UNIVERSITE NICE SOPHIA ANTIPOLIS

DOCTORAL SCHOOL STIC
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

Mémoire d'Habilitation à Diriger des Recherches

Defended by
Adrien BOUSSEAU

Depicting shape, materials and lighting: observation, formulation and implementation of artistic principles

Inria Sophia Antipolis
defended on September 3rd, 2015

Jury :

Reviewers :	Kavita BALA	-	Professor, Cornell University
	Marie-Paule CANI	-	Professor, Institut Polytechnique de Grenoble
	Ariel SHAMIR	-	Professor, Interdisciplinary Center Herzliya
President :	George DRETTAKIS	-	Research Director, Inria
Examiner :	David SALESIN	-	Professor, University of Washington

Abstract

The appearance of a scene results from complex interactions between the geometry, materials and lights that compose that scene. While Computer Graphics algorithms are now capable of simulating these interactions, it comes at the cost of tedious 3D modeling of a virtual scene, which only well-trained artists can do. In contrast, photographs allow the instantaneous capture of a scene, but shape, materials and lighting are difficult to manipulate directly in the image. Drawings can also suggest real or imaginary scenes with a few lines but creating convincing illustrations requires significant artistic skills.

The goal of my research is to facilitate the creation and manipulation of shape, materials and lighting in drawings and photographs, for laymen and professional artists alike. This document first presents my work on computer-assisted drawing where I proposed algorithms to automate the depiction of materials in line drawings as well as to estimate a 3D model from design sketches. I also worked on user interfaces to assist beginners in learning traditional drawing techniques. Through the development of these projects I have formalized a general methodology to observe how artists work, deduce artistic principles from these observations, and implement these principles as algorithms.

In the second part of this document I present my work on relighting multiple photographs of a scene, for which we first need to estimate the materials and lighting that compose that scene. The main novelty of our approach is to combine image analysis and lighting simulation in order to reason about the scene despite the lack of an accurate 3D model.

Acknowledgments

While part of the exercise of defending an Habilitation is to provide an overview of one's work, the research presented in this document should be understood as the result of a collaborative work with a number of students, postdocs and researchers. Each collaborator gave me new insights and new perspectives on the problems we tackled, and the overall methodology I advocate was greatly influenced by exchanges with others. It is also thanks to fruitful collaborations that I opened my research interests to new fields, such as geometry processing, computer vision and HCI.

First of all, I would like to thank the great mentors who taught me to do research in the first place. Joëlle Thollot who always found the delicate balance between guidance and freedom to help me gain self-confidence during my Ph.D; François Sillion who led the group where I did my Ph.D; David Salesin who hosted me at Adobe and guided me through my first SIGGRAPH submissions; Frédo Durand and Sylvain Paris who made me part of MIT CSAIL for a few months; Maneesh Agrawala and Ravi Ramamoorthi who provided me with very complementary guidance during my postdoc at UC Berkeley, and who had a significant influence on the methodology I describe to formalize and automate artistic principles; and finally George Drettakis who continuously advises and supports me since I joined his group.

I am also grateful to Marie-Paule Cani, Kavita Bala and Ariel Shamir for accepting to review the present manuscript, and for taking the time to travel to my defense. Thanks also to David Salesin for attending both my Ph.D defense and my Habilitation defense.

I am now trying to be a good advisor for the PhD students and postdocs I work with. Several of them made large contributions to the research presented here: Pierre-Yves Laffont, Sylvain Duchêne, Emmanuel Iarussi, Jean-Dominique Favreau, Jorge Lopez-Moreno, Christian Richardt, Fumio Okura.

I also want to thank all my other recent collaborators: Emmanuelle Chapoulie, Pierre Bénard, Pascal Barla, Cloud Shao, Alla Sheffer, Karan Singh, James P. O Shea, Stefan Popov, Theophanis Tsandilas, Adrian Jarabo, Belen Masia, Fabio Pellacini, Diego Gutierrez, Baoxuan Xu, William Chang, James McCrae, Kenneth Vanhoey, Florent Lafarge, Clement Riant, Gaurav Chaurasia, David Bommes, Wilmot Li, Lora Oehlberg, Wendy Mackay. Many thanks also to Sophie Honnorat for her every day support.

Finally, many thanks to my sisters and parents for their encouragements, and to Alex for her love and trust that balances my professional life with an equally fulfilling personal one.

Contents

Chapter 1

Introduction

The appearance of a scene results from complex interactions between the geometry, materials and lights that compose that scene. With the progress of Computer Graphics, rendering algorithms are now capable of efficiently and accurately simulating these interactions to produce realistic images from a virtual description of a scene. However, creating a virtual scene is a tedious and time-consuming task as it involves modeling 3D objects and lights, positioning them in a 3D world, specifying material properties, adjusting camera position, all of which necessitates careful adjustment of numerous parameters through dedicated user interfaces. For these reasons, 3D rendering is reserved to a well-trained elite.

In contrast, photographs and drawings are much more accessible and as such remain very popular ways of creating images. People capture and share millions of photographs everyday, and children are able to represent imaginary worlds through drawings before knowing how to write. However, photographs and drawings also have their limitations. A photograph captures the conflated interactions of light and matter, and modifying its content after capture often requires solving the inverse problem of recovering the geometry, materials and lights of the captured scene. Similarly, while anybody can draw abstract doodles, conveying the appearance of a scene through a few lines and brush strokes requires advanced artistic skills.

The long term goal of my research is to lift the restrictions of drawings and photographs in order to ease the creation and manipulation of images for professional artists and laymen alike. Specifically, my work on computer-assisted drawing aims at assisting the drawing of shape, materials and lighting, as well as interpreting drawings to convert them into 3D models. My work on photograph manipulation aims at recovering the material and lighting information of a scene in order to allow the independent editing of these two ingredients, for instance to visualize a scene at a different time than the time of capture. I next describe these two research directions in more detail.

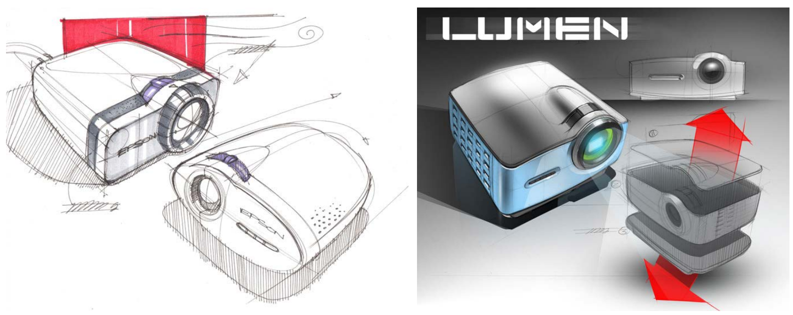
1.1 Computer-Assisted Drawing

My work on computer-assisted drawing is primarily motivated by applications in product and graphic design, where sketches are an integral part of the creation process. While studying the work of professional artists, I also identify techniques that can be automated or assisted to make them accessible to amateurs.

1.1.1 Context and Challenges

Product design, from the inception of an idea to its realization as a 3D concept, is extensively guided by freehand sketches. Sketching is visceral and quick, providing a direct connection between an idea and its visual representation. However, while concept sketches facilitate viewer understanding of 3D shapes, they require a mental leap to imagine the appearance of the drawn object, imbued with specific material properties and lighting.

Designers usually convey a more comprehensive 3D-look by artistically shading their sketches (Figure 1.1). This shaded imagery, often referred to as production drawings, is the traditional mode of communicating 3D concepts between designers and their patrons (Pipes 2007; Eissen and Steur 2008). However, production drawings require significantly more time, effort and expertise to create than the concept sketches on which they are overlaid. First, artists need to acquire a deep understanding of the way light interacts with shapes and materials in order to produce convincing depictions of an object (Powell 1986). Second, existing drawing tools require artists to depict material appearance by carefully applying color brushes and gradients over the empty regions of the drawing. For these reasons, only expert illustrators are even capable of creating compelling 3D-looking illustrations. Finally, shaded illustrations need to be laboriously redrawn for different shading configurations, such as changes in colors, materials or lighting conditions. In that context, I proposed several algorithms to **automate shading of line drawings**, either by recovering surface orientation to compute realistic shading or by generating color fills and gradients for stylized shading.



(a) Rough ideation sketch (b) Shaded presentation sketch

Figure 1.1: Typical design sketches by Spencer Nugent on sketch-a-day.com. (a) Designers draw rough ideation sketches to explore early concepts. (b) Shading is subsequently painted for presentation to decision makers.

Modeling the objects in 3D can substitute the manual shading process, allowing the use of rendering algorithms to automatically simulate the interaction of light with shape and materials. However, 3D modeling is often more distracting than direct sketching as users need to sculpt and navigate in a 3D world through a complex 2D user interface. Much of this 3D modeling effort is also wasted, given the frequency of iteration in the early design stages. Finally, once a 3D model has been created, lighting needs to be carefully configured to best depict the material properties of the objects. While well designed lighting configurations can enhance shading, reflections and refractions, poor lighting design can lead to misinterpretation of image content (Hunter et al. 2012). To address these challenges, I contributed to several algorithms to **estimate 3D models from concept sketches** and I proposed a method to **automate lighting design for material depiction.**

The above workflows reflect common practice in professional design but require drawing skills that many people feel out of reach. While drawing books and tutorials instruct how to acquire such skills, the techniques are often only illustrated on a few examples and require significant practice to be generalized to arbitrary models. In addition, books do not provide feedback to the aspiring artist about her performance. I thus proposed an interactive system to **assist novices in their practice of traditional drawing techniques**.

1.1.2 Methodology and Contributions

Through the development of my research I have developed a general methodology to formulate and automate artistic techniques. This methodology is inspired by the work of M. Agrawala (Agrawala et al. 2011), with whom I did my postdoc at UC Berkeley. In their work, Agrawala and colleagues relate principles of visual communication with insights from cognitive science to formulate algorithms that facilitate the creation of effective instructions (Agrawala et al. 2003), technical illustrations (Li et al. 2007; Li et al. 2008; Mitra et al. 2010) and maps (Agrawala and Stolte 2001; Grabler et al. 2008). We have extended this approach to encompass principles of geometry, material and lighting depiction. The resulting methodology is composed of four main steps:

1. First, we observe the techniques of novice and professional artists, either through design books and tutorials or by performing field studies and interviews.

2. The second step of our approach consists in distilling our observations in a coherent set of principles. The main challenge in this task comes from the fact that design techniques are often described by different artists in their own vocabulary and illustrated on specific examples. We formalize and generalize these techniques by relating them to findings in other scientific fields. In particular, the way artists draw shape, material and lighting is strongly related to the way shape, material and lighting interact in reality – as studied in geometry, computer vision and rendering – and how we perceive these interactions – as studied in visual perception.

3. From the design principles, we then derive algorithms to facilitate the artistic tasks. To do so, we commonly express the principles as energy terms in an optimization, balancing the sparse and often inaccurate user inputs with the automatic generation of virtual content.

4. Finally, we validate our results by comparing them to traditional illustrations or by running user studies to evaluate the effectiveness of our tools.

The first part of this document describes the main steps of this methodology in detail, from initial observations, to formalization, implementation and evaluation. I illustrate each step with examples from my research, such that each specific project is actually discussed multiple times through the document. Specifically, my work on this topic resulted in the following contributions:

- A study on how people perceive materials in stylized images (Bousseau et al. 2013). We used non photo-realistic rendering algorithms to create images of a scene under different styles (painting, cartoon), from which we evaluate how stylization alters the perception of gloss.

- An interactive drawing tool that provides automated guidance to practice traditional drawing-by-observation techniques (Iarussi et al. 2013). Our tool extracts constructions lines from a user-provided photograph and generates visual feedback to help users understand their mistakes.

- A compact *shade tree* representation to automatically generate stylized depictions of materials in vector graphics (Lopez-Moreno et al. 2013). This representation encapsulates the creation of vector primitives that artists routinely use to convey material effects, which allows even inexperienced users to quickly turn a line drawing into a fully colored illustration. I also proposed a user-assisted method to convert bitmap images into vector graphics with editable layers (Richardt et al. 2014), which allows a variety of edits, such as modifying the shape of highlights, adding texture to an object or changing its diffuse color.

- Several algorithms to estimate normal fields and 3D shapes from concept sketches. These algorithms leverage specific construction lines that designers draw to convey surface curvature (Shao et al. 2012; Xu et al. 2014; Iarussi et al. 2015). Two of the algorithms take a single vector drawing of the shape as input, while the third method works from a bitmap drawing and is robust to sketchy lines made of many overlapping strokes.

- An optimization framework to design lighting that enhances material appearance in realistic renderings of 3D scenes (Bousseau et al. 2011). Our algorithm generates environment maps that reveal material-specific features, such as sharp highlights on shiny objects or high contrast along contours of transparent objects.

1.2 Multi-View Intrinsic Images and Relighting

My work on photograph manipulation also aims at allowing users to obtain the material and lighting they wish to convey in an image. However, in contrast to drawings where I leveraged artistic techniques to create plausible material and lighting effects, photographs require the inversion of the image formation model in order to recover the geometry, material and lighting components that explain the captured image. I proposed several algorithms to perform this inversion in the context of image-based-rendering, where we take as input multiple images of a scene.

1.2.1 Context and Challenges

Recent progress on automatic multi-view 3D reconstruction (Snavely et al. 2006; Furukawa and Ponce 2010) and image-based-rendering (Chaurasia et al. 2013) greatly facilitate the production of realistic virtual walkthroughs from a small number of photographs of a scene. However, while existing algorithms can render the scene from novel viewpoints, lighting remains fixed to the conditions at the time of capture. Fixed lighting is a major limitation of image-based rendering, preventing its use for traditional applications of image synthesis such as video games and special effects.

Under the assumption that the materials are diffuse, an image can be expressed as the product of an illumination component that represents lighting effects and a reflectance component that is the color of the observed material (Figure 1.2). This decomposition forms the so-called *intrinsic components* of the image. The main challenge in modifying lighting in an image resides in the fact that a pixel color aggregates the effect of both reflectance and lighting, so that standard color manipulations are likely to affect both components. We thus need to first separate reflectance from lighting, which is an ill-posed problem since an infinity of material and illumination configurations can produce the same image.

(a) Original photograph (b) Reflectance (c) Illumination (d) Re-texturing

Figure 1.2: Assuming diffuse materials, a photograph can be expressed as the product of its reflectance and illumination components (b-c). This decomposition facilitates advanced image editing such as re-texturing (e) and re-lighting. This decomposition was computed using the user-assisted method described in (Bousseau et al. 2009).

Existing techniques tackle this problem by including additional constraints, such as the assumption that the illumination component is monochrome and smooth. Such approaches remain challenged by outdoor scenes where sun and sky light result in a mixture of colored lighting with hard shadows. Other methods deal with hard shadows by leveraging detailed geometry, varying illumination from a fixed or restricted viewpoint, or user assistance. The main novelty of the approaches I have proposed is to leverage a sparse 3D reconstruction of the scene to constrain the **decomposition of multiple images into their reflectance and illumination components**. I also described how to use the resulting decompositions to **transfer lighting between photographs** and **render new lighting with cast shadows**.

1.2.2 Methodology and Contributions

The main idea behind our methods is to combine approximate geometric reconstruction with image processing, thus leveraging their respective strengths. We exploit the geometric reconstruction to compute lighting information for a sparse set of pixels, and use image-guided propagation and segmentation to decompose all pixels of the photographs into their intrinsic components.

In this context, we proposed two alternative approaches that differ in the data they take as input:

- The first family of algorithms we proposed estimates lighting and reflectance in outdoor scenes from images captured at the same time of day, i.e. under the same lighting condition (Laffont et al. 2013; Duchêne et al. 2015). We use the geometric reconstruction to compute smooth shading due to sky and inter-reflections, while we use image analysis and processing to locate accurate shadow boundaries.
- The second approach we proposed focuses on photocollections of well-known touristic landmarks for which we can easily collect multiple photographs under varying lighting conditions (Laffont et al. 2012). The geometric reconstruction provides us with sparse correspondences between images, on which we factor out the varying lighting from the constant reflectance. We then propagate the lighting information to all pixels using image processing.

We demonstrate the benefits of our intrinsic decompositions for image relighting. In the first case, we approximately reconstruct shadow casters which we use to render new shadows in the images. In the second case, we transfer illumination between images of the same scene taken from different viewpoints and under different lighting conditions.

Chapter 2

Computer-Assisted Drawing

This chapter covers my research on computer-assisted drawing. I first provide a brief description of the main tools and algorithms I have proposed in order to position them with respect to related work. I then describe the main steps of the methodology that underpins my work: observation of artistic practice, formulation of artistic principles, implementation of these principles as algorithms and evaluation.

2.1 Overview and Related Work

I discuss in this section existing work that is most related to my research on assisting drawing for novices, on estimating 3D models from sketches and on depicting materials and lighting in stylized and realistic rendering. I refer the interested reader to the individual papers for extended discussions.

2.1.1 Drawing Assistance

Figure 2.1 shows an overview of the system we developed to help beginners practice traditional drawing-from-observation techniques (Iarussi et al. 2013). Our *drawing assistant* provides guidance and feedback over a model photograph that the user reproduces on a virtual canvas. We use computer vision algorithms to extract visual guides that enhance the geometric structures in the image. In the example of Figure 2.1, the user first sketched the block-in construction lines (c, blue) before drawing the regions and adding details. This guidance helps users produce more accurate drawings.

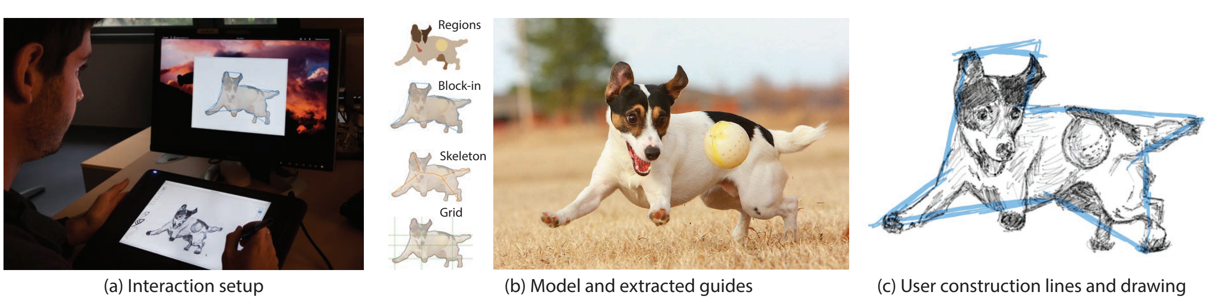


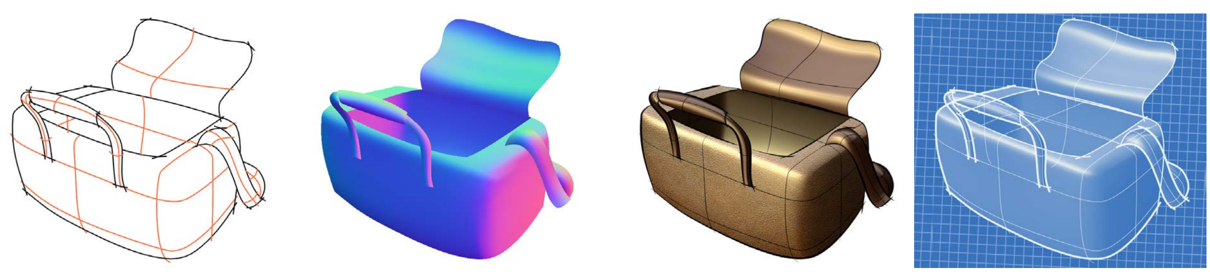
Figure 2.1: Our drawing assistant in use (a), the user observes a model photograph on the computer monitor and reproduces it on the pen display. Our system automatically extracts visual guides (b) that help users structure their drawings (c). (Iarussi et al. 2013)

Several existing systems assist the process of drawing by displaying guidance on the drawing surface using projectors (Flagg and Rehg 2006; Laviole and Hachet 2012) or pen tablets (Lee et al. 2011). All these methods are reminiscent of the traditional "paint-by-number" and "connect the dots" drawing books that guide people in placing individual strokes until completing complex artworks. While these approaches can give people confidence in their ability to draw, they do not help them observe and understand the underlying shapes, relationships and proportions of the drawn models. In contrast, we designed our drawing assistant to offer guidance and feedback on the model photograph, which encourages users to observe the subject they want to draw before reproducing it on the drawing surface.

A complex drawing can be easier to achieve if it is decomposed into a succession of simple steps. Sketch-Sketch Revolution (Fernquist et al. 2011) allows expert users of sketching software to generate step-by-step tutorials for novice users. However, such tutorials illustrate drawing techniques on pre-recorded examples rather than images of the user's choice. Closer to our work is the iCanDraw? system (Dixon et al. 2010) that assists users in drawing faces thanks to face recognition algorithms. We draw inspiration from this approach, incorporating some of its design principles. However, our drawing assistant implements a different set of guides to draw arbitrary models rather than faces. We also provide visual feedback that highlights alignments and proportions on the model photograph, helping users to see and correct the relationships between different parts of a shape.

2.1.2 Sketch-Based Modeling

My work on sketch-based modeling focuses on product design and leverages specific drawing techniques popular in that domain. In particular, designers frequently use *cross-sections* to convey the curvature of smooth shapes, as illustrated in Figure 2.2(a). We derived the mathematical properties of cross-section curves and leverage them to automatically estimate 3D information about the drawn shape. One such property is that cross-section curves align with curvature lines and as such should be orthogonal in 3D. We first exploited this property to estimate normal fields for shading (Shao et al. 2012), and later extended our algorithm to recover a complete 3D model (Xu et al. 2014). These two algorithms take a single vector drawing as input, as illustrated in Figure 2.2 and 2.3. More recently we have proposed an algorithm to estimates normal fields from rough bitmap drawings (Iarussi et al. 2015), as shown in Figure 2.4. This algorithm performs scattered data interpolation between the strokes to estimate curvature directions at each pixel, which we then use to compute the surface normal.



(a) Input curves (b) Estimated normals (c) Shading

Figure 2.2: Designers frequently draw cross-sections (drawn in orange) to convey surface curvature (a). Our *CrossShade* algorithm exploits those curves to estimate the surface orientation at each pixel (b). The resulting normal field allows users to shade the objects using a variety of shading styles and setups (c). (Shao et al. 2012)

Sketch-based modeling has matured over the past two decades; I refer readers to (Olsen et al. 2009) for a survey of existing methods. Sketching interfaces can be roughly described as based on a single-view or a multi-view metaphor. Using multi-view tools, artists sketch strokes from different viewpoints onto existing 3D geometry (Igarashi et al. 1999; Nealen et al. 2007) or use strokes to define transient construction surfaces on which 3D curves are drawn (Bae et al. 2008). Such tools combine the fluidity of sketching with a typical 3D CAD workflow based on frequent view changes. In contrast, our algorithms recover 3D information from a single sketch. Such a single-view approach is closer to traditional pen-on-paper sketching and allows 3D recovery from existing drawings.

Recent single-view methods rely on user indications or construction lines to model smooth objects. Schmidt et al. (2009) require users to specify polyhedral scaffolds as a support for 3D recovery from input sketches. Olsen et al. (2011) and Sýkora et al. (2014) combine user indications and shape inflation to model smooth shapes from existing drawings and photographs, while Gindgold et al. (2009) let users position parameterized primitives on an existing sketch, using various annotations to enforce alignment, equal length and symmetry. Instead, our approaches build directly upon the descriptive power of artist-drawn curves to recover piecewise-smooth shapes with minimal annotation.

Lipson and Shpitalni (1996) estimate 3D models from engineering drawings dominated by straight orthogonal lines. Their algorithm detects regularity constraints in the 2D drawing, including parallelism and orthogonality of the lines, and enforces these constraints on the 3D reconstruction. While we draw inspiration from this work, our algorithms aim at a more ambitious set of input sketches, in particular piecewise-smooth shapes that dominate modern product design. Such shapes have a much greater degree of freedom and few if any of the regularities listed above, and thus cannot be handled by existing methods.

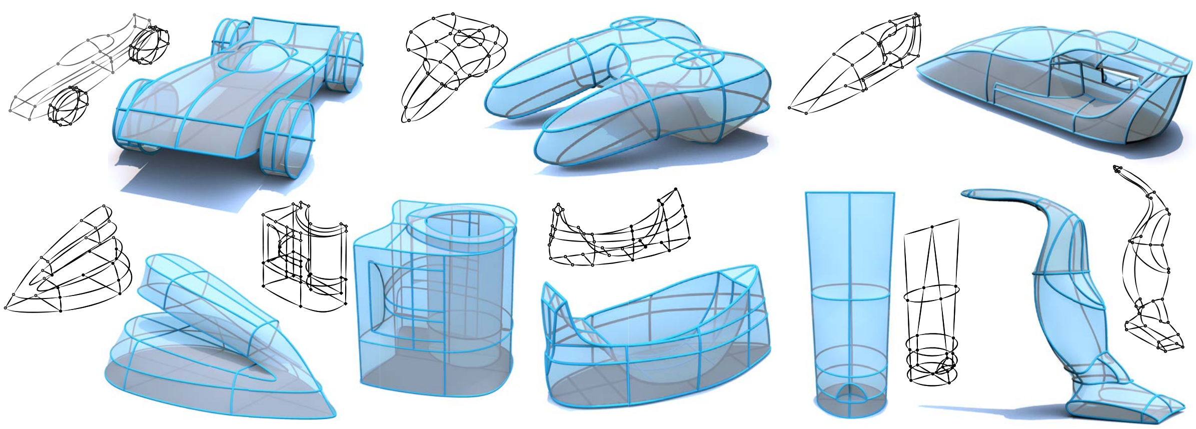


Figure 2.3: Our *True2Form* algorithm takes as input a single 2D vector drawing. We formulate a set of local 3D regularity properties that our algorithm detects and applies selectively to lift the curves off the page into 3D. Note that our interface allows users to draw half of the shape that is then mirrored to create the complete object. (Xu et al. 2014)

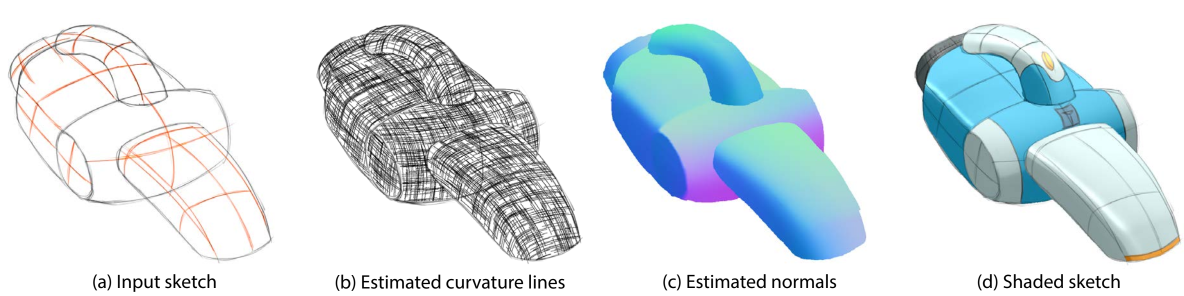


Figure 2.4: Our *BendFields* algorithm estimates normal fields from rough bitmap sketches by extrapolating curvature directions from the sketched cross-sections (red strokes in (a)). (Iarussi et al. 2015)

2.1.3 Material Depiction

While the appearance of an object results from complex interaction between light, materials and shape, artists have accumulated a number of techniques to draw convincing illustrations without explicitly simulating light transport. In the domain of vector graphics, skilled artists combine multiple layers, each composed of simple color and transparency gradients, to achieve vivid material appearance. I have worked on several methods to facilitate this task. First, we introduced *Vector Shade Trees* that represent stylized materials as a combination of basic shade nodes composed of vector graphics primitives (Lopez-Moreno et al. 2013). Vector shade trees allow the depiction of a variety of materials while preserving traditional vector drawing style and practice. We integrated this representation in a vector drawing tool that allows users to easily apply stylized shading effects on vector line drawings, as illustrated in Figure 2.5. In a follow-up work, we proposed an interactive vectorization technique to convert a bitmap image into a stack of opaque and semi-transparent vector layers composed of linear or radial gradients (Richardt et al. 2014). Users can manipulate the resulting layers using standard tools to quickly produce new looks, as shown in Figure 2.6.

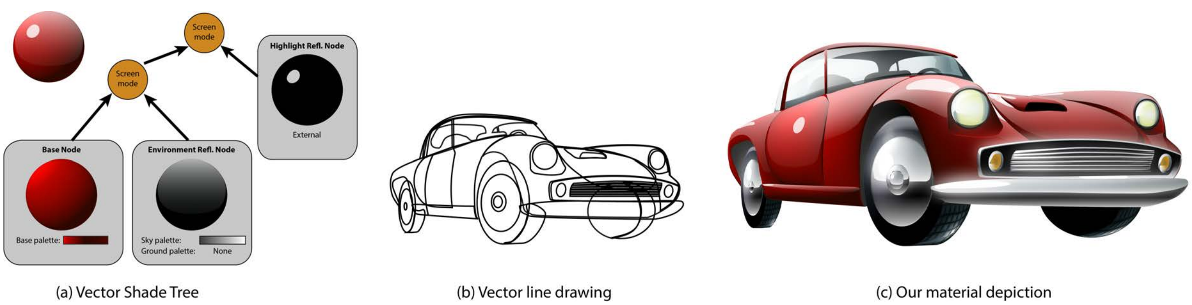


Figure 2.5: A Vector Shade Tree represents a stylized material by combining basic shading nodes (a). By assigning various shade trees to the regions of a line drawing (b), users can quickly colorize vector illustrations with rich material appearance (c). (Lopez-Moreno et al. 2013)

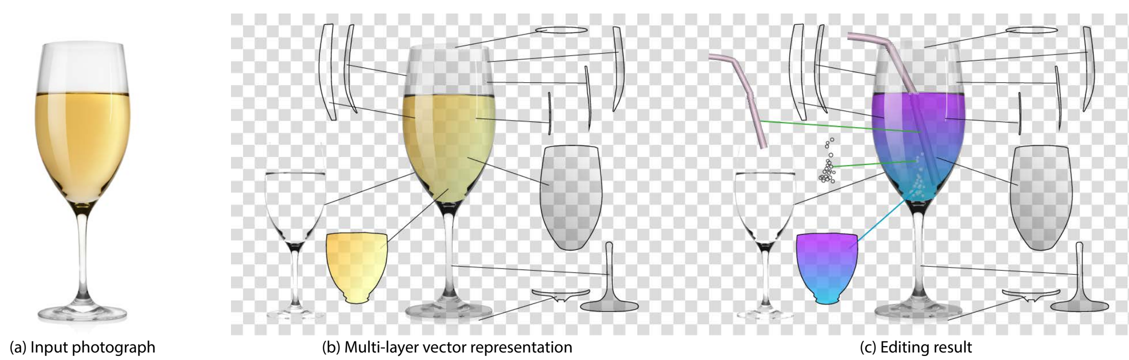


Figure 2.6: Starting with an input photograph (a), users of our interactive tool can produce a vector drawing composed of overlapping opaque and semi-transparent layers (b). Each layer can then be edited with standard vector graphics tools (c). (Richardt et al. 2014)

In the field of non-photorealistic rendering, Gooch et al. (1998) take inspiration from technical illustration to design a stylized lighting model that conveys metallic appearance. We have adopted a similar methodology since we analyzed illustrator practice to identify material depiction guidelines. However, we focus on vector graphics rather than 3D rendering and we describe new guidelines for the depiction of other common material properties such as glossy reflections, mirror reflections, transparency and translucence. Combined together in a *shade tree*, these properties allow the generation of a much larger set of stylized materials. The seminal Shade Trees paper by

Cook (1984) also introduced the concept of shading languages to the realistic rendering community; this approach was a rich source of inspiration for our work. Grabli et al. (2010) adapt this concept to the stylization of strokes in line drawings and describe programmable shaders to control style attributes such as color, texture and thickness. We adopt a complementary approach by focusing on the stylized depiction of materials in vector graphics, giving control to material attributes like shading and reflections.

Most existing vectorization methods segment the input image into smooth color regions that are then represented by vector gradients. Lecot and Lévy (2006) fit linear and radial gradients to generate vector images in Art Deco style. Gradient meshes represent complex gradients by interpolating colors over the faces of a quad mesh, making them a powerful primitive to capture the smooth color variations of natural images (Sun et al. 2007). Finally, Diffusion Curves (Orzan et al. 2008) vectorize images by storing colors on each side of strong edges while computing smooth color variations in-between the edges using a diffusion process. All of these algorithms are designed to vectorize opaque objects into a single layer. Our approach complements these techniques by decompositing the image into multiple transparent and opaque layers, each layer being simpler than the composed image. Our approach is also related to alpha matting (Smith and Blinn 1996), although existing matting algorithms estimate bitmap layers rather than vector gradients and therefore do not provide a small set of parameters suitable for editing the matted layers. Instead, our decompositing algorithm exploits the parametric nature of vector gradients to jointly separate and vectorize semi-transparent layers.

2.1.4 Lighting Design

Shading, reflections and refractions are important visual features for understanding the shapes and materials in an image. Figure 2.7 compares two renderings of the same scene under different lighting conditions. In Figure 2.7a, the lighting enhances the thickness variations in the subsurface scattering wax candle, it accentuates the Fresnel reflections at grazing angles of the porcelain vase and it adds strong edges in the specular highlights of the chrome sculpture. In Figure 2.7b however, poor lighting greatly reduces these visual features and makes it more difficult to correctly identify the materials. The candle appears more like solid plastic, the vase appears to be made of diffuse clay and the sculpture no longer looks like it is made of chrome. Figure 2.7a was produced using our algorithm for optimizing and synthesizing environment maps that emphasize distinctive visual features of the materials in a scene (Bousseau et al. 2011). We define a set of image quality metrics that measures how well a given lighting accentuates material features like contrast and sharpness, and present a general optimization framework to solve for the environment map that maximizes these metrics.

(a) Our optimized lighting emphasizes materials

(b) Poor lighting de-emphasizes materials

Figure 2.7: Our method (a) automatically optimizes the lighting to enhance material-specific visual features such as bright highlights and subsurface scattering. Poorly designed lighting (b) diminishes these characteristic visual features of the materials. (Bousseau et al. 2011)

Researchers have developed a variety of techniques for optimizing lighting to enhance shape (Shacked and Lischinski 2001; Gumhold 2002; Lee et al. 2006). In contrast, we focus on enhancing the visual features that emphasize materials rather than shape. In addition, our system generates complex environment lighting rather than a small set of point lights. Kerr and Pellacini (2009) present an excellent survey and evaluation of lighting design interfaces. However, these interfaces do not incorporate the guidelines expert photographers and lighting designers commonly use to enhance the appearance of materials. Users must rely on their own training and experience to set up the lighting. In contrast, our automated system optimizes the lighting based on design principles for material depiction.

Our work on material depiction and lighting design is also strongly inspired by research on material perception in realistic rendering. Fleming et al. (2003) show that the recognition of surface reflectance is improved when objects are illuminated under natural environments. These results suggest that natural image statistics such as color and derivative histograms provide strong cues for material perception. Ramanarayanan et al. (Ramanarayanan et al. 2007) evaluate if transformations of the lighting environment such as blurring and warping are perceivable given various geometries and materials. They observed that blurring the illumination is harder to perceive for diffuse materials, and that warping is harder to perceive for bumpy surfaces. We made similar observations in the context of stylized rendering, where brush strokes and cartoon quantization can be seen as forms of blurring and warping (Bousseau et al. 2013).

2.2 Observing How Artists Work

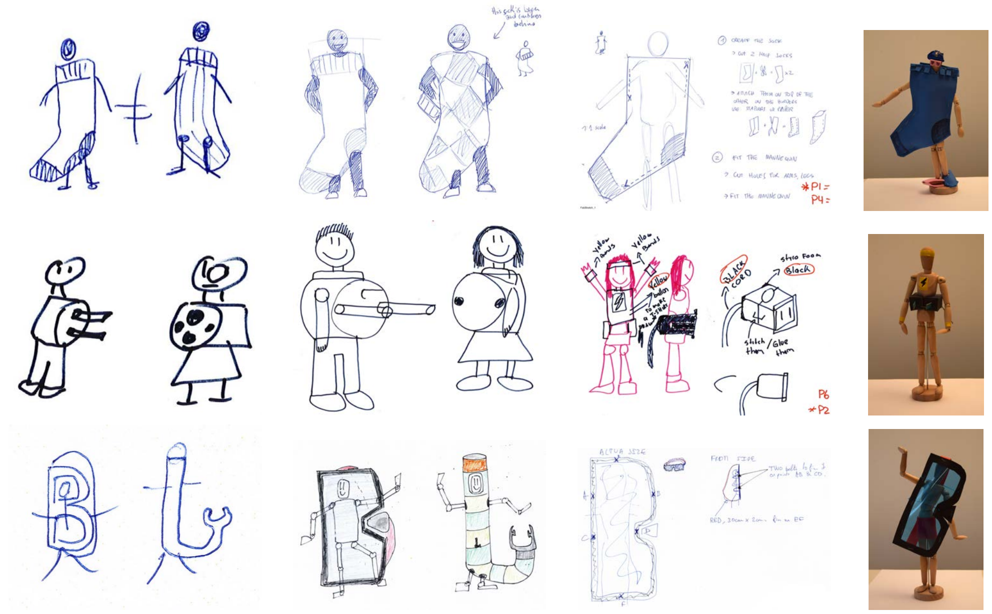
Most of the work presented in this chapter has been motivated by observations on how artists perform specific tasks. While I conducted a few interviews and field studies to better understand how artists work, I found that studying art books is often a more effective way to collect relevant guidelines suitable for implementation in a computer-assisted tool. I describe the advantages and drawbacks of the two approaches in this section.

2.2.1 Observational Studies

In the domain of Human Computer Interaction (HCI), field observations and interviews are common ways of understanding how a class of users perform specific tasks. In particular, observational studies allow researchers to obtain first-hand information about practices that have not yet been well documented. However, conducting meaningful observational studies in our context requires specific care.

A first challenge is to recruit artists with representative skills in order to draw conclusions that can generalize to other users. Unfortunately, skilled artists are rare and often have little time to participate in extensive interviews. Confidentiality may also prevent the observation of professional designers in their working environment. As an example, we had the chance to conduct a few interviews at Toyota ED^2 design studio, located in Sophia Antipolis. In the absence of an official collaboration, we could only interview four designers for one hour each, in an isolated room. While these interviews provided us with valuable insights on design practices, longer interviews and field observations with more participants would be needed to formulate well-grounded findings.

A second challenge stems from the fact that design is a complex, collaborative process that often takes place over a long period of time. I am currently collaborating with HCI experts to acquire a greater expertise in conducting observational studies of designers at work (Bousseau et al. In Progress). Previous studies of design behavior have set up time-constrained laboratory observations of a specific, focused design task (Brereton and McGarry 2000), or taken an longitudinal ethnographic field study approach (Henderson 1998) to look at the evolution of design behavior over time. We combine these approaches to observe how novices perform design tasks during a one-day design charette. This approach offers a middle-ground between an open field observation and a tightly-controlled lab experiment. On the one hand, the study follows a well-defined structure that we designed to stimulate the creativity of the participants. The structure of the study also facilitates data collection and comparison. On the other hand, participants perform ecologically-valid tasks that are close to the ones they would perform when solving a real design problem. The goal of this structured approach is to generate novel



(a) Ideation (b) Presentation (c) Fabrication (d) Prototype

Figure 2.8: Representative sketches and physical prototypes produced by novices during a costume design task. We instructed the participants to brainstorm about a concept (a), present the concept to a jury (b), describe how to fabricate the concept (c) and finally build a prototype in collaboration with another participant (d).

hypothesis that could later be tested with more constrained experiments.

Figure 2.8 shows sketches and physical prototypes produced by our participants when performing a costume-design task. Our initial analysis of this data reveals that while participants had no or little training in design, they tended to adopt similar strategies as professional designers. In particular, they used small and simple drawings to generate ideas (Figure 2.8a), big and detailed drawings for presentation (Figure 2.8b) and multiple drawings with annotations to document fabrication (Figure 2.8c). We also observed the use of common drawing techniques, such as focus+context and close-ups on important parts, step-by-step assembly instructions, transparency to show inner parts, 2D cutout of unfolded 3D shape, front/top/side views of 3D shapes. Finally, to our surprise, participants used very little sketching when discussing how to fabricate a concept. Instead, participants favored gesture and manipulation of physical materials, for example by grabbing a piece of foam and showing how it could fit on a mannequin. Based on these observations, we are now in the process of formulating recommendations for computer-aided design tools adapted to the needs and skills of novices.

2.2.2 Studying the Design Literature

The challenges of observational studies outlined in the previous section can be partly addressed by studying the art literature. Good art books and tutorials are written by professional artists with years of experience, and as such are representative of standard practices of their discipline. Well-known books are also likely to influence future generations of practitioners. In addition, art books compile numerous long-standing techniques that would be hard to discover solely from field observations. These techniques are described from the artists' perspective and as such do not suffer from the experimenter biases. Finally, the art literature offers many examples of what artists and designers aim at producing. In addition to motivating novel computer-assisted tools, these examples can be used to validate the performance and pertinence of the algorithms we propose.

Figure 2.9: Some of the books that inspired my research on drawing shape (Dodson 1985; Eissen and Steur 2008), materials (Powell 1986) and lighting (Hunter et al. 2012).

Figure 2.9 shows some of the books that inspired my research. Most of these books have been highly popular and re-edited, which indicates their relevance and impact in their respective domains. A number of Computer Graphics papers have similarly been directly inspired by design and art books, such as early work on pen-and-ink rendering (Winkenbach and Salesin 1994), watercolor rendering (Curtis et al. 1997), illustrative rendering (Gooch et al. 1998), sketch-based modeling with construction lines and annotations (Schmidt et al. 2009; Gingold et al. 2009), illustration of mechanical assembly (Li et al. 2007; Li et al. 2008; Mitra et al. 2010), character posing and animation (Guay et al. 2013) and many more. Each of these papers implies the formulation and implementation of design principles. I detail the challenges of such an approach in the next section.

2.3 Formulating Artistic Principles

Part of the difficulty in formulating general artistic principles stems from the fact that the techniques artists use are based on accumulated artistic knowledge. The description of these techniques are spread across a variety of books and tutorials, and are often illustrated with specific examples where information on how to perform a general task is intertwined with information specific to the subject at hand. Many of the techniques also share common principles implemented in different variations. Finally, art books and tutorials often describe drawing techniques from the artist's perspective, without necessarily justifying these techniques from a scientific point of view.

To build a tool facilitating an artistic task we first need to select, classify and generalize a coherent set of guidelines that artists combine together to achieve their goals. Our guidelines describe artists' current workflows (the tools they use, their vocabulary), and decompose these workflows into components suitable for integration into functional systems. We often reinforce these guidelines by relating artistic principles with perceptual principles. This approach is based on the premise that the principles guiding artists in their choices are the same principles that aid viewers in understanding an illustration.

2.3.1 Collecting Principles

During my past years of research, my colleagues and I have collected and formalized artistic principles on how to depict shape (Shao et al. 2012; Xu et al. 2014; Iarussi et al. 2013; Iarussi et al. 2015), materials and lighting (Bousseau et al. 2011; Lopez-Moreno et al. 2013; Richardt et al. 2014). Figure 2.10 illustrates some of the guidelines we have identified, which the next paragraphs explain in more detail.

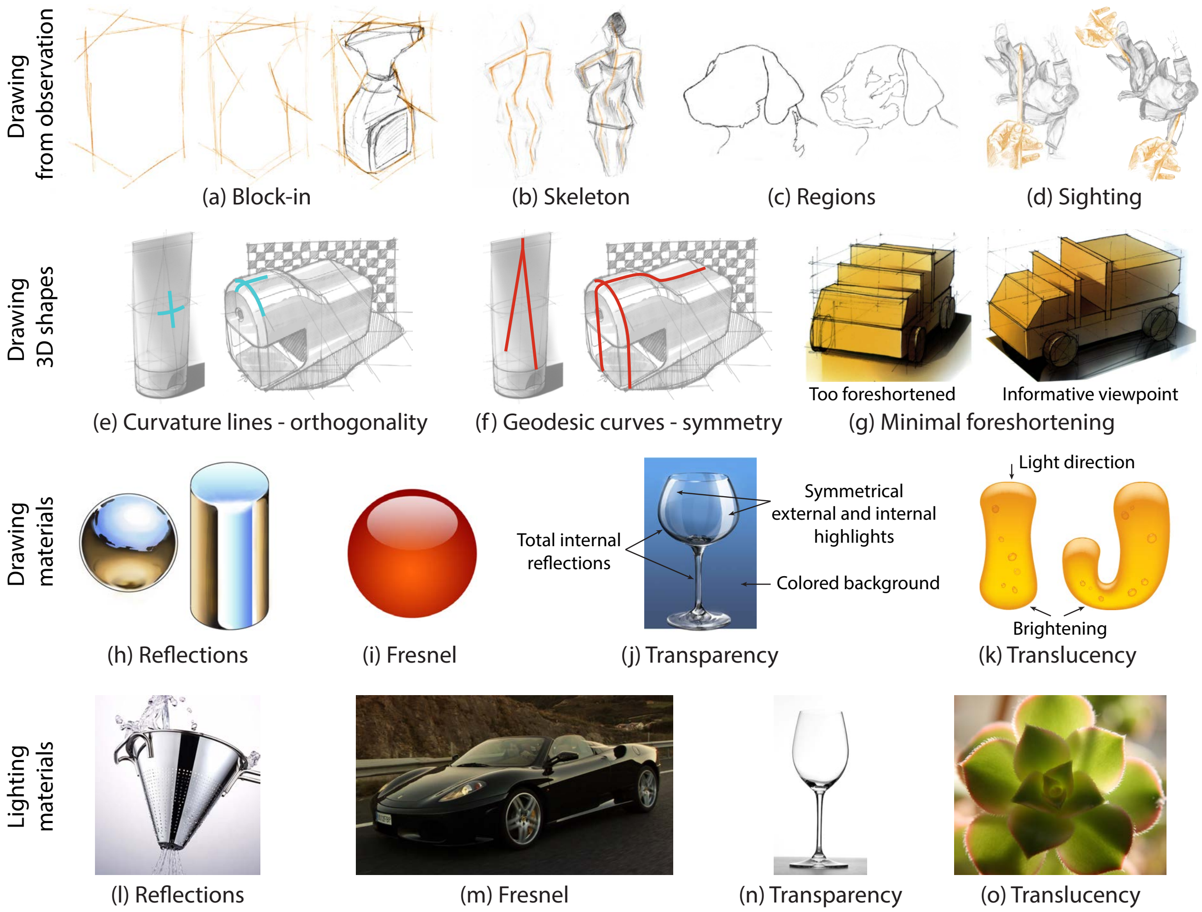


Figure 2.10: A selection of the guidelines we have collected on shape, material and lighting depiction.

Drawing shape from observation. A major challenge in drawing from observation is to trust what we *see* rather than what we *know* (Nicolaides 1969; Edwards 1979; Dodson 1985). Our mental image of common objects is iconic and conflicts with the particular instance that we observe, resulting in distorted or simplistic drawings (Eitz et al. 2012). Drawing books and tutorials provide simple techniques to help learners gain consciousness of shapes that they observe and their relationships. While most authors only present a subset of techniques and describe them in their own vocabulary and style, we distilled from these resources three main principles suitable for integration in a computer-assisted tool:

- Drawers should first lay down the main structure of the drawing with a coarse approximation of the shape. The *block-in* technique approximates the shape with a polygon (Figure 2.10(a)) or with a collection of geometrical primitives like disks and rectangles. Skeleton lines depict the principal internal directions and are more suitable to elongated structures and characters (Figure 2.10(b)).

- The coarse structure forms a scaffold to guide contour drawing. Drawers should draw contours of large regions first and then details (Figure 2.10(c)).

- Proportions and alignments should be verified to avoid distortions. It is often hard to judge and measure the distortions in a drawing with a naked eye. Artists make use of the ”sight” (or ”thumb-and-pencil”) method to facilitate this task. They hold their pen or pencil at arms length between their eye and the object of interest and sight along it (Figure 2.10(d)).

Drawing 3D shapes. Through years of practice, designers have learned to strategically sketch curves that are descriptive of the geometry they want to convey (Eissen and Steur 2008; Eissen and Steur 2011). These curves trigger our perception of geometric regularities that aid the inference of depth in line drawings. We combined observations from the design and perceptual literature to formulate properties of design sketches that explain their effectiveness in conveying complex, smooth 3D shapes. In particular, we identified the following regularity cues:

- Perceptual studies indicate that observers interpret intersecting smooth curves as aligned with the lines of curvature of an imaginary surface (Stevens 1981; Mamassian and Landy 1998), and thus having orthogonal tangents at these intersections. Figure 2.11, reproduced after (Stevens 1981), nicely illustrates this perceptual effect. Designers leverage this perceptual bias to depict smooth shapes effectively by using *cross-section* curves aligned with lines of curvature, as illustrated in Figure 2.10(e).

- Designers also position curves to emphasize intrinsic shape properties like local symmetries (Figure 2.10(f)). Differential geometry tells us that curves delineating local symmetries are geodesics of the surface. This observation is consistent with viewer tendency to interpret intersecting curves as geodesics over smooth surfaces (Knill 1992).

- Pizlo and Stevenson (1999) have shown the importance of planar contours on our ability to recognize the same shape under different viewpoints. Designers exploit this perceptual effect by drawing globally planar cross-section curves over smooth surfaces (Figure 2.10(e,f)).

- Design books recommend using viewpoints that "optimize shape information" and instruct designers to minimize foreshortening over most faces of the object (Figure 2.10(g)). This recommendation is consistent with the perceptual notion of *general* or *non-accidental* viewpoints (Nakayama and Shimojo 1992; Mather 2008), which suggests that observers interpret 2D geometric properties as strongly correlated with 3D geometry rather than being caused by a particular choice of viewpoint.

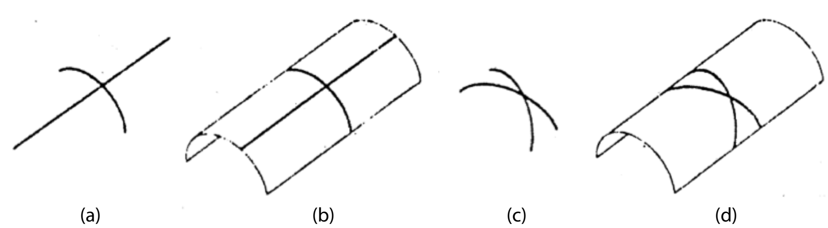


Figure 2.11: Most people interpret the drawing in (a) as the principal directions of curvature of a cylinder (b). Similarly, in the absence of context, (c) tends to be interpreted as curvature lines over an ellipsoid, even though the lines were drawn on a cylinder (d). After (Stevens 1981).

Depicting materials and lighting. Figure 2.10(h,i,j,k) illustrates guidelines we have collected on material depiction in illustrations. Note that these guidelines closely match lighting techniques used by photographers to enhance material appearance in photography (Figure 2.10(l,m,n,o)). We have identified different techniques for different classes of materials:

- To avoid cluttering the illustration with lots of highlights, artists typically depict reflections with simple abstract environments composed of a sky dome and a ground separated by an horizon line (Figure 2.10(h)) (Powell 1986). Similarly, photographers convey shiny materials by designing the lighting such that it produces reflections of sharp high contrast edges (Figure 2.10(l)) (Hunter et al. 2012). These observations are in line with perceptual studies that have shown that real-world illumination contributes to accurate perception of shiny materials because it contains edges and bright light sources (Fleming et al. 2003).

- For glossy materials such as plastic, artists only depict the reflection of the strongest light sources in the environment such as the sky dome or the rectangular shape of a window (Martin 1989a). Artists and photographers also exaggerate the strength of the reflections at grazing angle to convey the Fresnel effect (Robertson 2003) (Figure 2.10(i,m)).

- Artists depict transparent materials like glass and ice by drawing highlights in multiple locations due to multiple scattering within the object (Martin 1989b). For simple shapes, these internal highlights are often placed symmetrically to the external highlights with respect to the center of the shape (Figure 2.10(j)). In addition to bright internal highlights, artists often draw dark bands near the silhouette of transparent objects to depict total internal reflections. These bands delineate the contours of the object with a strong contrasting edge and the width of the bands suggest the thickness of the material. To obtain a similar effect, studio photographers place the object in front of a bright background and position dark plates (called *gobos*) around the object, outside the field of view, to produce dark reflections and refraction along contours (Hunter et al. 2012). Figure 2.10(n) illustrates this technique called *bright-field* lighting.

- A distinctive visual feature of translucent objects is their ability to scatter light through multiple internal reflections. Artists convey light exiting the object after internal scattering by brightening the parts of the object that are directly opposite from the point at which the light strikes the surface (Figure 2.10(k)). Photographers also enhance the appearance of translucent materials by illuminating the object from behind (Figure 2.10(o)). Fleming et al. (2004) show that subsurface scattering is better perceived when objects are illuminated from behind, since backlighting provides more visual cues of light scattered through the thin parts of the object.

2.3.2 Conducting Perceptual Studies

Existing design and perceptual literature is not always sufficient to formulate a coherent set of principles suitable for automation. We have thus run several dedicated perceptual studies to complement existing observations and reinforce weak hypothesis.

We based our work on 3D reconstruction from concept sketches on the hypothesis that intersecting cross-section lines provide strong cues of surface orientation (Shao et al. 2012; Xu et al. 2014; Iarussi et al. 2015). We validated this hypothesis by asking subjects to interactively specify the surface normal they perceive at each intersection of cross-section lines in a sketch, as illustrated in Figure 2.12. This task is inspired by the one of Cole et al. (Cole et al. 2009), who studied how people perceive shape from feature lines like contours and silhouettes. We found that viewers are persistent (median 2D difference of 6°), consistent with each other (median 2D difference of 10.6°) and accurate in their perception of surface normal from cross-section curves (median 2D error of 8°).

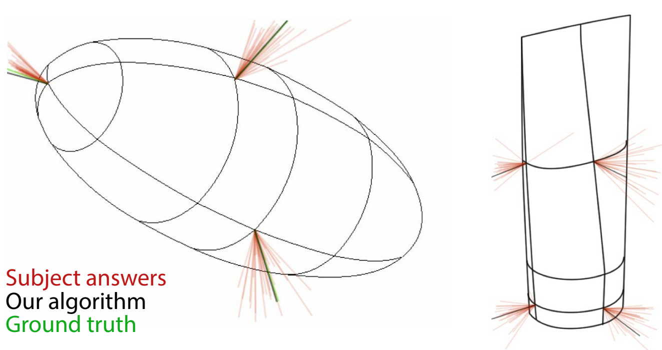


Figure 2.12: We asked subjects to specify the surface normal they perceive at intersections of cross-section curves. Overall, subjects are accurate with respect to ground truth and consistent with each other. (Shao et al. 2012)

Another of our studies aimed at understanding the perception of materials in illustrations (Bousseau et al. 2013). Artists often rely on their experience of their media to depict materials in different styles. However, this artistic knowledge is often implicit and while high-level rules exist to depict light and shade in a given style, no guidelines exist to vary low-level material properties such as the amount of gloss. We focused on painterly and cartoon images and used non-photorealistic rendering as a tool to systematically study the effects of style parameters on gloss perception. Specifically, we used a matching task to measure how brush size, brush opacity, brush texture and cartoon quantization softness alter the perception of materials ranging from very shiny to almost diffuse. Our study reveals a compression of the range of representable gloss in stylized images so that shiny materials appear more diffuse in painterly rendering, while diffuse materials appear shinier in cartoon images. From our measurements we estimated the function that maps realistic gloss parameters to their perception in a stylized rendering. This mapping allows users of non-photorealistic rendering algorithms to predict the perception of gloss in their images. The inverse of this function exaggerates gloss properties to make the contrast between materials in a stylized image more faithful.

2.4 Automating Artistic Principles

Our analysis of design principles informs the implementation of algorithms and user interfaces to facilitate design tasks. The main challenge we face is to automate tedious aspects of the artistic workflow while preserving the strength of the tools that artists are familiar with. While some artistic principles can directly be automated, others need to offer a balance between automation and user control.

2.4.1 Direct Automation

Drawing from observation. In an ideal scenario, the design principles can directly be automated to assist users in their tasks. We have achieved such an automation in the interface we proposed for assisting the practice of drawing-from-observation techniques (Iarussi et al. 2013). Inspired by our survey of the drawing literature, we set the following design goals for our user interface:

- Encourage users to focus their attention on the model they need to observe, rather than their drawing.
- Help users to practice observation techniques proposed by the drawing literature. These techniques should allow users to identify the shapes and their relationships on a model and to structure their drawings.
- Support corrective feedback to help users understand their errors and refine their drawings.

We articulated our interface around *visual guides* that help users construct their drawing following the principles listed in the previous section. We automatically extracted these visual guides from a photograph using well known computer vision algorithms for image segmentation, skeletonization, feature point detection (Figure 2.13). Note that our guides do not aim to match the style of a particular artist but rather to capture the common idea of drawing from coarse to fine.

(a) Block-in (b) Skeleton (c) Regions (d) Sighting

Figure 2.13: We automatically extract drawing guides by using simple computer vision algorithms. Our guides mimic the ones used in traditional drawing, as shown in Figure 2.10(a,b,c,d).

Figure 2.14 shows our user interface, which is composed of two distinct areas that we display on two separate monitors. The *model* area shows the photograph, which acts as the model for the drawing task, while the *canvas* is the drawing area where the user interacts with the pen. We display the model on a vertical computer monitor and the canvas on a pen display, which mimics traditional drawing where the drawer alternates between observing the model and drawing on paper. We display visual guides in the model area to encourage users to observe the model before structuring their drawing by themselves. We also use computer vision algorithms to register the user drawing with the reference model. This registration allows us to provide corrective feedback by highlighting in the model alignments and equal proportions that are violated in the drawing.

Figure 2.14: Our interface for assisting drawing-from-observation is composed of two display areas (a): the model area with the photograph and the visual guides, and the canvas area with the tools and the user's drawing. The user has used the drop-down list of tools (4) to activate a coarse block-in guide. The block-in guide is displayed over the model in blue (2). The user has reproduced the block-in guide over the canvas in the corresponding blue layer (5) and used these construction lines as a scaffold to reproduce a detailed contour (1,3). We offer simple drawing tools including a pencil, a pen and a small and big eraser (6). Our system registers the drawing in the active layer — block-in in this example — to estimate distortions (b) and shows on the model the erroneous alignments and proportions (c). In this example, the red dashed line shows a vertical alignment that has not been respected by the user and the dark blue segments show two distances that should be made equal. (Iarussi et al. 2013)

Material depiction. We also performed a direct automation of artistic principles in our work on Vector Shade Trees (Lopez-Moreno et al. 2013). This work, aimed at novices and professionals alike, strives to ease the task of depicting materials in vector graphics while preserving the traditional workflow of vector artists. Using our guideline classification we derived a compact yet expressive set of *shade nodes* that bring to vector graphics the flexibility of modular shading representations as known in the 3D rendering community, as illustrated in Figure 2.15. In contrast to traditional shade trees that combine pixel and vertex shaders, our shade nodes encapsulate the creation and blending of vector primitives that vector artists routinely use, such as gradient fills, gradient meshes and paths. We provide high-level user controls over each node, such as light direction or elevation of horizon line, as well as finer level controls using existing vector drawing tools.

Our approach automates the process of stylized material depiction for vector art and allows inexperienced users to take a line drawing and obtain a fully colored figure by applying a few clicks. Our solution also allows more experienced artists who are familiar with vector graphics software to easily refine the illustration, adding more details and visual features using the tools they already know. Figure 2.16 provides illustrations created by users with and without artistic training.

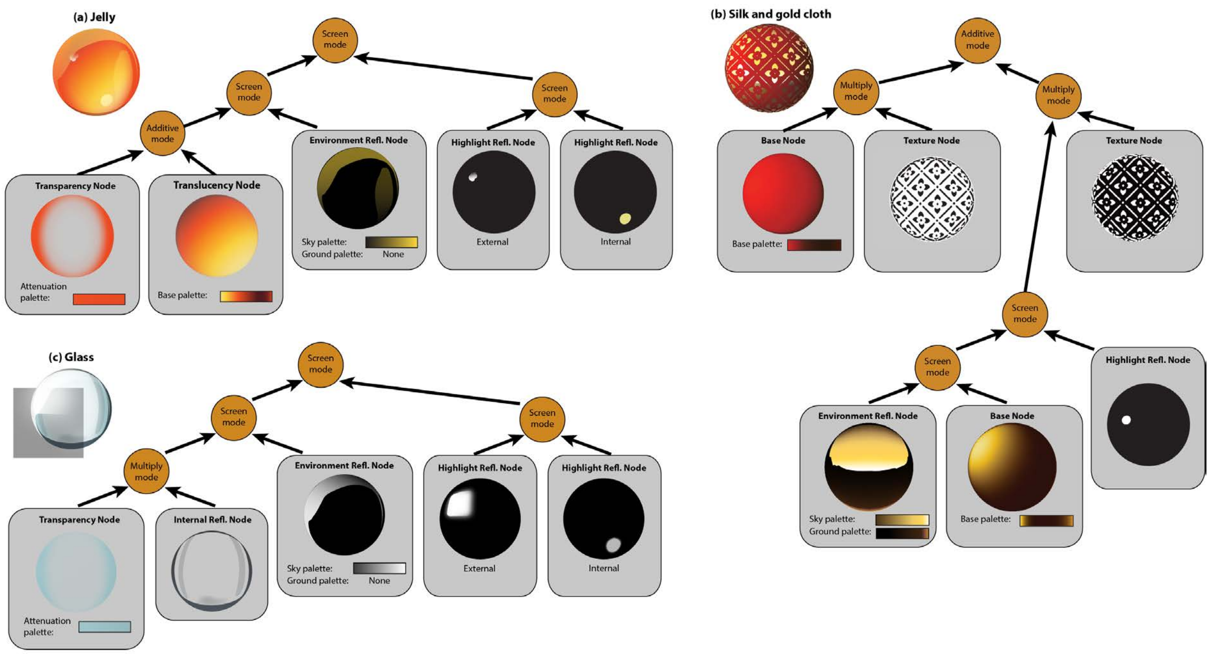


Figure 2.15: Vector Shade Trees for three materials: jelly, silk and gold cloth and glass, obtained by combining nodes for transparency, translucency, reflections, highlights, textures, base shading. Each node generates vector shapes filled with color gradients. (Lopez-Moreno et al. 2013)

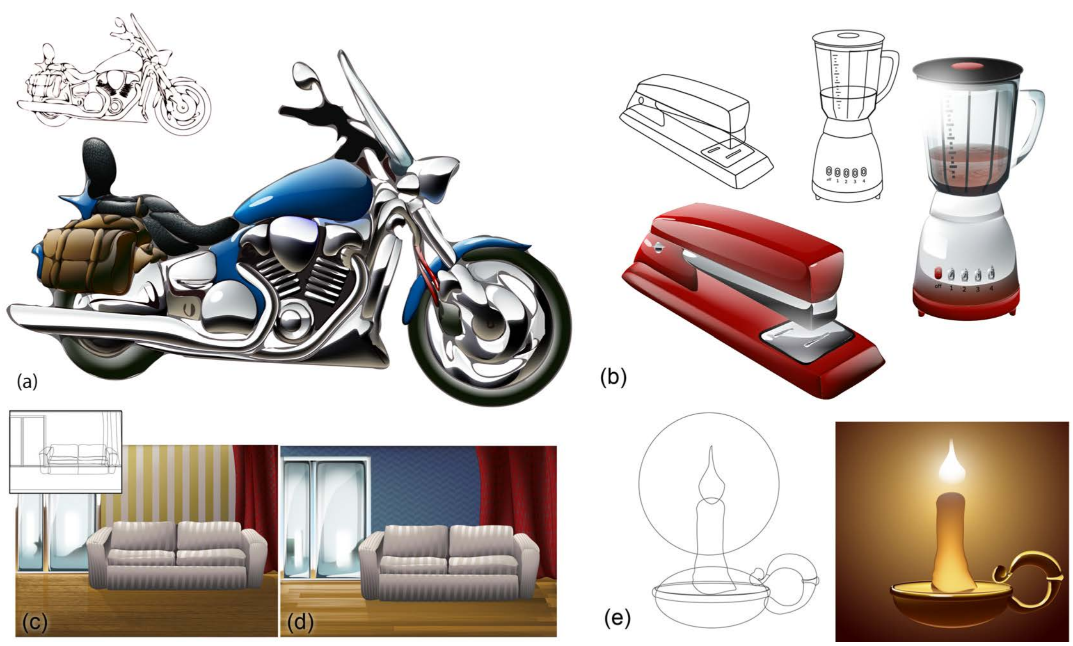


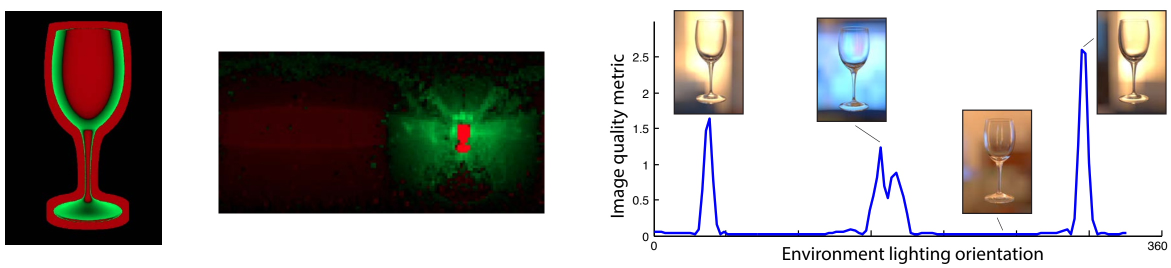
Figure 2.16: Different drawings colored with vector shade trees by an artist (a,c,d,e) and a novice user (b). The flame in (e) was added with standard vector graphics tools. (Lopez-Moreno et al. 2013)

2.4.2 Optimization

In many cases, the design principles translate in multiple, possibly concurrent goals. We typically express these goals as objectives in an optimization, which can also incorporate user input for guidance.

Lighting design. Lighting designers strategically position lights to delineate the outline of objects and reveal their gloss, transparency or translucency. Since shading, shadows and highlights concurrently affect these different objectives, we expressed the problem of lighting design as an optimization, aiming for the lighting environment that best distributes light and shade in the image according to shape and material-specific energy terms (Bousseau et al. 2011).

Figure 2.17a illustrates an energy term that measures contrast along the contours of transparent objects (green and red pixels represent positive and negative weights respectively). Using pre-computed light transport we deduce the areas of the lighting environment that contribute to contrast at contours (Figure 2.17b). Our optimization then searches for the orientation of an existing environment that maximizes contrast (Figure 2.17c), which can be efficiently evaluated in lighting space as a dot product between the environment map and the vector of weights. Using the same approach we can also solve for an artificial environment map that maximizes the metric.



(a) Image-space contrast measure

(b) Lighting-space contrast measure

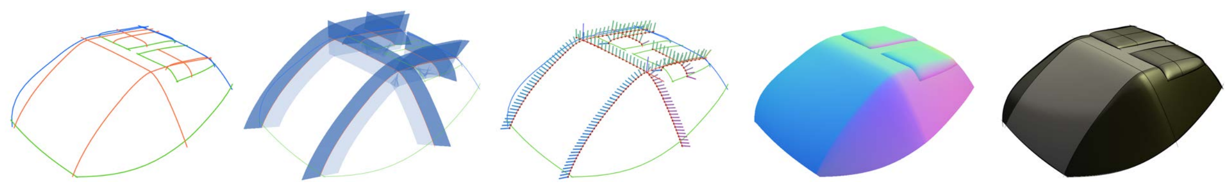
(c) Search for optimal orientation of the lighting

Figure 2.17: We express lighting design guidelines as image-space metrics, which we transpose to lighting space using pre-computed light transport. In this example, green pixels have positive weights and red pixels have negative weights. A dot product between these weights and the environment map gives a measure of contrast along contours. We then search for the best lighting orientation according to this metric.

3D shapes from line drawings. Optimization methods are also well suited for interactive tools where the design principles act as constraints that complement the often ambiguous user inputs. We adopted this approach in our work on sketch-based modeling (Shao et al. 2012; Xu et al. 2014), where we deduce strong regularity cues about a shape from lines that designers draw to convey curvature and local symmetry. These cues translate into orthogonality and parallelism constraints on the intended 3D curves. Our optimizations solve for the 3D curves that best satisfy these constraints while projecting well on the input 2D curves. Figure 2.18 shows the main steps of our CrossShade algorithm that follows this approach to estimate normal fields for shading.

One challenge we faced in estimating 3D shapes from line drawings is that many of the regularity cues are inherently context-based and should only be applied when consistent with each other. In addition, designers' drawings often contain distortions and inaccuracies. Our first algorithm, called *CrossShade* (Shao et al. 2012), treats the regularity cues as soft or bounded constraints in the optimization to allow for inaccuracy and weak cues. While this approach proved to be sufficient to estimate surface normals, it is not accurate enough to produce faithful 3D models.

Our second algorithm, called *True2Form* (Xu et al. 2014), applies the regularity cues selectively. We first generate an initial 3D reconstruction by enforcing orthogonality of cross-section curves, which is the cue in which we have most confidence. We next note that the degree to which a regularity cue is satisfied in this



(a) Input curves (b) Estimated curve planes (c) Normals along curves (d) Normals over the sketch (e) Resulting shading

Figure 2.18: Our CrossShade algorithm takes as input an annotated sketch (a), where orange curves denote cross-sections. We first optimize for the supporting plane of each cross-section and compute the corresponding 3D curves based on those (b). The supporting planes are constrained to be orthogonal to each other and to result in orthogonal curves with a preference for geodesics. We use the resulting 3D curves to compute 3D normals at each intersection and interpolate normals along the curves (c). We finally generate a normal field in between the curves using Coons' interpolation (d). (Shao et al. 2012)

reconstruction is strongly indicative of the likelihood of its applicability. However, thresholding the likelihood of all regularity cues at once results in inconsistent reconstructions as it ignores interconnections over the global network of curves. Instead, we cast regularity detection as a rounding problem, that progressively drives the continuous applicability likelihoods of individual cues to binary values. We express regularities as soft constraints weighted by their likelihood and progressively round likelihoods within an ϵ of 1 or 0, determining the regularities to be applicable or not. Each rounding yields a new 3D curve network that optimizes re-projection to the drawing subject to the regularity cues, where the resolved cues form hard constraints while the unresolved ones are soft and keep updating. The process is repeated until all regularity cues are resolved. Figure 2.19 illustrates the main iterations of this optimization that progressively detects and enforces the regularity cues. Our detection and strict enforcement of regularities is key to correcting the inevitable inaccuracy in sketches.

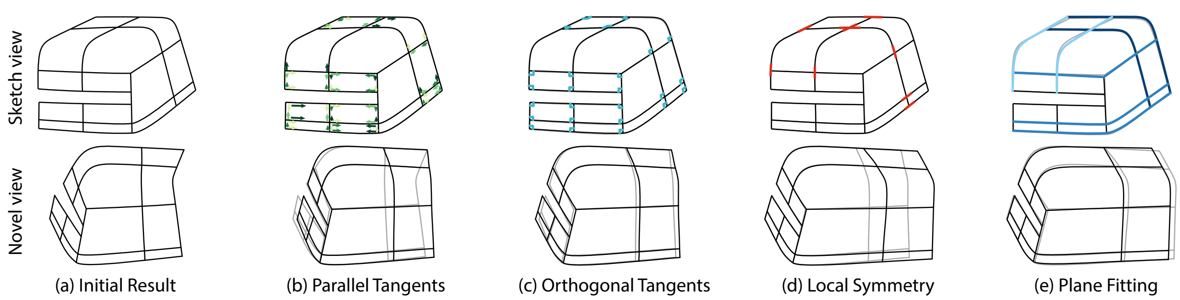


Figure 2.19: Our algorithm progressively detects and enforces a set of regularity cues to lift the sketch to 3D while correcting for drawing inaccuracy. (Xu et al. 2014)

Our CrossShade and True2Form algorithms both take clean vectorial curves as input. In a subsequent work we aimed at solving the same problem for rough bitmap sketches. While such sketches are more representative of the ones encountered in product design, they raise numerous challenges for interpretation since the curves are now drawn with multiple strokes and discretized over a pixel grid. State-of-the-art vectorization algorithms are also challenged by sketchy drawings (Noris et al. 2013) and as such cannot be used to convert rough sketches into curves suitable for CrossShade and True2Form. Our solution is to cast the problem of shape estimation from rough sketches as a scattered data interpolation of the curvature lines drawn in the sketch (Iarussi et al. 2015). To do so, we first formulated mathematical properties of principal directions of curvature when projected in image space. In particular, we derived that the two families of curvature lines that form the curvature network are smooth along each-other, which we measure using the so-called *covariant derivatives* of the corresponding vector fields. We incorporated this new measure of smoothness in an optimization that extrapolates a dense curvature network from the sparse curvature lines of a sketch, as illustrated in Figure 2.20. This geometric information allows us to estimate normals and texture coordinates from a single rough sketch. The main strength of this algorithm is that it is very robust to various levels of sketchiness of the sketch, as illustrated in Figure 2.21. However, this robustness comes at the cost of an expensive optimization, which takes several minutes with our current implementation.

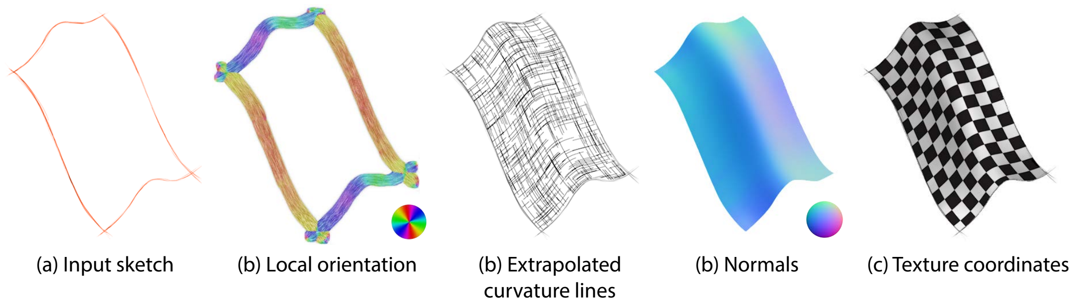


Figure 2.20: Our BendFields algorithm takes as input a bitmap sketch (a) from which we estimate the local orientation of the strokes (b). Assuming that the strokes represent curvature lines on a surface, our formulation extrapolates them as a non-orthogonal *cross field* that mimics the behavior of projected curvature lines (c). We lift the cross field to 3D by imposing orthogonality constraints (d). We finally compute normals and parameterization from the cross field (e,f). (Iarussi et al. 2015)

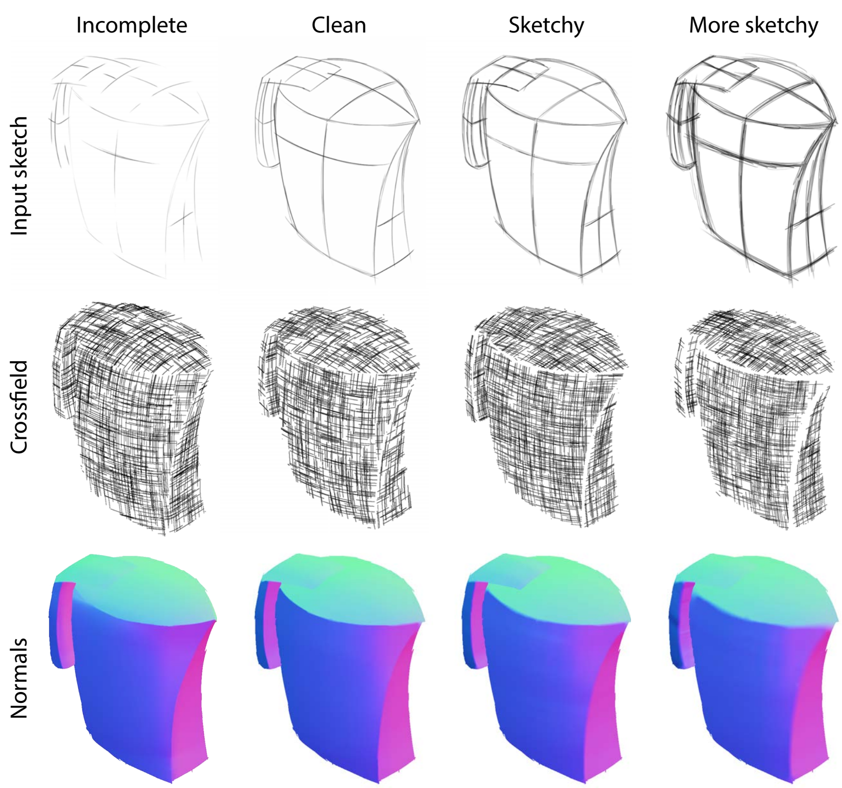


Figure 2.21: Our BendFields algorithm is robust to different levels of sketchiness, from sparse strokes with holes (left) to many overlapping strokes (right). Despite these drastic differences in input style, our method produces consistent curvature lines and normals. (Iarussi et al. 2015)

Layered vectorization. Finally, our work on image vectorization (Richardt et al. 2014) also relies on a user-guided optimization to convert a bitmap image into multiple semi-transparent vector layers. This work was motivated by the observation that skilled vector artists commonly blend multiple layers, each composed of simple color and transparency gradients, to represent the appearance of an object. Each layer typically corresponds to a single aspect of the shading, such as diffuse shading, specular highlights, shadows or Fresnel reflections. The challenge is to fully capture complex shading effects while also maintaining a small number of layers and gradient parameters so that the vector representation remains compact and easy to edit.

Our interactive vectorization algorithm assists the task of converting a bitmap into a small, editable set of vector layers. Users progressively select smooth color regions in the image, which our system separates into a foreground region filled with a vector gradient and a background bitmap layer, as illustrated in Figure 2.22. Our decompositing algorithm exploits the parametric nature of vector gradients to jointly separate and vectorize semi-transparent layers. In particular, we constrain the foreground colors to vary according to linear or radial gradients, allowing us to solve for a small set of parameters per layer instead of the thousands of unknowns over all pixels in a region. We have integrated our decompositing algorithm in Adobe Photoshop and we export our gradients as Illustrator layers, allowing vector artists to create and edit semi-transparent layers with standard tools. Figure 2.23 shows photographs and drawings vectorized with our approach.

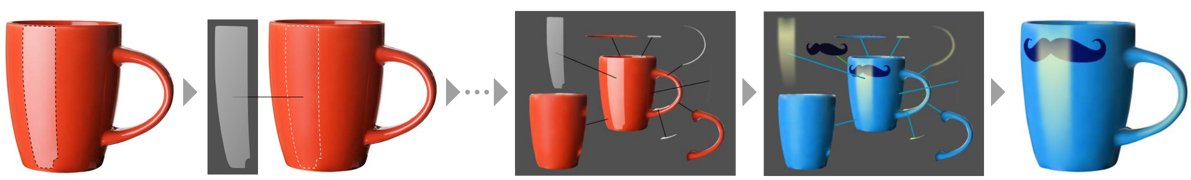


Figure 2.22: Interactive workflow for image vectorization. A user selects a region in an image (a), which our algorithm decomposites into a vector foreground layer and a background bitmap (b). This process is repeated and opaque layers are handled with existing tools to create a complete set of vector layers (c). Layers can be edited easily (shown as blue lines in d) and added (green lines), and re-composited to enable powerful editing applications (e). (Richardt et al. 2014)

2.5 Evaluation

The last step of our methodology consists in evaluating the effectiveness of the tools and algorithms we propose for computer-assisted drawing tasks. Given that our research aims at facilitating the work of artists, one could think that it is sufficient to let artists use our prototypes and give feedback. However, collecting informal opinions in a meaningful way is as difficult as running studies to understand how artists work. Hertzmann discusses several common pitfalls when running such evaluations (Hertzmann 2010), such as the fact that different artists often have very different tastes and preferences, or that people often try to please the questioner rather than provide an honest opinion. Finally, the user experience can be negatively impacted by the clunky interface of research prototypes, which is often beyond the scope of our algorithmic contributions.

For these reasons, we often run user studies where we complement subjective questionnaires with objective tasks in order to compare what people produce with our tool and a baseline method. When possible, we also compare our results with a ground truth reference or with existing imagery.

2.5.1 User Studies

Figure 2.24 illustrates the results of a small study where six trained artists were asked to draw materials in a vector drawing, both using our Vector Shade Trees (Lopez-Moreno et al. 2013) and manually using standard Illustrator functionality. In the instructions, we provided images of stylized materials which they should try to reproduce.

Figure 2.23: Decompositing and editing results of our interactive vectorization. For each result, we show the input photo or drawing on the left, and on the right an exploded view of all layers as well as editing results for some of the images. We added outlines to semi-transparent layers for visualization; these lines are not part of our results. We use blue lines to show edited layers, green lines for added layers, and the re-composited result in the center. (Richardt et al. 2014)

Our tool allowed users to achieve results of a comparable quality to the baseline solution, in a fraction of the time. User 1 took approximately the same time for both, yet the visual quality with our plugin is much richer. Feedback from these users suggests that achieving a quality similar to our results using only standard vector drawing tools is very difficult and requires great expertise. The standard process first requires the observation of the desired materials, or previous training on depicting them, and then rendering the materials using a combination of vector primitives. Our tool automates this process to quickly provide an initial depiction that artists can refine. One user claimed that a major benefit of our tool compared to the standard vector graphics workflow was the predictability of the result, which avoids tedious trial and error.

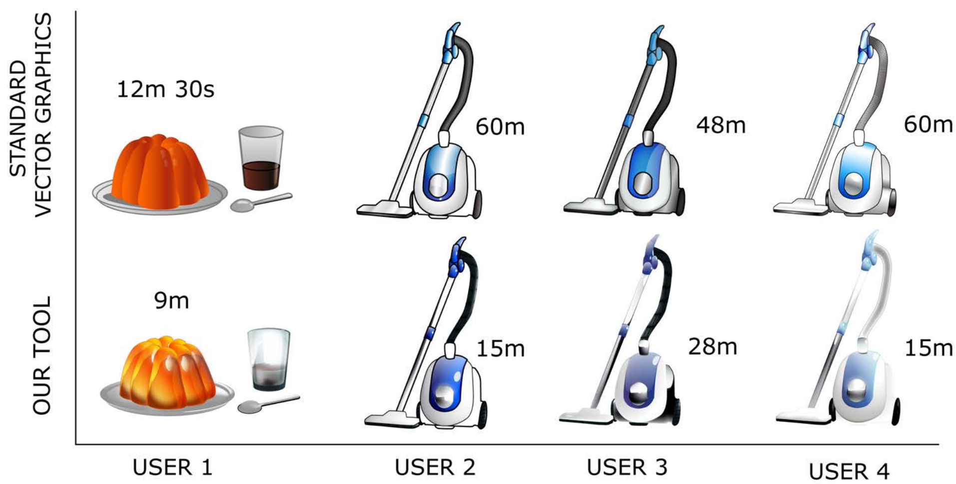


Figure 2.24: Images created using both our Vector Shade Trees and standard vector graphics tools and guidelines. The corresponding times for completion are shown with each illustration. (Lopez-Moreno et al. 2013)

Figure 2.25 presents the results of a more formal study evaluating the benefits of our drawing assistant for a drawing-from-observation task (Iarussi et al. 2013). We asked eight participants to produce line drawings from model photographs. Each participant was exposed to two versions of the drawing interface: our interface with guidance and a base interface with no guidance. The order of their presentation was counterbalanced among participants, i.e., four participants started with the base interface, and four participants started with the guided one. After the end of the session, participants completed a questionnaire to evaluate their experience with the tool.

We used a contour-matching algorithm to compute the mean error between the drawing and the reference. This error provides an objective measure of the overall distortion of the drawing that we aim to correct for. Most participants produced better drawings using our tool, with more accurate proportions and alignments. A notable exception is Participant 5, who reported having previous training in drawing and explained *“The guide is very clear and I understood quickly what I was supposed to do [but] following the guide was hard for me because I’m used to drawing in a different way. I think this is very useful for a beginner.”* Our system was more effective for participants 1, 6 and 7 that had poor drawing experience. A close examination of their drawings without guidance reveals significant errors. For example, Participant 1 made the roller shoe too tall and the wheels too close apart and Participant 6 made the torso of the character too long, the right leg too short and the left leg too low. No such distortions appear in their guided drawing.

Overall, participants reported that guidance and corrective feedback helped them better understand how to draw: *“I discovered new concepts like regions and block-in. I learned a fairly easy technique to improve my drawings.”* Interestingly, participants who were first exposed to our drawing tool tried to apply the techniques practiced through the first task to the second non-guided task, despite the fact that the base interface did not

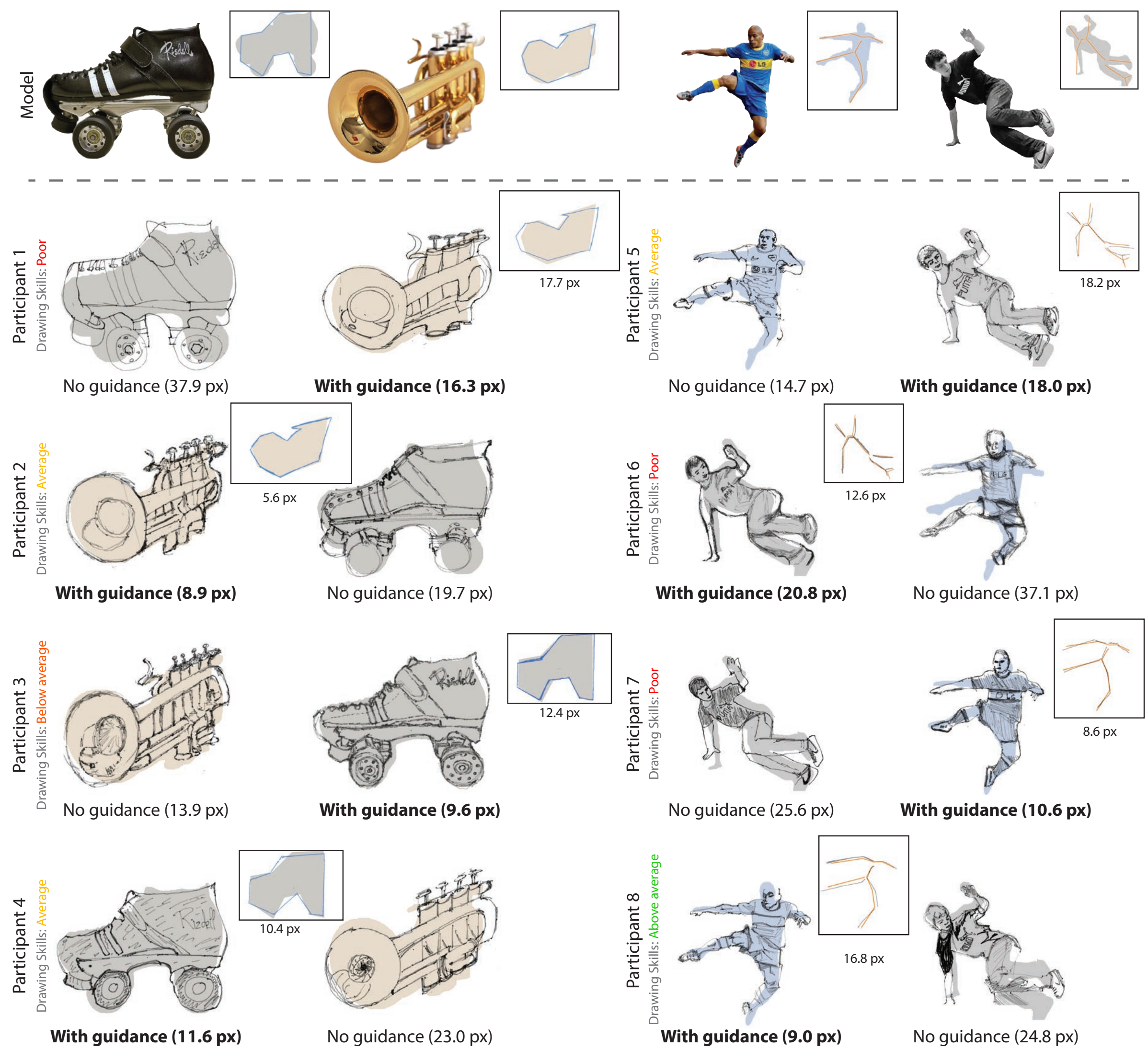


Figure 2.25: Drawings produced by the eight participants, with and without our automated assistance, in the order of completion. We provide as inset the drawn construction lines for the drawings performed with our tool. We display under each drawing the average error of the main contour, in pixels. (Iarussi et al. 2013)

encourage their use: *"I could apply the methods on my second drawing, and I think they were very useful to better reproduce the photo. I understood clearly the interest of the explained method."* Finally, some participants identified limitations and proposed areas for improvement. A participant observed that the *"first stages of the process provided a lot of help for learning to draw the volumes with right proportions"* but commented that she *"had some problems adding details"* as this stage of drawing was *"less assisted"*. Other participants pointed to the lack of support for different drawing habits, especially assistance adapted to more experienced drawers.

2.5.2 Comparisons to Ground Truth and Existing Imagery

Our work on sketch-based modeling is more amenable to ground-truth comparisons as we can use an existing 3D model to generate a sketch from which our algorithms reconstruct the 3D information. Figure 2.26 shows the normal field that our CrossShade algorithm produces on simple geometric shapes, along with ground truth normals and a naive normal diffusion from silhouettes (Shao et al. 2012). We measured a median difference at curve intersections of less than 3 degrees between our result and ground truth. Figure 2.27 provides a similar evaluation for the True2Form algorithm (Xu et al. 2014), where we visualize our 3D reconstruction from a different viewpoint than the sketch input. In addition to the ground truth 3D curves, we also show the 3D models and line drawings that artists produced when asked to represent the 3D shape they perceive in the input sketch. Qualitatively, ground truth, artists' 3D models, drawings and our results are visually similar in the view orthogonal to the input sketch.

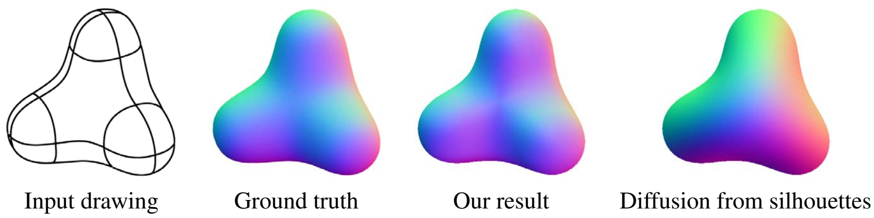


Figure 2.26: Comparison between CrossShade normals, ground truth normals and normals from silhouettes (Johnston 2002). Our normal field matches ground truth closely. Small deviations from ground truth appear in the center of the shape where midpoint subdivision is applied. (Shao et al. 2012)

Finally, we also evaluate our methods by applying them to reproduce existing traditional imagery. Such visual comparisons show how well our tools perform on the content that artists are interested in. As an example, Figure 2.28 contains 3D reconstructions that we obtained by tracing curves over the lines of existing concept sketches. Note that most of the lines we used were already present in the existing drawing, which suggests that our True2Form algorithm is well adapted to traditional practice.

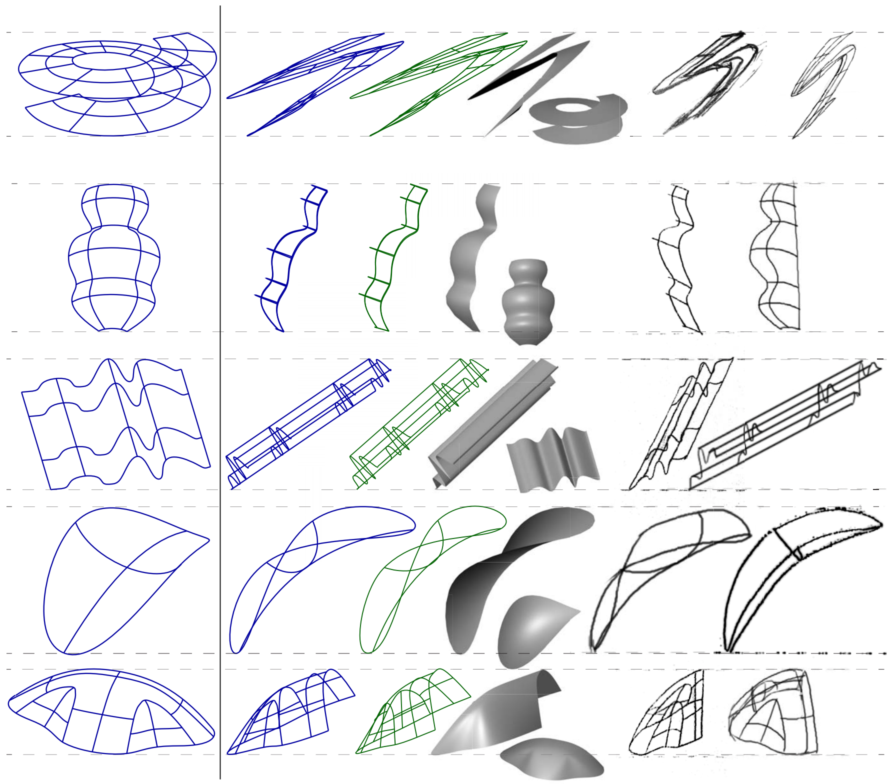


Figure 2.27: The curves in the extreme left column (ground truth) were shown to artists as the input sketch. The other columns show a view orthogonal to the sketch to illustrate perceived depth. From left to right we show ground truth curves (blue), our algorithmic output (green), artist modeled 3D surfaces and alternate view curves sketched by two artists. (Xu et al. 2014)

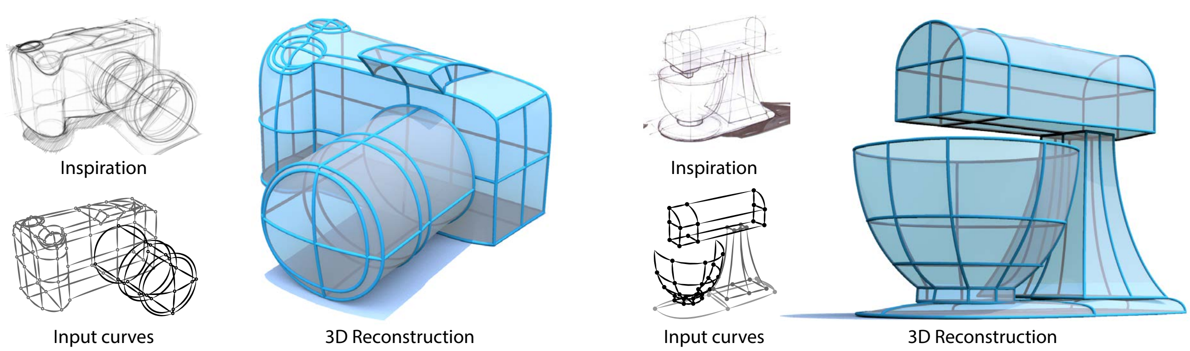


Figure 2.28: Our single-view modeling system allows us to reconstruct 3D models by tracing curves over existing sketches and photographs. Camera sketch by Spencer Nugent on www.sketch-a-day.com, mixer sketch from (Eissen and Steur 2008). (Xu et al. 2014)

Chapter 3

Multi-View Intrinsic Images and Relighting

This chapter covers my research on recovering material and lighting information from multiple photographs of a scene, and the application of such a decomposition to image relighting. I first briefly describe the methods I worked on and their relation to previous work. I then describe in more detail the decomposition model we used – called *intrinsic images*, the methods we developed to treat scenes captured under fixed or varying lighting, and how we performed relighting from the estimated quantities.

3.1 Overview and Related Work

Editing materials and lighting is a common image editing task that requires significant expertise to achieve consistent results. Since pixel color records the interaction of light and materials at the surface of objects, simply changing the color of a pixel is likely to modify both the perceived material and lighting.

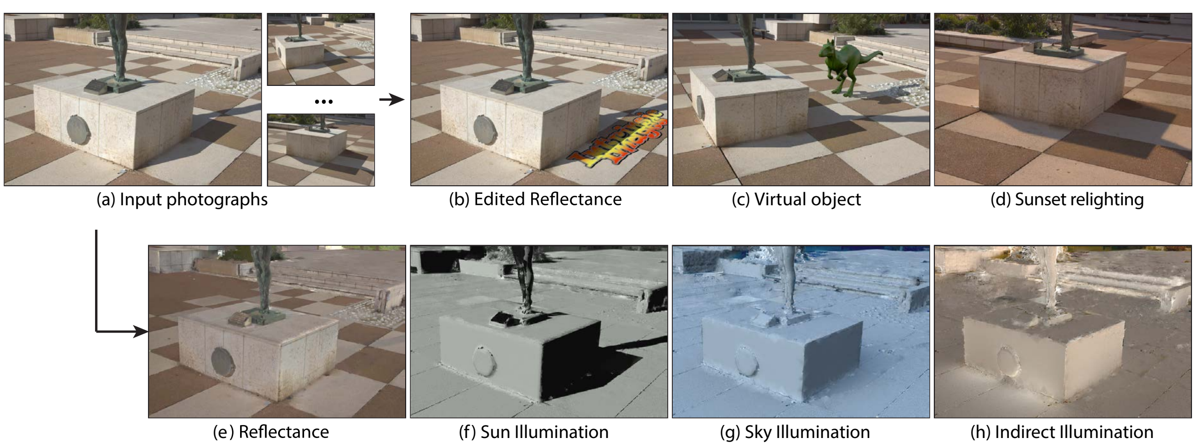


Figure 3.1: Starting from multiple views of the scene captured at the same time (a), our method decomposes photographs into four intrinsic components – the reflectance (e), the illumination due to sun (f), the illumination due to sky (g) and the indirect illumination (h). Each intrinsic component can then be manipulated independently for advanced image editing applications (b-d). (Laffont et al. 2013)

Under the assumption that the materials are diffuse, the pixel color can be expressed as the product of the lighting, also called shading, and the material color, also called reflectance. Reflectance and shading form the so-called *intrinsic layers* of the image. I have worked on several methods to estimate shading and reflectance layers from multiple images of a scene. A first group of approaches focuses on outdoor scenes for which all images are captured at the same time of day (Laffont et al. 2013; Duchêne et al. 2015). We first apply multi-view stereo reconstruction on the input images to recover a sparse 3D point cloud of the scene. We also recover an environment map of the sky as well as sun direction and intensity, either through user intervention (Laffont et al. 2013) or using automatic image analysis (Duchêne et al. 2015). While approximate, this information is sufficient to compute plausible diffuse shading due to sun, sky and indirect lighting. Once these values are computed, the only remaining unknowns are the reflectance of the materials and the visibility of the sun, i.e. shadows. We proposed two algorithms to estimate these unknowns, both consisting in solving for the values of sun visibility that result in a small number of reflectances in the scene. Since the 3D reconstruction is sparse, we use image-guided propagation to assign the estimated quantities to all pixels. Figure 3.1 shows results of our first algorithm (Laffont et al. 2013).

We have also proposed a method that exploits the multiple lighting conditions that are typically found in photocollections of touristic landmarks (Laffont et al. 2012). We again rely on a multi-view reconstruction of the scene, although we now use it to identify pairs of 3D points likely to receive the same illumination. We show that for such points, the ratio of their reflectance is equal to the ratio of their radiance, i.e. their pixel color. Combining such constraints within and across images allows us to recover a reflectance that is coherent over the photocollection, as shown in Figure 3.2.

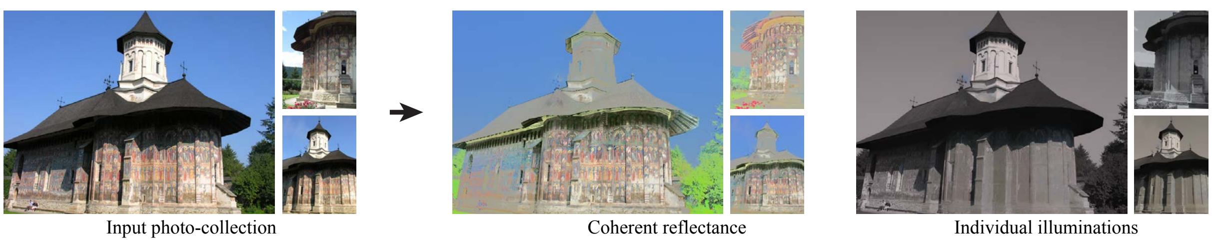


Figure 3.2: Our method leverages the heterogeneity of photo collections to automatically decompose photographs of a scene into reflectance and illumination layers. The extracted reflectance layers are coherent across all views, while the illumination captures the shading and shadow variations proper to each picture. Here we show the decomposition of three photos in the collection. (Laffont et al. 2012)

The three methods I proposed build on related work in the domain of computer vision and rendering. On the one hand, existing intrinsic image methods and shadow classifiers optimize for the best decomposition of an image given specific assumptions about materials, lighting and shadows in the scene. On the other hand, inverse rendering methods rely on precise geometry to perform lighting simulation. We build on the strengths of both types of methods to complement approximate geometric computation with image-guided regularization.

3.1.1 Inverse rendering and Relighting

A comprehensive survey of inverse rendering methods can be found in (Jacobs and Loscos 2006). Early work (Yu and Malik 1998; Yu et al. 1999; Loscos et al. 1999) required geometry which was of sufficient quality for pixel-accurate cast shadows; this is also true for more recent work (Debevec et al. 2004; Troccoli and Allen 2008) that relies on manually-constructed or scanned geometry. In contrast, we target the often imprecise and incomplete 3D geometry reconstructed from casual photographs by automatic algorithms. Haber et al. (Haber et al. 2009) estimate materials and distant illumination in 3D scenes reconstructed with multi-view stereo. However, as stated by the authors, manual intervention remains necessary to correct the geometry and ensure accurate visibility computation for shadow removal. In contrast, our methods combine approximate geometric computation and image-guided propagation to automatically estimate reflectance from incomplete 3D reconstructions, even when shadow casters are not observed in the input photographs. While our methods assume diffuse reflectance, they produce pixel-accurate decompositions that are well suited for image editing and image-based rendering.

3.1.2 Intrinsic images

While inverse rendering methods support complex material models, intrinsic image decompositions (Barrow and Tenenbaum 1978) assume that the materials are diffuse. The recent technical report of Barron and Malik (2013a) provides a good review of intrinsic image methods. Automatic single image methods rely on assumptions or classifiers on the statistics of reflectance and shading (Land and McCann 1971; Tappen et al. 2005; Shen and Yeo 2011; Zhao et al. 2012; Bell et al. 2014). In particular, most methods assume a sparse or piece-wise constant reflectance and smooth grey illumination – the so-called *Retinex* assumptions. Closer to our work is the method of Garces et al. (2012) who group pixels of similar chrominance to form clusters that are encouraged to share the same reflectance. These methods work best on objects captured under white lighting (Grosse et al. 2009) and on indoor scenes (Bell et al. 2014) but tend to fail on outdoor scenes where sun, sky and indirect illumination produce a mixture of colored lighting and cast shadows. Our algorithms for fixed lighting handle such cases by explicitly modeling the influence of sky and indirect illumination and by detecting shadow areas. Our method for internet photocollections relies on multiple lighting conditions to separate color variations due to lighting from color variations due to reflectance. User-assisted methods (Bousseau et al. 2009; Shen et al. 2011) can handle colored shading but would be cumbersome for the multi-view image sets we target.

Existing methods for multi-image datasets exploit multiple lighting conditions captured under a fixed or restricted viewpoint, typically from timelapse sequences (Weiss 2001; Sunkavalli et al. 2007) or photo-collections (Liu et al. 2008). We rely on multiview stereo algorithm to build a 3D reconstruction of the scene from multiple images, which allows us to relate images taken from an extended set of viewpoints. Geometric information has also been used to compute intrinsic images from image+depth data, where the additional depth information helps identifying points that have the same orientation and as such should share the same lighting (Lee et al. 2012; Barron and Malik 2013b; Chen and Koltun 2013). We follow a similar strategy when working with images taken under varying lighting conditions. In the case of a single lighting condition we also use the geometric information to perform lighting computation and initialize shadow detection.

3.1.3 Shadow detection

Shadow detection and removal have been studied extensively (Sanin et al. 2012) and most methods take a single image as input. Early approaches include automatic methods (e.g., (Finlayson et al. 2004)) which were demonstrated on images of uncluttered scenes with isolated shadows. More recent automated approaches include (Lalonde et al. 2010; Zhu et al. 2010; Panagopoulos et al. 2013; Guo et al. 2012). Similarly to these methods, our algorithms for fixed lighting detect shadows by identifying pairs of lit and shadow points sharing the same reflectance. However, existing work detects such pairs using machine learning (Guo et al. 2012) or by approximating shading and reflectance with brightness and hue (Panagopoulos et al. 2013). In contrast, we rely on multiple images to estimate an approximate 3D geometry, which we use to explicitly compute sun, sky and indirect lighting. This additional information provides us with more accurate estimation of reflectance values between pairs of points, which in turn yields more robust shadow classification.

3.2 Image Formation Model

The intrinsic images model assumes diffuse surfaces and expresses the image values $\mathbf{I}$ at each pixel as the product between the incident illumination $\mathbf{S}$ and the object reflectance $\mathbf{R}$. Formally, the radiance towards the camera at each non-emissive, visible point corresponding to a pixel is given by the equation

$$\mathbf{I} = \mathbf{R} * \int_{\Omega} \cos\theta_\omega \mathbf{L}(\omega) d\omega \tag{3.1}$$

$$\mathbf{I} = \mathbf{R} * \mathbf{S} \tag{3.2}$$

where lighting is integrated over the hemisphere Ω centered on the normal at the visible point. $\mathbf{L}(\omega)$ is the incoming radiance in direction ω, θ_ω is the angle between the normal at the visible point and direction ω. Capital bold letters represent RGB color values and $*$ denotes per-channel multiplication.

We can further separate out the incoming radiance into three components: the radiance due to the sun, that due to the sky and that due to indirect lighting. To simplify notation, we define two subsets of the hemisphere: Ω_{sky}, i.e., the subset of directions in which the visible point sees the sky, and Ω_{ind} the subset of directions in which another object is visible, and thus contributes to indirect lighting. We model the sun as a directional light source subject to the visibility term v_{sun} to obtain

$$\mathbf{I} = \mathbf{R} * \left(v_{sun} \cos\theta_{sun} \mathbf{L}_{sun} + \int_{\Omega_{sky}} \cos\theta_\omega \mathbf{L}_{sky}(\omega) d\omega + \int_{\Omega_{ind}} \cos\theta_\omega \mathbf{L}_{ind}(\omega) d\omega \right) \tag{3.3}$$

$$\mathbf{I} = \mathbf{R} * \left(v_{sun} \mathbf{S}_{sun} + \mathbf{S}_{sky} + \mathbf{S}_{ind} \right) \tag{3.4}$$

where $\mathbf{R}$ is the object RGB reflectance, $\mathbf{S}_{sun}$, $\mathbf{S}_{sky}$ and $\mathbf{S}_{ind}$ are the RGB incident illumination (or shading) from the sun, sky and indirect lighting respectively, v_{sun} indicates points visible from the sun and as such captures shadows.

3.3 Intrinsic Images under Fixed Lighting

In this section I describe two methods we have proposed to recover each component of Equation 3.4 when the scene is captured by multiple photographs taken at the same time of day. The first algorithm we have proposed takes as input the multiple images as well as an environment map of the scene captured with a chrome ball and a photographer's gray card that we use to calibrate sun color (Laffont et al. 2013). We improved this capture setup in our second algorithm, which automatically estimates the environment map from the input images and detects pairs of points of same reflectance in light and shadow to perform sun calibration (Duchêne et al. 2015).

Given the multiple images of the scene, the first step of our methods is to reconstruct a 3D model of the scene using multi-view stereo algorithms (Snavely et al. 2006; Furukawa and Ponce 2010). The resulting sparse 3D reconstruction only provides an imprecise and incomplete representation of the scene. Nevertheless, this reconstruction is sufficient to compute plausible sky and indirect illumination at each reconstructed 3D point. The coarse geometry is however unreliable for sun illumination because it typically contains high-frequency features due to cast shadows.

Our key observation is that sun visibility (i.e. shadows) can be estimated jointly with reflectance once all other unknowns have been computed. Given the now known $\mathbf{S}_{sun}$, $\mathbf{S}_{sky}$ and $\mathbf{S}_{ind}$, we rewrite Equation 3.4 to express reflectance as a function of sun visibility:

$$\mathbf{R}(v_{sun}) = \frac{\mathbf{I}}{v_{sun}\mathbf{S}_{sun} + \mathbf{S}_{sky} + \mathbf{S}_{ind}}. \tag{3.5}$$

With this formulation, each point now has multiple candidate reflectance values depending on the value of v_{sun}. A guiding principle of our approach is to select the value of v_{sun} that favors a small number of reflectances in the scene, a heuristic also used in prior work on intrinsic image decomposition and segmentation (Omer and Werman 2004; Barron and Malik 2013a; Bell et al. 2014).

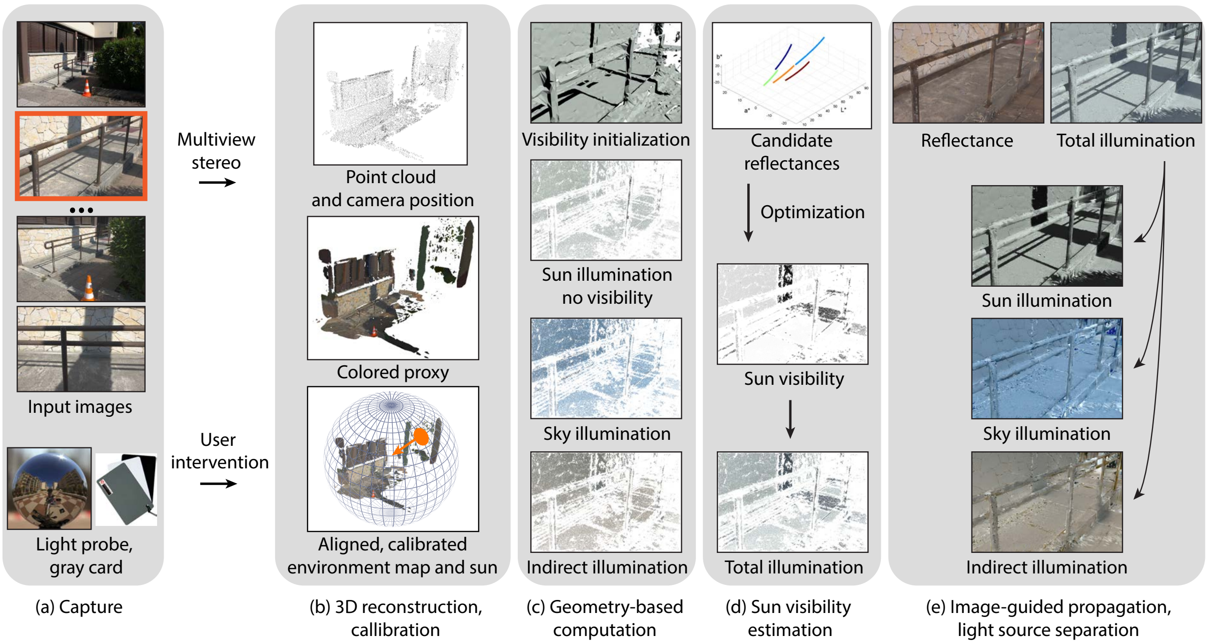


Figure 3.3: Overview of our first approach to decompose images captured under fixed lighting. Users capture a small set of pictures of the scene, along with an environment map and two pictures of a gray card in sun light and in shadow (a). We illustrate our intrinsic image decomposition with the picture highlighted in orange. We use multi-view stereo to reconstruct a point cloud of the scene and a coarse proxy geometry (b). Users align the environment map and the sun with this point cloud and use the gray card to calibrate their intensity. Once this calibration is performed, all the remaining steps are automatic. We use the reconstructed 3D geometry to compute sun, sky and indirect lighting over the point cloud (c). We also compute an initial guess of the sun visibility using the coarse proxy. These lighting values give us the necessary information to compute a set of candidate reflectances for each 3D point. The candidate reflectances form curves in color space parametrized by the sun visibility (d). We introduce an iterative optimization that identifies the reflectance of each 3D point from these candidates, along with an estimation of the sun visibility. The final step of our method consists in propagating the illumination values computed at 3D points to every pixel in the image (e). We decompose the illumination into the sun, sky and indirect lighting components. (Laffont et al. 2013)

The first algorithm we proposed (Laffont et al. 2013) solves for *continuous* values of $\mathbf{S}_{sky}$. In a nutshell, varying v_{sun} in $[0, 1]$ generates curves of candidate reflectances in color space and points of the scene sharing the same reflectance result in intersecting curves. We use an iterative mean-shift algorithm to progressively update sun visibility at each point to create clusters of points with similar reflectances. However, we observed that allowing v_{sun} to take continuous values tends to let this term absorb errors from all other terms. In our subsequent work we instead solve for a *binary* sun visibility (Duchêne et al. 2015). We solve the resulting binary labelling problem with a Markov Random Field formulation, where an energy term measures the likelihood of pairs of points to share the same reflectance under each of the four possible labelling configurations. Once sun visibility is estimated over the sparse 3D reconstruction, the last step of our approach consists in propagating illumination values to all pixels. To do so, we rely on image-guided propagation methods (Levin et al. 2008; Bousseau et al. 2009). These methods assign similar values to neighboring pixels sharing similar colors in the input image, which effectively preserves image discontinuities. In our context, these approaches ensure that the propagated values will respect object and shadow boundaries.

Figure 3.3 provides an overview of our first algorithm. Figure 3.4 shows the results of our second algorithm applied on a variety of outdoor scenes.

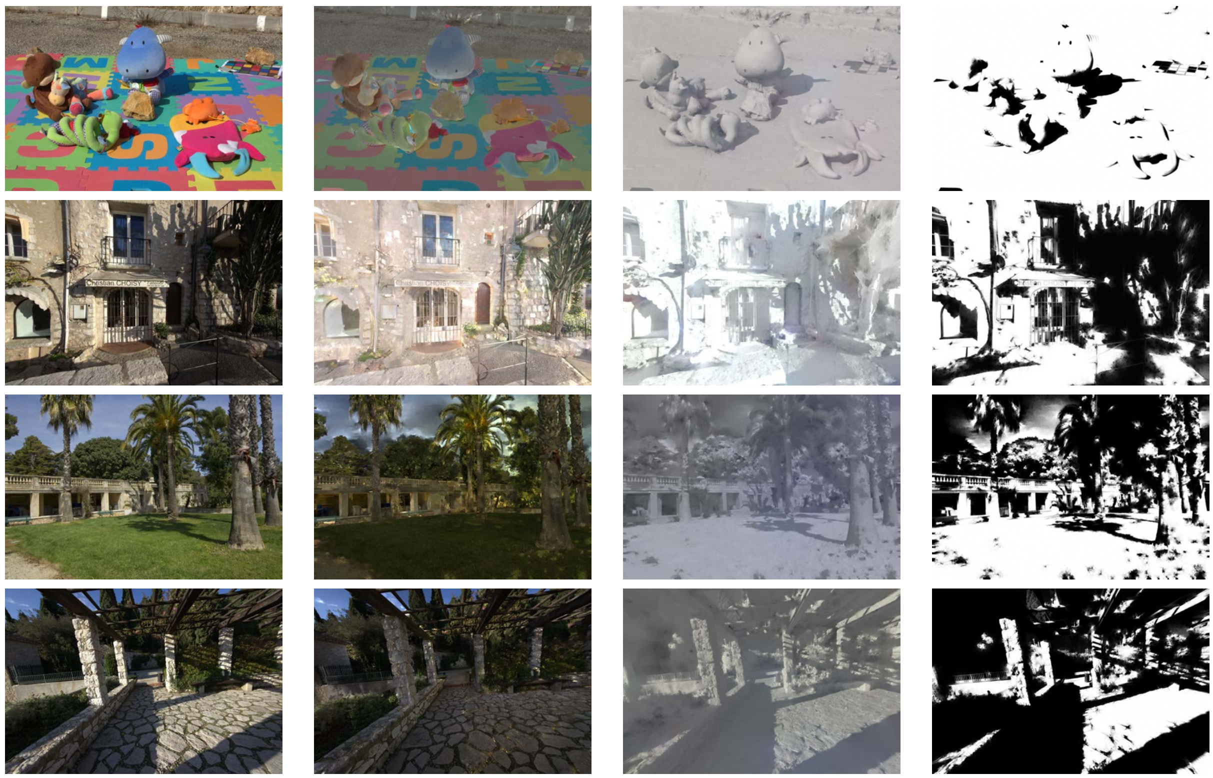


Figure 3.4: Our extracted layers on a variety of scenes: toys, urban (top), vegetation (middle), thin structures (bottom). (Duchêne et al. 2015)

3.4 Intrinsic Images under Varying Lighting

Photocollections provide multiple photographs of well-known touristic landmarks captured from different viewpoints and under varying illumination. The variation of illumination in a collection has often been seen as a nuisance that is distracting during navigation or, at best an interesting source of visual diversity. Inspired by existing work on time-lapse sequences (Weiss 2001) we proposed to exploit these variations as a rich source of information to identify the reflectance, that is constant over all images of a point, and the illumination, that varies across images (Laffont et al. 2012).

In this context, we use the sparse 3D reconstruction provided by multi-view stereo to relate scene points in different images. However, any triplet R,G,B is a possible reflectance solution for which the illumination of the point in each image is its pixel value divided by R,G,B. We overcome this difficulty by processing pairs of points likely to receive the same illumination. If two points p and q have the same normal and receive the same incoming radiance, then the difference between the observed radiances $\mathbf{I}(p)$ and $\mathbf{I}(q)$ are only due to variations between the scene reflectances $\mathbf{R}(p)$ and $\mathbf{R}(q)$. In such configurations, the ratio of reflectance between the two points is equal to the ratio of observed radiance:

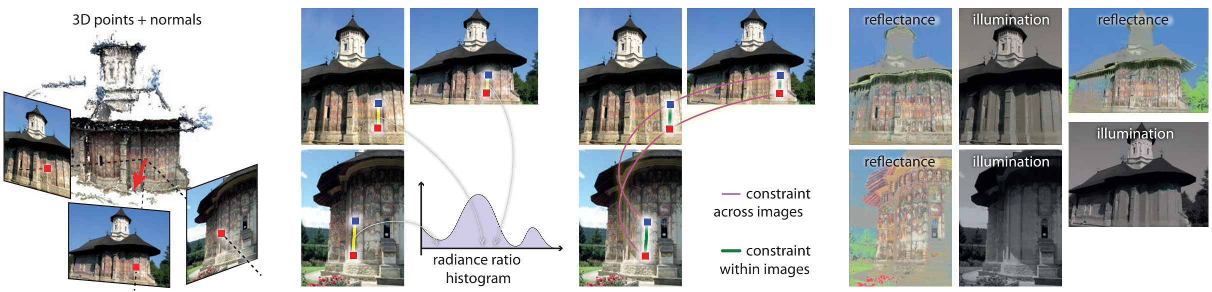
$$\mathbf{I}(p) = \mathbf{R}(p)\mathbf{S}, \quad \mathbf{I}(q) = \mathbf{R}(q)\mathbf{S}, \quad \frac{\mathbf{I}(p)}{\mathbf{I}(q)} = \frac{\mathbf{R}(p)}{\mathbf{R}(q)}. \tag{3.6}$$

From multiview stereo we have a normal estimate for each point, and it is straightforward to find points with similar normals. We next find an image where lighting conditions at p and q match. For points p and q which are close, the likelihood that a shadow boundary falls between them is low. Thus for most images in which these points are visible, the radiance ratio is equal to the reflectance ratio. However, lighting may still not match in a few images. Inspired by the work of Weiss (Weiss 2001) in the context of timelapse sequences, we use the median operator as a robust estimator to deal with the rare cases where two neighboring points are separated by a shadow boundary in some images

$$\frac{\mathbf{R}(p)}{\mathbf{R}(q)} = \text{median}\left(\frac{\mathbf{I}_i(p)}{\mathbf{I}_i(q)}\right) \tag{3.7}$$

where $\{\mathbf{I}_i\}$ is the set of images where both points are visible.

Similarly to the fixed lighting case, the constraints we obtain from pairs of 3D points are sparse and leave many pixels unconstrained. We again propagate information to all pixels using image-guided methods (Levin et al. 2008). Figure 3.5 shows the main components of our algorithm and Figure 3.6 provides result on several scenes.



(a) off-the-shelf multiview stereo (b) reflectance ratio inference (c) least-squares propagation within and across images (d) output illumination and reflectance

Figure 3.5: Our method for varying lighting infers reflectance ratios between points of a scene and then expresses the computation of illumination in all images in a unified least-square optimization system. (Laffont et al. 2012)

Figure 3.6: Our extracted layers on captured (left) and internet photographs (right). In each case, the first row contains the input image, the second row is the reflectance and the third row is the illumination. (Laffont et al. 2012)

3.5 Relighting Images

We designed our multi-view intrinsic decomposition in the goal of automatically extracting lighting from multiple images of the same scene. Given such multi-view datasets we then proposed several methods to allow relighting in the context of image-based rendering.

(a) - 2 hours (b) - 1 hour (c) Input image (d) + 1 hour (e) + 2.5 hours

Figure 3.7: Relit images for different times of day. While our method can produce drastic motions of the shadows (a-d), the shadow of the central tree breaks apart after a deviation of more than 2 hours (e). (Duchêne et al. 2015)

In the case of images captured under a fixed lighting condition, we can use our decomposition to remove the illumination of an image and replace it with the rendering of a different shading and shadows. However, the coarse 3D reconstruction estimated using multi-view stereo is often not sufficient to render convincing novel shadows. In particular, the 3D geometry often misses small details and thin structures. Instead, we approximate a detailed shadow caster from the shadow layer of our decomposition. The resulting caster preserves the original shadow boundaries in the input image and allows some motion of the sun. Figure 3.7 shows relighting results on an outdoor scene, where our method (Duchêne et al. 2015) achieves convincing displacement of the cast shadows over a sun trajectory of more than 2 hours. Figure 3.8 provides a visual comparison between our relighting and the ground truth images that we captured over an hour. Note that our synthetic shadows have the same color as the shadow in the central input image because only this input was used to estimate the sun color and sky model. In reality the appearance of the sky changed over time, which explains why the real shadow is darker in some pictures.

When dealing with datasets captured under varying lighting, we can use our decomposition to transfer illumination between two pictures of a scene taken from different viewpoints under different illumination. Figure 3.9 illustrates the main steps of this process. We use the sparse 3D reconstruction as a set of correspondences for which the illumination is known in the two images. We then transfer the illumination of one image to the other image using image-guided propagation. In areas visible only in the target view, the propagation interpolates the illumination values from the surrounding points visible in both images. We generate a radiance image by multiplying the reflectance with the transferred illumination. In Figure 3.9(e) we compare our illumination transfer with direct transfer of radiance. Propagating the radiance produces smooth color variations in-between the correspondences. In contrast, our combination of transferred illumination with the target reflectance preserves fine details. In Figure 3.10 we apply our approach to harmonize lighting for multiple viewpoints. This harmonization allows stable transition during image-based navigation for virtual tourism applications (Snavely et al. 2006). We also used our approach to generate artificial timelapse sequences synthesized by transferring all illumination conditions on a single viewpoint.

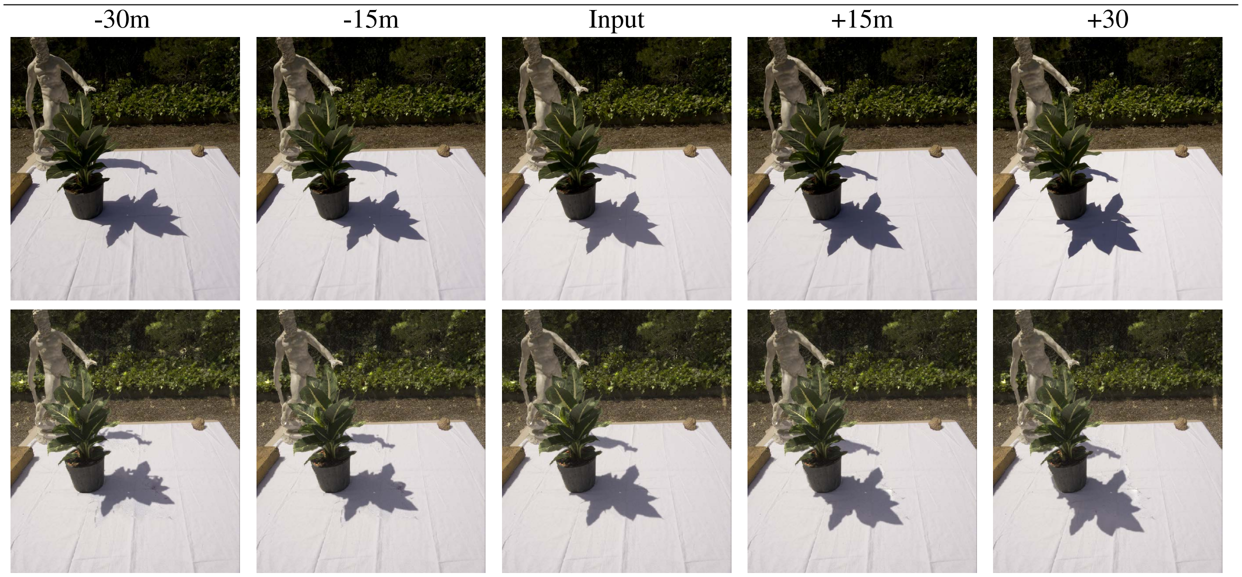


Figure 3.8: Above: real photographs taken at different times than those used for the algorithm. Below: relit images using our algorithm. (Duchêne et al. 2015)

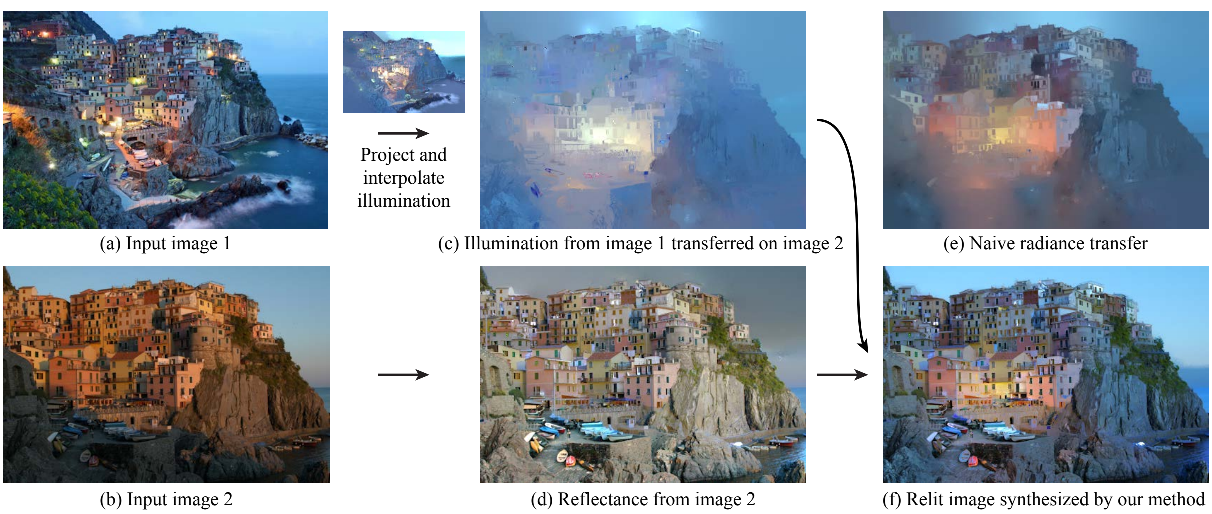
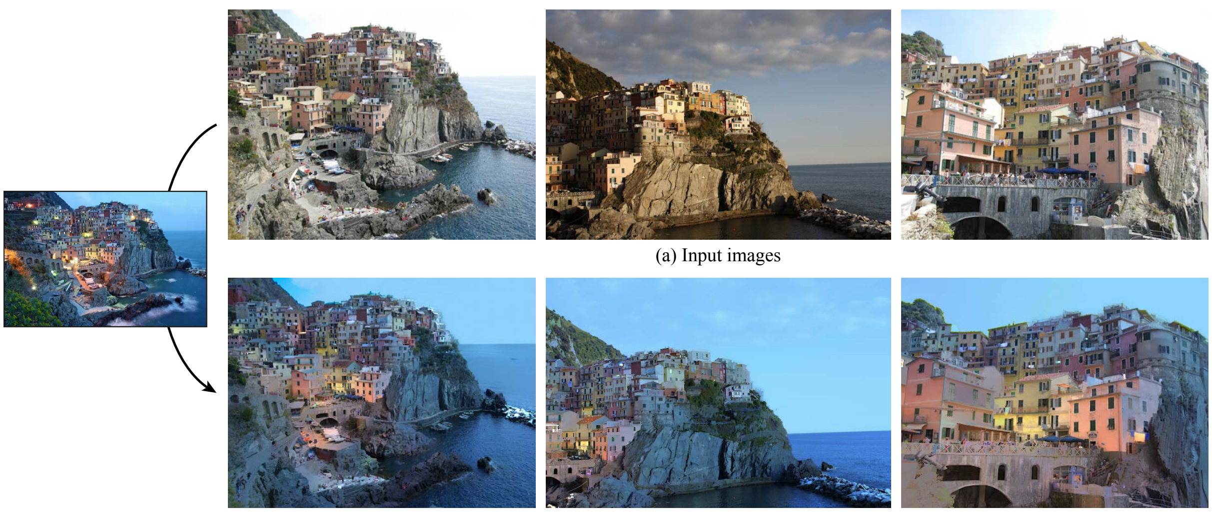


Figure 3.9: Given two views of the same scene under different lighting (a,b), we transfer the illumination from one view into the other view (c). We then multiply the transferred illumination by the reflectance layer (d) to synthesize the relit image (f). Transferring the radiance directly fails to preserve the fine details of the reflectance (e). (Laffont et al. 2012)



(a) Input images

(b) Relit images (illumination transferred from the lefthand image)

Figure 3.10: We use our lighting transfer to harmonize the illumination over multiple images of the same scene captured at different time of day. (Laffont et al. 2012)

Chapter 4

Conclusion and Future Work

I have presented my research on computer-assisted drawing and on image relighting. A significant part of this research has been motivated by observations on how people create images, from which I distilled and formalized common artistic techniques about shape, material and lighting depiction. Complementing these observations with knowledge in perception, graphics, vision and geometry allowed me to propose algorithms capable of assisting or automating tedious aspects of image creation. I strongly believe that the methodology I have described in this document can apply to other problems in content creation. In particular, I am interested in exploring the following research directions:

Physical prototyping. The advent of rapid prototyping technology such as 3D printers and laser cutters allows designers to quickly test the form and functionality of a concept in the physical world. However, these manufacturing technologies impose various restrictions on the type of shapes that can be created. In addition, designers often favor prototyping materials that are easy to manipulate but complex to assemble automatically, such as paper, cardboard, foam, metal wire and clay (Hallgrimsson 2012). Such rough prototypes form "physical sketches" that currently need to be converted into 3D models by hand. Inspired by my research on sketch-based modeling for product design, I plan to develop 3D reconstruction and modeling algorithms dedicated to the capture and editing of such physical prototypes. Traditional sculpture techniques, such as intermediate *scaffolds* and *jigs*, could also be automated to facilitate fabrication and assembly of prototypes.

Designers also often create new objects by considering the real context in which the object will be used. Such an application scenario suggests that context should be accounted for when interpreting design sketches. We recently proposed an algorithm in that spirit, which interprets architectural line drawings by exploiting information about the real 3D scene in which the building should integrate (Favreau et al. 2015).

Personal style. I have presented several sketch-based modeling algorithms that exploit the *cross-section lines* that designers draw to convey surface curvature. Similarly, other systems have been proposed to exploit *scaffolds* (Schmidt et al. 2009) and other forms of annotations (Gingold et al. 2009). While effective, these algorithms impose a specific drawing technique on users, at a stage where artists rather want to draw freely without external distraction. Instead of constraining users, I envision sketch-based modeling as a tool capable of conforming to the drawing techniques specific to each artist. I plan to explore the use of machine learning to predict the techniques used by a designer and adapt the 3D reconstruction algorithm accordingly. Similarly to speech recognition systems that require a training phase to adapt to the accent of an individual, users of such an approach would first train the system by drawing known shapes using their own techniques.

A major difficulty in developing a sketch-based modeling tool based on machine learning is to collect sufficient data to train the algorithms. Crowdsourcing is a common strategy to collect large training sets. While popular crowdsourcing websites like Amazon Mechanical Turk provide access to users with no training, dedicated websites such as Odesk, DesignCrowd and 99Designs could facilitate access to professionals.

Modeling variability. I have presented the True2Form algorithm (Xu et al. 2014) as a way to reconstruct a 3D model from a concept sketch. Similarly to prior work, our algorithm estimates the 3D model that best explains the drawing. However, designers often draw several sketches to convey multiple alternatives of a common concept. Each drawing can also be ambiguous and allow multiple interpretations. Because of the cost of 3D modeling, many of these alternatives are rejected as the concept undergoes further development. I wish to propose a sketch-based modeling tool capable of generating *distributions of objects* rather than solving for a unique solution. Such a probabilistic representation should allow designers to preserve multiple alternatives along all steps of the design process rather than committing to early decisions that impact all subsequent steps.

Manipulating appearance. We have presented initial solutions to the challenging problem of image relighting. However, our current solutions focus on outdoor scenes dominated by diffuse materials and are limited to a few hours of sun motion (Duchêne et al. 2015) or to the transfer of an existing lighting captured in another photograph (Laffont et al. 2012). Many challenges remain to be solved to handle a greater variety of material models and richer lighting setups as common in indoor scenes. In addition, while lighting greatly affects the mood of a scene, more dramatic temporal changes such as weather or seasons require modifying color and structure in an image. As a first step, we recently proposed to combine color and texture transfer to allow the synthesis of large temporal changes in photographs, such as converting a winter picture to summer or synthesizing the effects of snow storms and flooding (Okura et al. 2015).

References

AGRAWALA, M., AND STOLTE, C. 2001. Rendering effective route maps: Improving usability through generalization. *SIGGRAPH*, 241–249.

AGRAWALA, M., PHAN, D., HEISER, J., HAYMAKER, J., KLINGNER, J., HANRAHAN, P., AND TVERSKY, B. 2003. Designing effective step-by-step assembly instructions. *ACM Transactions on Graphics (Proc. SIGGRAPH) 22*, 3 (July), 828–837.

AGRAWALA, M., LI, W., AND BERTHOUZOZ, F. 2011. Design principles for visual communication. *Communications of the ACM 54*, 4 (Apr.), 60–69.

BAE, S., BALAKRISHNAN, R., AND SINGH, K. 2008. ILoveSketch: as-natural-as-possible sketching system for creating 3d curve models. In *Proc. User Interface Software and Technology*.

BARRON, J., AND MALIK, J. 2013. Shape, illumination, and reflectance from shading. Tech. rep., Berkeley Tech Report.

BARRON, J. T., AND MALIK, J. 2013. Intrinsic scene properties from a single RGB-D image. *CVPR*.

BARROW, H. G., AND TENENBAUM, J. M. 1978. Recovering intrinsic scene characteristics from images. *Computer Vision Systems 3*, 3–26.

BELL, S., BALA, K., AND SNAVELY, N. 2014. Intrinsic images in the wild. *ACM Transactions on Graphics (Proc. SIGGRAPH) 33*, 4.

BOUSSEAU, A., PARIS, S., AND DURAND, F. 2009. User assisted intrinsic images. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2009) 28*, 5.

BOUSSEAU, A., CHAPOULIE, E., RAMAMOORTHI, R., AND AGRAWALA, M. 2011. Optimizing environment maps for material depiction. *Computer Graphics Forum (Proc. Eurographics Symposium on Rendering) 30*, 4 (07).

BOUSSEAU, A., O SHEA, J. P., DURAND, F., RAMAMOORTHI, R., AND AGRAWALA, M. 2013. Gloss perception in painterly and cartoon rendering. *ACM Transactions on Graphics 32*, 2 (April).

BOUSSEAU, A., TSANDILAS, T., OEHLBERG, L., AND MACKAY, W. In Progress. Fabsketch: Studying how novices sketch, prototype and collaborate.

BRERETON, M., AND MCGARRY, B. 2000. An observational study of how objects support engineering design thinking and communication: Implications for the design of tangible media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, New York, NY, USA, CHI '00, 217–224.

CHAURASIA, G., DUCHÊNE, S., SORKINE-HORNUNG, O., AND DRETTAKIS, G. 2013. Depth synthesis and local warps for plausible image-based navigation. *ACM Transactions on Graphics 32*. to be presented at SIGGRAPH 2013.

CHEN, Q., AND KOLTUN, V. 2013. A simple model for intrinsic image decomposition with depth cues. In *ICCV*, IEEE.

COLE, F., SANIK, K., DECARLO, D., FINKELSTEIN, A., FUNKHOUSER, T., RUSINKIEWICZ, S., AND SINGH, M. 2009. How well do line drawings depict shape? In *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 28.

COOK, R. L. 1984. Shade trees. *SIGGRAPH*, 223–231.

CURTIS, C. J., ANDERSON, S. E., SEIMS, J. E., FLEISCHER, K. W., AND SALESIN, D. H. 1997. Computer-generated watercolor. *SIGGRAPH*, 421–430.

DEBEVEC, P., TCHOU, C., GARDNER, A., HAWKINS, T., POULLIS, C., STUMPFEL, J., JONES, A., YUN, N., EINARSSON, P., LUNDGREN, T., FAJARDO, M., AND MARTINEZ, P. 2004. Estimating surface reflectance properties of a complex scene under captured natural illumination. Tech. rep., USC Institute for Creative Technologies.

DIXON, D., PRASAD, M., AND HAMMOND, T. 2010. iCanDraw: Using sketch recognition and corrective feedback to assist a user in drawing human faces. In *Proc. of the International Conference on Human Factors in Computing Systems (CHI)*, ACM.

DODSON, B. 1985. *Keys to drawing*. North Light Books.

DUCHÊNE, S., RIANT, C., CHAURASIA, G., LOPEZ-MORENO, J., LAFFONT, P.-Y., POPOV, S., BOUSSEAU, A., AND DRETTAKIS, G. 2015. Multi-view intrinsic images of outdoors scenes with an application to relighting. *ACM Transactions on Graphics*.

EDWARDS, B. 1979. *Drawing on the right side of the brain*. Tarcher.

EISSEN, K., AND STEUR, R. 2008. *Sketching (5th Printing): Drawing Techniques for Product Designers*. Art and Design Series. Bis.

EISSEN, K., AND STEUR, R. 2011. *Sketching: The Basics*. Bis Publishers.

EITZ, M., HAYS, J., AND ALEXA, M. 2012. How do humans sketch objects? *ACM Transactions on Graphics (Proc. SIGGRAPH) 31*, 4, 44.

FAVREAU, J.-D., LAFARGE, F., AND BOUSSEAU, A. 2015. Line drawing interpretation in a multi-view context. In *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE.

FERNQUIST, J., GROSSMAN, T., AND FITZMAURICE, G. 2011. Sketch-sketch revolution: an engaging tutorial system for guided sketching and application learning. In *ACM Symp. on User Interface Software and Technology (UIST)*, 373–382.

FINLAYSON, G. D., DREW, M. S., AND LU, C. 2004. Intrinsic images by entropy minimization. In *ECCV*, 582–595.

FLAGG, M., AND REHG, J. 2006. Projector-guided painting. In *ACM Symp. on User Interface Software and Technology (UIST)*, 235–244.

FLEMING, R. W., DROR, R. O., AND ADELSON, E. H. 2003. Real-world illumination and the perception of surface reflectance properties. *Journal of Vision 3*, 5.

FLEMING, R. W., JENSEN, H. W., AND BÜLTHOFF, H. H. 2004. Perceiving translucent materials. In *APGV '04*.

FURUKAWA, Y., AND PONCE, J. 2010. Accurate, dense, and robust multi-view stereopsis. *IEEE Trans. on Pattern Analysis and Machine Intelligence 32*, 8, 1362–1376.

GARCES, E., MUNOZ, A., LOPEZ-MORENO, J., AND GUTIERREZ, D. 2012. Intrinsic images by clustering. *Computer Graphics Forum (Proc. EGSR) 31*, 4.

GINGOLD, Y., IGARASHI, T., AND ZORIN, D. 2009. Structured annotations for 2D-to-3D modeling. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 28*, 5.

GOOCH, A. A., GOOCH, B., SHIRLEY, P., AND COHEN, E. 1998. A non-photorealistic lighting model for automatic technical illustration. *SIGGRAPH*, 447–452.

GRABLER, F., AGRAWALA, M., SUMNER, R. W., AND PAULY, M. 2008. Automatic generation of tourist maps. *ACM Transactions on Graphics (Proc. SIGGRAPH) 27*, 3 (Aug.), 100:1–100:11.

GRABLI, S., TURQUIN, E., DURAND, F., AND SILLION, F. X. 2010. Programmable rendering of line drawing from 3d scenes. *ACM Transactions on Graphics 29*, 2 (Apr.), 18:1–18:20.

GROSSE, R., JOHNSON, M. K., ADELSON, E. H., AND FREEMAN, W. T. 2009. Ground-truth dataset and baseline evaluations for intrinsic image algorithms. In *ICCV*.

GUAY, M., CANI, M.-P., AND RONFARD, R. 2013. The Line of Action: an Intuitive Interface for Expressive Character Posing. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia 32*, 6.

GUMHOLD, S. 2002. Maximum entropy light source placement. In *Proc. IEEE Visualization*.

GUO, R., DAI, Q., AND HOIEM, D. 2012. Paired regions for shadow detection and removal. *IEEE Transactions on Pattern Analysis and Machine Intelligence 99*.

HABER, T., FUCHS, C., BEKAER, P., SEIDEL, H.-P., GOESELE, M., AND LENSCH, H. 2009. Relighting objects from image collections. In *CVPR*, 627–634.

HALLGRIMSSON, B. 2012. *Prototyping and Modelmaking for Product Design*. Portfolio skills. Laurence King Publishers.

HENDERSON, K. 1998. *On Line and on Paper: Visual Representations, Visual Culture, and Computer Graphics in Design Engineering*. MIT Press, Cambridge, MA, USA.

HERTZMANN, A. 2010. Non-photorealistic rendering and the science of art. In *Proc. International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*.

HUNTER, F., BIVER, S., AND FUQUA, P. 2012. *Light Science and Magic*, fourth edition ed. Focal Press.

IARUSSI, E., BOUSSEAU, A., AND TSANDILAS, T. 2013. The drawing assistant: Automated drawing guidance and feedback from photographs. In *ACM Symposium on User Interface Software and Technology (UIST)*, ACM.

IARUSSI, E., BOMMES, D., AND BOUSSEAU, A. 2015. Bendfields: Regularized curvature fields from rough concept sketches.

IGARASHI, T., MATSUOKA, S., AND TANAKA, H. 1999. Teddy: a sketching interface for 3D freeform design. *Proc. SIGGRAPH*.

JACOBS, K., AND LOSCOS, C. 2006. Classification of illumination methods for mixed reality. In *Computer Graphics Forum*, vol. 25, Wiley Online Library, 29–51.

JOHNSTON, S. F. 2002. Lumo: illumination for cel animation. In *Proc. Symp. on Non-Photorealistic Animation and Rendering*.

KERR, W. B., AND PELLACINI, F. 2009. Toward evaluating lighting design interface paradigms for novice users. *ACM TOG (Proc. of SIGGRAPH) 28*.

KNILL, D. C. 1992. Perception of surface contours and surface shape: from computation to psychophysics. *Journal of Optical Society of America 9*, 9, 1449–1464.

LAFFONT, P.-Y., BOUSSEAU, A., PARIS, S., DURAND, F., AND DRETTAKIS, G. 2012. Coherent intrinsic images from photo collections. *ACM Transactions on Graphics (SIGGRAPH Asia Conference Proceedings) 31*.

LAFFONT, P.-Y., BOUSSEAU, A., AND DRETTAKIS, G. 2013. Rich intrinsic image decomposition of outdoor scenes from multiple views. *IEEE Transactions on Visualization and Computer Graphics 19*, 2 (February), 210 – 224.

LALONDE, J.-F., EFROS, A. A., AND NARASIMHAN, S. G. 2010. Detecting ground shadows in outdoor consumer photographs. In *European Conference on Computer Vision*.

LAND, E. H., AND MCCANN, J. J. 1971. Lightness and retinex theory. *Journal of the optical society of America 61*, 1.

LAVIOLE, J., AND HACHET, M. 2012. Papart: interactive 3d graphics and multi-touch augmented paper for artistic creaton. In *3DUI - IEEE Virtual Reality Conference*.

LECOT, G., AND LÉVY, B. 2006. Ardeco: automatic region detection and conversion. In *Eurographics Symposium on Rendering*.

LEE, C. H., HAO, X., AND VARSHNEY, A. 2006. Geometry-dependent lighting. *IEEE Trans. on Visualization and Computer Graphics 12*.

LEE, Y., ZITNICK, C., AND COHEN, M. 2011. Shadowdraw: real-time user guidance for freehand drawing. *ACM Trans. on Graphics (Proc. SIGGRAPH) 30*, 4, 27.

LEE, K. J., ZHAO, Q., TONG, X., GONG, M., IZADI, S., LEE, S. U., TAN, P., AND LIN, S. 2012. Estimation of intrinsic image sequences from image+depth video. In *Proceedings of European Conference on Computer Vision (ECCV)*, 327–340.

LEVIN, A., LISCHINSKI, D., AND WEISS, Y. 2008. A closed-form solution to natural image matting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 30*, 2, 228–242.

LI, W., RITTER, L., AGRAWALA, M., CURLESS, B., AND SALESIN, D. 2007. Interactive cutaway illustrations of complex 3d models. *ACM Transactions on Graphics (Proc. SIGGRAPH) 26*, 3 (July).

LI, W., AGRAWALA, M., CURLESS, B., AND SALESIN, D. 2008. Automated generation of interactive 3d exploded view diagrams. *ACM Transactions on Graphics (Proc. SIGGRAPH) 27*, 3 (Aug.), 101:1–101:7.

LIPSON, H., AND SHPITALNI, M. 1996. Optimization-based reconstruction of a 3d object from a single freehand line drawing. *Computer-Aided Design 28*, 651–663.

LIU, X., WAN, L., QU, Y., WONG, T.-T., LIN, S., LEUNG, C.-S., AND HENG, P.-A. 2008. Intrinsic colorization. *ACM Transactions on Graphics (proc. of SIGGRAPH Asia) 27*, 152:1–152:9.

LOPEZ-MORENO, J., POPOV, S., BOUSSEAU, A., AGRAWALA, M., AND DRETTAKIS, G. 2013. Depicting stylized materials with vector shade trees. *ACM Transactions on Graphics (Proc. SIGGRAPH) 32*, 4.

LOSCOS, C., FRASSON, M.-C., DRETTAKIS, G., WALTER, B., GRANIER, X., AND POULIN, P. 1999. Interactive virtual relighting and remodeling of real scenes. In *Proceedings of the 10th Eurographics conference on Rendering*, 329–340.

MAMASSIAN, P., AND LANDY, M. S. 1998. Observer biases in the 3D interpretation of line drawings. *Vision research 38*, 18, 2817–2832.

MARTIN, J. 1989. *Rendering Highlights*. Airbrush Artist's Library. North Light Books.

MARTIN, J. 1989. *Rendering Transparency*. Airbrush Artist's Library. North Light Books.

MATHER, G. 2008. *Foundations of sensation and perception*. Taylor and Francis.

MITRA, N. J., YANG, Y.-L., YAN, D.-M., LI, W., AND AGRAWALA, M. 2010. Illustrating how mechanical assemblies work. *ACM Transactions on Graphics (Proc. SIGGRAPH) 29*, 4 (July), 58:1–58:12.

NAKAYAMA, K., AND SHIMOJO, S. 1992. Experiencing and Perceiving Visual Surfaces. *Science 257*, 1357–1363.

NEALEN, A., IGARASHI, T., SORKINE, O., AND ALEXA, M. 2007. Fibermesh: designing freeform surfaces with 3d curves. *ACM Trans. Graph. 26*.

NICOLAIDES, K. 1969. *The Natural Way to Draw: A Working Plan for Art Study*. Houghton Mifflin Co.

NORIS, G., HORNUNG, A., SUMNER, R. W., SIMMONS, M., AND GROSS, M. 2013. Topology-driven vectorization of clean line drawings. *ACM Transactions on Graphics (TOG) 32*, 1, 4.

OKURA, F., VANHOEY, K., BOUSSEAU, A., EFROS, A. A., AND DRETTAKIS, G. 2015. Unifying color and texture transfer for predictive appearance manipulation. *Computer Graphics Forum (Proc. of the Eurographics Symposium on Rendering)*.

OLSEN, L., SAMAVATI, F., SOUSA, M., AND JORGE, J. 2009. Sketch-based modeling: A survey. *Computers & Graphics 33*.

OLSEN, L., SAMAVATI, F., AND JORGE, J. A. 2011. Naturasketch: Modeling from images and natural sketches. *IEEE Computer Graphics and Applications 31*, 6, 24–34.

OMER, I., AND WERMAN, M. 2004. Color lines: Image specific color representation. In *Proc. IEEE Computer Vision and Pattern Recognition*, IEEE Computer Society, Washington, DC, USA, 946–953.

ORZAN, A., BOUSSEAU, A., WINNEMÖLLER, H., BARLA, P., THOLLOT, J., AND SALESIN, D. 2008. Diffusion curves: a vector representation for smooth-shaded images. *ACM Transactions on Graphics (Proc. SIGGRAPH) 27*, 3, 92:1–8.

PANAGOPOULOS, A., WANG, C., SAMARAS, D., AND PARAGIOS, N. 2013. Simultaneous cast shadows, illumination and geometry inference using hypergraphs. IEEE, vol. 35, 437–449.

PIPES, A. 2007. *Drawing for designers*. Laurence King.

PIZLO, Z., AND STEVENSON, A. 1999. Shape constancy from novel views. *Perception & Psychophysics 61*, 7, 1299–1307.

POWELL, D. 1986. *Design rendering techniques: a guide to drawing and presenting design ideas*. North Light.

RAMANARAYANAN, G., FERWERDA, J., WALTER, B., AND BALA, K. 2007. Visual equivalence: Towards a new standard for image fidelity. *ACM Transactions on Graphics (Proc. SIGGRAPH) 26*, 3.

RICHARDT, C., LOPEZ-MORENO, J., BOUSSEAU, A., AGRAWALA, M., AND DRETTAKIS, G. 2014. Vectorising bitmaps into semi-transparent gradient layers. *Computer Graphics Forum (Proc. Eurographics Symposium on Rendering) 33*, 4.

ROBERTSON, S. 2003. *How to Draw Cars the Hot Wheels Way*. MBI.

SANIN, A., SANDERSON, C., AND LOVELL, B. C. 2012. Shadow detection: A survey and comparative evaluation of recent methods. *Pattern recognition 45*, 4, 1684–1695.

SCHMIDT, R., KHAN, A., SINGH, K., AND KURTENBACH, G. 2009. Analytic drawing of 3d scaffolds. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 28*, 5.

SHACKED, R., AND LISCHINSKI, D. 2001. Automatic lighting design using a perceptual quality metric. *Computer Graphics Forum 20*.

SHAO, C., BOUSSEAU, A., SHEFFER, A., AND SINGH, K. 2012. Crossshade: Shading concept sketches using cross-section curves. *ACM Transactions on Graphics (Proc. SIGGRAPH) 31*, 4.

SHEN, L., AND YEO, C. 2011. Intrinsic images decomposition using a local and global sparse representation of reflectance. In *CVPR*, 697–704.

SHEN, J., YANG, X., JIA, Y., AND LI, X. 2011. Intrinsic images using optimization. In *CVPR*.

SMITH, A. R., AND BLINN, J. F. 1996. Blue screen matting. 259–268.

SNAVELY, N., SEITZ, S. M., AND SZELISKI, R. 2006. Photo tourism: Exploring photo collections in 3d. In *SIGGRAPH Conference Proceedings*, ACM Press, New York, NY, USA, 835–846.

STEVENS, K. A. 1981. The visual interpretation of surface contours. *Artificial Intelligence 17*.

SUN, J., LIANG, L., WEN, F., AND SHUM, H.-Y. 2007. Image vectorization using optimized gradient meshes. *ACM Transactions on Graphics (Proc. SIGGRAPH) 26*, 3, 11.

SUNKAVALLI, K., MATUSIK, W., PFISTER, H., AND RUSINKIEWICZ, S. 2007. Factored time-lapse video. *ACM Transactions on Graphics (proc. of SIGGRAPH) 26*, 3.

SÝKORA, D., KAVAN, L., ČADÍK, M., JAMRIŠKA, O., JACOBSON, A., WHITED, B., SIMMONS, M., AND SORKINE-HORNUNG, O. 2014. Ink-and-ray: Bas-relief meshes for adding global illumination effects to hand-drawn characters. *ACM Trans. Graphics 33*.

TAPPEN, M. F., FREEMAN, W. T., AND ADELSON, E. H. 2005. Recovering intrinsic images from a single image. *IEEE Trans. PAMI 27*, 9.

TROCCOLI, A., AND ALLEN, P. 2008. Building illumination coherent 3d models of large-scale outdoor scenes. *International Journal of Computer Vision 78*, 2-3, 261–280.

WEISS, Y. 2001. Deriving intrinsic images from image sequences. In *ICCV*, 68–75.

WINKENBACH, G., AND SALESIN, D. H. 1994. Computer-generated pen-and-ink illustration. *SIGGRAPH*, 91–100.

XU, B., CHANG, W., SHEFFER, A., BOUSSEAU, A., MCCRAE, J., AND SINGH, K. 2014. True2form: 3d curve networks from 2d sketches via selective regularization. *Transactions on Graphics (Proc. SIGGRAPH) 33*, 4.

YU, Y., AND MALIK, J. 1998. Recovering photometric properties of architectural scenes from photographs. In *SIGGRAPH*, 207–217.

YU, Y., DEBEVEC, P., MALIK, J., AND HAWKINS, T. 1999. Inverse global illumination: recovering reflectance models of real scenes from photographs. In *SIGGRAPH*, 215–224.

ZHAO, Q., TAN, P., DAI, Q., SHEN, L., WU, E., AND LIN, S. 2012. A closed-form solution to retinex with nonlocal texture constraints. *IEEE Trans. PAMI 34*, 1437–1444.

ZHU, J., SAMUEL, K. G. G., MASOOD, S., AND TAPPEN, M. F. 2010. Learning to recognize shadows in monochromatic natural images. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2010)*.