence the lighting or shadow information. However, most images in need of shadow removal, for example, are images captured by off-the-shelf digital cameras in unknown and complex lighting conditions. Although it is possible to perform transformation to achieve spectral sharpening [21]

to help satisfy some of the aforementioned imaging constraints, such spectral sharping method still requires either the camera sensors to be known or the presence of a set of calibration images taken with an unknown camera, which still restricts or affects the applicability of their approach.

The second issue concerns the types of shadow their approach can handle. Note that steps (1) and (2) rely on the detection of well-defined shadow boundaries. While hard shadows have well-defined shadow boundaries, soft shadows, which are ubiquitous, are spatially smooth with fuzzy shadow boundaries. Binary labeling of such soft boundaries is problematic and so the aforementioned Poisson-based image reconstruction will fail to remove soft shadows. More importantly, because the shadow attenuation inside a soft shadow is spatially varying, the use of a constant scale factor does not apply, making their method fail to produce satisfactory results on removing soft shadows. As a matter of fact, in many cases, even for a hard shadow, both the intensity contrast and attenuation are spatially varying inside the shadow, especially when the input image is of high dynamic range.

The third issue is that their approach focused on shadow removal. There is no shadow extraction in their method, which is desirable for applications such as image and shadow matting. While one may treat the constant factor as the shadow matte estimated in the shadowed region, this is only a rough approximation.

In summary, the power of the work by Finlayson et al. lies in a fully automatic way to detect and remove hard shadows. However, the assumptions associated with hard shadows as well as those concerning lighting and camera configurations have limited the method's applicability in shadow removal. In addition, the proposed removal algorithm (step (2)) is not suitable in removing shadows whose attenuation is spatially varying in the interior of the shadow.

On the other hand, these issues can be effectively addressed by our EM-based method, specifically section 3.3.2: the shadow formation process can be explained by Eqn. 3.2, where the input shadowed image (that is, I') is obtained by scaling the intensity of the shadowless background image (that is, B) by an attenuation map β which can be a

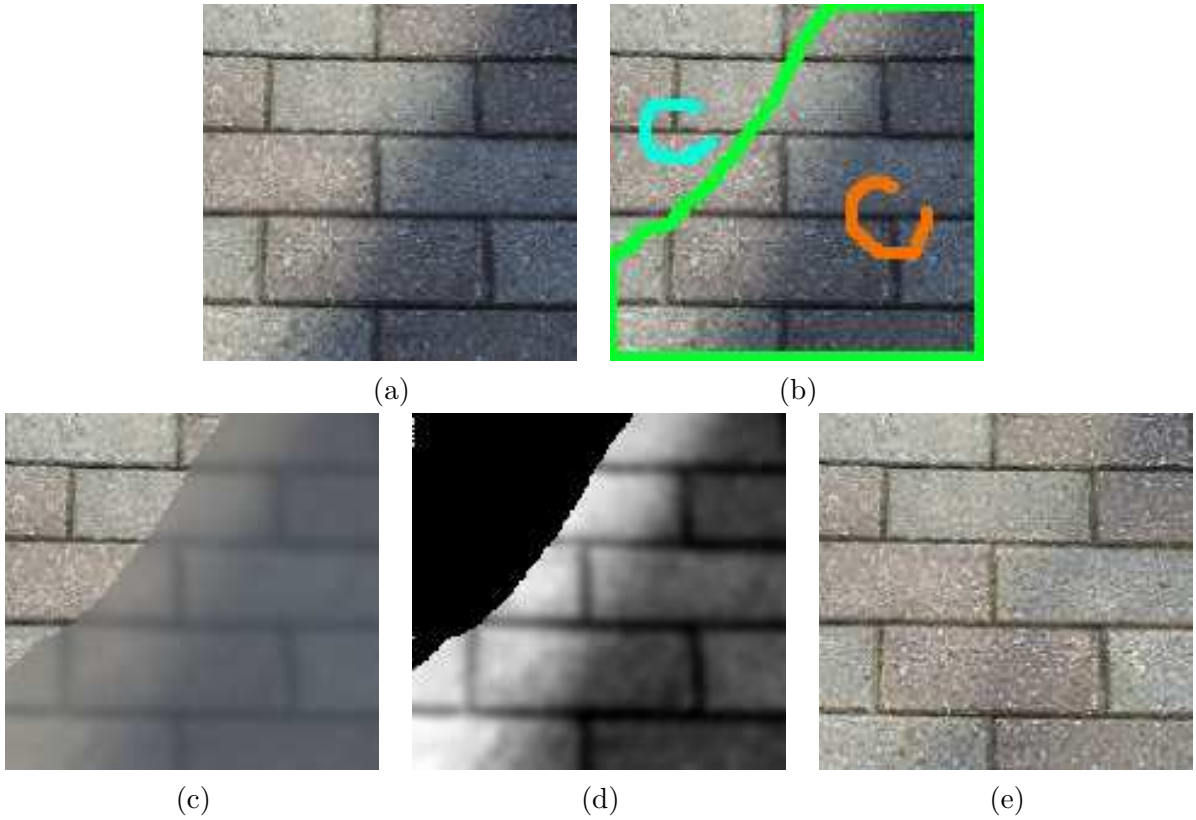


Figure 3.12: Brick wall. (a) Input image, (b) user scribbles (see Fig. 3.2 for color codes), (c) the optimized c_i, (d) the optimized w_i, (e) the result of shadow removal by our method.

spatially-varying. Take Figure 3.2 as an example:

(1) Rather than relying on a fully automatic algorithm, by observing that not every part of a single image is always shadowed, the user can easily scribble on it to obtain shadow and non-shadow color samples to help achieve better results. These two types of scribbled regions correspond to $\mathcal{S}$ and $\overline{\mathcal{S}}$, respectively.

(2) We do not use the image reconstruction Finlayson et al. adopted to recover the shadowless image. The result of their image reconstruction is a constant scale-up of the intensity of the shadowed region, which is localized after hard shadow boundary detection. As shown in our previous examples, because our EM algorithm produces soft labels (that is, α_{ij}), it can extract shadows of spatially-varying intensity which are not limited to either soft or hard shadows.

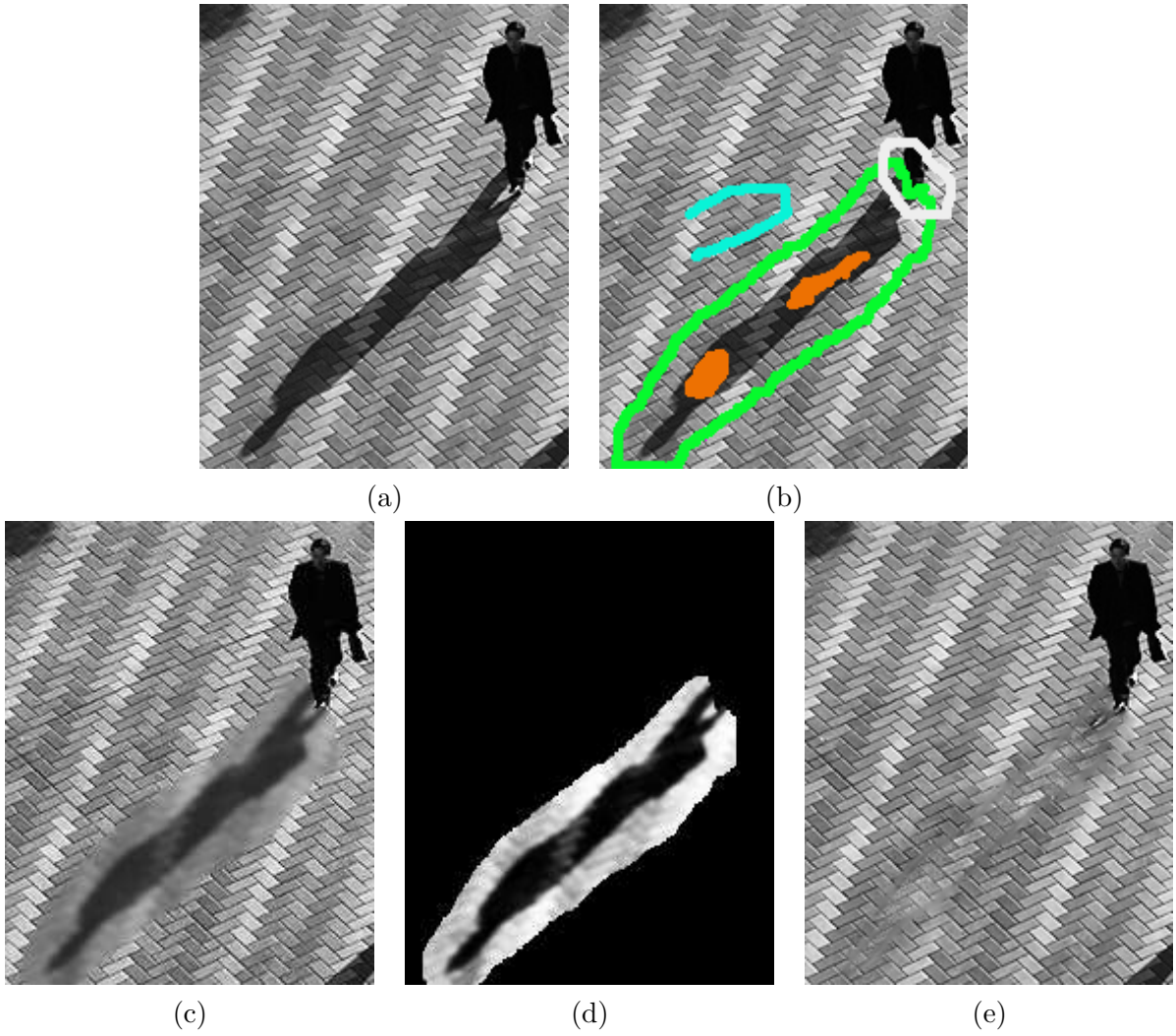


Figure 3.13: Long shadow: (a) Input image, (b) user scribbles (see Fig. 3.2 for color codes), (c) the optimized c_i, (d) the optimized w_i, (e) the result of shadow removal by our method.

3.5.2 Results

This section is organized into three subsections. We first apply our method to shadow removal. Then, we compare our shadow removal results with those by Finlayson et al. [22] which is an automatic method based on several assumptions about the light source and the camera, as well as our own previous works on shadow extraction and matting [97, 95].

Shadow removal and extraction

We show the results produced by our method under different scenarios. The input images were also used in our previous works [97, 95]. We produce comparable and better results while the amount of user interaction is smaller compared with the use of quadmap in [97,

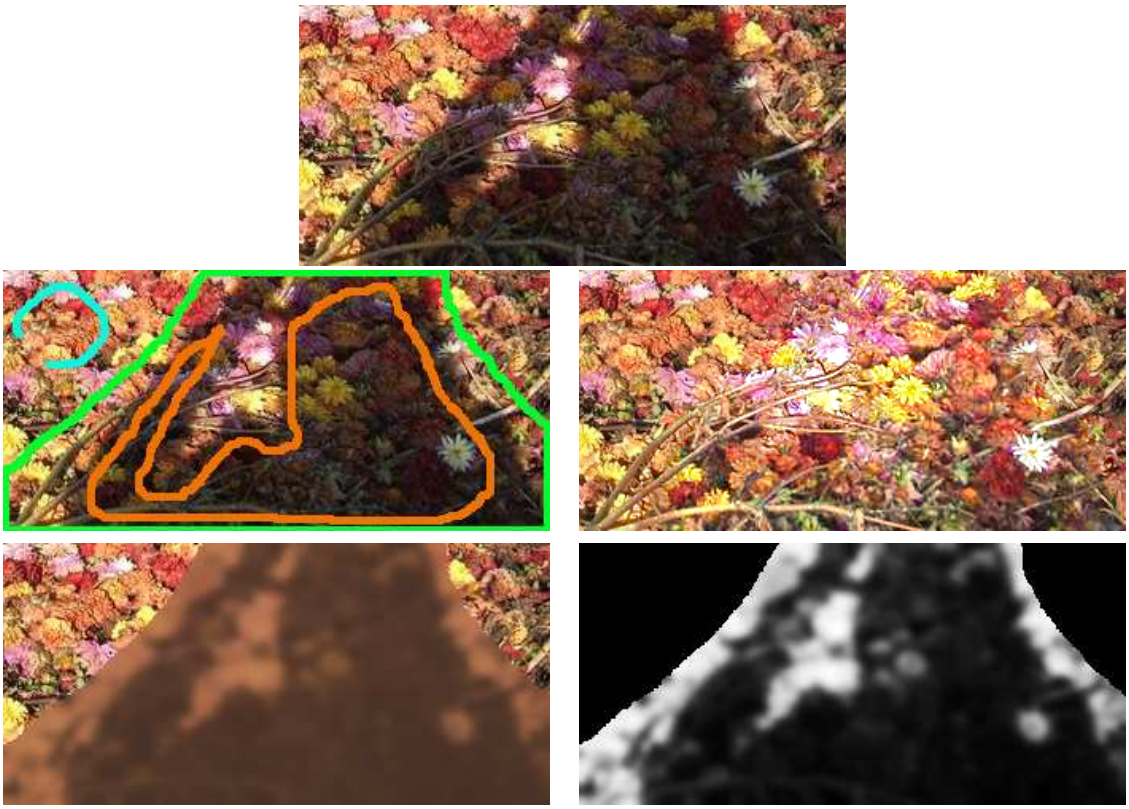


Figure 3.14: Flower. Top: input image. Middle: user scribbles (see Fig. 3.2 for color codes) and result of shadow removal by our method. Bottom: the optimized c_i and w_i.

95]. In each example, we show the input image, the user scribbles, the optimized c_i and w_i obtained by our EM algorithm, and the results of shadow removal (that is, B) we obtained. Fig. 3.12 shows an example where structure and texture are present under the shadowed area. The shadow is smooth, soft, and of varying intensity. Fig. 3.13 shows another shadow removal result where textures and salient structured patterns are under the hard shadow. Both results are very reasonable considering the small amount of user assistance.

Fig. 3.14 shows an example where complex textures are present under the shadow, while the shadow is smooth. Note that the unique texture pattern of the ground is maintained and the shadowless image is visually acceptable.

Fig. 3.15 shows an example where the shape of the shadow is very complex. The shadow is soft while the background contains some wood-like texture. Our method can remove all the shadow while preserving the texture under the complex shadow.

Fig. 3.16 show two examples on image and shadow compositing. Note that the textures

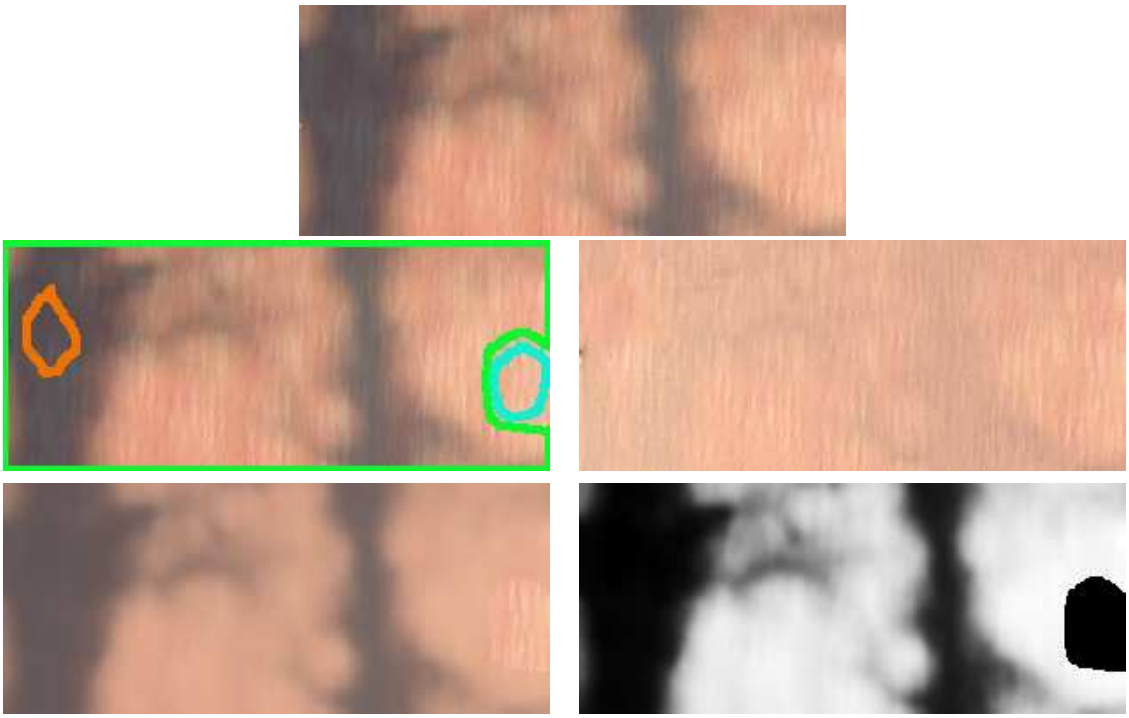


Figure 3.15: Wood. Top: input image. Middle: user scribbles (see Fig. 3.2 for color codes) and result of shadow removal by our method. Bottom: the optimized c_i and w_i.

	[97]	[95]	Our approach
Input	quadmap	quadmap	scribbles
Optimization	Energy Minimization	Bayesian + Poisson	EM
Problem modeling	Ad-hoc	Somewhat ad-hoc (see section 3.5.2)	Probabilistic
Running time 256 ×256 image	5–90sec	<2min	<2min

Table 3.1: Comparion of [97, 95] and our approach.

under the shadows are preserved after shadow extraction, and that the extracted shadow is smooth which is suitable for compositing in a new scene, where the extracted shadow is brightened or darkened to make it look natural. Fig. 3.17 shows three additional shadows we extracted. Note that, in the case of the bird, the background under the shadow contains significant textures. The extracted shadows shown in Fig. 3.17(c) are smooth and contain no texture contribution from the grounds, so they can be used as shadow mattes for shadow compositing. In Fig. 3.17(d), we composite the extracted shadows and the foregrounds onto new background images to create visually plausible image composite.

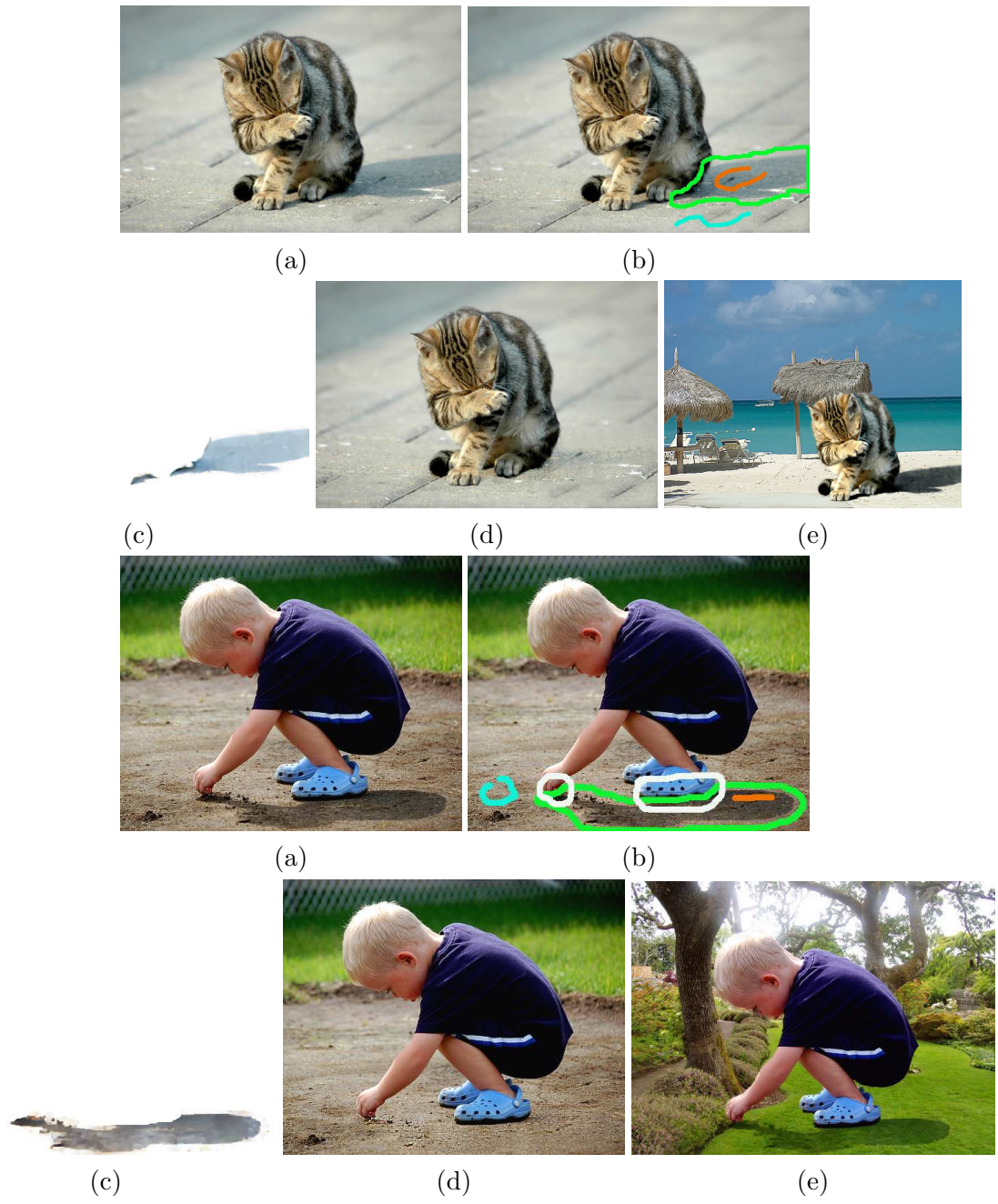
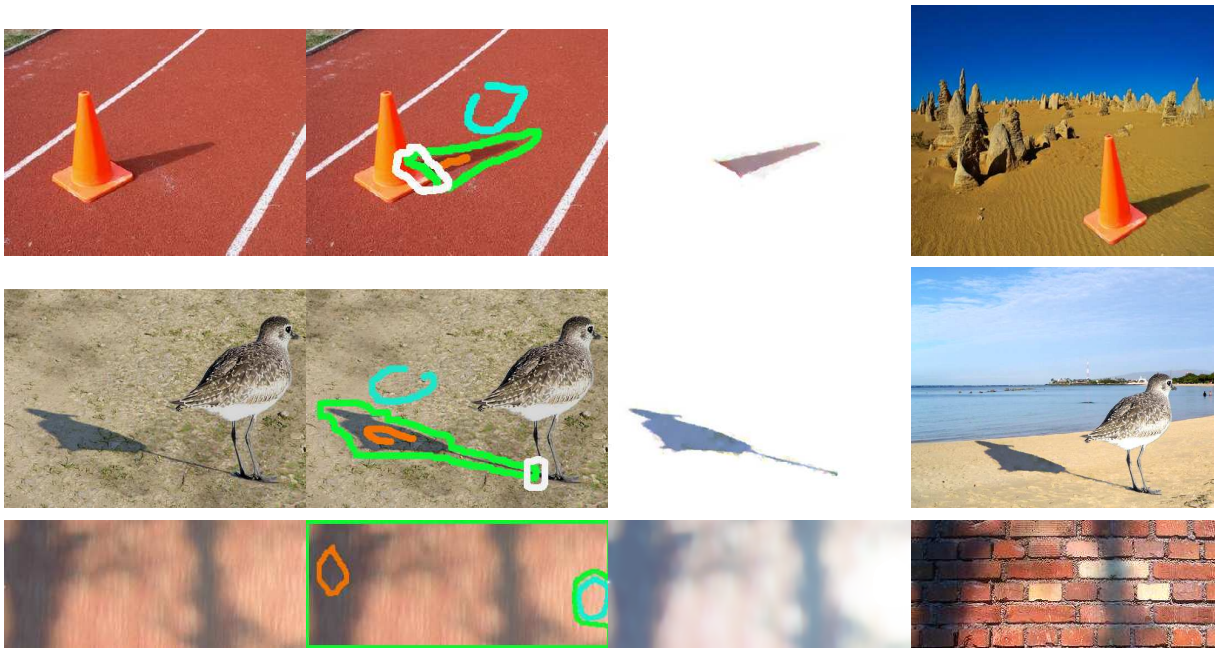


Figure 3.16: Examples on image and shadow compositing. (a) Input image, (b) user scribbles (see Fig. 3.2 for color codes), (c) the extracted shadow β, (d) the shadowless B, (e) the composite with shadow (brightened in the top and darkened in the bottom). Observe the preserved textures in the extracted shadowless image B and shadow smoothness is maintained in the extracted shadow β.



(a) (b) (c) (d)

Figure 3.17: Application in image composition: (a) input image, (b) user scribbles (see Fig. 3.2 for color codes), (c) shadow image β extracted by our method, (d) image composite with shadow.

Comparison with [22, 97, 95]

We select some complex results produced by Finlayson et al., [22] for comparison. Note that [22] is an automatic method assuming Planckian light source and narrow-band camera, while ours is a user-assisted approach without any assumption on the light source and the camera. After accepting the user-scribbled input, the rest of the processing is left to the computer.

The first example in Fig. 3.18 contains three noticeable shadows from the man, the plant and the lamp (on the left hand side of the input). The output by [22] smoothes out the textures while our method faithfully preserves them. Note, however, we cannot remove the lamp shadow well due to the dominant image noises, which is caused by the inadequate resolution of the image we obtained.

The second example depicted in Fig. 3.18 contains complex textures under hard shadows. Our method produces visually better results along the shadow boundary while the textures are not smoothed out.

Fig. 3.19 shows a complex example where the hard shadow straddles between two

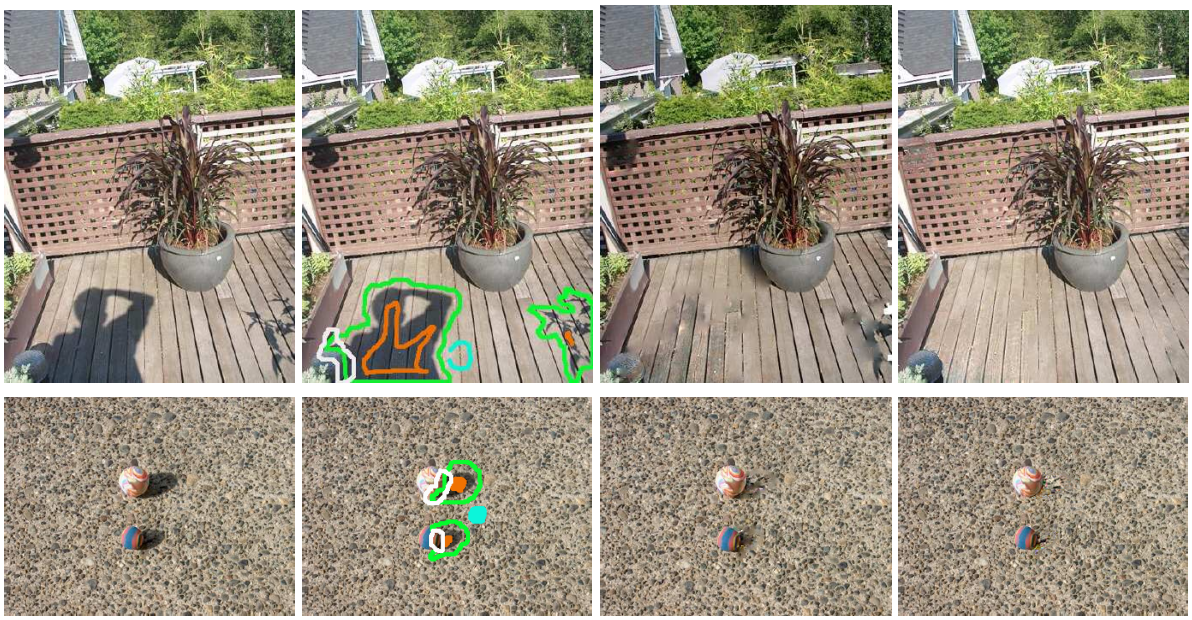


Figure 3.18: Shadow removal in complex scene. From left to right: input images from [22], user scribbles (see Fig. 3.2 for color codes), result of shadow removal by [22], result of shadow removal by our method.

different textured regions. This example shows that our approach may require repeated user's marking on all the shadows that should be removed. So if the captured scene contains a lot of fragmented hard shadows, and if the image is captured using a narrow-band camera under a Planckian light, the Finlayson et al. method should be more preferable.

Next, we use Table 3.1 to compare our EM method with our previous works on shadow matting and extraction [97, 95].

There are two stages in [95]. In the first stage, the system requires the user to input some hints about shadow and non-shadow samples (in the form of a quadmap [95]). In the second stage of [95], the user-supplied hints are translated into pertinent information for shadow extraction, where the information is represented by relevant color statistics. After extracting β, the shadowless image is simply I/β. Two problems lie in the second stage.

In the second stage, we formulated the shadow extraction problem by a Bayesian framework which was broken down into three sub-problems for simplification and approximation. The detail procedures are [95]:

(1) Estimate the probability that a pixel is shadowed. The resultant collection of per-

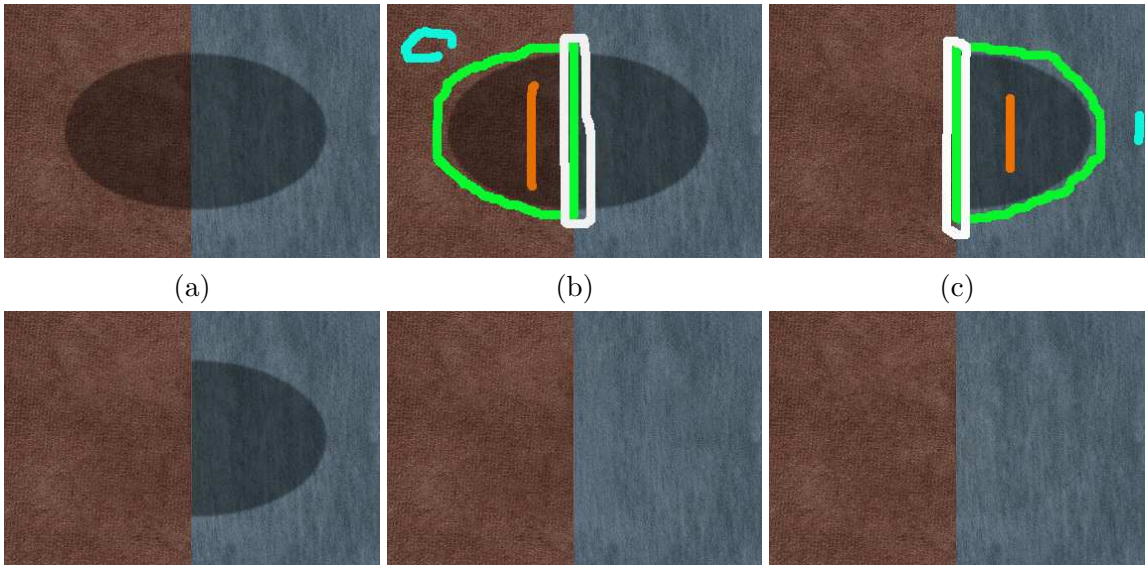


Figure 3.19: Hard shadow straddles between two different textured regions. (a) Input image. Because the shadows are fragmented, our approach requires the user to specify more than one processing regions ((b) and (c)), with different samples of $\mathcal{S}$ and $\overline{\mathcal{S}}$. (d) and (e) respectively show the shadow removal results successively obtained in different processing regions. (f) The expected shadow removal result by automatic method such as [22], given that a calibrated image where the sensor's spectral properties are known.

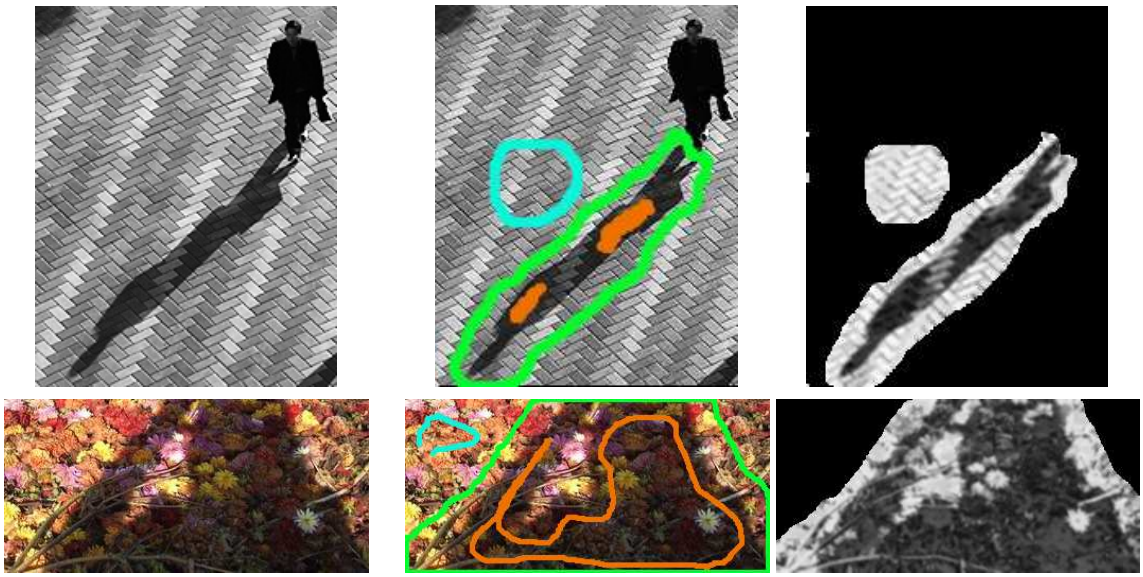
pixel probabilities constitute a probability map called $\mathcal{P}$.

(2) Calculate a coarse estimation of the shadowless image B, given $\mathcal{P}$.

(3) Optimize the shadow image β, given a coarse shadowless image and certain shadow priors.

The first problem concerns the estimation of $\mathcal{P}$. In [95], the calculation of $\mathcal{P}$ is based on model affinity: Given the color statistics (in the form of Gaussian Mixture Models (GMMs)) of the shadowed and non-shadowed regions, we estimate $\mathcal{P}(x)$ by calculating the model distance between the intensity at pixel x, $I(x)$, and the shadow and non-shadow GMMs respectively. The calculation is based on heuristic measures and the resultant map is not a true probability map. The drawback of using this method is that the resultant $\mathcal{P}(x)$ is very noisy (Fig. 3.20).

The second problem is that the estimation equation in the original Bayesian framework is rather complicated. So, the optimization was broken down in two, namely, Step (2)



(a) (b) (c)

Figure 3.20: (a) Input image, (b) user scribbles (see Fig. 3.2 for color codes), (b) the normalized map $\mathcal{P}$ is used as a rough β estimate in [95] (black pixels are excluded from calculation). Note that $\mathcal{P}$ is adversely affected by complex materials and colors.

and Step (3), where the intermediate result of Step (2) is only an approximation.

In this thesis, instead of estimating $\mathcal{P}$, we estimate certain useful measurements in the form of probabilities by our EM algorithm where the MRF assumption is used. These measurements are then used in our shadow optimization algorithm. Consequently, a unified Bayesian framework can be used as described in section 3.3.2 to replace Steps (2) and (3) where approximation or intermediate result are no longer needed.

3.6 Summary

Layer separation from a single image is a massively ill-posed problem in its general form. In this chapter, we showed how we solve an easier but useful alternative, and presented an EM algorithm to separate smooth layers and the substantially-textured background from a single image. The EM alternately optimizes the soft label and the expected color at each pixel, where the MRF is used to maintain spatial coherency of the smooth layers. The image textures of the background layer are explicitly preserved by solving the Bayesian MAP estimation problem. Our proposed algorithm is demonstrated to produce good

results in various computational photography applications.

We apply our EM algorithm to perform shadow extraction from a single image. Our image-based approach is user-assisted, with no simplifying assumption made on the camera or the light source other than the Lambertian assumption. The user supplies a few input scribbles to mark roughly on the input image to collect color statistics of definitely shadowed and nonshadowed regions. Using the relevant color statistics as observation, we use the same EM algorithm to estimate the shadow probability at each pixel. This probabilistic measurement indicates whether and how much the observed image gradient is affected by shadow, thus providing a confidence measurement in the process of shadowless image reconstruction.

Using the derived layers extraction algorithm, we attempted to transfer objects from the original image to a new background image by considering the transparent object as a set of layers. Figure 3.11 shows two examples in which we transfer a transparent monument and a champagne glass. It is obvious to see that for a planar object like the monument, the result is very good. However, for objects like a champagne glass, there is a critical issue in the result: light refraction due to the transparent object is not present. It can be easily detected that the difference between Figure 3.21(a) and (b) and feel that the appearance of Figure 3.21(b) is more natural. In the next chapter, we will present a new image model which encodes the refractive deformation inherent in the transparent objects. We will also show how we transfer transparent objects in general to a new image and discuss the related issues.

Figure 3.21: A champagne glass without refraction does not look right.

Chapter 4

Matting of Transparent and Refractive Objects

4.1 Introduction

In the previous chapter we described an approach to extract simple transparent layers from a single image. In general, the shape of the transparent object is not planar but can be of any shape. It creates a phenomenon which is not considered in the last chapter: the light refraction induced by the transparent object. For example, given any image in Figure 4.1, can we extract the transparent object and composite them on a new image?

The ability to cut and paste objects in photographs is a prerequisite for photo editing. While many effective approaches for segmenting and matting objects from a single image exist [49, 65, 89], these approaches assume the foreground objects are opaque. In many of these approaches, a user marks a trimap that consists of definite foreground, definite background, and uncertain region. The opacity assumption simplifies the matting problem in the uncertain regions by reducing the recovery of the foreground object to an estimation of each pixel's fractional contribution (α) of its color to the foreground (with the remainder being the background).

Transparent and refractive objects, on the other hand, have three properties that complicates their extraction and pasting. First, the fractional α associated with the

Figure 4.1: With only a single image of a transparent object without any prior 3D model and object's information, can we *cut* the object out and *paste* it onto a new scene? Notice that some objects are complex and the underlying light transport is very complicated.

transparent object is distributed over the *whole* object, as opposed to opaque objects where the fractional α values are mostly at the boundaries. Second, transparent objects commonly exhibit light attenuation through the object that affects each color channel differently. As a result, the conventional matting equation involving only a single scalar per pixel is insufficient. Finally, the refractive nature of these objects results in a warped appearance of the background. Since the object's 3D shape is typically unknown, this warping function is also unknown. To produce realistic composites, the refractive and transparent properties of these objects must be taken into consideration.

This thesis describes a new approach for matting transparent and refractive objects from a photograph and compositing the extracted object into a new scene. To accomplish this task we have modified the opaque image matting and compositing equation to fuse refractive deformation, color attenuation, and foreground estimation. We term this extracted information the *attenuation-refraction matte* (ARM). In general, a single photograph is insufficient to extract accurate refractive properties of a transparent object. Our approach instead recovers *plausible* light-transport properties of the matted object, exploiting our visual system's tolerance to inaccuracies in refractive phenomena as previously demonstrated by work targeting image-based material editing [42].

In this chapter we will show how user markup can be used to extract each component

of the ARM. By employing the unique properties of an object's ARM, the specular and attenuation components of the ARM can be optimized efficiently. Plausible refractive deformations are specified via control lines that are drawn on the image. One benefit of targeting *plausible* versus *accurate* light refraction is that our refractive markup need not be accurate. In the next chapter we will show how to make use of the information from the extracted ARM components to perform the composition and produce realistic results.

To the best of our knowledge, this is the first approach to address matting of transparent objects and their plausible refractive properties directly from a photograph.

4.2 Related Work

The basic challenge for opaque object image matting is to minimize user involvement without sacrificing output quality. Several techniques have been proposed and a survey can be found in [89]. Our approach applies matting for targeting specularities and foreground pixels, with the vast majority of the object being either transparent or attenuated background. In addition, the need for color attenuation is not considered in opaque matting.

The process of characterizing the light transport properties of transparent and refractive objects has been referred to as *environment matting*. Techniques for environment matting use either objects with known backgrounds [107, 15, 61], or multiple images [93] to extract the light transport properties. For example, Matusik et al. [55] used a turntable, multiple cameras, multiple lights, and monitors (as backgrounds) to capture an environment matte all around a transparent object. Our approach differs in that we target matting directly from an input photograph and use 'plausible light transport' to produce visually similar results.

Khan et al. [42] demonstrated an effective material editing technique that approximated light transport through coarse 3D object reconstruction. Using the object's approximated 3D shape, the object's material properties were changed, including the simulation of transparent and refractive effects. This technique, however, requires the original object to be opaque in order to approximate the 3D shape. Furthermore, material editing

is applied on the original image itself; matting and compositing are not performed.

If the shape of the transparent object is known, rendering the object with refractive effects in a given scene is well studied [26]. However, obtaining the real object's 3D shape and corresponding refractive properties is not a trivial task. Recent techniques for shape recovery of transparent objects are available, e.g., [4, 57, 58]. In addition, refractive deformation can be computed from a video [1]. However, these existing approaches operate under specific conditions, such as multi-camera capture, custom calibration, and often assume restrictions in 3D shape. This makes them inapplicable in our tool that is designed to operate entirely on photographs with no explicit 3D information.

Given a matted opaque object and a new background, multiple approaches exist for minimizing the object-background seam when compositing the object (e.g., [63, 39]). As previously mentioned, these existing matting and compositing approaches assume an opaque object with fractional pixel contributions at the object boundary. Compositing in our case requires more consideration of the refractive and transparent nature of the object. This will be discussed in the next chapter.

4.3 Attenuation-Refraction Matte (ARM)

We begin by defining a new matting and compositing equation that accounts for the appearance of the transparent and refractive object observed from an image. We start with a light scattering model similar to that proposed in [55]. Assuming a distant scene, the amount of light recorded at the camera is

$$C = \int_{\Omega} W(\omega) L(\omega) d\omega, \tag{4.1}$$

where $L(\omega)$ is the illumination from direction ω, $W(\omega)$ is the contributing weight, and Ω is the entire hemisphere. Note that $\int_{\Omega} W(\omega) d\omega < 1$ because of transmission loss or attenuation due to material absorption of light.

Consider a transparent object in the image bounded by a mask M. Within this object,

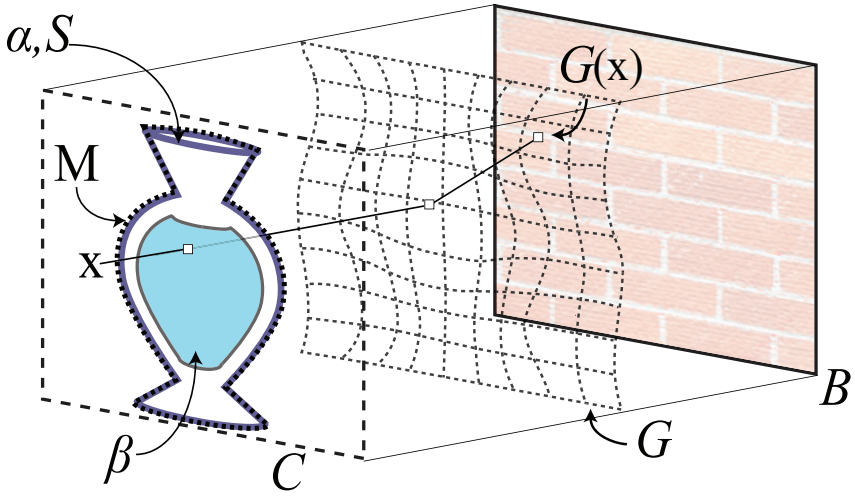


Figure 4.2: Generating the composite image $C(\mathbf{x})$ along the line of sight passing through pixel $\mathbf{x}$.

we make the simplifying assumption that C consists of two items:

$$C = \int_I W(\mathbf{p})L(\mathbf{p})d\mathbf{p} + \int_{\Omega - I} W(\omega)L(\omega)d\omega = C_I + C_S, \tag{4.2}$$

where $\mathbf{p}$ is the continuous 2D coordinate of the image. More specifically, we assume that the warped color seen through the transparent object comes from locations within the camera's field of view I. This is represented by the term C_I. Next, we assume that the specularities S are caused by illumination not directly visible to the camera (i.e., $\Omega - I$). The specularities on the object are represented by the term C_S.

For an object in the discrete image, we decompose these terms as

$$C_S = \alpha S, \quad C_I = (1 - \alpha)\beta B_G, \tag{4.3}$$

where α is the relative contribution of the appearance of specularities to the object's image (specular matte), β is 3-channel color transmission factor (< 1 in each channel because of attenuation), and B_G is the warped background. In discrete form, we write the image formation equation as

$$C(\mathbf{x}) = \alpha(\mathbf{x})S(\mathbf{x}) + (1 - \alpha(\mathbf{x}))\beta(\mathbf{x})B(G(\mathbf{x})). \tag{4.4}$$

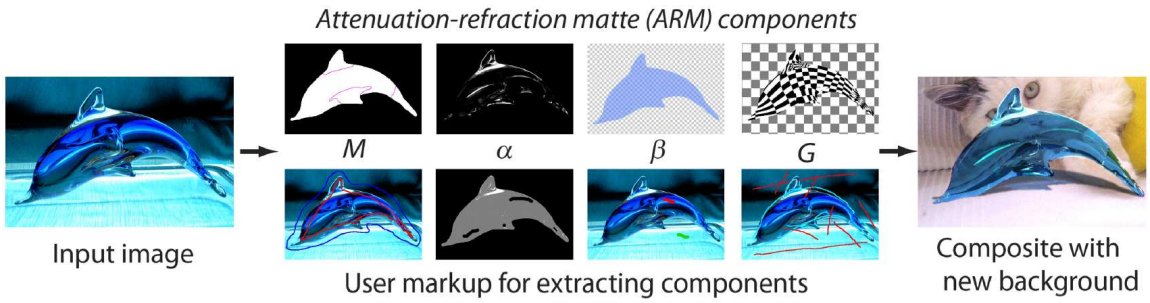


Figure 4.3: Overview of attenuation-refraction matte (ARM) extraction.

Here, $\mathbf{x}$ is pixel location within object M, $B(\mathbf{x})$ the appearance of the background without the object, and $G(\mathbf{x})$ the warping function. Figure 4.2 illustrates the discrete version with a single transparent object in the scene. The terms M, α, β, and G, collectively make up an object's *attenuation-refraction matte*, or ARM. For an input image, the terms of ARM are all unknowns.

4.4 ARM Extraction

The ARM extraction from a *single* image of a transparent object is an ill-posed problem. Even without considering the complex deformation G, we have seven unknowns (one for α, three for S, and three for β) to solve in the equation. We therefore make the following simplifying assumptions on our ARM extraction:

1. Strong, white specular highlights are observed on the transparent object. This simplifies the computation of α.

2. The transmission factor distribution is piecewise smoothly-varying within a refractive medium. This simplifies the computation of β.

3. The refractive deformation G need only be plausible.

These assumptions are also consistent with Eqn (4.3), where the C_S component incorporates highlights which are largely white and opaque, while the C_I component mostly accounts for the transparency observed. Moreover, some ARM components are independent of others. As a result, it is possible to simplify the complex extraction problem by

	image	M		α		β		G		total
	size	#strokes	sec	#strokes	sec	#pairs	sec	#pairs	sec	sec
glass	602×400	2	1	0	6.3	1	10	12	14	31.3
jug	183×286	3	1	2	5.1	2	14	6	6	26.1
elephant	495×295	5	2	5	18	1	17	16	18	55
chandelier	339×451	2	1	0	7.3	1	15	13	15	38.3
dolphin	463×306	2	1	5	11.5	1	46	8	8	66.5
lion	359×294	3	1	5	3.8	0	0	14	16	20.8
fish	416×457	1	1	16	22.7	1	24.8	9	10	58.5
swan	266×512	2	1	2	2.6	0	0	6	6	9.6
dragon	353×269	4	2	5	3.5	0	0	9	10	15.5
globe	308×213	4	0.5	0	3.2	0	0	2	1	4.7

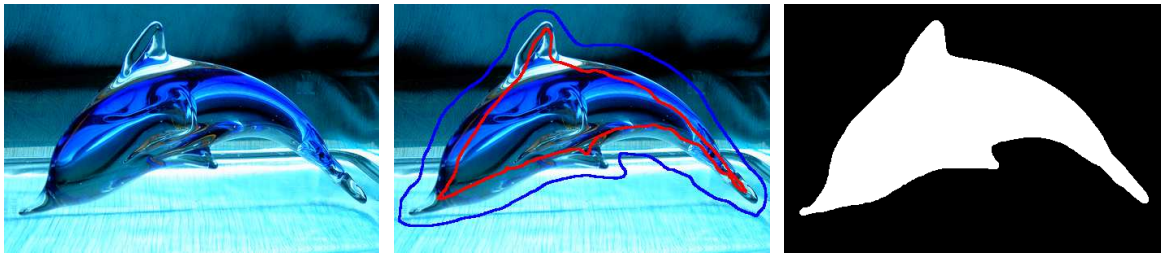
Table 4.1: Summary of user interaction (number of strokes or stroke pairs marked) and processing time (in seconds) for extracting the ARMs shown in this thesis. It typically takes 1–5 seconds to add a stroke. In general, for M and G, processing after each interaction (stroke or curve pair markup) takes less than 1 second. For α and β, processing is done once after all user markups are made. The program is run on a 3.6GHz PC with 2G RAM.

breaking it into several steps while achieving high-quality results.

Figure 4.3 shows the overview on ARM extraction. Specifically, we first extract M which defines the processing region for the extraction of α, β and G. Since α and β are related to pixel color whereas G concerns with pixel movement, we assume that the extraction of (α, β) is independent of that of G. Table 4.1 summarizes the user interaction and the processing time for most of the ARM examples in this thesis. In the following, we discuss the extraction of each component.

4.4.1 M-extraction

M specifies the "footprint" of the object in the image as a binary mask. We use lazy-snapping [49] to extract M (other techniques such as grab-cut [65] may be used as well). The user draws scribbles on the inside and outside of the object, shown respectively in red and blue in Figure 4.4. Although the color samples inside and outside the object can be similar due to transparency, color inconsistency at the object border allows the technique to work. M can be further partitioned to mask out non-transparent regions, and separate different refractive mediums or deformation regions. It typically takes seconds to two



(a) Input (b) Markup (c) Silhouette M

Figure 4.4: Extracting the object silhouette M. The user draws scribbles on the inside and outside of the object to collect the color statistics.

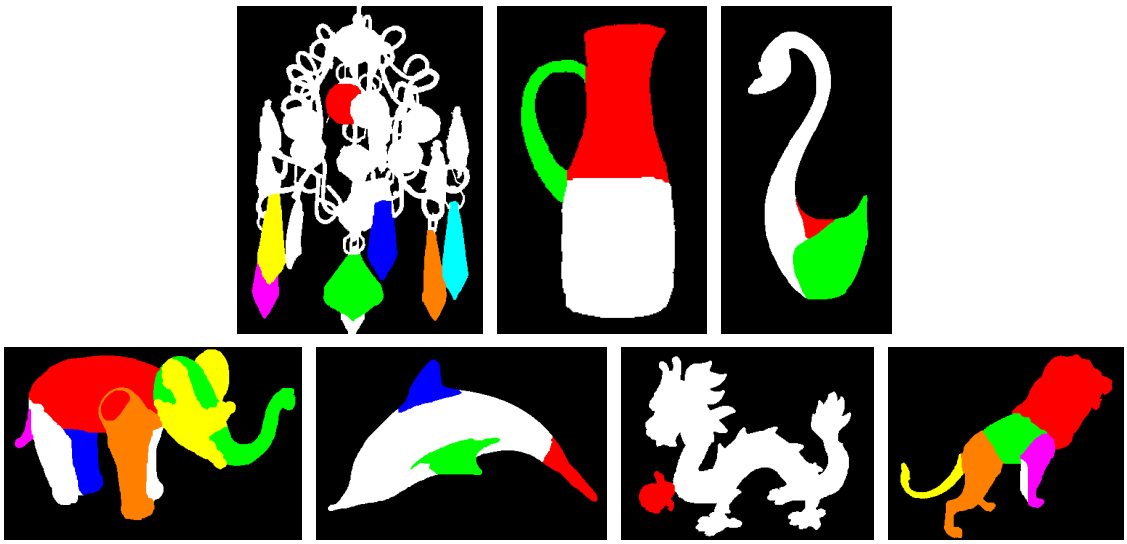


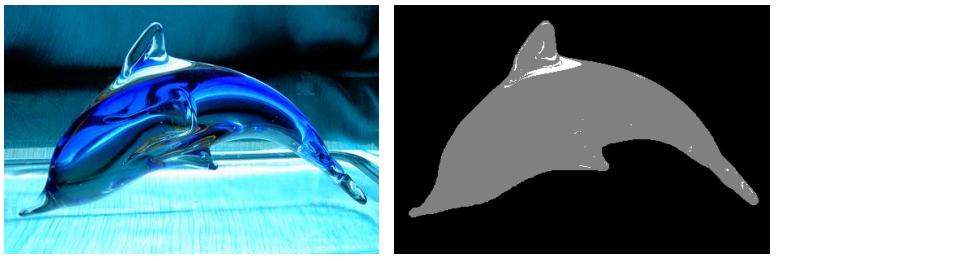
Figure 4.5: Partition of the object silhouette M into different refractive mediums or deformation regions.

minutes to specify a region mask; the region mask corresponding to the examples can be found in Figure 4.5.

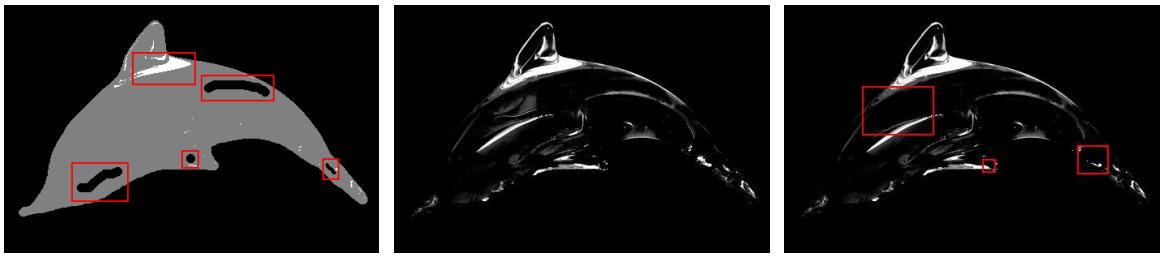
4.4.2 (α, β)-extraction

We extract α and β within M. Eqn (4.3) shows that we have more unknowns than equations to solve.

However, note that S (specularities) is largely white and opaque, and β (attenuation) is largely transparent and homogeneous within each refractive medium. S contributes to the "foreground colors" and β attenuates the "background colors" of the input image respectively. The extraction of α and S is therefore relatively easy and largely insensitive



(a) Input (b) Initial trimap



(c) Edited trimap (d) Specular α matte (e) Touched-up α matte

Figure 4.6: Since our target is to represent the specularities by the α matte, the trimap can be obtained automatically by simple thresholding. User can then add definite foreground or background samples on the trimap as shown. The resultant specular α matte can also be touched up for better visual effects.

to the extraction of β, which is smooth and transparent. The extraction of β can be improved after the specular and opaque highlights are removed.

Solving for α and S. To solve α and S, a trimap is automatically generated. Because the "definite foreground" consists of specular highlight, given the object mask M, definite foreground (shown as white within M in Figure 4.3) can automatically be labeled by thresholding (we set the threshold as all the RGB values > 220, in 0-255 scale in the input image), "definite background" is taken as pixel outside the mask (shown as black), with all remaining pixels within the mask labeled as "uncertain" (shown as gray). This automatic trimap construction is amenable to objects with a great deal of subtle structures that result in many small highlights (Figure 4.6(b)). Additional strokes within M can be marked to specify definite background (black) or foreground (white) if necessary (Figure 4.6(c)). The gray region is the uncertain region. Poisson matting [77] is performed using the trimap as input to extract the α. Occasionally, strong highlights in the background are extracted in the resulting α component (Figure 4.6(d)). Such unwanted artifacts usually come in blocks, and can be quickly edited away in a few seconds (Figure 4.6(e)). The mattes before and after editing of *swan*, *elephant*, *chandelier*, *jug*

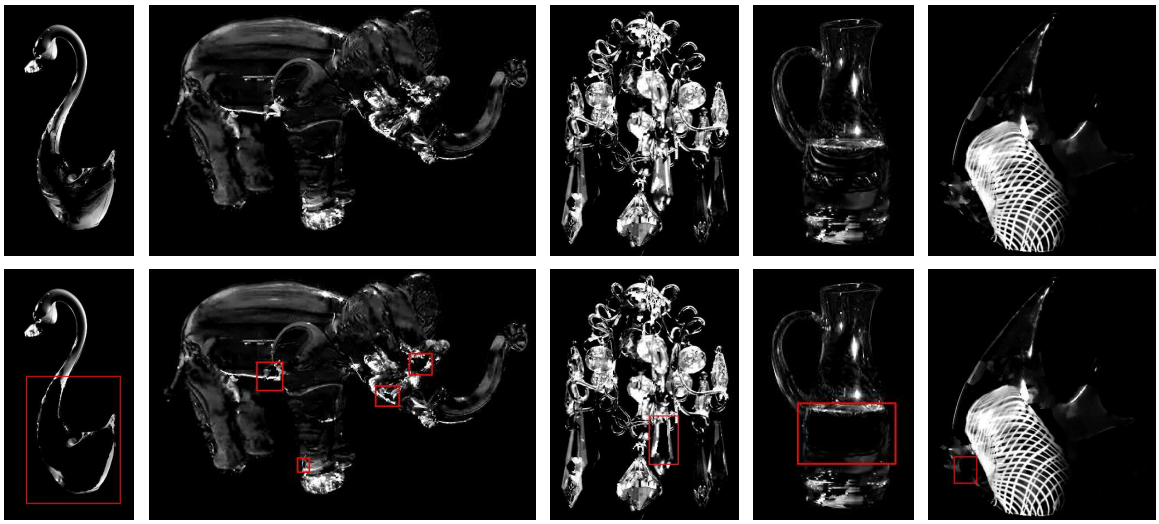


Figure 4.7: Specular α matte before and after user touch up. In our experiments, not all the α matte requires editing. The editing, even if needed, is very simple in all the cases.

(a) Markup (b) $\overline{C}_I$ (c) Color attenuation β

Figure 4.8: The user marks over the input image attenuated and unattenuated background to gather color statistics. After collecting these color samples, S's contribution will first be removed from the input image using α, i.e., $\overline{C}_I = (C - \alpha S)/(1 - \alpha)$. The system then optimizes a smooth β free of textures.

and *fish* are shown in Figure 4.7, showing that the amount of additional editing of the specular mattes is acceptable.

Solving for β. To solve β, S's contribution will first be removed from the input image using α. The user then marks up on the image to collect a number of attenuated and unattenuated background color samples, within and outside of the object's mask M. This is done using simple scribbles (red and green), as shown in Figure 4.8, where we assume that the scribbled background colors on the object are not severely affected by refractive deformation. To collect color statistics that are primarily affected by β-attenuation, the user should mark up sample pairs that are similar in texture. Here, the problem is translated into one similar to natural shadow matting [98], except that we do

not need to handle hard (shadow) boundaries.

Removing S's contribution using α leads to $\overline{C}_I = (C - \alpha S)/(1 - \alpha) = \beta B$. So, $\overline{C}_I$ can be regarded as the color-attenuated version of B. We propose to solve the following Bayesian optimization:

$$\beta^* = \arg\max_{\beta} P(\hat{B}|\beta) + P(\beta), \tag{4.5}$$

where $P(\hat{B}|\beta)$ is the likelihood, and $P(\beta)$ is the prior and $\hat{B}$ is a rough estimation of the unattenuated background B, which can be estimated using the technique of [98]. The likelihood is defined as

$$P(\hat{B}|\beta) = \exp(-\frac{\sum_{\mathbf{x}\in M} ||\overline{C}_I(\mathbf{x}) - \beta(\mathbf{x})\hat{B}(\mathbf{x})||^2}{2\sigma_1^2}), \tag{4.6}$$

where σ_1^2 is the variance of the measurement error ($\sigma_1 = 1000$ for $\beta = [0, 255]$ in our experiments). The smoothness prior of $P(\beta)$ is defined by

$$P(\beta) = \exp(-\frac{\sum_{(\mathbf{x},\mathbf{y})\in N} ||\beta(\mathbf{x}) - \beta(\mathbf{y})||^2}{2\sigma_2^2}), \tag{4.7}$$

where σ_2^2 is the variance on the smoothness of β, and N is the set of first order pixel neighbors in M. ($\sigma_2 = 5000$ in our experiments.)

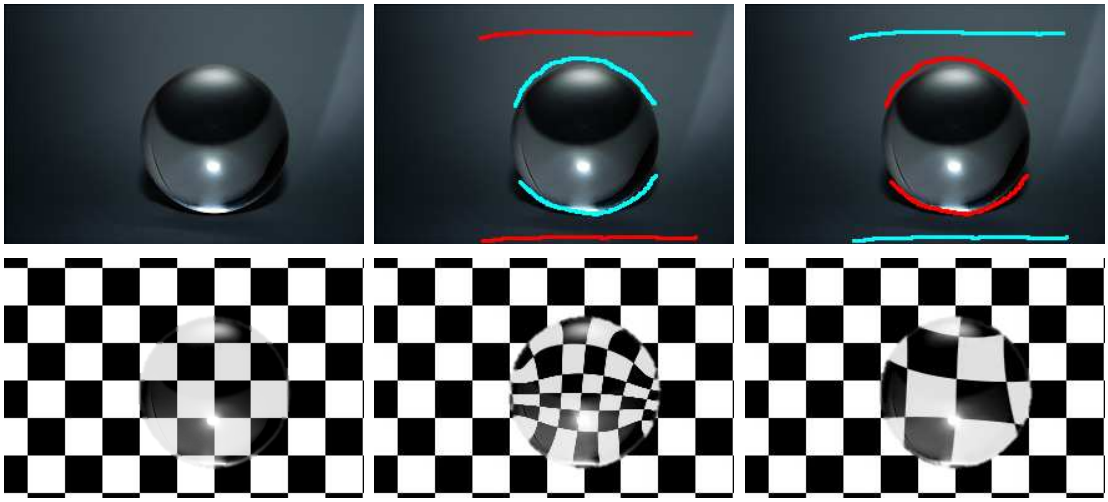
4.4.3 G-extraction

In this section we describe how simple markup can be used to specify plausible refraction.

A typical input image consists of a transparent object, placed in front of a background scene which has undergone refractive deformation observed within the transparent object. Since we only target a plausible effect, for the markup we assume orthographic projection in image capture, where the background is planar and parallel to the image plane.

When marking up the refraction G, visual cues from either the object or background help to specify the deformation.

Object cue. When a background region is largely homogeneous or the deformed structure/texture is too complex to mark up, the shape of the transparent object itself



(a) (b) (c)

Figure 4.9: Object cue. Three basic markups of refractive light-transport: (a) no markup, (b) markup for simulating light convergence, (c) markup for simulating light divergence. c_{ref} is in red and c_{target} is in cyan. c_{target} is where c_{ref} is perceived to distort to.

(a) (b)

Figure 4.10: Background cue. (a) Markup drawn using the background as a cue. As with object cue, c_{target} (cyan) is where c_{ref} (red) is perceived to distort to. (b) Example of the resulting deformation.

provides the main cue for the user to mark up G. We first consider three basic cases of refractive light-transport (Figure 4.9):

1. No change (planar object requires no markup).

2. Convergence (convex object).

3. Divergence (concave object, the opposite of (2)).

Depending on whether the perceived shape is convex or concave, the user first draws a rough line (c_{ref}), followed by drawing a curve (c_{target}) that roughly follows the 2D shape of the object (refer to Figure 4.9). This object cue markup is quite effective in producing visually plausible deformation of the background. These basic cases when combined can

(a) (b) (c) (d) (e) (f) (g)

Figure 4.11: *Can you spot the original?* We captured two images: one without the glass (I_0, (a)), and another with the glass (I_1, one of (b–g)). We extract the glass' ARM directly from I_1. Several versions of the glass' ARM are extracted using different user markup. We paste the ARMs onto I_0 in (a) to produce the other 'fake' images in (b–g). All the edited examples look visually compelling and are not easy to distinguish from the real one. See Figure 4.12 to see which (b–g) is I_1.

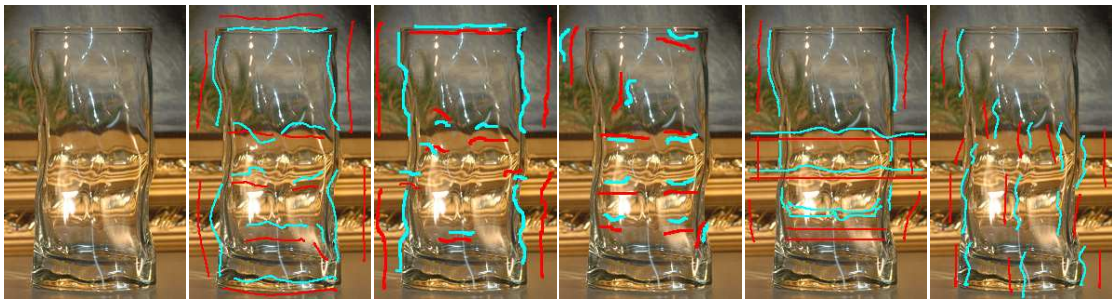
be used to markup complex shapes. In addition, such markup can be combined with the markup specified based on background cues which is described next.

Background cue. Deformation markup can also use the background to provide cues. Figure 4.10 shows an example where the background scene contains salient structure whose deformation is observable within the transparent object. In this case, the user can draw c_{ref} to roughly indicate how this structure looks before deformation (e.g., a straight line), and then draw c_{target} to follow the deformed structure in the object as shown in Figure 4.10(a). Figure 4.10(b) shows the effect of this stroke pair on the deformation field.

4.4.4 Deformation Warping

When a c_{ref} and c_{target} curve pair is drawn, 2D points along the curves are sampled and serve as the input landmark-pairs for thin-plate-spline (TPS) warping [9]. We use TPS because it is computationally efficient and is amenable to a 2D editing interface: given a set of K corresponding 2D point pairs sampled from the G markup, the TPS warp is specified by $2(K+3)$ parameters (6 global affine motion parameters and $2K$ coefficients for K control points). Also, TPS requires no manual tuning and has a closed-form solution.

After each curve-pair markup, the deformation map (shown as a checkerboard pattern) is updated in real-time to provide instant visual feedback.



(b) (c) (d) (e) (f) (g)

Figure 4.12: Simple markups are sufficient for producing visually plausible results. Here, we show different scribbles marked on the original input image (b) and the corresponding ARM composites on the same background.

Markup for G often involves stroke pairs drawn using both object and background cues and need not be that accurate. In fact, many different markups of an object can produce visually plausible refraction effects. This is attributed to our tolerance to errors in *complex* refractive light-transport as noted by Khan et al. [42].

Figure 4.11 shows several versions of ARMs extracted from a photo of a glass. Figure 4.12 shows the rough scribbles marked on the input image produce the corresponding "fakes" shown in Figure 4.11. This markup is casually performed and uses both object and background cues. Although they visually look dissimilar, all of them are visually quite plausible and it is difficult to detect refractive inaccuracies due to the inherent complexity.

4.4.5 Examples of Extracted ARMs

Figures 4.13 and 4.14 shows several example input images (column 1), markup (column 2), and the extracted ARMs (column 3). How the compositing results (column 4) are obtained will be explained in the next chapter. Both *elephant* and *dragon* have very complex shapes. The *chandelier* has a complex surface with both opacity and transparency. A simple mask for separating opaque and transparent parts is available. The opaque part of the chandelier is extracted by conventional matting. The *Swan* has an interesting shape, where the light transport produces an image inversion effect. In the *fish* example, the internal orange material is extracted as opaque colors. There is apparent color attenuation in the *fish* image and some in *elephant* and *chandelier* images. The images of colorless transparent

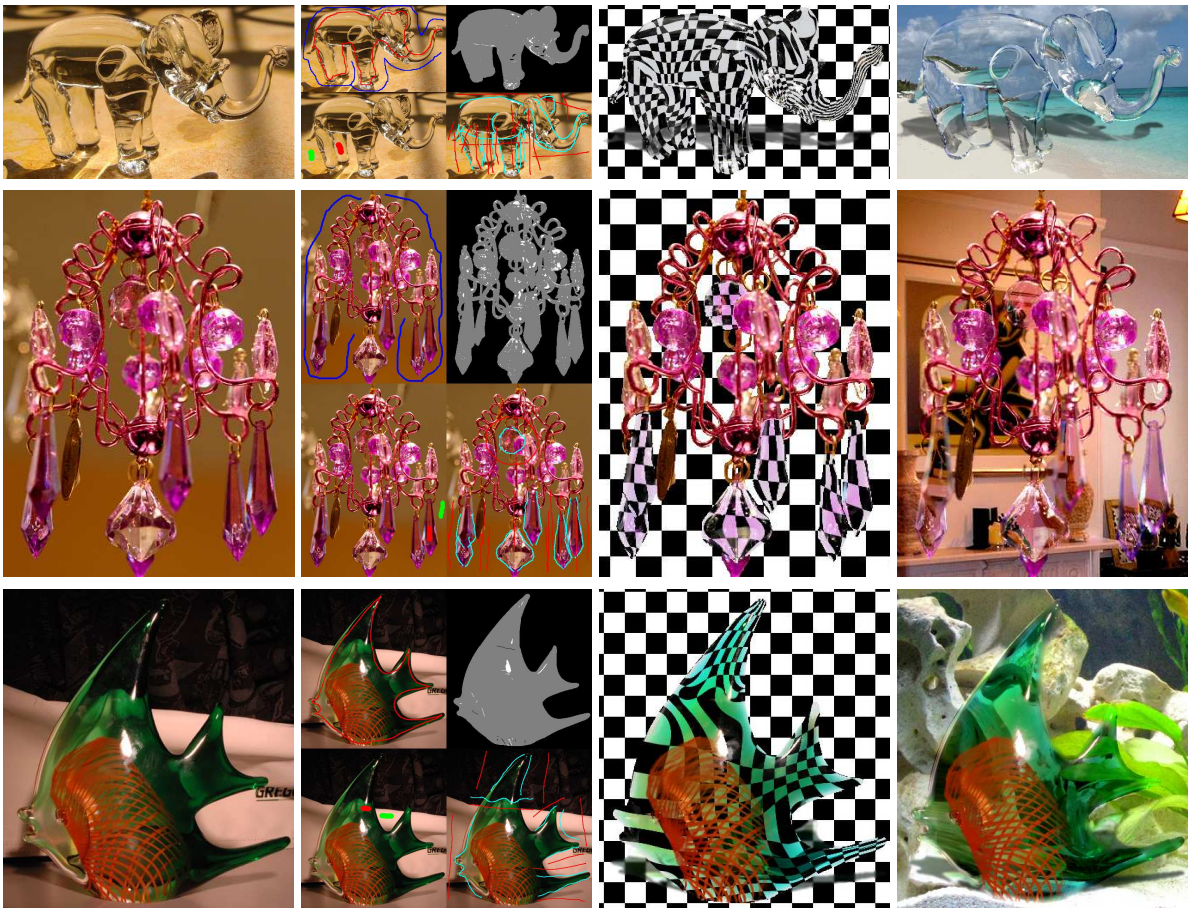


Figure 4.13: Results of extracting ARM. From left to right: Input, markup (from left in clockwise direction: M, α, G, and β), ARM and composite. The examples are (from top to bottom): *elephant*, *chandelier*, *fish*. All the markup for M, α, β and G shown here are complete. Refer to Table 4.1 for total number of stroke markups. The procedure to achieve the composite results will be discussed in the next chapter.

objects (e.g., *lion*, *dragon*, *swan*) do not require the corresponding markup for β.

4.5 Summary

In this chapter we have developed a new matting equation that accounts for the appearance of the transparent and refractive object observed from an image. We termed this extracted information the *attenuation-refraction matte* or ARM. We showed that with simpler user markup, the three properties of the transparent object: light reflection (specularities), light attenuation and light refraction can be represented by our ARM formulation and extracted given only a single image but no 3D information.

With the extracted ARM components, we are ready to composite the transparent

Figure 4.14: Results of extracting ARM. From left to right: Input, markup (from left in clockwise direction: M, α, G, and β), ARM and composite. The examples are (from top to bottom): *lion*, *dragon*, *swan*. All the markup for M, α, β and G shown here are complete. Refer to Table 4.1 for total number of stroke markups. The procedure to achieve the composite results will be discussed in the next chapter.

object onto a new background image. However, the visual effects by simply compositing the α, β and G on the new scene will not be compelling. Additional steps are required to achieve realism of the composite results. With the availability of ARM, these steps can be readily performed and will be discussed in detail in the next chapter.

Chapter 5

Compositing of Transparent and Refractive Objects

In the previous chapter we have discussed how to extract a transparent object from a single image by introducing a new matting equation. By simple user markup, we extract the plausible refractive deformation, color attenuation and foreground specular reflection from a given image and termed this extracted information the *attenuation-refraction matte* or ARM.

However, simply compositing the extracted ARM may not produce the best effect. In reality, the occluding boundary of a transparent object exhibits rich reflection behavior instead of being transparent. This phenomenon is called Fresnel effect [35]. In order to produce a compelling composite, the boundary of the matted transparent object must be enhanced with Fresnel effect [35]. While this effect is inherently captured by environment matting approaches, in our case it needs to be imitated realistically. In addition, compositing a transparent object without its caustic shadow will make the result look unrealistic. Traditional shadow matting techniques are not applicable here because a video is required [13], or the single-image formulation does not account for caustic shadows [98]. In [34], caustic shadows are simulated from a single image by detecting phase symmetry on the 3D depth map recovered from the image using the "dark is deep" assumption, which is not applicable to transparent objects.

In addition, in some cases we may need to composite more than one transparent object on the new background scene. A transparent object will be in front of another transparent object so compound refraction occurs. The refractive effect will also be changed if the transparent object is placed in a different depth position towards the background.

In this chapter we are going to show how to produce a range of effects when compositing an object into a new background. These effects include a Fresnel effect to enhance boundary realism, scene depth, multiple-object pasting, and the simulation of caustic shadows.

Because no comparable systems are able to provide similar matting and compositing capabilities, we perform comparisons between transparent objects transferred using Photoshop and those using the ARM representation. Our work suggests that not only extraction using ARM is faster, but the results are visually more believable.

5.1 ARM Compositing

After extracting the ARM from a photograph, a wide range of compositing effects can be produced to add visual realism.

5.1.1 Fresnel Effect

The occluding boundary of a transparent object produces a silhouette that exhibits rich reflection behavior. This phenomenon is characterized by the *Fresnel* effect: along the object's silhouette, the viewing angle to the transparent object becomes minimal. Light glancing off this boundary makes it look as if it is opaque to the viewer [35]. Without the proper simulation of the Fresnel effect, the composited object often does not appear solid as shown in Figure 5.1(c).

In our image-based approach, the target background image can be regarded as the surrounding light. Hence, the resultant appearance due to the Fresnel effect depends on both the transparent object and the background image. To simulate the Fresnel effect, we apply Poisson blending [63] *along boundaries* to incorporate the information from *both*

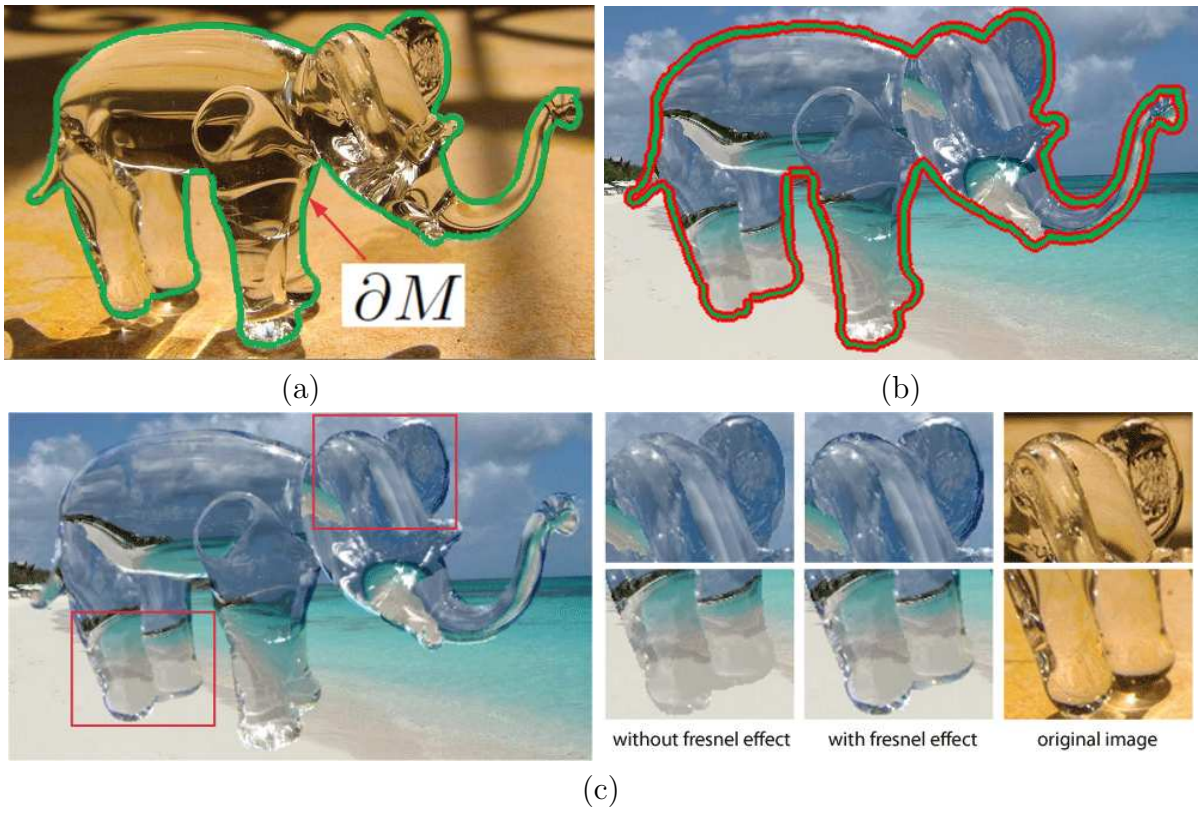


Figure 5.1: Simulating Fresnel effect using *Poisson boundary blending*. (a) The image gradient in ∂M (green) in the input image serves as the guidance field. (b) Dirichlet boundary condition derived from the pixel colors of the composited image excluding ∂M (red). (c) Solving the Poisson equation on ∂M using (a) and (b) produces a Fresnel effect that helps make the composited object looks like a 3D transparent solid.

the input and target background image.

Our approach amounts to solving a Poisson equation by taking into consideration: 1) the image gradient, ∇C, along the silhouette (∂M) of the input image (C) as the guidance field (Figure 5.1(a)), and 2) Dirichlet boundary condition derived from the pixel colors of the *composited image* surrounding ∂M (Figure 5.1(b)). In addition to the object's outline, this technique can also be applied along boundaries of the parts observed within the same object.

The width of the ∂M boundary can be adjusted given the image resolution and object shape. We found that widths from 4 to 6 pixels produced good results.

(a) (b) (c)

Figure 5.2: Simulating scene depth without 3D. In (a) and (b), the object size is fixed. The G is scaled to make the background appear farther away and closer to the object, respectively. In (c), all the ARM's components are scaled to simulate object movement toward the background.

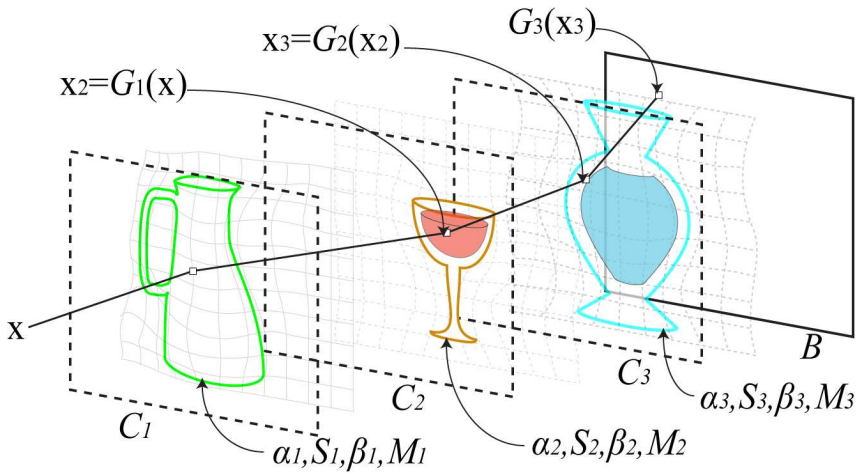


Figure 5.3: Compositing three overlapping image-based transparent objects.

5.1.2 Scene Depth

Although we have no 3D or depth information, we can simulate different scene depths when compositing a transparent object. Keeping the size of the object unchanged, we can make the background scene appear farther or closer to the object (shown in Figure 5.2(a) and (b)), by scaling the G component of the object's ARM. Alternatively, if we want to animate the object by moving it towards the background (see Figure 5.2(c) and the submission video), a scale factor is applied to all components of the ARM (M, α, β, G) at its desired scene location. This scaling also simulates a change in the object's apparent distance from the scene.

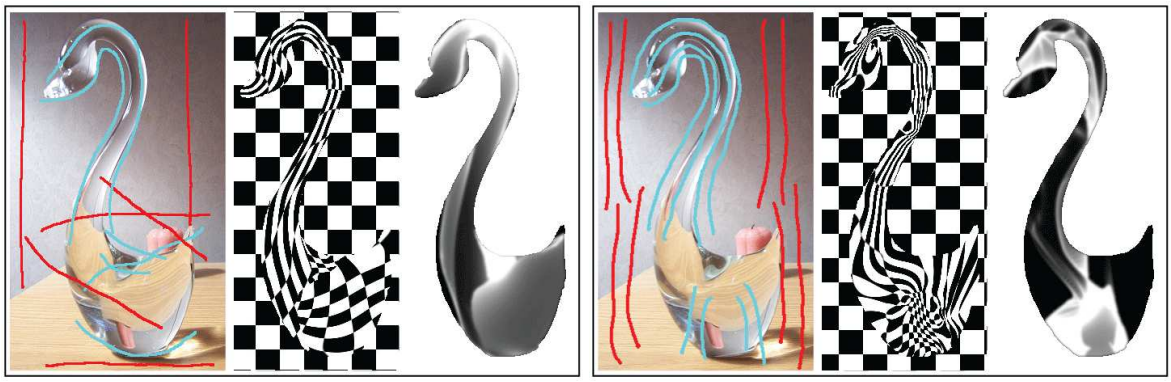
Figure 5.4: Examples on compound composition of multiple objects. Note that all results are entirely image-based; in our photo-editing application, no 3D models are available.

5.1.3 Compound Compositing

To produce the effect of compound refraction involving multiple objects, the user assigns a depth order for each object. Each object's ARM will be scaled as described in the previous section. Eqn (4.4) can be generalized to allow compound compositing of multiple and overlapping transparent objects, by writing it as a recurrence relation: Let S_n, α_n, β_n, and G_n be respectively the S, α, β, and G for object n, $n = 1, \cdots N$ and N is the total number of overlapping transparent objects, and M_n be the corresponding object mask. By letting $B = C_{N+1}$ be the original background without any transparent object, we have

$$C_n(\mathbf{x}) = \alpha_n(\mathbf{x})S_n(\mathbf{x}) + \beta_n(\mathbf{x})C_{n+1}(G_n(\mathbf{x})) \tag{5.1}$$

for $\mathbf{x} \in M_n$. Figure 5.3 illustrates the recurrence relation where $N = 3$. The final composite is given by C_1. Figure 5.4 shows several examples on compound compositing. Some of these examples show the extracted objects with different colors. This was achieved by rotating the hue of the input, the extracted β and the simulated caustic shadow.



(a) (b)

Figure 5.5: Two results for simulating caustic shadows using our procedure. In one result (a), markup (left) is added to produce deformation G (middle) and the simulated shadow (right). In the result (b), a new deformation G' is created by simulating the convergent lens effect.

Figure 5.6: Examples of results with and without caustic shadow. The caustic shadow adds an additional touch of realism.

5.1.4 Caustic Shadows

Since our ARM generation is non-geometric (no 3D model or explicit depth distribution), obtaining a depth map to simulate caustic shadows is not feasible. Instead, we exploit the availability of the refractive deformation G to aid in our shadow construction. Our method is intended to add an additional touch of realism but not to compete with accurate simulation of caustic shadows that use 3D models.

The procedure is as follows: at each pixel location inside M, the number of pixels mapped from the background based on refractive deformation G is accumulated. This estimates how light converges from the background to each pixel in the foreground object. The simulated shadow $\mathcal{T}$ is given by

$$\mathcal{T}(\mathbf{x}) = p + \beta H(\mathbf{x}, G)q, \tag{5.2}$$

where p is a small constant serving as the ambient intensity for the shadow, $H(\cdot)$ is a histogram function for tabulating at each pixel $\mathbf{x}$ the number of pixels mapped to $\mathbf{x}$ via the deformation warp G, and q is a user-supplied constant to control the apparent amount of light passing through the object.

In practice, directly using the H in Eqn (5.2) will result in a shadow very bright at a few points but very dark overall. We re-map the histogram by applying a non-linear logarithmic compression (other re-mapping functions may be used as well).

While the original G marked up for the ARM can be used, the user can also specify a new G' for a more realistic caustic shadow effect, as shown respectively in Figure 5.5(a) and (b). In our examples, the *fish*, *jug*, *globe*, *lion* and *swan* (Figure 4.13 and Figure 4.14) have new G' specified for the caustic shadows. The simulated shadows are warped by projective transformation before compositing. The shadow's contrast can also be adjusted to match the background image.

5.2 Comparison with Photoshop

We are not aware of any comparable matting and compositing system for editing transparent objects. The closest tool we can think of is Photoshop; hence, we used it to perform comparisons of results. In addition, we performed a user study involving 147 subjects to assess preference for our results or those produced by a Photoshop expert.

5.2.1 Transparent Object Transfer

Our ARM system is easy enough for a novice to be able to generate plausible results. Unfortunately, this is not so for the Photoshop. As a result, we hired a professional graphic artist. We contacted four professional agencies, with two responding (they characterized our task as a 'challenge'). In the end, only one was able to complete our request. This particular person was an accomplished veteran digital artist in addition to a Photoshop expert. We also asked an intermediate Photoshop user (who has 5 years of experience) to generate results.

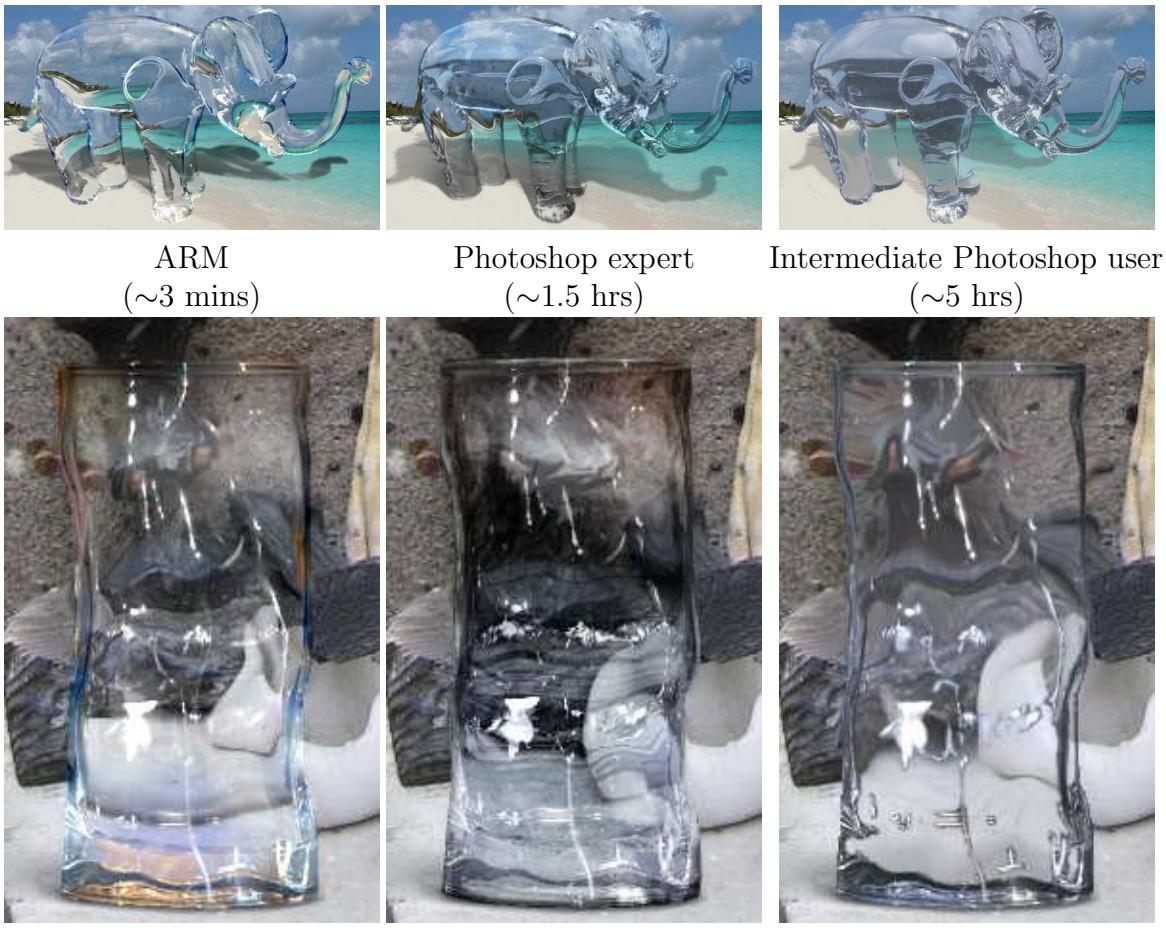


Figure 5.7: An expert and an intermediate user of Photoshop were asked to produce comparable ARM results. The ARM extraction requires significantly less time (the times shown here include interaction time and overheads such as looking at the photo and deciding what to do). No caustic shadow and Fresnel effect were simulated by both Photoshop users. Notice that the glass produced by our intermediate Photoshop user is the poorest despite requiring the largest amount of effort.

Both persons were given the original *elephant* and *glass* input image and the new backgrounds (same input used to produce our ARM composites). They were also shown our results merely to indicate the level of realism to achieve or exceed. Note that we did not ask them to replicate our results. The side-by-side comparisons are shown in Figure 5.7.

Photoshop expert. Interestingly, similar to our ARM representation, layers were used by the Photoshop expert to produce the transparent object transfer (Figure 5.8). However, unlike the ARM representation where each component has a semantic meaning (specular

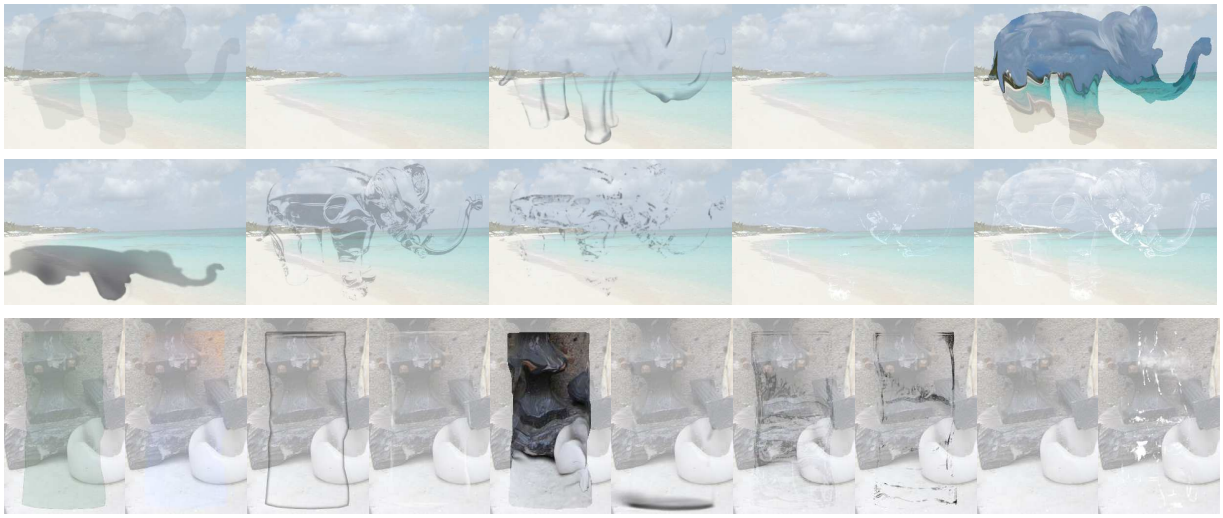
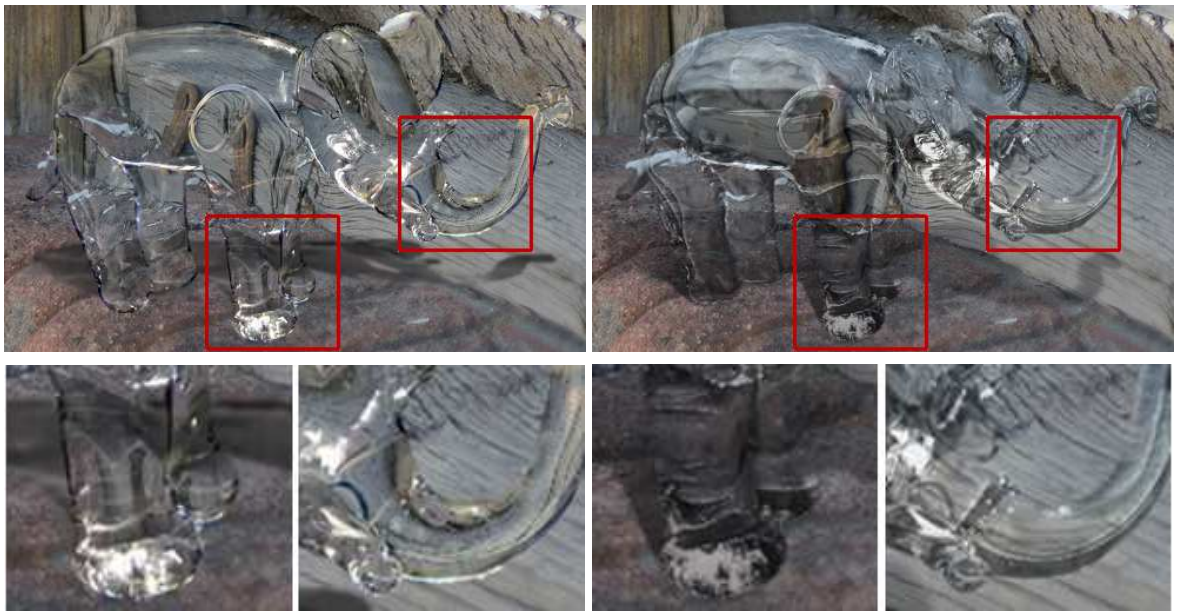


Figure 5.8: Layers derived by the Photoshop expert to transfer the transparent object (the backgrounds have been made lighter to make the layers more apparent). As opposed to ARM extraction, the derived layers rely on the expert's artistic sense and color perception on seeing the input and the background photo we provided; thus the approach may vary from other Photoshop experts.

highlight, attenuation, deformation), the layers produced by the Photoshop expert are ad-hoc in nature, involving several layers of color range selection that were subsequently blended with the background. There were approximately 10 layers for each example as shown in Figure 5.8. The silhouette of the objects were drawn manually. Refraction was performed using the 'liquify' filter in both examples and has roughly the same effect for both results. The Fresnel effect is not apparent around the elephant legs and trunk. The Photoshop expert did not attempt caustics in the elephant shadow.

With the majority of time spent on the *elephant* example, it took the Photoshop expert approximately two hours to produce results comparable to our ARM results. In several places, our Photoshop expert painted in pixels that were not extracted from the original. We note that layer mixing is tuned for the given background. Unlike our ARM representation, the results by our artist would need to be tuned for each new background. This is demonstrated in Figure 5.9 that shows the comparison of the Photoshop expert's extracted *elephant* placed onto a different background. As opposed to the Photoshop expert's layers, our ARM representation does not require fine tuning for a new background.

Intermediate Photoshop user. For our intermediate Photoshop user, 5 hours were



ARM Photoshop expert

Figure 5.9: Comparison of the ARM result against Photoshop expert's result when composited onto another background. Note in the ARM result the apparent Fresnel effect in the elephant legs and trunk, which are automatically generated when our ARM adapts to a new background image. The Photoshop expert's layers do not blend naturally and requires fine-tuning.

spent on the *elephant* and 2 hours on the *glass*. This user's photo-editing approach was first to remove pixels from the input image which were considered as background pixels using the eraser brush. The hues of the remaining pixels were then rotated to adapt to the new background. Because of the complex shape and illumination effects, a great deal of manual editing was necessary to make the result visually acceptable. Similar to the Photoshop expert, the liquify filter was applied to simulate the refractive deformation, however, the effect by the intermediate user is not as visually appealing as that done by the Photoshop expert. The caustic shadow was not attempted either.

5.2.2 User Study

Our user study is performed to evaluate viewers preference between the composited results generated by our ARM approach or the Photoshop expert. In our study we did not include the results produced by the intermediate Photoshop user because they were not comparable in terms of visual quality as shown in Figure 5.7.

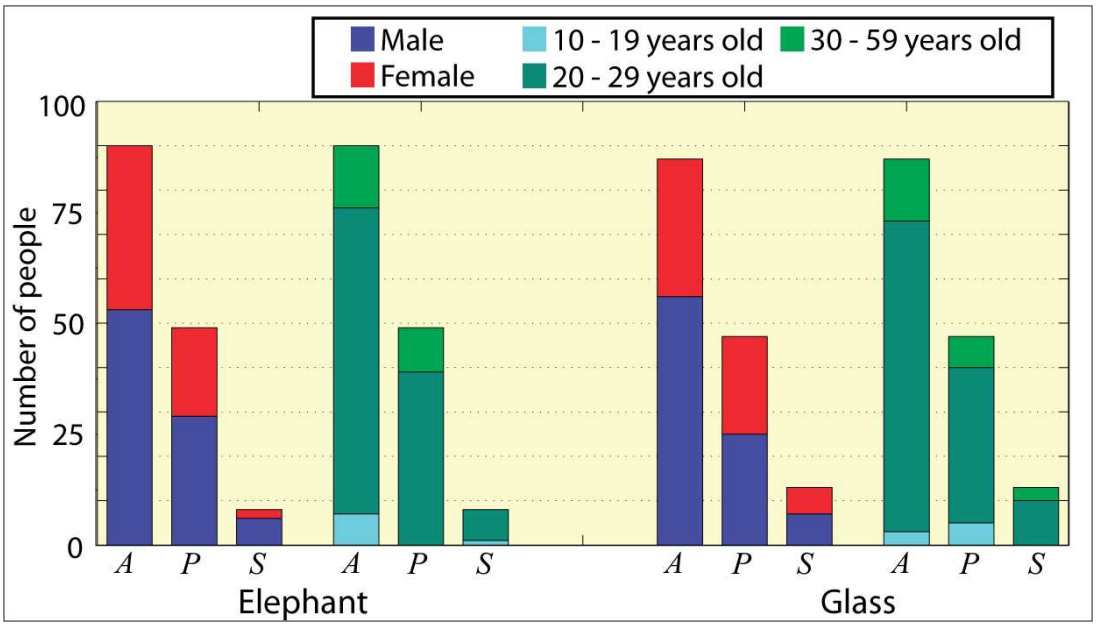


Figure 5.10: Survey result on 147 subjects. Their preferences are indicated by A: ARM result, P: Photoshop expert's result, and S: both results look similar.

We sent a distribution list via email to invite people in our international campus community to participate in our survey. They were also invited to forward our invitation to their relatives and friends. In the end, a total of 147 persons in different age groups and genders responded to our online survey.

In our online survey, the *elephant* and *glass* images produced by the ARM approach and the Photoshop expert were presented to a viewer. The image pairs in each example were shown side by side for direct comparison, and we randomized the order shown for *elephant* to *glass* across different subjects to avoid bias. Each subject was asked to make a decision within a time limit of 30 seconds on each of the following questions: 1) "Please select your preferred image of a transparent and refractive elephant standing on a sandy beach." 2) "Please select your preferred image of a glass placed in front of a background". The user could also indicate that both results were visually similar, indicating no preference for either. Additional comments could be input in a text box available in the survey.

The statistics of our user study were collected and plotted in Figure 5.10, which indicates that our results are preferred. Comments by subjects indicate that the shadow and the refraction of the legs of the *elephant* in the Photoshop expert's result look artificial. Some suggested that our results look clearer and are more pleasing. Some commented

that results may depend on certain material properties and thus difficult to tell which one is better.

5.3 Discussion

While 3D object reconstruction requires measurement against ground-truth for validation, our ARM approach targets plausible refraction effects. This can be a subjective matter; in this thesis we showed numerous examples and conducted a user study to demonstrate that our system is successful in achieving this goal. Our efficient system provides interactive photo-editing capability with almost instant feedback for each stroke the user marks up on the image, thereby allowing the user to easily experiment the effects of different deformation such as those depicted in Figure 4.11.

Because we make some simplifying assumptions in our single-image scenario to facilitate the ARM extraction, our technique cannot be readily applied to extract complex transparent objects from photos. Examples include multi-colored transparent objects (that is, violation of the smooth β assumption). Specularities on the transparent object extracted cannot be adapted to a different lighting environment where the background image was captured. This limitation is inherent in all matting and compositing techniques using single images.

Our ARM cannot handle translucent objects which scatter light (e.g., jade). Such objects can be regarded as largely reflective and conventional matting is sufficient for their extraction if their apparent foreground colors are known. For objects with both transparency and opacity (e.g., *chandelier*) or consisting of different refractive mediums (e.g., *jug*), a rough segmentation on M (object mask) is needed to separate the processing regions.

While transparent object transfer can be performed by an experienced Photoshop user, our ARM representation has several advantages. First, its extraction is significantly faster and geared towards novice users. In addition, the ARM produces a meaningful and complete transparent object representation that can be used to paste onto a new background without fine tuning. The explicit ARM representation also allows for effects

such as multiple object compositing, depth manipulation, and a procedural mechanism for producing caustic shadows.

Finally, knowing the 3D shape of the object offers much more opportunity for producing more realistic compositing effects, however, we emphasize that our work operates directly on a single image without 3D shape information and demonstrates a significant first attempt at photo-editing for transparent and refractive objects.

5.4 Summary

We have introduced the attenuation-refraction matte (ARM) which provides an image-based model to encode the visual effects associated with transparent and refractive objects. In this thesis, we described how the ARM can be extracted from a single image, and how to use the ARM to paste the object into a new background. We show that plausible refractive deformation suffices in producing visually compelling results. We believe our work is the first to allow photo-editing of transparent and refractive objects in cases where only a single image is available. In addition, we have shown a variety of compositing results that cannot be easily replicated using existing single-image editing tools.

Chapter 6

Modeling and Rendering of Impossible Figures: an Image-Based Approach

In the previous three chapters we present an image-based approach to extract transparent objects. Under the theme of human-computer interaction where the user is involved to make the extraction possible, by exploiting our human visual system and taking hints from human user, we made an important pass in image-based transparent objects extraction. In the following two chapters, we will investigate the role of human in modeling and rendering of *impossible figures*. Again, the remarkable human visual ability will be very useful in solving the problem. Let us first define the notion of impossible figure.

6.1 What is an Impossible Figure?

The human visual system has a remarkable ability to make instant connection to 3D on things we see. Even though what we look at are just paintings and 2D drawings on paper, we can effortlessly infer the third dimension and evaluate depths and distances. This connection, however, can lead to interesting problems. In 1934, a Swedish artist Oscar Reutersvard introduced *impossible figures* [17]. These are 2D drawings that can confuse our visual system, see Figure 6.1(b) for the nine-cube arrangement that was first drawn

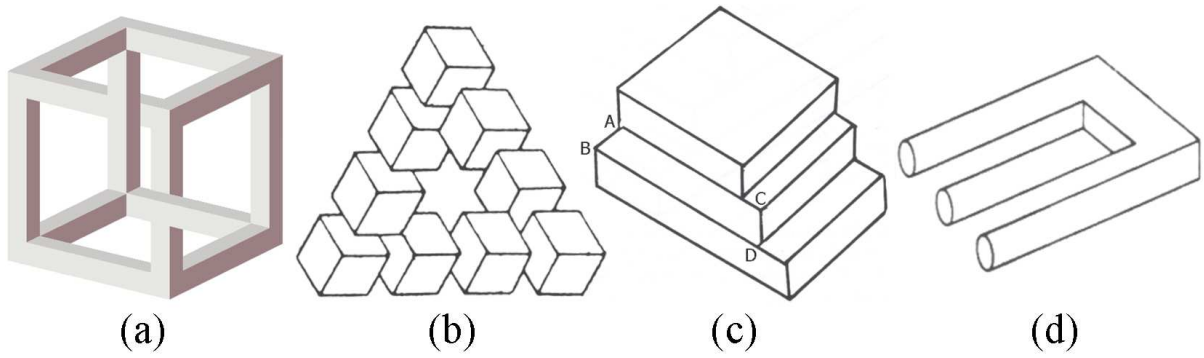


Figure 6.1: Representative examples of the four basic classes of impossible figures: (a) the impossible cuboid, (b) the nine-cube arrangement, (c) the impossible stairs, and (d) the impossible bar.

by Reutersvard. When this kind of drawings is presented before our eyes, we can only distinctly perceive local 3D structure of individual parts in the drawing. When we are asked to view the entire drawing as a whole, structural inconsistency will arise and confuse our minds when we attempt to build a globally-consistent 3D structure. The following summarizes the four classes of impossible figures:

- **Depth interposition** See the impossible cuboid in Figure 6.1(a), where the optical illusion is caused by structural inconsistency due to problematic depth ordering.
- **Depth contradiction** Refer to the nine-cube arrangement and also the Penrose triangle by the Penroses [62] in Figure 6.1(b), where propagation of local 3D information gives rise to global structural inconsistency.
- **Disappearing normals** See Figure 6.1(c) for an example of impossible stairs, where the plane $ABCD$ depicted in the figure could appear to be horizontal at one side while vertical at the other, making it impossible to assign a consistent normal across the whole plane.
- **Disappearing space** See Figure 6.1(d) where the silhouette of the impossible trident is not closed.

Figure 6.1 shows four fundamental impossible figures. The nine-cube arrangement (Figure 6.1(b)) can induce a similar figure, known as the *Penrose triangle* (see the teaser figure). It was created by the Penroses [62] in 1958. Similar to the nine-cube arrangement,

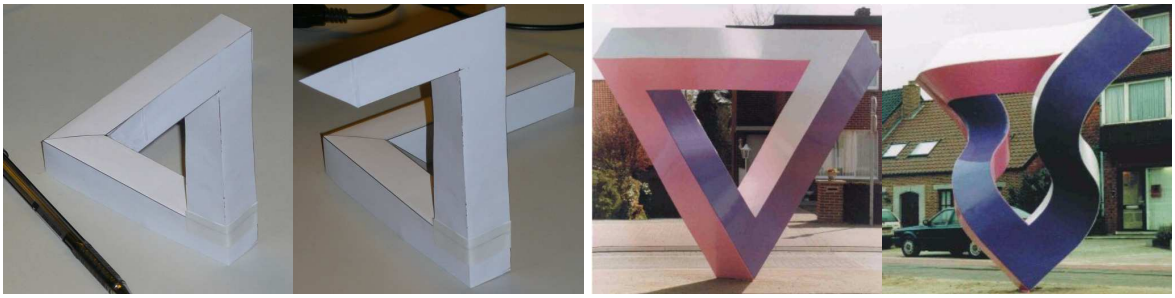


Figure 6.2: Only one view of the impossible figure can be produced from the corresponding 3D model with parts disconnection (left) and structure twisting (right).

depth contradiction is resulted when we attempt to propagate local 3D information over the entire Penrose triangle. The *impossible cuboid*, shown in Figure 6.1(a), demonstrates a slightly different situation. Rather than contradiction brought by depth propagation, the idea behind the impossible cuboid is to mess up the depth ordering; such a technique is also known as the interposition. In the figure, we can see that the (expected) far bottom corner in the cuboid weirdly occludes the corner supposedly closer to the viewer. Ernst [17] wrote a book on the subject of impossible figures, where he described various key elements in the world of impossible figures.

Here, we can see that when we look at local regions around triangle corners, we can instinctively reconstruct local 3D structures such as face orientation, but when we attempt to spread these local 3D information over the 2D drawing, depth estimation contradiction will occur.

The impossible cuboid is created by the *interposition* technique, where the depth ordering between parts in the drawings confuses our mind when our visual system tries to infer global structure in the 2D drawings. Other than occlusion, the Penrose triangle demonstrates another interesting effect known as the depth-estimation contradiction; this happens when our visual system tries to propagate local 3D structure over the picture.

Impossible figures have been receiving much less attention in computer graphics, possibly due to these difficulties which make geometry proxy assignment impossible. This stems from the very nature of impossible figures, where automatic detection of global impossibility remains elusive and limited to line drawings [37, 36]: specifically, an impossible figure is a special kind of 2D drawing that has *locally-possible* 3D parts, but the overall

geometry is *globally-inconsistent* as we have discussed in the previous paragraphs.

It can be noticed that, given only a single image of impossible figure, not even to locate the impossible parts, it is very difficult for the computer to tell it is an impossible figure. But we, human beings, can detect such global-inconsistency in the overall geometry easily. So again, we propose to bring human users into the loop to locate and segment the local possible parts. The computer can then do the job to let the user predict the appearance of the impossible figures in the novel view. With the predicted view from the human as a hint, the computer can generate the desired view. The approach in this chapter will in image-based. In the following section we will review some related works on image-based modeling and rendering and also how different scholars work on impossible figures.

6.2 Related Work

6.2.1 Image-Based Modeling and Rendering

Image-Based Modeling and Rendering (IBMR) [41, 73] has significantly contributed to the efficient rendering of complex real scenes, bypassing the modeling bottleneck of building complex 3D models. To reduce visual artifact, the plenoptic sampling theory [11] indicates that different amount of geometry proxy (e.g., in layered depth images [71] and unstructured lumigraphs [10]) should be employed, depending on the number of sampled images, where the depth relationship among segmented layers is delineated.

In terms of goals and challenges in image-based modeling and rendering, a real image also shares a great deal with impossible figure, which has been widely used in various applications (see practical examples in computer games, non-photorealistic rendering, image synthesis, and photo-montage [3], where imaginary and geometrically impossible scenes are rendered). First, the common goal is to provide plausible novel views given a very limited number of input images. Second, typically only a single image of an impossible figure is provided. Third, while it is tedious to build a 3D model of a complex real scene, it is indelibly impossible to build a 3D model that corresponds to the impossible figure without severe structure twisting or parts disconnection, as shown in Figure 6.2.

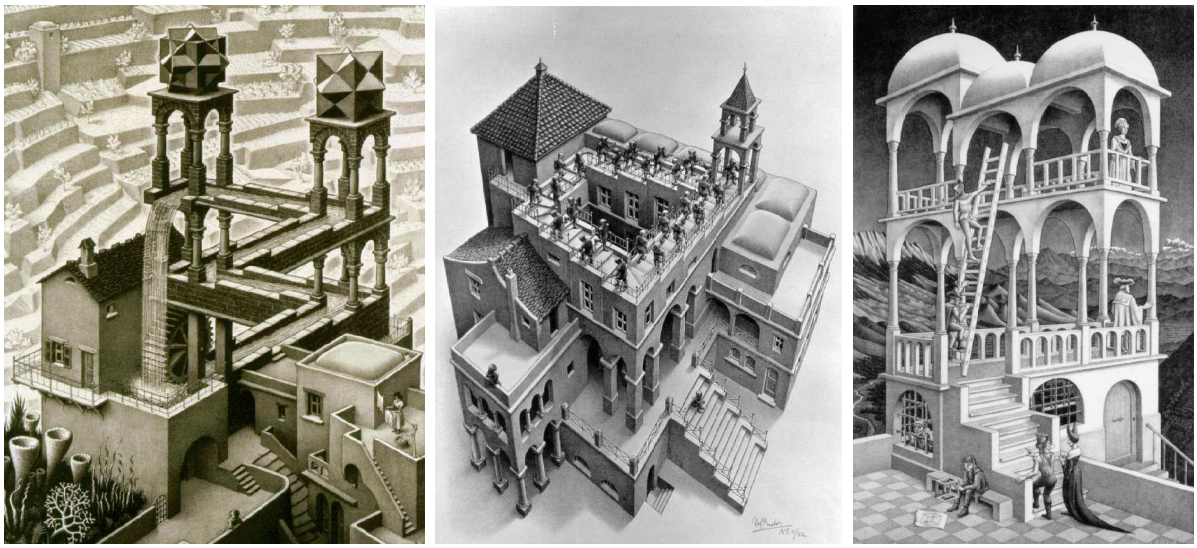


Figure 6.3: M.C. Escher's original drawings: Waterfall (left), Ascending and Descending (middle), and Belvedere (right).

Even though such a 3D structure is built, it can provide one view only [16, 50].

In this chapter, we attempt to extend conventional IBMR to render impossible figures from a single image. Similar in spirit to layered depth images [71] for real imagery, our approach consists of a) *segmenting* the figure into a set of layered images, b) assigning normals and geometric proxies, followed by c) *connecting* individual parts to produce a new impossible figure. While strikingly similar to conventional IBMR except the last connection step, our working "impossible-figure-in impossible-figure-out" solution is designed in parallel with how an impossible figure is perceived: while we have no problem in perceiving each *segmented* locally-possible part as a 3D sub-structure, when all parts are *connected* in 2D, a globally-impossible 3D structure is perceived.

6.2.2 Modeling of Impossible Figures

M.C. Escher [56, 68] is a renowned art master in making artwork of impossible figures. He successfully popularized impossible figures by skillfully embedding models of impossible figures in architectural drawings. Figure 6.3 presents three of his masterpieces: *Waterfall*, *Ascending and Descending*, and *Belvedere*. *Waterfall* and *Ascending and Descending* were created based on depth contradiction, while *Belvedere* was created based on the depth interposition technique demonstrated in the impossible cuboid.

To produce novel views of an impossible figure, 2D artists have experimented simple tricks such as 2D scaling and stretching, but this fails to make the transformed figure look like a 3D solid. Other than that, 3D geometric approaches were proposed, such as the creation of tangible 3D models for impossible figures. Elber [16] created physical models for a wide range of impossible figures, including also those by M.C. Escher. In a similar fashion, Lipson [50] made use of LEGO bricks to build physical models of various impossible figures rendered by Escher. Common to these approaches is that in order to make possible the corresponding 3D geometry, which is a non-trivial task by itself, continuous linear structures have to be disconnected or otherwise heavily twisted. These tricks somehow ruin the "magic" of impossible figures, which lie in the way how straight lines are connected in 2D to make the overall figure globally-impossible in 3D. Moreover, the resultant 3D models are very contrived, and resemble an impossible figure only when viewed from restrictive directions.

Khoh and Kovesi [43] generated novel views of impossible figures using the idea of two complementary halves, where the two halves are 3D models related by an inversion transform in the image plane. However, their approach works only for a particular subset of impossible figures, and the thickness of the two complementary halves may not match after inversion transform. A similar approach was adopted by Tsuruno [85], where a 3D model of *Belvedere* is pre-constructed to create different views of *Belvedere*.

For artwork of impossible figures, Simanek [74] presented a proposal to create false perspectives on impossible figures; simple tricks such as prolonging the horizontal dimension are used to create stereo pairs of impossible figures. Though this method lacks sufficient geometric reasoning, it is nevertheless the first proposal we noticed on creating stereo views for impossible figures. In particular, Savransky [67] described impossible 3D scenes by encoding linear transformations between the neighboring 3D parts in the impossible 3D world, by solving a correct viewpoint (a projection transformation) that optimally projects the impossible scene. Uribe [86] proposed to use a set of triangular tiles to design and create 2D impossible figures.

6.3 Image-Based Modeling

Koenderink *et al.* [44] introduced *pictorial relief*, which is a surface in the 3D 'pictorial space' perceived in a single, flat picture. Although largely applied to real imagery, the theory can be applied to impossible figures, as conjectured by the author, where the perception results are locally-consistent but globally inconsistent. In particular, the study involved a large number of human subjects, and it was found that dense normal assignment among all subjects is very consistent. Our interactive image-based modeling approach contains these two crucial ingredients: 1) segmentation of locally-consistent parts, and 2) normal assignment for generating proxy and subsequent rendering, except for 2) we only require sparse markup. We summarize below the steps:

1. We segment an impossible figure into individual parts which are possible in 3D. See Figure 6.4 for some examples.

2. We also segment an impossible figure into individual regions which is the 2D bounded regions in the original figure, also see Figure 6.4.

3. Compute normal maps over the image regions, and derive height maps for segmented parts by leveraging existing height-from-normal algorithms such as [28].

4. Rotate the local possible parts in 3D. Once these heights are available, they act as geometry proxies of the local shapes. As shown in Figure 6.5(a), although the resulting views correspond only to those of the local shapes, they together provide very good cues on how the impossible solid object would have appeared after rotation.

5. Straight lines are fitted by user based on the rotated impossible part to form a completed impossible figure in line drawing.

6. The final impossible impossible figure at novel view is obtained by warping the original impossible figure to the fitted straight lines by the user (Figure 6.5(b)) to preserve linearity and intersections. The results are something like (Figure 6.5(c)).

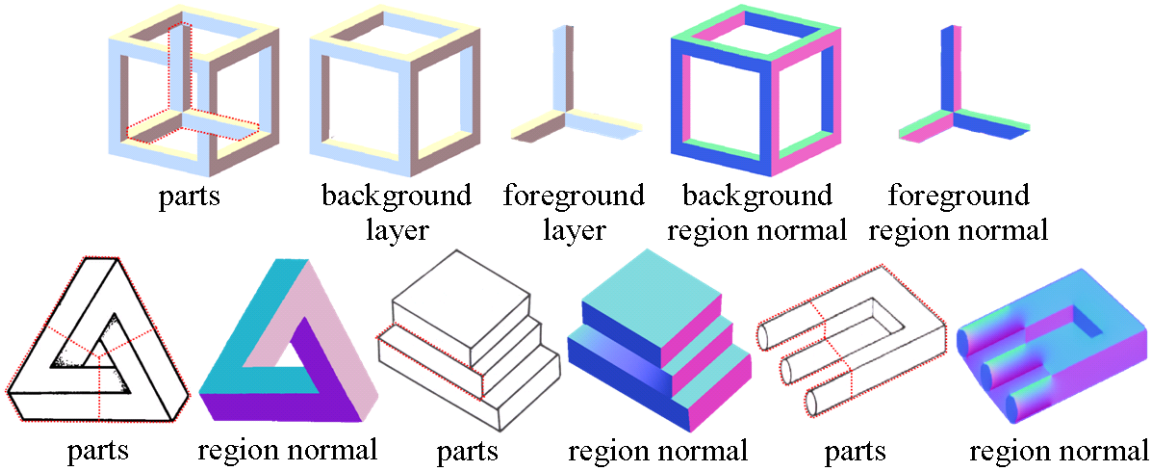


Figure 6.4: Parts segmentation (segmentation boundaries are shown in red) and region-based normals for the input figure. Top: impossible cuboid. Bottom, from left to right: Penrose triangle, impossible stairs, and impossible trident.

6.3.1 Segmentation

An impossible figure is segmented into *parts* and *regions*:

Parts segmentation is used to partition an input impossible figure, as shown in Figure 6.4, into a set of locally-possible units: the problematic corner of the *impossible cuboid* is disconnected from the cuboid, making the two independent parts simple 3D objects. To handle occlusion, we designate the occluding corner as the foreground layer, and the rest of the cuboid as background layer. The *Penrose triangle* is decomposed into three possible polyhedral objects (parts), where the parts segmentation introduces new edges. Similar to impossible cuboid, the problematic plane is segmented out from *impossible stairs*, making the resulting surfaces feasible in 3D. For *impossible trident*, the problematic bars are disconnected from the object body, making all segmented parts possible in 3D.

Regions segmentation is usually given by the 2D bounded regions in the original figure, where normals can be assigned in each segmented region, see Figure 6.4. The exception is *impossible trident*, where the silhouette is not closed, and we need parts segmentation to form regions before the normal assignment, refer to the figure.

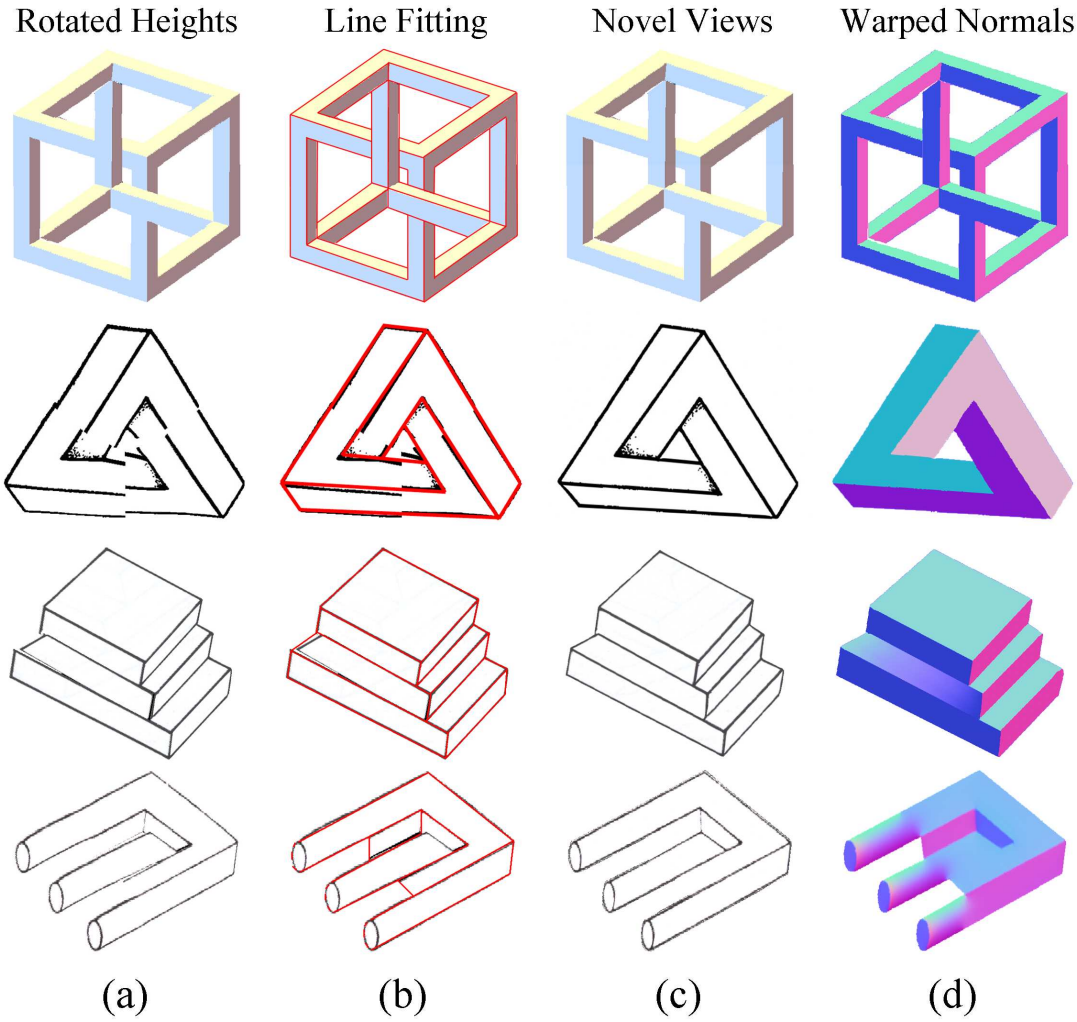
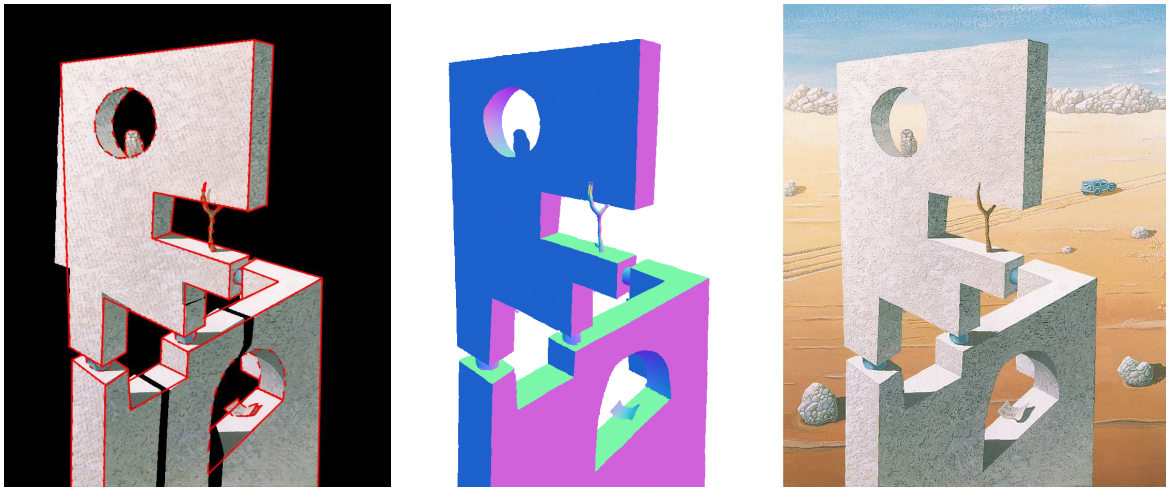


Figure 6.5: Synthesis of novel views and the corresponding normal maps. (a) We rotate height maps of all locally-possible parts to derive plausible novel views of impossible objects when viewed from a different viewpoint. (b) New impossible figures are generated by line fitting (red lines) such that linearity and line intersections are preserved. (c) The impossible figure at novel views. (d) The corresponding per-pixel normal maps warped from the region normals at the input view.

6.3.2 Normal Assignment and Height Generation

Dense per-pixel region normals are automatically generated after using a normal markup tool, such as shape palettes [96], where the user assigns sparse normals to each segmented region in the input image, see Figure 6.4. It is interesting to note how the normal orientations change from one end to another on the problematic plane of *impossible stairs*. The normal map of *impossible trident* consists of four components as shown in Figure 6.4 due to the parts partitions.

Dense per-pixel heights for each locally-possible segmented part will also be automati-



(a) (b) (c)

Figure 6.6: *Construction*. We rotate height maps of the partitioned parts to derive a plausible novel view of the impossible object. (a) Line fitting for generating a new impossible figure such that linearity and line intersections are preserved. (b) The corresponding normal map. (c) The impossible figure at a novel view.

cally estimated, again by shape palettes, where the parts segmentation boundaries provide the necessary constraints to disconnect locally-possible parts from the figure. The resulting height maps act as 3D geometric proxies for rotating the segmented parts as shown in Figure 6.5(a). For complex artwork, such as *Belvedere*, height proxies will be simplified into planar structures, that is, at least three identical normals will be assigned before height generation. The details of the *Belvedere* results are given in the next section.

6.3.3 Line Fitting and Warping

The rotated view provides useful cues for deriving the desired novel view by fitting straight lines, as shown in Figure 6.5(b), where we connect rotated parts to preserve linearity and intersections.

Now, the input figure and the normal map are warped toward the user-drawn lines as shown respectively in Figures 6.5(c) and (d). This warping procedure is automatic after specifying lines of correspondences [6] as part of the line fitting procedure. Note that one-to-one mappings between input regions and warped regions are available, because in practice the angle difference between two views is not allowed to exceed 30 degrees, which is considered quite large given that the object is geometrically impossible.

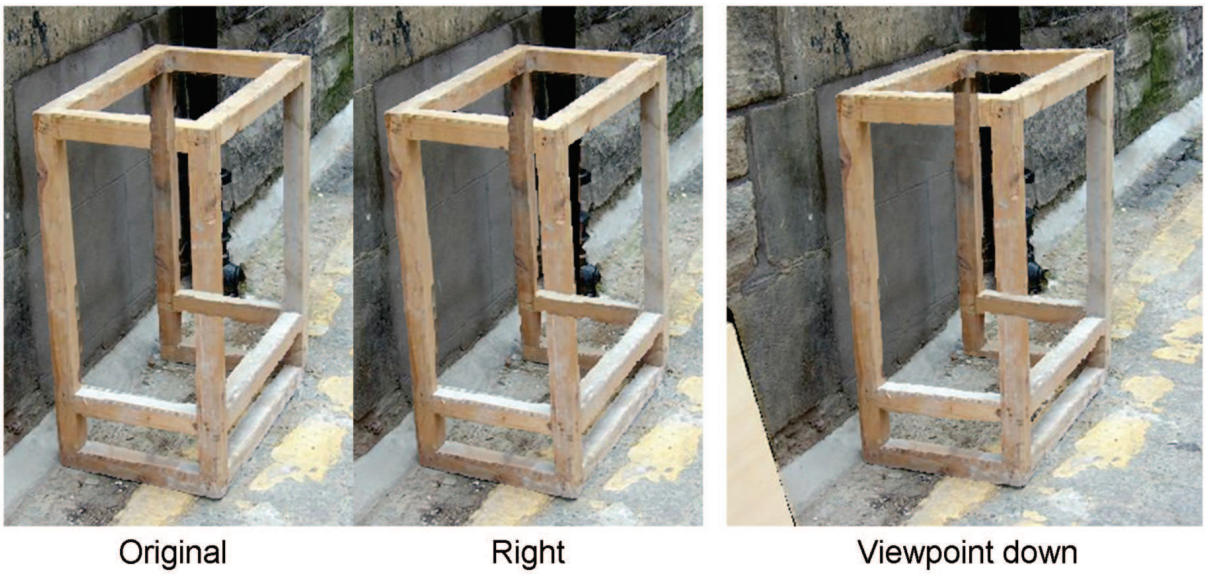


Figure 6.7: Real images of an impossible figure.

If occlusion occurs, warping is first performed on the background layer, followed by the foreground layer. For example, in *impossible cuboid*, the foreground layer consists of the regions making up the problematic corner, whereas the background layer is comprised of the rest of the cuboid. The occluded portions in the input figure and normal maps will be completed using the clone brush in PhotoShop or [78] before warping.

6.4 Image-Based Rendering

The dense per-pixel region normals and part heights constitute the image-based model, which can be employed to render impossible figures in a variety of ways.

Complex artworks Figure 6.6 shows the results on *Construction* (1997). Similar to the *Penrose triangle*, *Construction* can be decomposed into several locally-possible parts, where each part is a simple 3D object. A novel view rendered using a similar procedure to *Penrose triangle* is also shown here.

Stereopsis We apply the derived image-based information to reproduce left-views and right-views for various impossible figures so as to create stereo drawings or views. Figure 6.8 shows a set of three images for each impossible figure: the impossible cuboid, the Penrose triangle, the impossible stairs, and the impossible trident. The same technique can be applied to real images of impossible figure. Figure 6.7 shows the stereo-pair

consisting of the input image, the right-view as well as another novel view. For best stereo viewing, the viewing distance is around 3–4 inches from the paper.

6.5 Summary

In this chapter, we have introduced the class of impossible figures and some of their properties. We propose an IBMR framework for impossible figures that match with how an impossible figure is constructed and subsequently perceived. In line with conventional IBMR for real imagery, our extended approach does not construct the corresponding 3D models.

Presently, we segment the parts and regions by hand. Although it would be possible to automate the line fitting process by formulating linearity and line intersections as optimization constraints.

In the next chapter, we will materialize the above idea and present another approach based on view-dependent modeling where manual line fitting is not needed. Once again, we will make use of our human visual system to help in solving the problems by reducing the amount of user input.

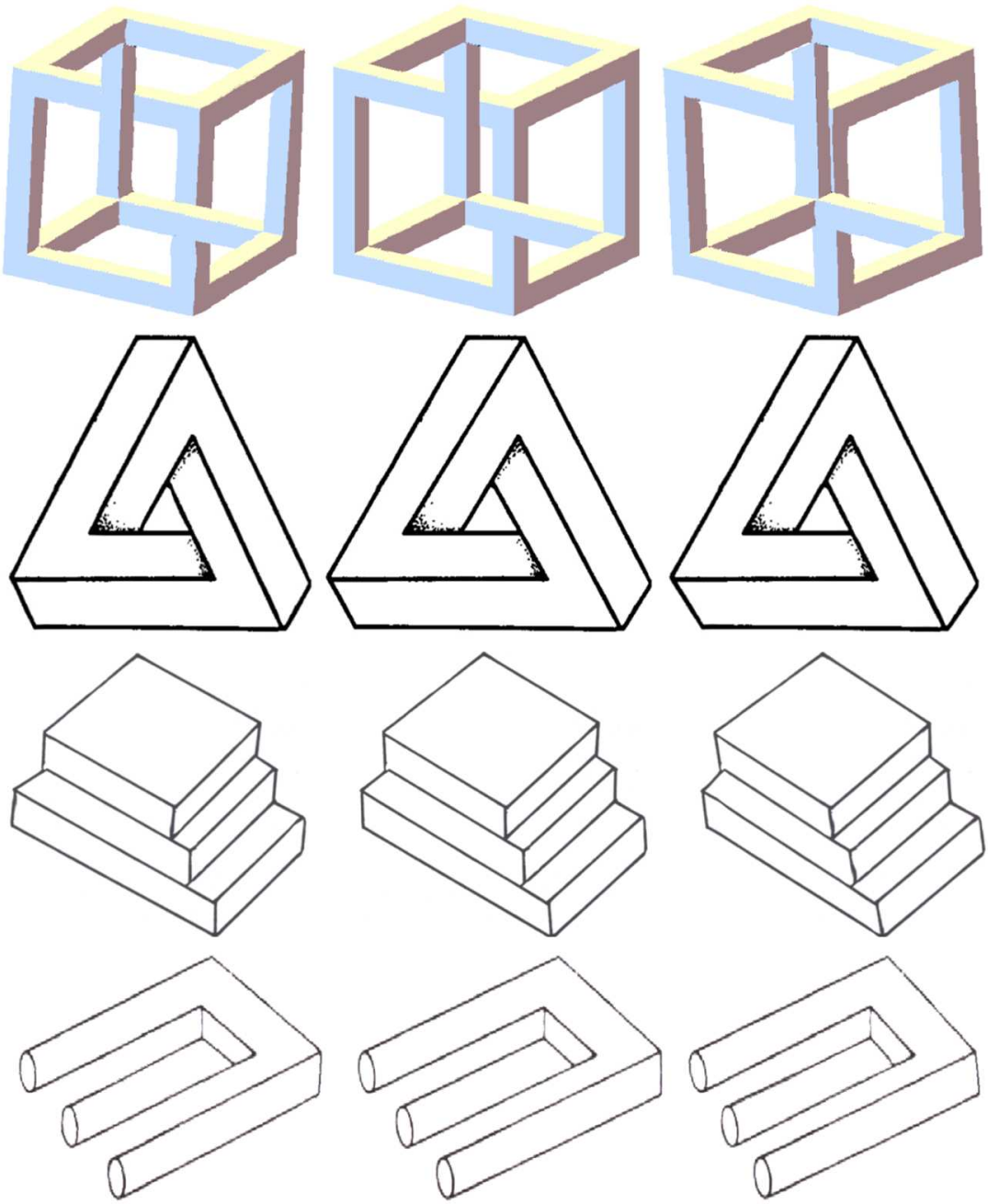


Figure 6.8: Stereopsis: left, middle and right views of impossible cuboid, Penrose triangle, impossible stairs, and impossible trident.

Chapter 7

Modeling and Rendering of Impossible Figures: a View-Dependent Modeling Approach

In the last chapter, we have extended image-based modeling and rendering to the class of impossible figures. We show how layered segmentation, normal assignment, and 2D line fitting can be used to generate plausible novel views from a single image of impossible figure. In fact, 2D artists have experimented with simple tricks such as 2D scaling and stretching. However, when the novel view is far from the reference view, 2D warping fails in making the transformed figure look like a 3D solid.

To tackle this, we formulate a view-dependent model of the impossible figure, subject to the necessary 3D constraints for rendering the impossible figure at the desired novel viewpoints. One advantage of a 3D approach is that temporal coherence in animating an impossible figure is automatically and smoothly maintained, as a 3D model is available for animation. Let us first analyze the 3D structure of an impossible figure.

7.1 Analysis

When presented with an impossible figure before our eyes, we can perceive local 3D structure of the individual parts in the drawing. When we view the entire drawing as a

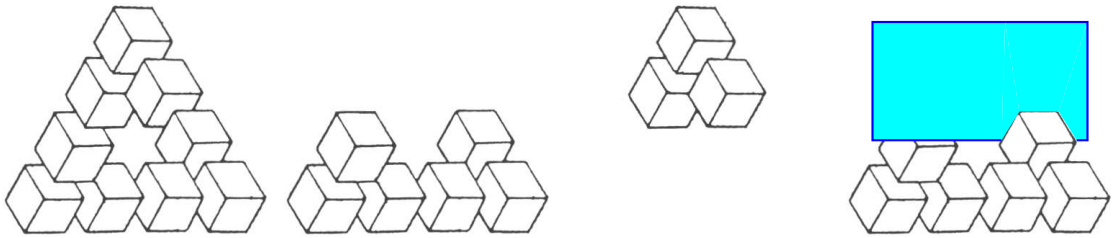


Figure 7.1: (a) *Nine-cube Arrangement*. (b) and (c) are the two possible parts of (a). (d) A tilted plane is placed as shown in the 3D model. This plane serves as the image plane where (c) is projected to produce the impossible figure in (a).

whole, structural inconsistency arises and we become aware that the figure is invalid as we attempt to mentally build a globally-consistent 3D structure from the impossible figure.

7.1.1 Dissecting an Impossible Figure

We believe that the visual confusion caused by viewing an impossible figure lies in the presence of multiple 3D structures projected by using *different* transformations, while some of them violate the laws of rigid camera transform but the resulting projected 2D structures still retain 3D semantics.

For example, if we separate the upper three cubes from the rest in Figure 7.1(a), as shown in (b) and (c), the two subfigures correspond to 3D rectilinear structures and can be respectively obtained using an orthographic camera. Now, suppose we put a tilted plane between the two cubes in the 3D model of Figure 7.1(b), thereby producing (d). Then, we project orthographically the 3D model of Figure 7.1(c) onto the plane. Combining the resulting image with Figure 7.1(b), we now obtain Figure 7.1(a), the nine-cube arrangement.

In addition to using a rigid standard camera, this example demonstrates that non-standard projective transformation is needed to create the impossible figure. While the orthographic camera above is one form of rigid transformation, the above tailor-made procedure (that is, the usage of a tilted plane and combination of images) can be characterized by a more general, non-rigid transformation.

To the best of our knowledge, there is no camera model available to generate an

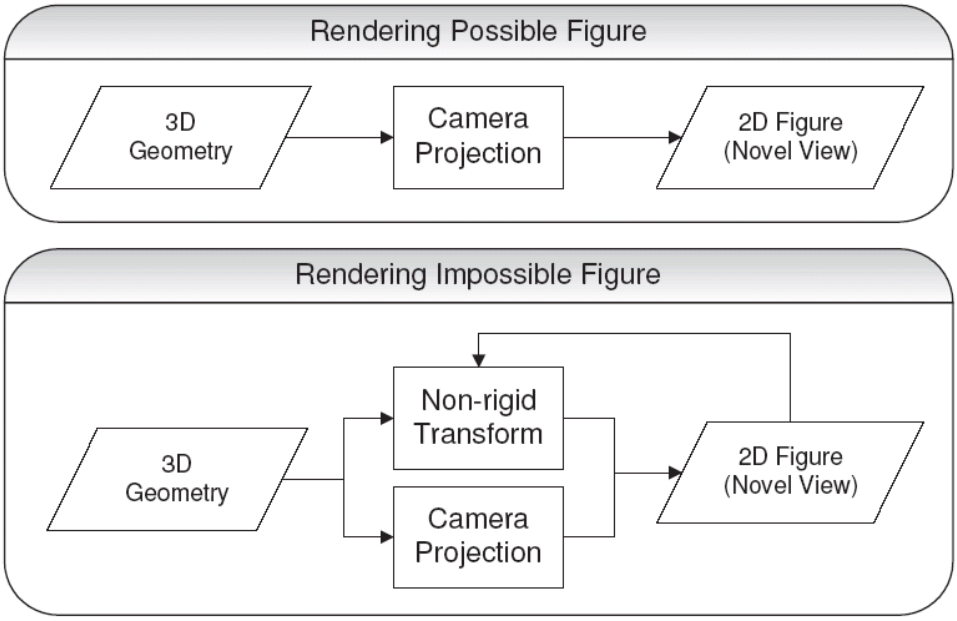
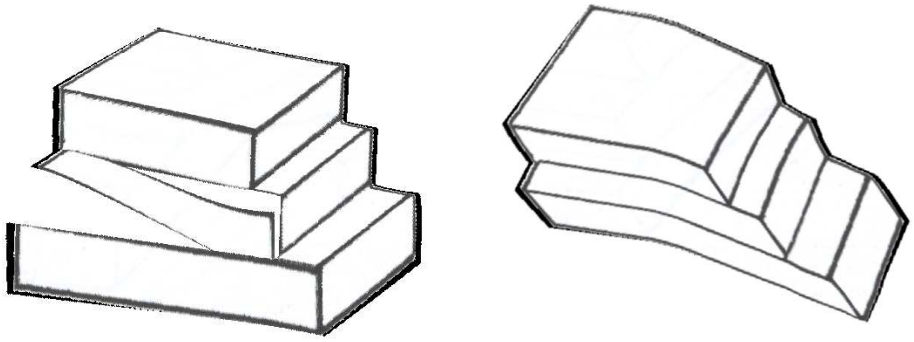


Figure 7.2: Rendering possible and impossible figures.



(a) (b)

Figure 7.3: Two renderings respectively dominated by (a) rigid transformation and (b) non-rigid transformation.

impossible figure by appropriately combining rigid and non-rigid transformation.

7.1.2 Rigid and Non-rigid Transformation

The above analysis suggests that a reconciliation process is required of rigid and non-rigid transformation in producing an impossible figure. Specifically, while a novel view should be specified with respect to a rigid camera, it is necessary to incorporate non-rigid transformation to connect multiple 3D structures so as to produce the global structural inconsistency. The non-rigid transformation, on the other hand, must be constrained so

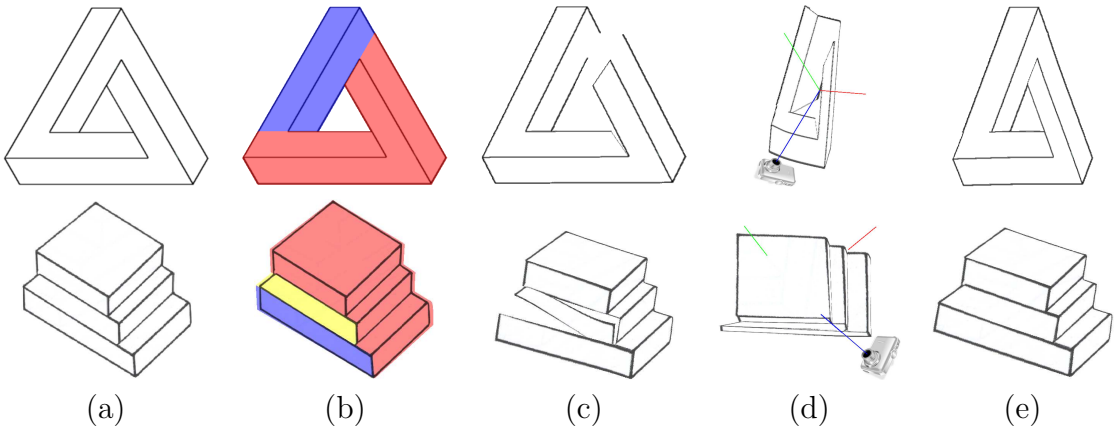
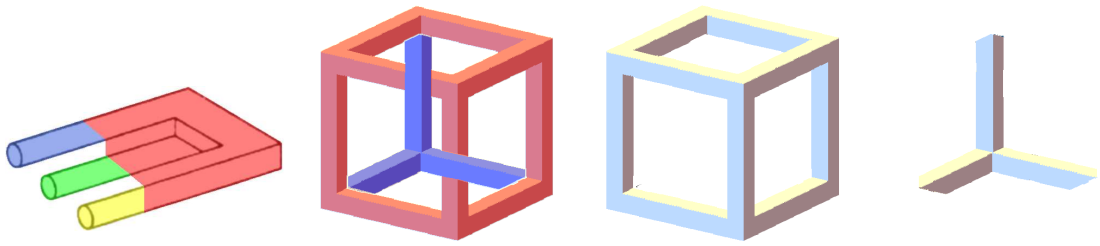


Figure 7.4: Segmenting impossible figures into 3D possible parts. (a) The impossible figure is segmented into parts (as shown in (b)), which are possible in 3D. (c) When the camera viewpoint is changed, the figure starts to collapse. Our system automatically optimizes at interactive rate a *view-dependent* 3D model for rendering the impossible figure at the novel view. Such a model looks distorted, as shown in (d), except at the camera viewpoint specified by the user, as shown in (e).

that straight line connections are maintained at the novel view. We term such constrained non-rigid transform *view-dependent modeling*.

Figure 7.2 illustrates the differences between the possible and impossible figure in terms of modeling and rendering. Note that the former is a standard image formation process, while in the latter (view-dependent modeling) the projected 2D view provides constraints on how the 3D model should be transformed to achieve global structure inconsistency.

Enforcing appropriate constraints in non-rigid transformation is essential in constructing impossible figures. Figure 7.3 illustrates the situations where either rigid (projective camera) transformation or non-rigid transformation dominates the rendering process. Figure 7.3(a) shows the impossible staircase rendered at a novel viewpoint that maximizes the size of the rigid parts. The 3D perception is strong but the figure is no longer impossible in 3D. On the other hand, Figure 7.3(b) shows a novel view generated by free-form warping of the input figure. Though preserving the topology, the figure is curved and does not resemble a novel view derived from the same solid object that produces the original drawing.



trident parts cuboid parts background layer foreground layer

Figure 7.5: Parts segmentation for impossible trident and impossible cuboid. See Figure 7.4 for Penrose triangle and impossible staircase.

7.2 View-Dependent Modeling

Typically, we are only given a single view of an impossible figure as input. In this case, after segmentation, the user constructs the 3D model for each possible part (section 7.2.1). Then, the user marks up 3D constraints in the form of points and lines on the 2D figure. A view-dependent model will be automatically optimized for rendering the impossible figure at the novel view (section 7.2.2).

7.2.1 Segmentation and Modeling of 3D Possible Parts

We provide one simple segmentation strategy for each of the four classes of impossible figures. The goal is to produce a set of locally-possible parts that can be modeled using existing modeling tools such as Maya and 3D Studio Max, or systems like [40, 59, 96]. In some demonstrated examples, a height field rather than a full 3D model was used. The tools used in segmentation and modeling are not the focus of our work.

Referring to Figure 7.4, the Penrose triangle is decomposed into two possible polyhedral parts. The problematic plane is segmented from the impossible staircase, making both the resulting polyhedron and the plane feasible in 3D. To make the 3D model consistent with the original view, for the Penrose triangle, we need curved surfaces but flat normals, while for the impossible staircase, we need a flat surface but curved normals.

Next, referring to Figure 7.5, the problematic corner of the impossible cuboid (the inverted Y-shaped component) is disconnected from the cuboid, making the two independent parts simple 3D models. This corner will be treated as the foreground layer, and

the rest of the cuboid as the background layer. For the impossible trident, the problematic bars are disconnected from the object body, making all segmented parts individually possible in 3D.

It can be noticed that multiple segmentation strategies exist for an impossible figure. Ideally, we seek one which contains at least one region whose size is maximum. This largest region will be considered as the rigid part where rigid transformation will be applied. The rationale behind this strategy is that non-rigid transformation of other parts, which may reduce the effect of 3D solid perception, will thus be minimized as much as possible. The tradeoff between rigid and non-rigid transformation was illustrated in Figure 7.3. While more than one segmentations are feasible, our efficient system makes it easy for the user to experiment different strategies.

7.2.2 Automatic Optimization

Referring to Figure 7.4, the user selects one part as the *reference part* (shown in red), and moves the rigid camera to the desired viewpoint. The selected reference part is rendered as a rigid body in the novel view and no non-rigid deformation is applied to it.

The other parts will undergo non-rigid transformation to optimize a connected 3D model that produces the impossible figure at the desired novel viewpoint. Because a view-dependent model can provide only one view, to render a sequence of novel views, the computation needs to be automatic and efficient.

In this section, we describe our efficient algorithm which connects the 3D segmented parts so as to create the impossible figure at the novel view. We propose to implement such parts connection by applying Thin-Plate Spline (TPS) warping [9] in the 3D domain, subject to the criteria necessary for a 2D impossible figure. We choose TPS because it minimizes the Laplacian (or curvature) of the warping energy, where a natural and smooth deformation can be obtained. This smoothness property also allows graceful degradation when the solution does not exist at certain viewpoints (more detail in the discussion section). Besides, it is well-known that the deformation of TPS is smooth and stable with respect to the change of the input even without explicit enforcing on temporal coherence,

so we have less degree of freedom to consider. Moreover, the TPS model is computationally efficient. Also, note that we warp the 3D parts in the 3D space instead of warping in the projected 2D domain because, as we will see, operating in 3D allows us to readily handle depth ordering.

In the following, we first provide a concise review of TPS, and then define the constraints to the TPS solution to achieve view-dependent modeling.

Review of Thin-Plate Spline Warping

Let $\mathbf{p} = (x, y, z)^T$ and $f(\mathbf{p}) : \mathbb{R}^3 \longmapsto \mathbb{R}^3$ be a mapping function, TPS warping in 3D is defined by:

$$f(\mathbf{p}) = \mathbf{a}_1 + x\mathbf{a}_2 + y\mathbf{a}_3 + z\mathbf{a}_4 + \sum_i^s \mathbf{w}_i U(||\mathbf{p}_i - \mathbf{p}||) \tag{7.1}$$

where $U(r) = -|r|$ in 3D [87], s is the number of input sites which is equal to the number of matching point pairs. $\{\mathbf{a}_j \in \mathbb{R}^3 | j = 1, 2, 3, 4\}$ and $\{\mathbf{w}_i \in \mathbb{R}^3 | i = 1, \cdots, s\}$ are the model parameters to be estimated.

Denote $\mathbf{T} = (\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4, \mathbf{w}_1, \cdots, \mathbf{w}_s)$, which is a $3 \times (s + 4)$ matrix, to be the unknown parameters. Denote $\mathbf{v} = (1, \mathbf{p}^T, U(||\mathbf{p}_1 - \mathbf{p}||), \cdots, U(||\mathbf{p}_s - \mathbf{p}||))^T$, which is a $(s + 4)$ column vector, to be the known input. Eqn. 7.1 can be written as the following matrix form:

$$f(\mathbf{p}) = \mathbf{T}\mathbf{v}. \tag{7.2}$$

Suppose that we have a discrete set of matching samples $\{(\mathbf{p}_i, \mathbf{m}_i)\}$ such that $\mathbf{p}_i \in \mathbb{R}^3 \longrightarrow \mathbf{m}_i \in \mathbb{R}^3$, where $\{\mathbf{p}_i\}$ is the set of input sites and $\{\mathbf{m}_i\}$ is the set of mapping targets, and $\mathbf{v}_i = (1, \mathbf{p}_i^T, U(||\mathbf{p}_1 - \mathbf{p}_i||), \cdots, U(||\mathbf{p}_s - \mathbf{p}_i||))^T$, we can estimate the model parameter $\mathbf{T}$ by solving the following set of linear equations:

$$\mathbf{T} \begin{pmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_s \end{pmatrix} = \begin{pmatrix} \mathbf{m}_1 & \cdots & \mathbf{m}_s \end{pmatrix} . \tag{7.3}$$

Standard TPS considers the null space of the input sites [9], that is, Eqn. 7.3 has to be

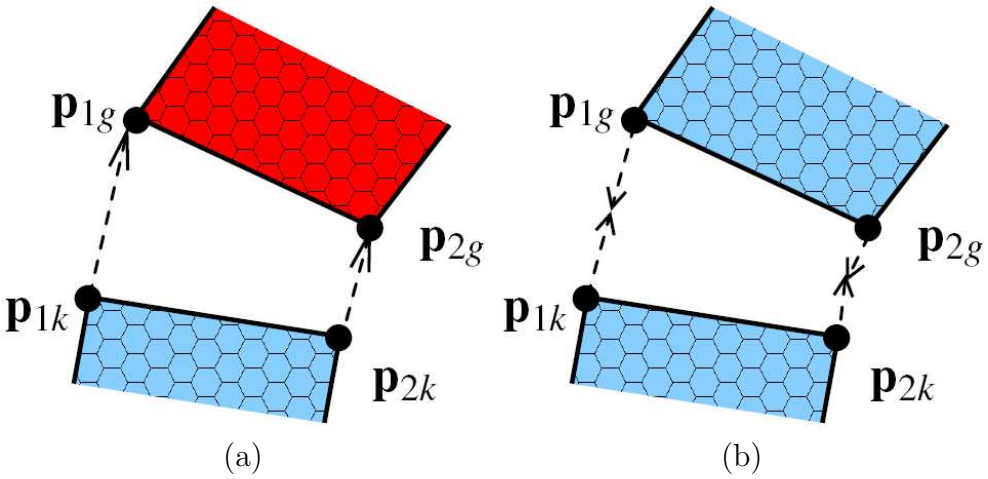


Figure 7.6: *Connection constraint.* (a) case 1: the points in the blue region approach to the points in the red region (the reference part). (b) case 2: the corresponding points are approaching toward each other.

solved subject to the following condition:

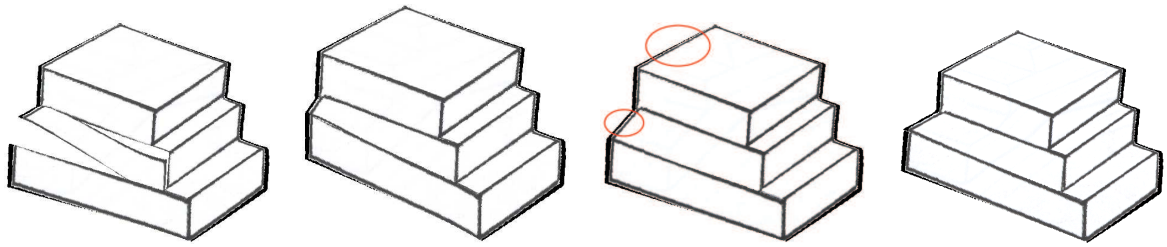
$$\mathbf{T}\begin{pmatrix} \mathbf{O} & \mathbf{p}'_1 & \cdots & \mathbf{p}'_s \end{pmatrix}^T = \mathbf{0} \tag{7.4}$$

where $\mathbf{O}$ is a 4×4 zero matrix and $\mathbf{p}'_i = \begin{pmatrix} 1 \\ \mathbf{p}_i \end{pmatrix}$. Given the matrix forms shown in Eqns 7.3 and 7.4, we are ready to derive the conditions for our view-dependent modeling.

Connection Constraint

The resulting impossible figure must be connected in the rendered 2D view. Without loss of generality, suppose we have a set of n matching point pairs $\{(\mathbf{p}_{ik}, \mathbf{p}_{ig})|i = 1, \cdots, n\}$ that corresponds to parts k and g. After TPS transformation, $\mathbf{p}_{ik}$ and $\mathbf{p}_{ig}$ have to be connected in 3D, which automatically enforces 2D connection in the projected camera view.

Let $\mathbf{T}_k$ and $\mathbf{T}_g$ be the model parameters corresponding to the mapping functions for parts k and g. We have two cases to consider (see Figure 7.6): 1) the connection between the reference part and a deformable part, and 2) the connection between two deformable parts.



(a) (b) (c) (d)

Figure 7.7: Results by enforcing additional constraints. (a) A novel view before optimization. (b) A novel view optimized subject to the connection constraint, where the yellow region deforms severely while the red region does not. (c) A novel view optimized subject to both the connection and collinearity constraints. Compared with the original figure, the two highlighted structures should be parallel. (d) A novel view optimized subject to the connection, collinearity, and parallel constraints.

Case 1 Suppose that part g is the reference part, according to Eqn. 7.3, we have:

$$\mathbf{T}_k \begin{pmatrix} \mathbf{v}_{1k} & \cdots & \mathbf{v}_{nk} \end{pmatrix} = \begin{pmatrix} \mathbf{p}_{1g} & \cdots & \mathbf{p}_{ng} \end{pmatrix} \tag{7.5}$$

Case 2 When no reference part is involved in the connection, we cannot assume either $\mathbf{p}_{ik}$ or $\mathbf{p}_{ig}$ to be the mapping target, as in Case 1. This is because parts k and g will be deformed simultaneously. Instead, we minimize the 3D distance between these two points:

$$\mathbf{T}_k \begin{pmatrix} \mathbf{v}_{1k} & \cdots & \mathbf{v}_{nk} \end{pmatrix} - \mathbf{T}_g \begin{pmatrix} \mathbf{v}_{1g} & \cdots & \mathbf{v}_{ng} \end{pmatrix} = \mathbf{0} \tag{7.6}$$

Similar to the derivation of Eqn. 7.4, we need to consider the null space of these input points:

$$\mathbf{T}_\Lambda \begin{pmatrix} \mathbf{O} & \mathbf{p}'_{1\Lambda} & \cdots & \mathbf{p}'_{s\Lambda} \end{pmatrix}^T = \mathbf{0} \tag{7.7}$$

where $\Lambda \in \{k, g\}$. Eqns 7.5, 7.6 and 7.7 together constitute a set of linear equations that enforces the basic connectivity in the optimized 3D model. Figure 7.7(a) and (b) respectively show the results before and after applying the connection constraint at a novel view.

Collinearity Constraint

Since each part may be deformed by different mapping functions, the resulting deformation can be biased to a certain part, resulting in uneven deformation and unwanted distorted appearance. Specifically, there is no guarantee that after TPS warping the transformed points in the contact area will have the same gradient. For example, in Figure 7.7(b), the corresponding yellow region shown in Figure 7.4(b) undergoes large distortion while the red region undergoes small deformation.

To eliminate such artifact on the resulting 2D figure, we need to minimize the changes of the mapping functions at the matching points. Let $\mathbf{C} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)^T$ be the known 3×4 camera projection matrix corresponding to the novel viewpoint. Then, for each matching point pair $(\mathbf{p}_{ik}, \mathbf{p}_{ig})$, we have:

$$\begin{aligned} \frac{\partial}{\partial x}\mathbf{c}_1^T \begin{pmatrix} f_k(\mathbf{p}_{ik}) - f_g(\mathbf{p}_{ig}) \\ 0 \end{pmatrix} = 0 \\ \frac{\partial}{\partial y}\mathbf{c}_2^T \begin{pmatrix} f_k(\mathbf{p}_{ik}) - f_g(\mathbf{p}_{ig}) \\ 0 \end{pmatrix} = 0 \end{aligned} \tag{7.8}$$

where $f_k(\cdot)$ is the mapping function for part k. This equation explicitly minimizes the difference in the 2D gradient across the contact area of the mapping functions on the screen space. Figure 7.7(c) shows the result after applying the collinearity constraint, where the deformation is not biased to any of the two non-reference parts (colored yellow and blue in Figure 7.4(b)).

Parallel Constraint

Since TPS warping is used, the underlying structure of the parts can be deformed severely. While this is allowed in 3D, we need to constrain the projected 2D structures to protect the shape from apparent distortion while achieving the global structure inconsistency. To this end, we impose the parallel constraint during TPS warping. For example, there is apparent distortion in the figure shown in Figure 7.7(c) when compared with the input:

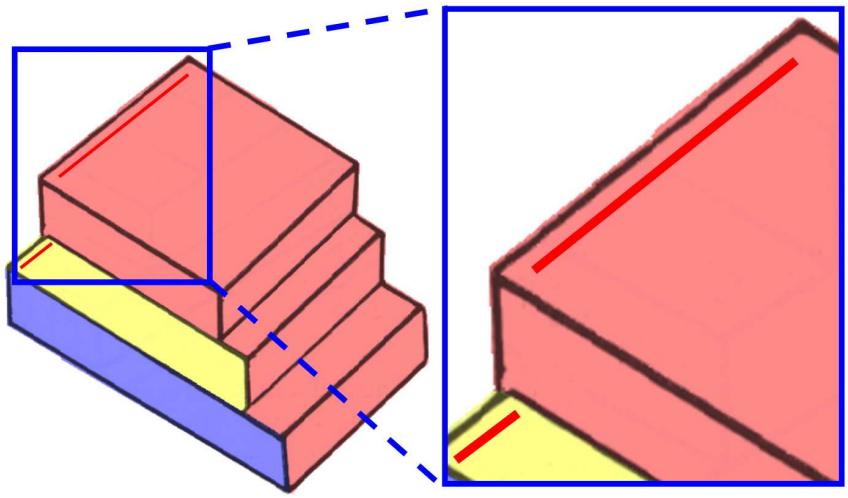


Figure 7.8: *Parallel (line) constraint.* The pair of red lines are corresponding.

the highlighted structures in (c) should be parallel to each other, similar to Figure 7.4(a).

Recall that the selected reference part remains rigid throughout the optimization; otherwise, enforcing parallelism will be elusive in TPS warping when all parts are allowed to deform simultaneously. To specify the parallel line constraint, the user marks up on the 2D view a pair of corresponding lines between the reference part (rigid) and the deformable parts (non-rigid). An example is shown in Figure 7.8. Along the pair of corresponding lines, similarity in 2D gradients will be maximized. Mathematically, we set up the following set of linear equations for achieving this goal:

$$\mathbf{c}_s^T \begin{pmatrix} f_k(\mathbf{l}_a) - f_k(\mathbf{l}_b) \\ 1 \end{pmatrix} = \mathbf{c}_s^T \begin{pmatrix} \mathbf{l}'_a - \mathbf{l}'_b \\ 1 \end{pmatrix} \tag{7.9}$$

where $\mathbf{C} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)^T$ is the 3×4 camera projection matrix, $s \in \{1, 2\}$, $\mathbf{l}_a$ and $\mathbf{l}_b$ are the two endpoints of a line in part k, $\mathbf{l}'_a$ and $\mathbf{l}'_b$ are the two endpoints of the line in the reference region. Figure 7.7(d) shows the final result after applying the parallel constraint.

Optimization

Mathematically, the set of linear equations in Eqns 7.5, 7.6, 7.7, 7.8 and 7.9 can be combined into the following form:

$$\mathbf{Ah} = \mathbf{b} \tag{7.10}$$

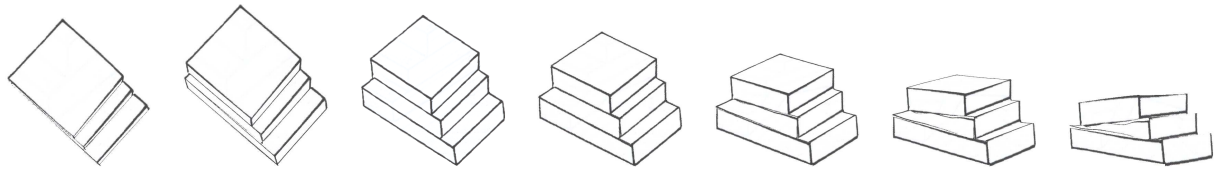


Figure 7.9: Error measurement of sample novel views of the *Impossible staircase*. The errors shown here are calculated by applying Eqn. 7.11, followed by multiplying the result by 1000.

where a *closed-form* solution exists. **A** is the matrix containing the set of coefficients for calculating the Gram matrix.

Directly expanding the linear system expressed by Eqn (7.10), on the other hand, is complicated and unnecessary: in our implementation, we derive the Gauss-Seidel solutions directly from Eqns 7.5–7.9 independently and combine them, which is a common strategy adopted for a simpler and faster implementation requiring less memory storage. To make the solution more stable, all the 3D inputs are normalized in the range of $[-1, +1]$ before computation.

Finally, a remark on depth ordering is necessary here. As mentioned in Section 7.2.2, we may need to handle objects like impossible cuboid. To restrict the depth of a region within some range, for example, the center portion of the inverted Y-shaped component has to be displayed in front of the background layer, we can impose *range constraints* on the z-coordinate (e.g. $z > q$ where q is a constant) in the solver when solving Eqn. 7.10. This can easily be done when a Gauss-Seidel solver (with/without successive over-relaxation) is used. In doing so, the center of the inverted Y-shaped component will always be located in front of the background layer, while the three extreme ends will be connected to the background layer.

Error Measurement

Different from standard TPS, our solution is a least-square solution with an over-constrained system. As a result, for some viewpoints, the optimized solution **h** to Eqn. 7.10 may not maintain the strict equality. This means that some of the constraints may not be ade-

	modeling (FPS)	rendering (FPS)	total
cuboid	0.05 (18.60)	0.02 (44.61)	0.08 (13.13)
penrose	0.07 (15.01)	0.02 (41.08)	0.09 (10.99)
staircase	0.04 (24.76)	0.03 (39.39)	0.07 (15.20)
trident	0.14 (7.32)	0.01 (87.56)	0.15 (6.76)

Table 7.1: Average modeling and rendering computing time (in seconds), measured on a PC (Intel Core 2 Quad CPU Q9400 running at 2.66GHz), with 4GB RAM and Geforce 9800 graphics board. The corresponding frame rates per second (FPS) are also shown.

quately satisfied, which causes the impossible figure to collapse.

In practice, we can quantify the validity of the optimized result by computing the root-mean-square (RMS) error between the optimized 3D model and the constraints encoded in Eqn. 7.10, that is,

$$\sqrt{\frac{||\mathbf{Ah}-\mathbf{b}||^2}{N}} \tag{7.11}$$

where N is number of rows in $\mathbf{A}$, which is equal to the number of constraint equations formed using Eqns 7.5–7.9. As mentioned, it is not necessary to expand matrix $\mathbf{A}$. To calculate the numerator of Eqn. 7.11, we sum up the squared errors of each equation formed using Eqns 7.5–7.9. The RMS errors for some sample novel views of the impossible staircase are shown below the subfigures in Figure 7.9. At extreme viewpoints, we may not be able to maintain the straight-line structure in the result where the impossible figure ceases to exist (indicated by Eqn 7.11). The viewpoints corresponding to large errors can therefore be pruned away and labeled as invalid.

7.2.3 Basic Impossible Figures

Table 7.1 tabulates the running time of our system on modeling and rendering basic impossible figures. Note again that the user marks up constraints only once. The view-dependent models are then automatically generated at the novel views.

Figure 7.4 shows two novel views for the penrose triangle and impossible staircase. For the penrose triangle, the collinearity constraint is instrumental in preserving the structure linearity in the projected figure, because the parts segmentation cuts the two silhouette edges as shown in Figure 7.4. We have demonstrated how the three constraints operate

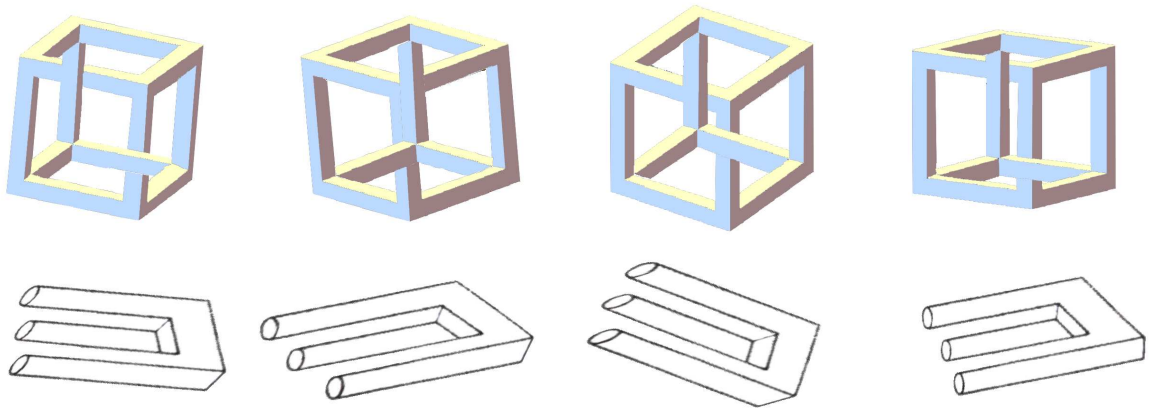


Figure 7.10: Novel views of basic impossible figures.

using the impossible staircase as the running example.

Figure 7.10 shows four novel views for the impossible cuboid and impossible trident. For the impossible cuboid, the foreground layer is optimized with the range constraint mentioned in Section 7.2.2. Similar to the Penrose triangle, the collinearity constraint is essential to the impossible trident in preserving the linear structure during TPS warping, because its parts segmentation cuts across the three bars (Figure 7.5) to produce the corresponding locally-possible parts. In addition, the parallel constraint can constrain the three bars to be parallel in the resulting novel view.

7.2.4 Survey

The human visual perception on impossible figures can be subjective and widely varies. We conducted a user survey to investigate the quality of novel views of impossible figures generated by our system. After an explanation of impossible figures is given, the subjects were first presented with the original input of the four basic impossible figures (Penrose triangle, impossible cuboid, impossible staircase, and impossible trident). The order of showing was randomized across different subjects to avoid bias. If they can articulate why the input figure is impossible in 3D, the experiment would proceed to the next phase, where they were presented four novel views of the respective four basic impossible figures generated by our system (some of them are shown in Figure 7.10). Again, the order of showing was randomized. The users were asked to label them either as "impossible,"

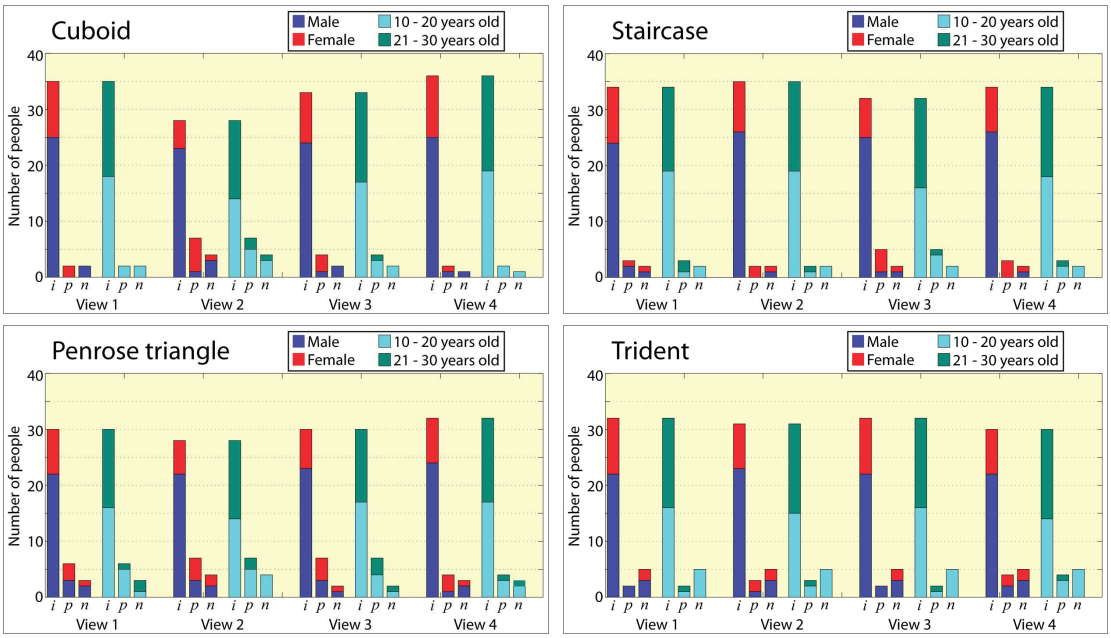


Figure 7.11: *Survey results.* For each basic impossible figure, five views were presented: the original view (Figure 6.1) and four novel views derived from the original view using our system. Users were asked to label each image as either *impossible* (i), *possible* (p) or *not sure* (n).

"possible," or "not sure" within a time limit of 20 seconds.

A total of 47 persons in different age groups and genders participated in our survey; 39 persons proceeded to the next phase. We found that over 76% of users label our figures as "impossible." A vote of "not sure" is cast when the user cannot decide within the time limit. In general, we found that if the user was able to articulate why the original figure is impossible in 3D, they would be able to label the novel views as impossible as well.

Among the four cases, we require the least effort for participants to label the impossible cuboid, which also scores the highest on being "impossible" among all the figures. On the other hand, a participant requires on average the most time in labeling the Penrose triangle, which also scores the lowest on being "impossible." Some users commented after the survey that the figures are "fantastic" and "interesting to view"; some are intrigued by the confused 3D perception (because they were actually seeing an image of a 3D model, albeit a view-dependent one optimized by our system), while others even proposed how to use overlay photography to create an impossible cuboid figure.

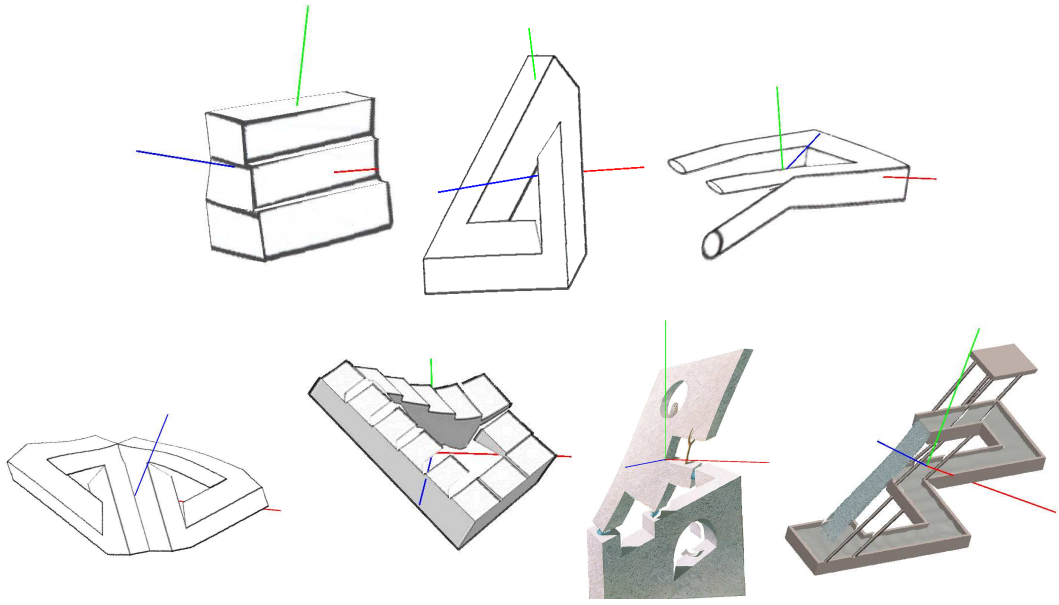


Figure 7.12: *View-dependent models.* While they look distorted when viewed at other camera viewpoints, the generated 2D impossible figures consist of straight lines when such models are viewed at the specified viewpoint.

7.3 Discussion

We demonstrated in Figure 7.9 that we can use our efficient system to stretch the rendering limit of the impossible staircase, by modeling and rendering the figure at viewpoints far away from the input view, which was quite difficult previously without the tedious modeling step.

When the figure starts to collapse, the degradation is observed to be quite graceful. This is due to the use of thin-plate spline in our constrained optimization. We have the same observation for other examples as well. The RMS error gives a quantitative measure on the distance between the result and the model, or in other words, how bad the failure cases are.

Similar to how 3D modeling artists build a scene to render impossible figure, the optimized view-dependent model can in fact be severely deformed, although at the user-specified novel camera view where the model is optimized, we observe an impossible figure with straight connections. Figure 7.12 shows the optimized 3D models viewed at other camera viewpoints.

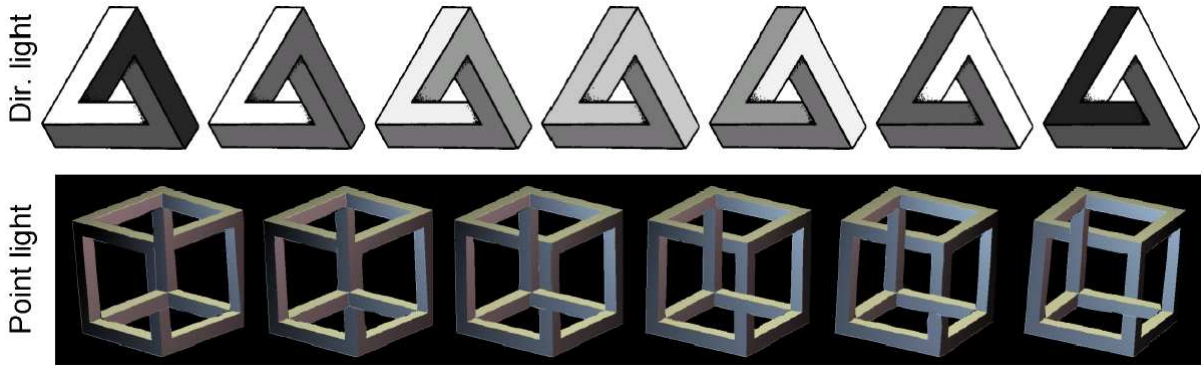


Figure 7.13: Top: directional lighting. Bottom: point source lighting with variable viewpoint changes.

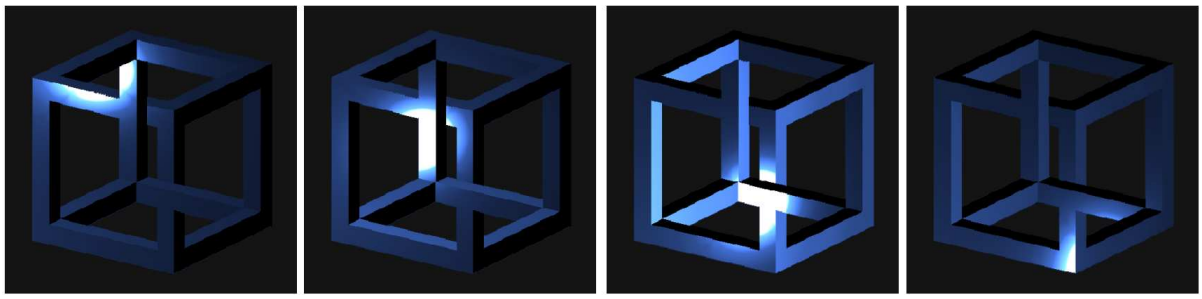


Figure 7.14: Dressing the impossible cuboid with an isotropic BRDF.

7.4 Results

The existing approaches for constructing 3D models for rendering impossible figures are expensive and provide limited views of the figure. Using our automatic modeling and rendering system, for the first time we are able to produce complex illumination effects on a large variety of impossible figures while they are being animated.

7.4.1 How to Shade an 'Impossible Object'?

Normal is a core component in shading. However, the surfaces optimized by our system are deformed. If we shade the normals obtained directly from the deformed surface produced after the TPS-optimization, we will perceive an unnatural surface. To tackle this problem, the normals used for shading are sampled from the original 3D model. These normals are rotated by the same amount that is applied to the rigid reference part. In doing so, the shaded surface will still look natural even though the model has been severely deformed. Figure 7.13 shows a series of renderings depicting a directional (and point) light source

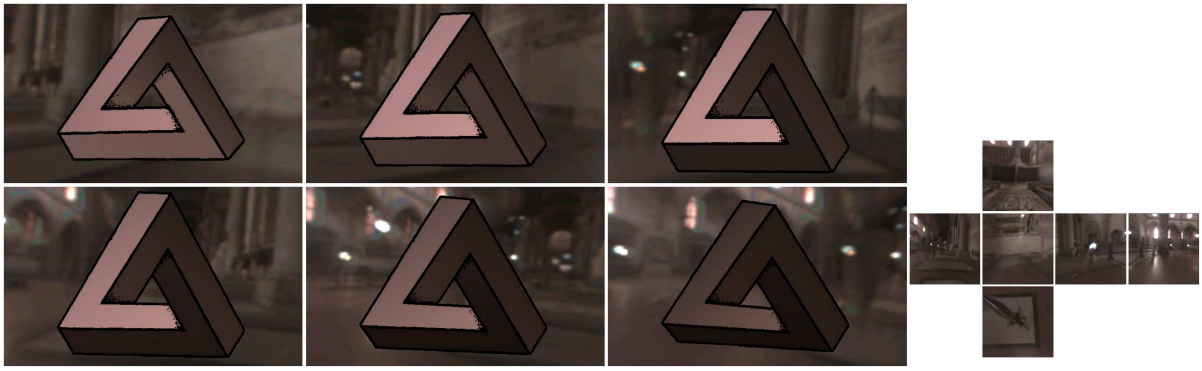


Figure 7.15: Viewing the impossible figure under a rotating distant environment.

being moved from left to right across the Penrose triangle and impossible cuboid.

7.4.2 BRDF

Using the optimized model, isotropic BRDF data (Matusik et al. [54]) can be arranged on impossible figures. More specifically, we first compute the light and view vectors for each pixel fragment on the impossible figure, and then compute the angles between: 1) the light vector and pixel's normal; 2) the view vector and pixel's normal; and 3) the projected light and view vectors (after projection onto the tangent plane of the pixel). Thus, we can look up the reflectance value in the isotropic BRDF data and shade each pixel accordingly. See Figure 7.14 for a rendering example with the specular blue phenolic BRDF covering the impossible cuboid. In the image sequence shown, we move a search light around the impossible cuboid.

7.4.3 Environment Lighting

In addition, we can also put an impossible figure and re-render it under a distant environment. Specifically, we employ the importance sampling approach [2] by approximating the illumination of a distant environment using a limited number of directional lights. Efficient sampling algorithms have been proposed, and in our implementation, around 200 lights are extracted. Then, for each sample (directional light), we render the impossible figure by local illumination, and produce the final result by summing up the rendering results from multiple passes of such local illumination. Figure 7.15 presents the com-

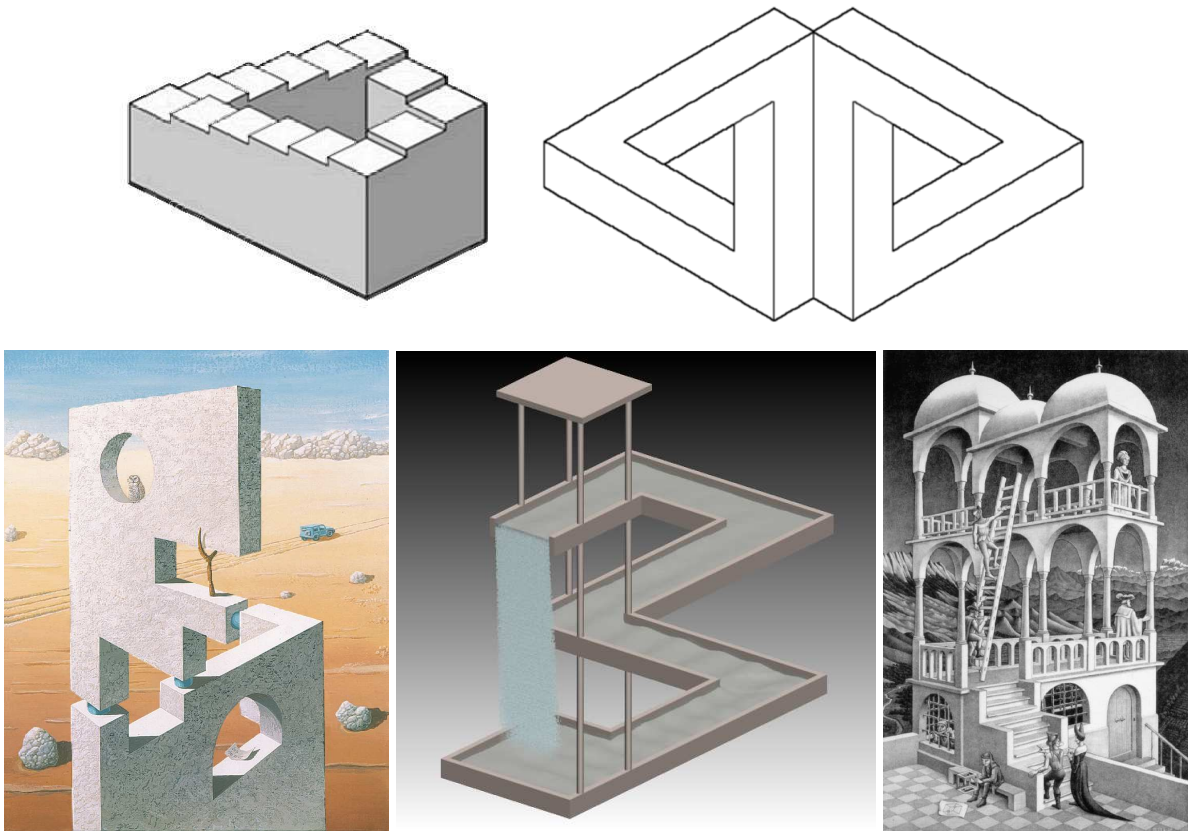


Figure 7.16: Top: Ascending and Descending (left) and Double Penrose Triangles (right). Bottom: Construction (left), Waterfall (middle), and M. C. Escher's original Belvedere (right).

posed rendering results of the Penrose triangle under the GRACE environment. Here, we gradually change the viewpoint on the Penrose triangle while simultaneously rotating the distant environment.

7.4.4 Rendering Artworks of Impossible Figures

Figures 7.17–7.21 show the results of rendering different impossible figures at a novel view, the corresponding input figures are shown in Figure 7.16.

Ascending and Descending Like the Penrose triangle, this figure belongs to the class of *depth contradiction*. Figure 7.17 shows the parts segmentation, constraint markups, and novel views. The red part is chosen as the reference, which undergoes rigid transform under normal camera projection (which can be perspective or orthographic). The connection constraints maintain the connectivity of the two parts at the novel view. The

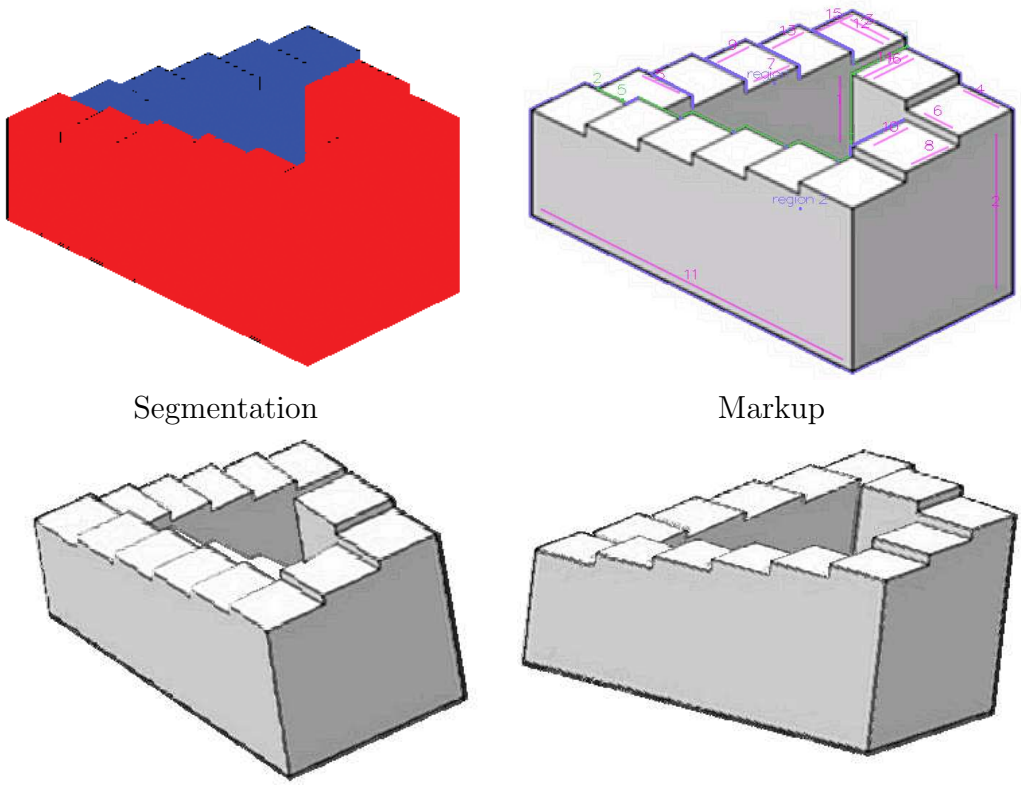


Figure 7.17: *Ascending and Descending.*

parallel constraints reduce the distortion effect on the blue part, which undergoes structure deformation for connecting to the transformed red part.

Double Penrose Triangles Using our system we can compose and render new impossible figures by blending existing impossible parts, where the view-dependent geometry will be optimized in exactly the same fashion, that is, by connecting the input parts in 3D to produce an impossible figure at the chosen view. In Figure 7.18, we juxtaposed two Penrose triangles. In the optimization process, the red part is the reference part. Constraints are specified to bind the two triangles together while all straight and parallel connections are maintained among the pertinent parts. This example is difficult because of the severe structural conflict within and between the two Penrose triangles.

Construction Similar to the Penrose triangle, *Construction* was constructed using two locally-possible parts, and each part was modeled as a height-field. Figure 7.19 shows the

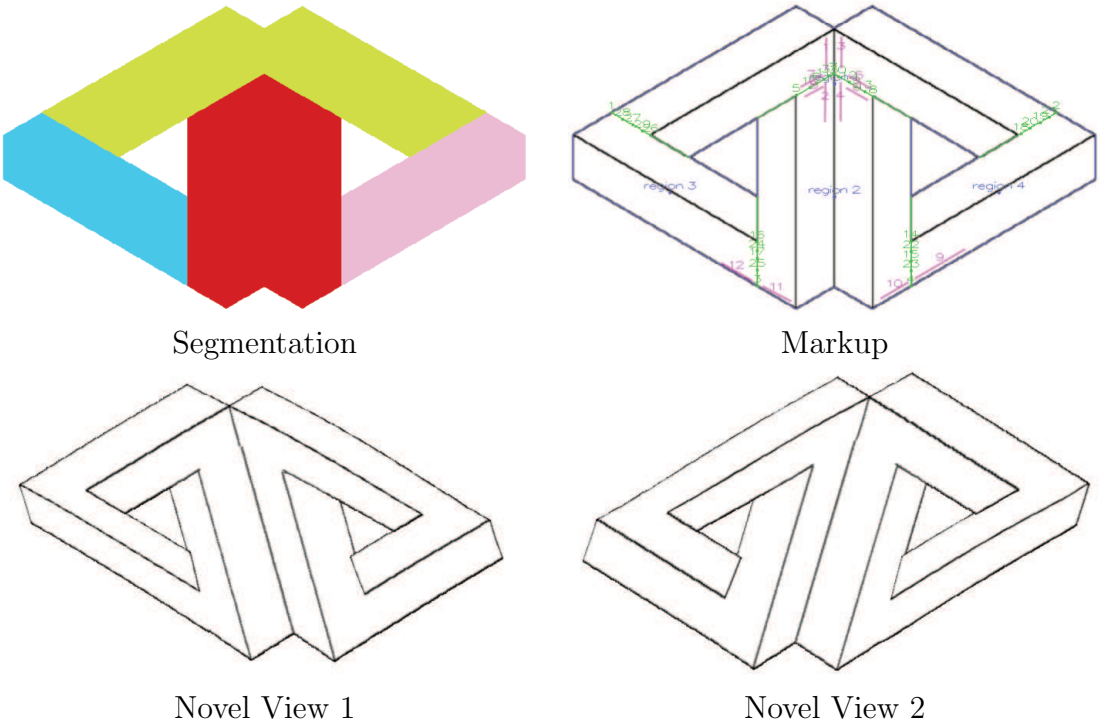


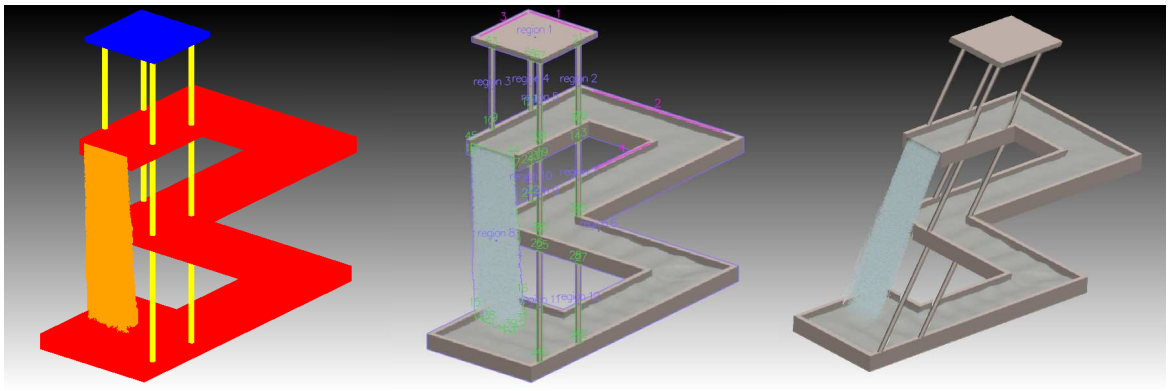
Figure 7.18: *Double Penrose Triangles.*

constraint markup, parts segmentation, and novel views. The lower part (the red part) is chosen as the reference part which undergoes rigid body transformation. Non-rigid transformation is applied to the upper part in the constrained-TPS optimization. The collinearity constraints protect the overall shape from severe distortion, while the parallel constraints enforce the left and right sides of both parts to be straight and parallel in the rendered novel views. Note that the figure only starts to collapse at novel views far from the input view. At these viewpoints, the rate of collapse is accelerated by the use height-field, which is not a full 3D representation.

Waterfall Similar to *Ascending and Descending*, this figure also belongs to the class of depth contradiction. This is one of the most difficult examples where the two "penrose triangles" are shackled together with multiple pillars. Figure 7.20 shows the parts segmentation, constraints, and novel views. The red part is chosen as the reference. The connection constraints are used to enforce the necessary connectivity to hold the figure as one fully-connected component at the novel view. The parallel constraints are used to preserve the overall shape of the top roof. Note that while the novel views still preserve

Segmentation Markup Novel View

Figure 7.19: *Construction*. A very difficult example because of the severe structural inconsistency inherent in the impossible figure.



Segmentation Markup Novel View

Figure 7.20: *Waterfall*, another very difficult example.

all straight connections, the waterfall model looks skewed at the novel view due to the severe structural inconsistency inherent in this impossible figure.

Belvedere This impossible figure belongs to the class of *depth interposition*. The height-fields of the possible parts, namely, the upper level and the lower level, are available. Connection constraints are used to enforce the pillars to be connected to both the upper and lower levels. Similar to *Construction*, parallel and collinearity constraints are specified to maintain the overall shape of the entire architecture. Figure 7.21 shows some novel views generated. Notice that we inpainted the background layer to further enhance the

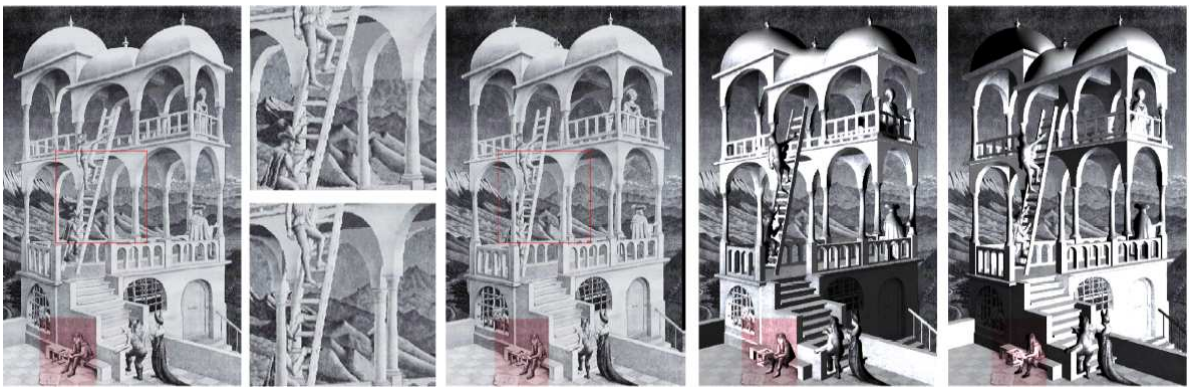


Figure 7.21: Novel views of *Belvedere*. Using the optimized view-dependent model, we can relight *Belvedere* under different lighting configurations.

rendering effect of this masterpiece by M. C. Escher.

Possible and Impossible Finally, we blend our impossible object (that is, the optimized view-dependent model) into a geometrically-possible 3D scene to create special effects. Figure 7.22 (top and middle row) shows several snapshots of the animation sequence of such scenes. The novel viewpoint are specified using a rigid camera, where standard perspective transformation is applied to the possible objects and also to the rigid part of view dependent model. Constrained-TPS optimization is applied to the non-rigid parts of the model as described in the thesis. Using previous 3D approaches it is difficult to produce these animations, because constructing per-frame 3D model was done by hand or else it required expensive computation.

Note in particular the bottom row of Figure 7.22 where we show the zoom-in views of a possible object (a ball) bouncing on an impossible object (*Ascending and Descending*). This involves collision detection and response handling. The problem is non-trivial when impossible objects are involved: a possible and an impossible object cannot interact directly in the 3D space because the latter is highly deformed. While this is future work to pursue, here we adopt a simple approach to produce this visual effect, by rendering each frame in two layers: the bouncing ball and the remainder of the scene.

7.5 Summary

Impossible figures have long been used in applications such as computer games, non-photorealistic rendering, and image synthesis. We investigate another practical approach for modeling and rendering impossible figures. Our approach is motivated by how a 3D modeling artist builds view-dependent models for rendering impossible figures. Modeling and rendering of impossible figures are coupled. This led to our *view-dependent modeling* approach which connects possible 3D parts for rendering novel views of impossible figure. Our mathematical formulation shows that a closed-form solution exists for view-dependent modeling, thus allowing us to implement an efficient system to model and render novel views of impossible figures at interactive speed. This formulation also provides a numerical means for pruning away invalid viewpoints where the impossible figure ceases to exist. Once optimized, the 3D model can be used to create compelling visual effects previously restricted to possible 3D graphics models.

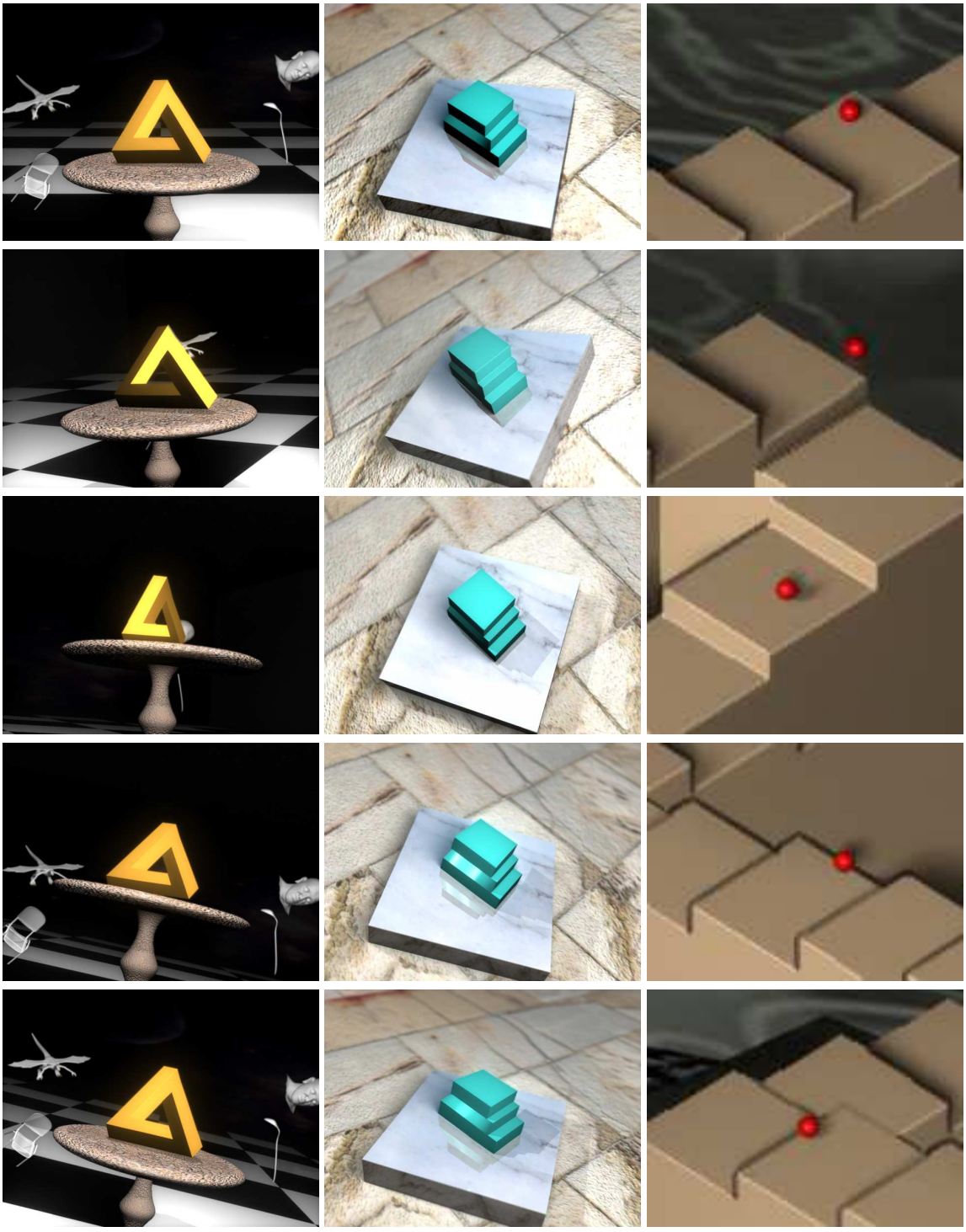


Figure 7.22: Modeling and animating 3D scenes with possible and impossible objects.

Chapter 8

Conclusion

This thesis proposes an HCI approach to address two difficult problems in vision and graphics: modeling and rendering the invisible (transparent) objects and impossible figures.

Human-computer interaction (HCI) has gaining more attention in recent years. With the successful development of different automation techniques in computer vision and modeling methods in computer graphics, researchers are putting more attention to bridging the two together. Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.

This thesis has made a significant pass on this important area in computer vision and graphics. Our proposed approaches do not advocate to manual processing, but an easy and interactive approach where the user only needs to provide a few simple hints for the computer algorithm to perform automatic processing.

First, a novel Expectation-Maximization (EM) framework over Markov-Random Field (MRF) was formulated to show how simple user interaction can tremendously help in solving a difficult problem of decomposing an image into overlapping transparent layers. By supplying few input scribbles from the user to collect different color statistics, we can extract the layers to perform various tasks such as local colorization, natural image matting and shadow matting.

Then, by making use of the remarkable human visual system to detect transpar-

ent object and the corresponding tolerance to inaccuracies in refractive phenomena, we have derived a new image-based matting model, termed the attenuation-refraction matte (ARM), that encodes the visual effects associated with transparent and refractive objects. We described how the ARM can be extracted from a single image with simple user markup, and how to use the ARM to paste the object into a new background. We show that plausible refractive deformation suffices in producing visually compelling results. We believe it is the first work to allow photo-editing of transparent and refractive objects in cases where only a single image is available which is traditionally impossible.

Finally, we introduce a novel approach to model and render impossible figures which have long been used in computer graphics applications such as computer games, non-photorealistic rendering, and image synthesis. Inspired by 2D and 3D modeling artists we have derived two approaches, one is image-based and the other is view-dependent modeling to render an input impossible figure at desired novel views. In both approaches, the user supplies constraints to the computer algorithms rather than tediously building 3D models or manually painting the images. We also created compelling visual effects using our system.

In the future, we are planning to investigate difficult problems which may be better solved using a human-computer interaction approach in vision and graphics. One specific problem is video editing involving transparent objects. How can we cut from a video sequence of a transparent object while maintaining temporal coherence in its light transport properties? This is one of the major challenges. Another future work is to investigate how the human visual system can help in the accurate 3D modeling of a transparent object from single images, rather than recovering plausible appearance-based models for graphical rendering purposes.

Bibliography

[1] S. Agarwal, S. P. Mallick, D. J. Kriegman, and S. Belongie. On refractive optical flow. In *ECCV (2)*, pages 483–494, 2004.

[2] S. Agarwal, R. Ramamoorthi, S. Belongie, and H. W. Jensen. Structured importance sampling of environment maps. *ACM Trans. Graph. (SIGGRAPH 2003)*, 22(3):605–612, 2003.

[3] V. Alexeev. Impossible world. http://im-possible.info/english/, 2001–8.

[4] M. B. Ezra and S.K. Nayar. What does motion reveal about transparency? In *ICCV03*, pages 1025–1032, 2003.

[5] H.G. Barrow and J.M. Tenenbaum. Recovering intrinsic scene characteristics from images. In *CVS78*, pages 3–26, 1978.

[6] T. Beier and S. Neely. Feature-based image metamorphosis. In *SIGGRAPH '92*, pages 35–42, 1992.

[7] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 1992.

[8] J. Bilmes. A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical Report ICSI-TR-97-021, ICSI, 1997.

[9] F.L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *PAMI*, 11(6):567–585, June 1989.

[10] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen. Unstructured lumigraph rendering. In *SIGGRAPH '01*, pages 425–432, 2001.

[11] J.-X. Chai, S.-C. Chan, H.-Y. Shum, and X. Tong. Plenoptic sampling. In *SIGGRAPH 2000*, pages 307–318, 2000.

[12] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski. A bayesian approach to digital matting. *CVPR'01*, pages 264–271, 2001.

[13] Y.-Y. Chuang, D. B. Goldman, B. Curless, D. H. Salesin, and R. Szeliski. Shadow matting and compositing. *ACM Trans. Graph.*, 22(3):494–500, 2003.

[14] Y.-Y. Chuang, D.B. Goldman, B. Curless, D. H. Salesin, and R. Szeliski. Shadow matting and compositing. *ACM Trans. Graph.*, 22(3):494–500, 2003.

[15] Y.-Y. Chuang, D. E. Zongker, J. Hindorff, B. Curless, D. H. Salesin, and R. Szeliski. Environment matting extensions: towards higher accuracy and real-time capture. In *SIGGRAPH '00*, pages 121–130, 2000.

[16] G. Elber. Escher for real. http://www.cs.technion.ac.il/ gershon/escherforreal, 2002.

[17] B. Ernst. *Adventures with Impossible Figures*. Tarquin, Stradbroke, England, 1987.

[18] H. Farid and E.H. Adelson. Separating reflections from images by use of independent component analysis. 16(9):2136–2145, 1999.

[19] G.D. Finlayson. Color constancy in diagonal chromaticity space. In *ICCV95*, pages 218–223, 1995.

[20] G.D. Finlayson, M.S. Drew, and B.V. Funt. Diagonal transforms suffice for color constancy. In *ICCV93*, pages 164–171, 1993.

[21] G.D. Finlayson, M.S. Drew, and B.V. Funt. Spectral sharpening: Sensor transformations for improved color constancy. *JOSA-A*, 11(5):1553–1563, May 1994.

[22] G.D. Finlayson, M.S. Drew, and C. Lu. Intrinsic images by entropy minimization. In *ECCV04*, pages Vol III: 582–595, 2004.

[23] G.D. Finlayson, C. Fredembach, and M.S. Drew. Detecting illumination in images. In *ICCV07*, pages 1–8, 2007.

[24] G.D. Finlayson, S.D. Hordley, and M. S. Drew. Removing shadows from images. *Proceedings of ECCV'02, Vol. IV, 823-836*, 2002.

[25] G.D. Finlayson, S.D. Hordley, and M.S. Drew. Removing shadows from images using retinex. In *IS&T/SID Tenth Color Imaging Conference*, pages 73–79, 2002.

[26] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practices (2nd edition in C).* Addison-Wesley., 1995.

[27] D.A. Forsyth. A novel approach to color constancy. *IJCV*, 5(1):5–36, August 1990.

[28] R.T. Frankot and R. Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 10(4):439–451, 1988.

[29] C. Fredembach and G.D. Finlayson. Hamiltonian path based shadow removal. In *BMVC05*, 2005.

[30] C. Fredembach and G.D. Finlayson. Simple shadow removal. In *ICPR06*, pages I: 832–835, 2006.

[31] W.T. Freeman and D.H. Brainard. Bayesian color constancy. *JOSA-A*, 14(7):1393–1411, July 1997.

[32] B.V. Funt, K. Barnard, and L. Martin. Is machine colour constancy good enough? In *ECCV98*, page I: 445, 1998.

[33] K. Gai, Z.W. Shi, and C.S. Zhang. Blindly separating mixtures of multiple layers with spatial shifts. In *CVPR08*, pages 1–8, 2008.

[34] D. Gutierrez, J. Lopez-Moreno, J. Fandos, F. Seron, M. P. Sanchez, and E. Reinhard. Depicting procedural caustics in single images. *ACM Trans. Graph.*, 27(5), 2008.

[35] E. Hecht. *Optics (2nd ed.)*. Addison Wesley., 1987.

[36] A. Heyden. On the consistency of line-drawings, obtained by projections of piecewise planar objects. *J. Math. Imaging and Vis.*, 6(4):393–412, 1996.

[37] D. A. Huffman. Impossible objects as nonsense sentences. *Machine Intelligence*, 6:295–323, 1971.

[38] M. Irani and S. Peleg. Image sequence enhancement using multiple motions analysis. In *CVPR92*, pages 216–221, 1992.

[39] J. Jia, J. Sun, C.-K. Tang, and H.-Y. Shum. Drag-and-drop pasting. *ACM Trans. Graph.*, 25(3):631–637, 2006.

[40] T. Ju, Q.-Y. Zhou, and S.-M. Hu. Editing the topology of 3d models by sketching. *ACM Trans. Graph.*, 26(3):42, 2007.

[41] S. B. Kang. A survey of image-based rendering techniques. Technical report, Digital Equipment Corporation, Cambridge Research Lab, Aug. 1997. Tech. Rep. CRL 97/4.

[42] E. A. Khan, E. Reinhard, R. W. Fleming, and H. H. Bulthoff. Image-based material editing. *ACM Trans. Graph.*, 25(3):654–663, 2006.

[43] C. W. Khoh and P. Kovesi. Animating impossible objects. http://www.csse.uwa.edu.au/ pk/impossible/impossible.html, 1999.

[44] J.J. Koenderink, H. C. Longuet-Higgins, W. Triggs, A. Fitzgibbon, D. Foster, and A. Johnston. Pictorial relief. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 356(1740):1071–1086, 1998.

[45] E. Land and J. McCann. Lightness and the retinex theory. *J. Opt. Soc. Am*, pages 1–11, 1971.

[46] A. Levin, D. Lischinski, and Y. Weiss. A closed form solution to natural image matting. In *CVPR06*, pages I: 61–68, 2006.

[47] A. Levin, A. Rav-Acha, and D. Lischinski. Spectral matting. In *CVPR*, 2007.

[48] A. Levin and Y. Weiss. User assisted separation of reflections from a single image using a sparsity prior. In *ECCV04*, volume I, pages 602–613, 2004.

[49] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. *ACM Trans. Graph.*, 23(3):303–308, 2004.

[50] A. Lipson. Andrew lipson's LEGO page. http://www.andrewlipson.com/lego.htm.

[51] F. Liu and M. Gleicher. Texture-consistent shadow removal. In *ECCV (4)*, pages 437–450, 2008.

[52] L.T. Maloney. Evaluation of linear models of surface spectral reflectance with small numbers of parameters. *JOSA-A*, 3(10):1673–1683, October 1986.

[53] Y. Matsushita, S. Lin, S.B. Kang, and H.-Y. Shum. Estimating intrinsic images from image sequences with biased illumination. In *ECCV04*, pages Vol II: 274–286, 2004.

[54] W. Matusik, H. Pfister, M. Brand, and L. McMillan. A data-driven reflectance model. *ACM Trans. Graph. (SIGGRAPH 2003)*, 22(3):759–769, 2003.

[55] W. Matusik, H. Pfister, R. Ziegler, A. Ngan, and L. McMillan. Acquisition and rendering of transparent and refractive objects. In *Eurographics Workshop on Rendering*, 2002.

[56] M.C. Escher Foundation. The official M.C.Escher website. http://www.mcescher.com.

[57] D. Miyazaki and K. Ikeuchi. Shape estimation of transparent objects by using inverse polarization ray tracing. *PAMI*, 29(11):2018–2030, November 2007.

[58] N.J.W. Morris and K.N. Kutulakos. Reconstructing the surface of inhomogeneous transparent scenes by scatter trace photography. In *ICCV07*, 2007.

[59] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa. Fibermesh: designing freeform surfaces with 3d curves. *ACM Trans. Graph.*, 26(3):41, 2007.

[60] S. Owada and J. Fujiki. Dynafusion: A modeling system for interactive impossible objects. In *Proc. of Non-Photorealistic Animation and Rendering (NPAR)*, pages 65–68, 2008.

[61] P. Peers and P. Dutre. Wavelet environment matting. In *14th Eurographics Workshop on Rendering*, pages 157–166, 2003.

[62] L. S. Penrose and R. Penrose. Impossible objects: A special type of illusion. *British J. of Psychology*, 49:31–33, 1958.

[63] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3):313–318, 2003.

[64] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE CG&A*, 21:34–41, 2001.

[65] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.

[66] B. Sarel and M. Irani. Separating transparent layers of repetitive dynamic behaviors. In *ICCV05*, pages I: 26–32, 2005.

[67] G. Savransky, D. Dimerman, and C. Gotsman. Modeling and rendering Escher-like impossible scenes. *Computer Graphics Forum*, 18(2):173–179, 1999.

[68] D. Schattschneider and M. Emmer, editors. *M.C.Escher's Legacy: A Centennial Celebration*. Springer, 2003.

[69] Y.Y. Schechner, N. Kiryati, and R. Basri. Separation of transparent layers using focus. *IJCV*, 39(1):25–39, August 2000.

[70] N. Sebe, M. S. Lew, and T. S. Huang. The state-of-the-art in human-computer interaction. In *ECCV Workshop on HCI*, pages 1–6, 2004.

[71] J. Shade, S. Gortler, L.-W. He, and R. Szeliski. Layered depth images. In *SIGGRAPH '98*, pages 231–242, July 1998.

[72] Y. Shechner, J. Shamir, and N. Kiryati. Polarization-based decorrelation of transparent layers: The inclination angle of an invisible surface. In *ICCV99*, pages 814–819, 1999.

[73] H.-Y. Shum, S.-C. Chan, and S. B. Kang. *Image-Based Rendering*. Springer, 2007.

[74] D. E. Simanek. The principles of artistic illusions – adding depth to illusions. http://www.lhup.edu/ dsimanek/3d/illus2.htm, 1996.

[75] D. Singaraju and R. Vidal. Interactive image matting for multiple layers. In *CVPR08*, pages 1–7, 2008.

[76] K. Sugihara. Three-dimensional realization of anomalous pictures–an application of picture interpretation theory to toy design. *Pattern Recognition*, 30(7):1061 – 1067, 1997.

[77] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum. Poisson matting. *ACM Trans. Graph.*, 23(3):315–321, 2004.

[78] J. Sun, L. Yuan, J. Jia, and H.-Y. Shum. Image completion with structure propagation. *ACM Trans. Graph.*, 24(3):861–868, 2005.

[79] R. Szeliksi, S. Avidan, and P. Anandan. Layer extraction from multiple images containing reflections and transparency. In *CVPR00*, volume 1, pages 246–253, 2000.

[80] Y.-W. Tai, J. Jia, and C.-K. Tang. Soft color segmentation and its applications. *PAMI*, 2007.

[81] M.F. Tappen, W.T. Freeman, and E.H. Adelson. Recovering intrinsic images from a single image. In *NIPS*, pages 1343–1350, 2002.

[82] J.-P. Thirion. Image matching as a diffusion process: an analogy with maxwell's demons. *Medical Image Analysis*, 2(3):243–260, 1998.

[83] Y. Tsin, R.T. Collins, V. Ramesh, and T. Kanade. Bayesian color constancy for outdoor object recognition. In *CVPR01*, pages I:1132–1139, 2001.

[84] M. Tsukada and Y. Ohta. An approach to color constancy using multiple images. In *ICCV90*, pages 385–389, 1990.

[85] S. Tsuruno. The animation of M.C. Escher's "Belvedere". In *ACM SIGGRAPH 97 Visual Proceeding*, page 237, 1997. Presented at Siggraph Electronic Theater 1997.

[86] D. Uribe. A set of impossible tiles. http://impossible.info/english/articles/tiles/tiles.html.

[87] G. Wahba. *Spline models for observational data*. SIAM, 1990.

[88] J. Wang and M. F. Cohen. Optimized color sampling for robust matting. In *CVPR*, 2007.

[89] J. Wang and M. F. Cohen. *Image and Video Matting*. Now Publishers Inc., Hanover, MA, USA, 2008.

[90] J. Wang and M.F. Cohen. An iterative optimization approach for unified image segmentation and matting. In *ICCV05*, pages II: 936–943, 2005.

[91] Y. Weiss. Deriving intrinsic images from image sequences. In *ICCV01*, pages II: 68–75, 2001.

[92] T. Welsh, M. Ashikhmin, and K. Mueller. Transferring color to greyscale images. *ACM Trans. Graph.*, 21:277–280, 2002.

[93] Y. Wexler, A. Fitzgibbon, and A. Zisserman. Image-based environment matting. In *EuroGraphics Workshop on Rendering*, pages 289–299, 2002.

[94] T.-P. Wu, P. C.-W. Fu, S.-K. Yeung, J. Jia, and C.-K. Tang. Modeling and rendering of impossible figures. *ACM Transactions on Graphics (TOG). – accepted with major revision.*

[95] T.-P. Wu and C.-K. Tang. A bayesian approach for shadow extraction from a single image. In *ICCV05*, pages I: 480–487, 2005.

[96] T.-P. Wu, C.-K. Tang, M. S. Brown, and H.-Y. Shum. Shapepalettes: interactive normal transfer via sketching. *ACM Trans. Graph. (SIGGRAPH 2007)*, 26(3):44, 2007.

[97] T.-P. Wu, C.-K. Tang, M.S. Brown, and H.-Y. Shum. Natural shadow matting. *ACM Trans. Graph.*, 26(2), 2007.

[98] T.-P. Wu, C.-K. Tang, M.S. Brown, and H.-Y. Shum. Natural shadow matting. *ACM Trans. Graph.*, 26(2), 2007.

[99] T.-P. Wu and C.K. Tang. Separating specular, diffuse, and subsurface scattering reflectances from photometric images. In *ECCV04*, pages Vol II: 419–433, 2004.

[100] T.-P. Wu and C.K. Tang. Separating subsurface scattering from photometric image. In *ICPR06*, pages II: 207–210, 2006.

[101] S.-K. Yeung and P. Shi. Stochastic inverse consistency in medical image registration. In *MICCAI (2)*, pages 188–196, 2005.

[102] S.-K. Yeung and P. Shi. Stochastic framework for symmetric affine matching between point sets. In *ICPR (3)*, pages 790–793, 2006.

[103] S.-K. Yeung, C.-K. Tang, M. S. Brown, and S. B. Kang. Matting and compositing of transparent and refractive objects. *ACM Transactions on Graphics (TOG). – accepted with major revision.*

[104] S.-K. Yeung, T.-P. Wu, and C.-K. Tang. Extracting smooth and transparent layers from a single image. In *CVPR*, 2008.

[105] S.-K. Yeung, C.-K. Tang, P. Shi, J. P. W. Pluim, M. A. Viergever, A. C. S. Chung, and H. C. Shen. Enforcing stochastic inverse consistency in non-rigid image registration and matching. In *CVPR*, 2008.

[106] B. Zitová and J. Flusser. Image registration methods: a survey. *Image Vision Comput.*, 21(11):977–1000, 2003.

[107] D. E. Zongker, D. M. Werner, B. Curless, and D. H. Salesin. Environment matting and compositing. In *SIGGRAPH '99*, pages 205–214, 1999.