

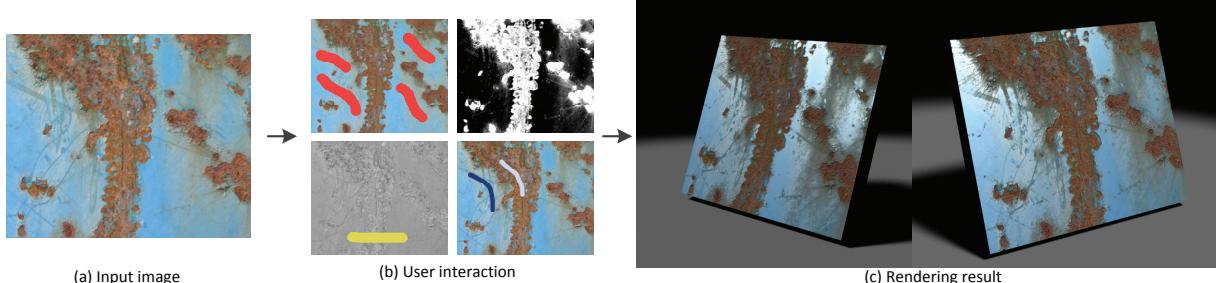
# AppGen: Interactive Material Modeling from a Single Image

Yue Dong<sup>\*†</sup>  
\*Tsinghua University

Xin Tong<sup>†</sup>  
† Microsoft Research Asia

Fabio Pellacini<sup>‡§</sup>  
‡ Dartmouth College

Baining Guo<sup>\*†</sup>  
§ Sapienza University of Rome



**Figure 1:** Given a single image (a), our system models the spatially varying reflectance properties and normals with a few strokes specified by the user (b). The resulting material can be rendered under different lighting and viewing conditions (c).

## Abstract

We present *AppGen*, an interactive system for modeling materials from a single image. Given a texture image of a nearly planar surface lit with directional lighting, our system models the detailed spatially-varying reflectance properties (diffuse, specular and roughness) and surface normal variations with minimal user interaction. We ask users to indicate global shading and reflectance information by roughly marking the image with a few user strokes, while our system assigns reflectance properties and normals to each pixel. We first interactively decompose the input image into the product of a diffuse albedo map and a shading map. A two-scale normal reconstruction algorithm is then introduced to recover the normal variations from the shading map and preserve the geometric features at different scales. We finally assign the specular parameters to each pixel guided by user strokes and the diffuse albedo. Our system generates convincing results within minutes of interaction and works well for a variety of material types that exhibit different reflectance and normal variations, including natural surfaces and man-made ones.

**CR Categories:** I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture;

**Keywords:** appearance modeling, user interaction

**Links:** [DL](#) [PDF](#)

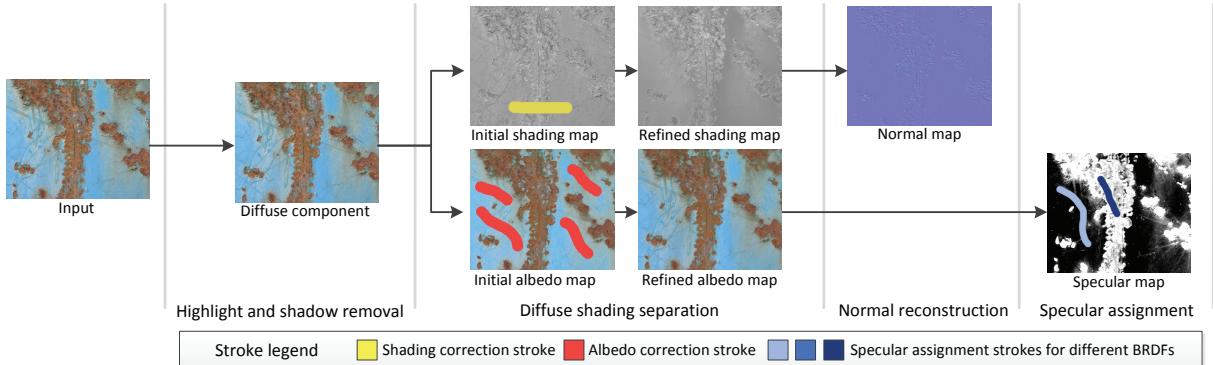
## 1 Introduction

**Modeling Realistic Materials.** The use of realistic materials is necessary when rendering high-quality images. Measured materials provide the highest quality datasets, but are cumbersome to use in practice due to complex acquisition setups, lengthy measurement

times and the size of the generated data [Weyrich et al. 2009]. Today, the vast majority of applications use materials painstakingly modeled by artists. Typically, artists start from a single texture image, and use that to generate spatially-varying diffuse, specular and roughness coefficients of an analytic reflectance model together with a bump map to enrich the surface details. For many materials, this process takes hours to perform, involving the use of image manipulation programs (e.g. Photoshop), inverse shading tools (e.g. CrazyBump), and shading networks in 3D software (e.g. Maya). Not only is this process cumbersome, but it often does not lead to the highest quality materials since no robust method can be used to easily derive detailed reflectance and normal maps from the image. Fig. 3 shows two example materials generated by an experienced artist in roughly one hour each, using the standard toolset that includes Photoshop and CrazyBump.

**AppGen.** In this paper, we present *AppGen*, an interactive system for modeling material from a single image. We focus on modeling spatially-varying reflectance (i.e. diffuse, specular and roughness parameters) and normal variations from a texture image that is captured from a nearly planar surface lit by directional lighting. Such images are easily found in texture collections since they are widely used by artists when manually modeling materials. Our goal is not to determine the exact reflectance and normals from such single images, which is a well-known ill-posed problem. Instead, we are interested in significantly speeding up the workflow of artists when modeling such materials. Our key idea is that we can keep user interaction minimal by asking the user to specify shading or reflectance information on a few pixels with sparse strokes, while our algorithm efficiently infers the reflectance and normal details for all pixels in the image. Fig. 1 shows one example of a material modeled using our system with just a few user strokes. Note the highly detailed, realistic look of the output material. Our experienced artist was able to regenerate the example materials in Fig. 3 in a few minutes using *AppGen*.

Our system consists of four steps, illustrated in Fig. 2. (1) First, we remove the highlight and shadow pixels in the input image and fill them by image inpainting. After that, we are left with an image of only the diffuse contribution. (2) We present an algorithm for interactively separating the texture image into the product of shading and diffuse albedo. We assume that in each local region pixels with the same chroma value belong to the same material and have the same albedo intensity, while groups of pixels with different chroma values share the same global geometry and thus have the same average shading. Based on these two assumptions, we formu-

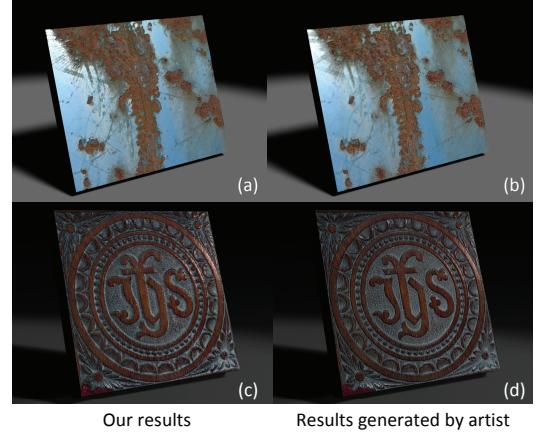


**Figure 2:** Overview of our system. In the preprocessing step, we remove highlight and shadow pixels. After that, with user assistance, we decompose the diffuse component into the product of a diffuse albedo map and a diffuse shading map. We then assign specular reflectance guided by the albedo map and user strokes. Finally, we reconstruct geometry details from the diffuse shading map. A legend of color coded strokes used in this paper is shown at the bottom.

late the separation as an optimization problem and solve it via an Expectation-Maximization (EM) like algorithm. We ask the user to interactively mark regions that violate our assumptions using a few rough strokes and augment our optimization procedure with these constraints to further refine the separation results. The result of this step is a diffuse color map and a shading map caused by normal variations. (3) We recover the per-pixel normals from the shading map by representing the underlying geometry as a height field (to capture the overall shape) with perturbed normals over it (to fit the remaining fine-scale shading variations). In general, splitting the shading map into a height field and perturbed normals contribution is an ill-posed problem. In our case though, since we assume that the contributions of the perturbed normals are subtle high frequency effects, we remove them from the shading map by smoothing. Based on this observation, we introduce a two-scale normal reconstruction algorithm. We first compute a height field that best fits the smoothed shading image, and then solve for a detailed normal map over the height field that best fits the detailed shading map. As a result, the geometry features at different scales are well recovered and generate consistent shading results under different lighting conditions. (4) Finally, we assign the proper specular properties to each pixel based on the diffuse albedo and the specular properties of a sparse set of pixels that are assigned by the user with rough strokes.

**Contributions.** Given an input image, our system can generate compelling materials within minutes of interaction using only a few user strokes. We found our system to work well for a variety of material types, including natural surfaces (metals, woods, rocks, leathers) and man-made ones (textiles, papers, concrete).

We believe that the main contribution of this paper is the overall interactive system. To the best of our knowledge, this is the first system that allows artists to use minimal interactions to model a wide-variety of material types from just a single texture image. The main technical contributions of this paper are our interactive diffuse shading separation algorithm and two-scale normal reconstruction algorithm. Both of them are specially designed for texture images that are characterized by detailed and coupled reflectance and normal variations. Our separation algorithm efficiently decouples reflectance and shading in a single image and preserves the details in both albedo and shading, while our reconstruction algorithm well recovers the height field and detailed normals from the shading map. Working together with the specular assignment algorithm, these two algorithms enable artists to quickly produce compelling material models.



**Figure 3:** Comparisons between the materials modeled by our method and materials manually modeled by an experienced artist. The results generated by the two methods are similar. Our system generates each result within 5 minutes, while the artist takes one hour to generate each result.

## 2 Related Work

**Image based Material Editing** modifies or alters the object material in a photograph to achieve different image editing effects. Fattal et al. [2007] use bilateral filtering of multi-light images into multiple scale images and enhance the shape and surface details of an object by manipulating its details in each scale. Fang and Hart [2004] and Zelinka et al. [2005] decorate an object in a photograph with synthesized texture, in which the object normal recovered via shape from shading is used to guide texture synthesis. Both methods assume the object geometry is smooth and ignore intensity variations caused by albedo. Khan et al. [2006] infer the shape and surrounding lighting of an object in a photograph and render its appearance with altered material. This method does not recover the object reflectance and directly maps the smoothed pixel intensity to depth. Xue et al. [2008] model the reflectance of weathered surface pixels in a photograph as a manifold and use it for editing the weathering effects in the image. All these methods only recover partial material information for editing object appearance under the same view and lighting of the original image. Our method is designed for modeling both reflectance and normal textures from a single image so that the results can be rendered under new viewing and lighting conditions.

**Image based Material Modeling** interactively recovers the geometry and reflectance of objects in a single input image. Oh et al. [2001] developed a set of tools for interactively modelling the depth layers in a single image. A decoupling filter is presented to decouple the illumination from uniformly textured areas. Their method is designed for modeling geometry of a scene or character but not material with the rich texture and geometric details we are interested in. Several interactive methods have been developed for modeling a bump map of structured textures [Dischler et al. 2002], displacement map of tree barks [Wang et al. 2003], and stochastic/procedural volumetric textures [Gilet and Dischler 2010] from single image input. All these methods are designed for specific kinds of textures and cannot easily be extended for modeling other textures. Our method has few constraints on the underlying geometry and texture distribution of the input image and thus provides a more general solution. In industry, CrazyBump [2010] is widely used by artists to generate bump maps from single images. For most texture inputs, it simply takes the image intensity as the shading map and ignores the albedo intensity variations of underlying materials, which leads to lots of manual work for refining results. Our approach well handles texture images with interwoven albedo and normal details and generates good results with few user interactions.

**Intrinsic Images** decomposes images into reflectance and illumination. Without prior knowledge, decomposition of intrinsic images from a single image cannot be solved due to its inherent ill-posedness [Grosse et al. 2009]. The method proposed by Horn [1986] assumes the illumination over the surface is smooth, which is not true for bumpy surfaces. Tappen et al. [2006] use color information and a pre-trained classifier to classify reflectance and albedo gradient in the input image. Shen et al. [2008] use detected textons as a non-local cue for better separation. However the texture clustering used in their method will lead to banding artifacts in the resulting shading map. Recently, Bousseau et al. [2009] decoupled intrinsic images from a single input image with user assistance. Their method assumes that the reflectance of pixels in a local window lies in a plane. Although this assumption is valid for natural images, it will fail in texture image with rich reflectance details. All these methods are designed for natural images and cannot work well for texture images that have rich variations in both albedo and shading. Our separation algorithm targets texture images. With few assumptions and sparse user input, our method optimally decomposes the reflectance from shading and keeps both albedo and shading details in the result. Xue et al. [2008] present a diffuse shading separation method for images of weathered surfaces where the surface reflectance forms a 1D manifold in color space. Our separation method has no assumption about the distribution of surface reflectance in color space and provides a general solution. We also introduce an efficient refinement scheme.

**Shape From Shading** tries to recover both albedo (i.e. diffuse color) and geometry from a single image, which is a well-known ill-posed problem. Horn and Brooks [1989] recover the normals from a shading image by regularizing the problem with smoothness constraints. The resulting normals are always noisy and biased toward the lighting direction. Later methods [Zhang et al. 1999; Durou et al. 2008] assume the underlying surface is integrable and solve a height field from the shading image. Although these methods work well for reconstructing the surface geometry, they fail to fit the shading details and thus cannot reproduce detailed normal variations over the surface. Our two step normal reconstruction approach well recovers both surface geometry and detailed normal variations from a single shading image and generates consistent shading results under different lighting directions.

Recently, Glencross et al. [2008] hallucinated a surface height field from an image captured under diffuse ambient lighting. Another image with flash illumination of the same scene is needed for separating the albedo from the input image. Our system uses a single input image under directional lighting for modeling both albedo variations and normal variations. Moreover, our normal reconstruction method well recovers detailed normal maps and preserves details in the shading map.

### 3 System Overview

Our system takes as input a texture image  $I$  of a nearly planar surface with spatially-varying reflectance and normal variations and lit with a directional light. Since the underlying surface is nearly planar, we further ignore occlusion and inter-reflections between geometry details on the surface. Although this lighting model is not physically accurate, it nevertheless produces plausible results for many input images, as shown in this paper.

Without losing generality, we model the BRDF  $\rho(x)$  at pixel  $x$  as the sum of a Lambertian component, with albedo  $\rho_d(x)$ , and a specular component, with specular coefficient  $\rho_s(x)$  and lobe shape  $f_r(x)$ . Under directional lighting, the image value  $\mathbf{I}(x)$  at  $x$  can be computed as the sum of the diffuse contribution  $\mathbf{I}_d(x)$  and the specular highlights  $\mathbf{I}_s(x)$  as

$$\mathbf{I}(x) = \mathbf{I}_d(x) + \mathbf{I}_s(x), \quad (1)$$

where

$$\mathbf{I}_d(x) = \rho_d(x)\mathbf{S}_d(x) = \rho_d(x)(\mathbf{N}(x) \cdot L)I_l \quad (2)$$

$$\mathbf{I}_s(x) = \rho_s(x)\mathbf{S}_s(x) = \rho_s(x)f_r(\mathbf{N}(x), L, V)(\mathbf{N}(x) \cdot L)I_l. \quad (3)$$

The diffuse shading  $\mathbf{S}_d(x)$  is determined by the local normal  $\mathbf{N}(x)$ , the light direction  $L$  and intensity  $I_l$ , while the specular shading  $\mathbf{S}_s(x)$  is also related to the viewing direction  $V$ .

Given input image  $\mathbf{I}$ , the goal of our system is to model spatially-varying diffuse albedos  $\rho_d(x)$ , specular coefficients  $\rho_s(x)$ , lobe shapes  $f_r(x)$ , and normals  $\mathbf{N}(x)$  with the help of a few user strokes. Fig. 2 shows an overview of our system that is composed of the following steps:

- *Highlight and shadow removal.* We first identify the highlight  $\mathbf{I}_s(x)$  and shadow pixels  $\mathbf{I}_o(x)$  by thresholding the pixel values ( $I(x) > 235$  for highlights and  $I(x) < 25$  for shadows for 8-bit images) and fill these pixels by image inpainting [Bertalmio et al. 2000]. After that, the image only contains the diffuse component  $\mathbf{I}_d$ . Any other shadow removal and specular separation methods can be used in this step.
- *Diffuse Shading Separation (Sect. 4).* We decompose  $\mathbf{I}_d$  obtained from the last step into the product of a diffuse albedo map  $\rho_d$  and a diffuse shading map  $\mathbf{S}_d$ . We formulate this separation as an optimization problem and compute the initial shading and diffuse albedo maps. After that, the user quickly refines the initial separation results by drawing sparse strokes in regions that violate our assumptions and thus exhibit artifacts.
- *Normal Reconstruction (Sect. 5).* We reconstruct a normal map  $\mathbf{N}$  from the diffuse shading map  $\mathbf{S}_d$  with a two-scale normal reconstruction approach. After the user specifies the lighting direction, we first compute a height field that fits a smoothed shading map. We then recover fine geometric variations, by fitting detailed normals over the height field to match the detailed input shading map  $\mathbf{S}_d$ .
- *Specular Assignment (Sect. 6).* We assign the specular behavior ( $\rho_s$  and  $f_r$ ) of a fixed set of specular basis BRDFs to each pixel, guided by user strokes that assign the basis BRDFs to just a few pixels. A material classification algorithm determines the

material type of all pixels, according to the diffuse color and stroke location, and uses this to assign a specular BRDF to each pixel.

## 4 User-Assisted Shading Separation

In this step, we decompose the diffuse components  $\mathbf{I}_d$  into a shading map  $\mathbf{S}_d$  and a diffuse albedo map  $\rho_d$ . These two components will serve as the input of the following steps for normal reconstruction and specular assignment respectively.

For this purpose, we represent the image value  $\mathbf{I}_d = (\mathbf{I}_d^r, \mathbf{I}_d^g, \mathbf{I}_d^b)$  by its intensity  $\mathbf{I}_d^i = (\mathbf{I}_d^r + \mathbf{I}_d^g + \mathbf{I}_d^b)/3$  and chroma value  $\mathbf{I}_d^c = (\mathbf{I}_d^r/\mathbf{I}_d^i, \mathbf{I}_d^g/\mathbf{I}_d^i, 3 - \mathbf{I}_d^r/\mathbf{I}_d^i - \mathbf{I}_d^g/\mathbf{I}_d^i)$ . We assume the light is white  $\mathbf{I}_l = (1.0, 1.0, 1.0)$  so that the image chroma comes from the chroma of the diffuse albedo  $\rho_d^c(x) = \mathbf{I}_d^c(x)$ , while the image intensity is the product of shading and albedo brightness

$$\mathbf{I}_d^i(x) = \rho_d^c(x), \quad \mathbf{I}_d^i(x) = \rho_d^i(x)\mathbf{S}_d^i(x). \quad (4)$$

Our goal is to decompose the diffuse intensity  $\mathbf{I}_d^i$  into an albedo intensity map  $\rho_d^i$  and a shading intensity map  $\mathbf{S}_d^i$ . To this end, we formulate the separation as an optimization problem and solve the initial albedo map and shading map by an EM (Expectation-Maximization) like algorithm. After that, we refine the results with the constraints specified by sparse strokes.

### 4.1 Separation as Optimization

We first assume that in each local region  $\Omega$ , pixels with the same chroma value  $\mathbf{I}_d^c(x) = \rho_d^c(x) = c$  belong to one material and thus have the same albedo intensity  $i_c$ . Based on this *local albedo assumption*, we have

$$\rho_d^i(x) = i_c \quad x \in \Omega_c, \quad (5)$$

where  $\Omega_c$  refers to the set of pixels that are in  $\Omega$  and have the same chroma value  $c$ . For shadings caused by the geometric details in  $\Omega$ , our key observation is that although the spatial patterns and amplitudes of the geometric details of each material (i.e. pixels in each  $\Omega_c$ ) may be different from one another, the large scale geometry in  $\Omega$  is almost flat. As a result, the average normals of the geometric details of all materials in  $\Omega$  are almost the same and the shading estimation of each material is equivalent to the shading estimation of all pixels in  $\Omega$ . Based on this *local shading assumption*, we have

$$E(\mathbf{S}_d^i(x)|x \in \Omega_c) = E(\mathbf{S}_d^i(x')|x' \in \Omega). \quad (6)$$

Given Equation 5 and Equation 6, the intensity estimation of pixels in  $\Omega_c$  can be computed as

$$E(\mathbf{I}_d^i(x)|x \in \Omega_c) = \frac{\sum_{x \in \Omega_c} (\rho_d^i(x)\mathbf{S}_d^i(x))}{N_{\Omega_c}} = \frac{i_c \sum_{x \in \Omega_c} \mathbf{S}_d^i(x)}{N_{\Omega_c}} \\ = i_c E(\mathbf{S}_d^i(x)|x \in \Omega_c) = i_c E(\mathbf{S}_d^i(x')|x' \in \Omega), \quad (7)$$

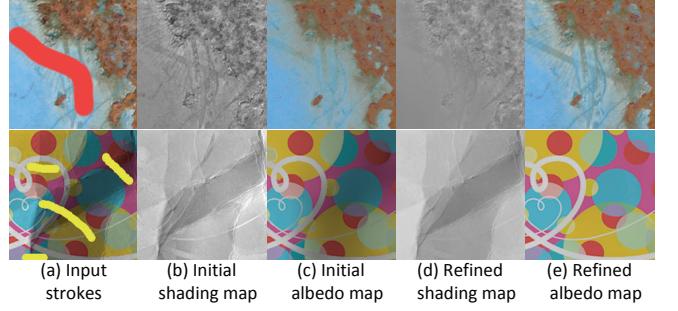
where  $N_{\Omega_c}$  is the number of pixels in  $\Omega_c$ . So the albedo intensity  $\rho_d^i(x)$  of a pixel  $x$  in region  $\Omega_c$  should satisfy

$$\rho_d^i(x) = \frac{E(\mathbf{I}_d^i(x')|x' \in \Omega_c)}{E(\mathbf{S}_d^i(x')|x' \in \Omega)} \quad x \in \Omega_c. \quad (8)$$

Since the shading intensity is  $\mathbf{S}_d^i(x) = \mathbf{I}_d^i(x)/\rho_d^i(x)$ , we can rewrite the right side of the equation as a function of image intensities and albedo intensities of pixels in  $\Omega$  as

$$E_0(\Omega, c, \mathbf{I}_d^i, \rho_d^i) = \frac{E(\mathbf{I}_d^i(x')|x' \in \Omega_c)}{E(\mathbf{S}_d^i(x')|x' \in \Omega)} = \frac{\frac{1}{N_{\Omega_c}} \sum_{x' \in \Omega_c} \mathbf{I}_d^i(x')}{\frac{1}{N_{\Omega}} \sum_{x'' \in \Omega} \frac{\mathbf{I}_d^i(x'')}{\rho_d^i(x'')}}. \quad (9)$$

Based on this local constraint, we formulate the separation as an optimization problem by minimizing the following energy function



**Figure 4:** Interactive refinement. Given the initial shading map (b) and albedo map (c) separated from the input image, the user draws sparse strokes in regions that violate our assumptions to refine the results. The top row shows the effect of Albedo correction stroke, and the bottom row shows the effect of Shading correction strokes. Artifacts in the initial results (b) and (c) are fixed in (d) and (e) after refinement.

of  $\rho_d^i(x)$ :

$$F_0(\rho_d^i(x)) = \sum_{\Omega \in \Omega^\dagger} \|\rho_d^i(x) - E_0(\Omega, \rho_d^c(x), \mathbf{I}_d^i, \rho_d^i)\|^2, \quad (10)$$

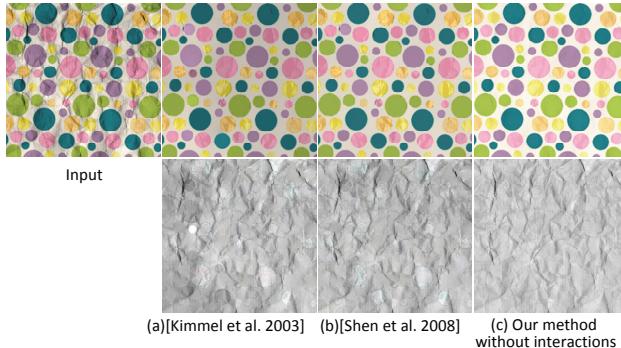
where  $\Omega^\dagger$  is the collection of all fixed sized local regions that contain  $x$ . In practice, we define  $\Omega$  as a  $W \times W$  window and solve this optimization using an iterative algorithm similar to Expectation Maximization. In the E-step, given the  $E_0(\Omega, \rho_d^c(x), \mathbf{I}_d^i, \rho_d^i)$  computed from the current albedo intensity map, we update  $\rho_d^i(x)$  by solving the linear equations that result from the differentiations of  $F_0(\rho_d^i(x))$  with respect to  $\rho_d^i(x)$ . Then in the M-step, we update the  $E_0(\Omega, c, \mathbf{I}_d^i, \rho_d^i)$  for each window and each chroma value from the new albedo intensity map according to Equation 9. We repeat these two steps iteratively until convergence. After obtaining the albedo intensity  $\rho_d^i(x)$ , we compute the shading intensity  $\mathbf{S}_d^i(x) = \mathbf{I}_d^i(x)/\rho_d^i(x)$ . Fig. 4.b and 4.c illustrate the shading intensity map and albedo intensity map separated by our optimization algorithm.

In practice, we set the region size to  $W = 20$  and initialize the optimization by setting the albedo intensity to the image intensity (i.e. the shading intensity is 1.0 everywhere). To determine whether two pixels have the same chroma value, we uniformly subdivide the first two channels of chroma vectors into 20 slots and quantize each pixel's chroma value to one of 400 quantized chroma values.

### 4.2 Interactive Refinement

Although our method generates reasonable results in most image regions, it will fail and generate artifacts in regions that violate our assumptions. Specifically, in regions that violate the *local albedo assumption*, pixels with the same chroma value have different albedo intensities. In this case, our method will leave image intensity variations of these pixels to the shading map and thus generate undesired detail shading variations in flat regions (shown in the top row of Fig. 4.b and 4.c). In regions that violate the *local shading assumption*, the shading estimation of each material is different from each other and thus is also different from the shading estimation of the local region. In this case, our method will compute a biased albedo intensity for each pixel and thus introduce undesired variations in regions with constant albedo (shown in the bottom row of Fig 4.b and 4.c). This case often happens in regions where the material distribution is correlated to large scale geometric structures.

**User Strokes** We design a user stroke for each type of artifact for users to quickly specify artifact pixels. Based on sparse strokes



**Figure 5:** Comparisons to automatic intrinsic image algorithms. (a) The color-retinex [Kimmel et al. 2003] has artifacts in the separated results. (b) Even if combined with non-local texture cues [Shen et al. 2008] these artifacts cannot be fully removed. (c) Without user interaction, our method already produces reasonable results.

specified by the user, our algorithm automatically removes the artifacts and refines the separation results.

To fix the artifacts in regions that violate the *local albedo assumption*, we ask the user to draw *albedo correction strokes* over artifact pixels to indicate that locally the underlying geometry is flat and the shading details of a pixel should be moved to its albedo intensities. As a result, each albedo correction stroke defines the following constraint:

$$\rho_d^i(x) = E_S(\Omega, x, \mathbf{I}_d^i, \rho_d^i) = \frac{\mathbf{I}_d^i(x)}{E(\mathbf{S}_d^i(x') | x' \in \Omega)} = \frac{\mathbf{I}_d^i(x)}{\frac{1}{N_\Omega} \sum_{x' \in \Omega} \frac{\mathbf{I}_d^i(x')}{\rho_d^i(x')}} \quad (11)$$

and an energy term  $F_S$  for optimization:

$$F_S(\rho_d^i(x)) = \sum_{\Omega \in \Omega^\dagger} w(x) \|\rho_d^i(x) - E_S(\Omega, x, \mathbf{I}_d^i, \rho_d^i)\|^2, \quad (12)$$

where  $w(x) = \lambda e^{-\|x-x_s\|^2/\sigma}$  is a weight function to control the importance of the stroke constraint at  $x$ , which is determined by the distance between  $x$  and its closest stroke pixel  $x'$ . In practice, we set  $\lambda = 10.0$  and  $\sigma = 3.0$ .

To fix artifacts in regions that violate the *local shading assumption*, we ask the user to draw *shading correction strokes* over artifact pixels to indicate that locally the albedo intensity of a pixel with chroma  $c$  should be the same as the albedo intensity of stroke pixels that have the same chroma  $c$ . If no pixel in the stroke has chroma  $c$ , the pixel keeps its original albedo. So each shading correction stroke defines the following constraint:

$$\begin{aligned} \rho_d^i(x) &= E_A(\Omega_s, c, \rho_d^i) = E(\rho_d^i(x') | x' \in \Omega_{sc}) \\ &= \frac{1}{N_{\Omega_{sc}}} \sum_{x' \in \Omega_{sc}} \rho_d^i(x') \quad \rho_d^c(x) = c, \end{aligned} \quad (13)$$

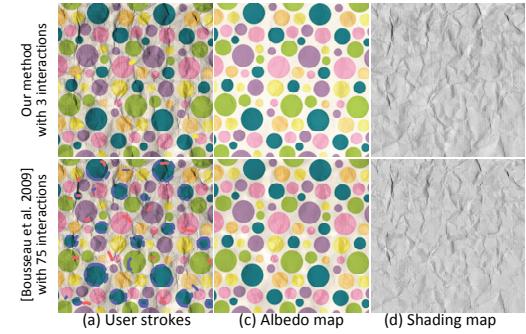
where  $\Omega_s$  is the set of all pixels in the stroke,  $\Omega_{sc}$  refers to all pixels in  $\Omega_s$  that have chroma  $c$ , and  $N_{\Omega_{sc}}$  is the number of pixels in  $\Omega_{sc}$ . We thus define the following energy term  $F_A$  for optimization:

$$F_A(\rho_d^i(x)) = \sum_{\Omega \in \Omega^\dagger} w(x) \|\rho_d^i(x) - E_A(\Omega_s, \rho_d^c(x), \rho_d^i)\|^2. \quad (14)$$

**Result Refinement** To refine the separation result with user specified strokes, we minimize the following energy function that combines the energy terms defined by all strokes and  $F_0$ :

$$F_0(\rho_d^i(x)) + \lambda_S \sum_{j=0}^{N_S} F_{S,j}(\rho_d^i(x)) + \lambda_A \sum_{k=0}^{N_A} F_{A,k}(\rho_d^i(x)), \quad (15)$$

where  $F_{S,j}$  denotes the for  $j$ -th albedo correction stroke, and  $F_{A,k}$  is the  $k$ -th shading correction stroke.  $N_S$  and  $N_A$  are the numbers of



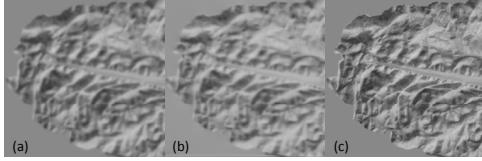
**Figure 6:** Comparisons to user-assisted intrinsic images. Our method is shown in the top row, while the results of [Bousseau et al. 2009] are shown in the bottom row. With only 3 interactions, our method can generate high quality separation results, while [Bousseau et al. 2009] requires many more input strokes to get a reasonable separation.

strokes of the two types.  $\lambda_S$  and  $\lambda_A$  weight the constraints specified by the two types of strokes respectively in optimization, both of which are set to 1.0 in practice.

We adapt our iterative solver for this new energy function. In the E-step, we update the albedo intensity  $\rho_d^i(x)$  by solving the sparse linear system resulting from the differentiations of the new energy function with respect to  $\rho_d^i(x)$ . In the M-step, we update the functions  $E_0$ ,  $E_S$  and  $E_A$  with the new albedo intensity values according to Equation 9, 11, and 13 respectively. We repeat these two steps iteratively until convergence. The spatial weights  $w$  for each stroke are computed before the optimization. Fig. 4.d and 4.e illustrate refined separation results, where the artifacts are removed with the help of sparse user strokes.

### 4.3 Discussion

Different from previous intrinsic images methods [Grosse et al. 2009] mainly designed for natural images, our method targets texture images with details in both reflectance and shading. On one hand, our local albedo assumption allows arbitrary chroma variations in the local region and thus can well handle complicated reflectance details. On the other hand, our local shading assumption only constrains the average shading of each material in the local region and thus well preserves the complicated shading details in the input. Moreover, we design two types of strokes for the user to quickly remove artifacts and refine results. We compare our method with two automatic intrinsic images methods [Kimmel et al. 2003; Shen et al. 2008] in Fig. 5. The color-retinex method [Kimmel et al. 2003] generates visible artifacts in the separated results. Although the non-local texture used in [Shen et al. 2008] improves the results, the separation artifacts still cannot be fully removed. On the contrary, our method can automatically recover the shading/albedo in images with both shading and albedo variations. Fig. 6 compares the results generated by our method with the ones generated by the user-assisted intrinsic images method in [Bousseau et al. 2009]. Note that in [Bousseau et al. 2009], the local reflectance plane assumption cannot guarantee the shading/albedo to be constant in regions with constant shading/albedo. User inputs are always necessary for generating reasonable results and become cumbersome as the detail in the input image increases, while in our method, the automatic solution already generates convincing results for most image inputs. User input is only needed to fix artifacts in the results.



**Figure 7:** Two-scale normal reconstruction. (a) Filtered shading map. (b) Result rendered from the reconstructed height field. (c) Result rendered from the final normal map.

## 5 Two-Scale Normal Reconstruction

To model the geometry details of the input surface, we reconstruct a normal map  $\mathbf{N}(x)$  from the diffuse shading map  $\mathbf{S}_d(x)$  and a lighting direction  $L$  roughly specified by the user.

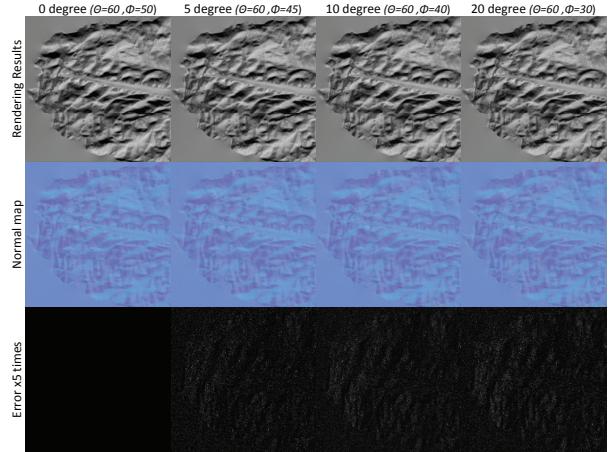
For this purpose, we represent the surface geometry as a height field with perturbed normals over it and develop a two-scale normal reconstruction algorithm. The key observation of our approach is that the shading details produced by the normal perturbations in  $\mathbf{S}_d(x)$  are always subtle and high-frequency, and as such can be filtered out by smoothing. Based on this observation, we first filter the input shading map  $\mathbf{S}_d$  with a  $3 \times 3$  Gaussian filter ( $\sigma = 0.4$  in our implementation) and recover a height field  $\mathbf{H}$  from the filtered shading map  $\mathbf{S}'_d$  via shape from shading. We follow the method in [Wu et al. 2008] to compute the height field  $\mathbf{H}$  in our current implementation, but other shape from shading methods can also be used here. After that, we compute the perturbed normals defined over the height field by minimizing the energy function:

$$E_n = \sum_x ||\mathbf{N}(x) \cdot L - \mathbf{S}_d(x)||^2 + \lambda \sum_x ||\mathbf{N}(x) - \mathbf{N}_h(x)||^2. \quad (16)$$

The first term constrains the shading result of  $\mathbf{N}$  under the lighting direction  $L$  to fit the input shading map  $\mathbf{S}_d$ , while the second regularization term minimizes the difference between the resulting normal  $\mathbf{N}$  and the normal  $\mathbf{N}_h$  computed from the height field  $\mathbf{H}$ . The weight  $\lambda$  is a regularization term, which is set to 0.001 for all results shown in the paper. This optimization can be done by solving a sparse linear system. In practice, we initialize the optimization by setting  $\mathbf{N}(x) = \mathbf{N}_h(x)$  and compute the normal via a Gauss-Seidel solver with successive over-relaxation. The results of these two steps are illustrated in Fig. 7. Note that the normal map recovered by our method preserves well the geometric details and generates convincing rendering results under different lighting conditions.

**Discussion** Our method is not sensitive to the accuracy of the light direction  $L$  specified by the user in that the error of light direction will only lead to a global rotation of all normals over the surface but has no effect on the relative normal variations. Since our method assumes the underlying geometry is almost flat, we rotate the initial normals and lighting in the first step so that the average normal of underlying geometry is always upward. Fig. 8 illustrates results generated from one input shading map but with different lighting directions specified by the user. As the error of the specified lighting direction becomes larger, the error of the resulting normal map is small and almost unchanged.

In Fig. 9, we compare the result generated by our method (Fig. 9.g) with the ones generated by other existing normal reconstruction methods. A straightforward solution would be to directly reconstruct the normal map from  $\mathbf{S}_d$  with a regularization term as in [Horn and Brooks 1989]. Although the resulting normal map well fits the input shading image, it is biased toward the lighting direction  $L$  and generates artifacts under other lighting directions (Fig. 9.c). Other shape from shading approaches assume the surface is integrable and reconstruct a height field from the input shading map. Although these methods can reconstruct the overall surface geometry, the detailed normal perturbations over the surface are



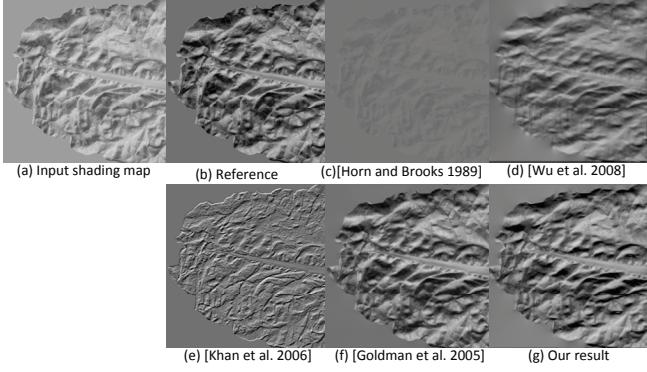
**Figure 8:** Normal maps reconstructed from the same shading map, but different lighting directions, rotated respectively at 0, 5, 10, and 20 degrees. The rendering results are shown in the top row. The middle row visualizes the reconstructed normal map. The bottom row shows the a 5 times magnified reconstruction error, measured as the length of the cross product between the resulting normal map and normal map at 0 degrees. Note how our method is robust to the input light directions.

smoothed out. Fig. 9.d shows a rendering result of the height field reconstructed by [Wu et al. 2008]. Although the height field generates reasonable results under different lighting directions, the shading details caused by detailed normal variations are lost. Fig. 9.e illustrates the result generated by the shading-as-depth method used in [Khan et al. 2006]. Although the shading-as-depth method works well for many smooth surfaces, it fails to model the geometric features in this input. We also compare the results generated by our method with the one (Fig. 9.f) generated by a photometric method in [Goldman et al. 2010]. While the photometric method can well recover the normal map, it needs many more images (12 images in this example) as input, which are not always available in our application. Instead, our method generates convincing results from a single input image.

## 6 User-Assisted Specular Assignment

In this step, we assign a specular coefficient  $\rho_s$  and lobe shapes  $f$  to each pixel. Based on the observation that the specular properties at each pixel mostly depend on the underlying material type (e.g. whether a pixel is metal or rust) rather than detailed diffuse color variations, we ask the user to assign the specular BRDF to a sparse set of pixels, using rough strokes, and then automatically determine the specular BRDF for other pixels. The user strokes not only determine the specular reflectance of the underlying pixels, but also assign the same material type to these pixels. With  $N_M$  BRDFs and  $N_M$  corresponding material types assigned by the strokes, we classify each pixel's material type by computing the probability that it belongs to each material type. After that, we assign the specular BRDF to each pixel based on the material classification results. This process is illustrated in Fig. 10.

**Material Classification** Given the set of pixels that belongs to each material type  $i$  (i.e. the pixels in the same-BRDF strokes), we construct the sets  $M_i$  of their diffuse colors. We remove the outliers in each set  $M_i$  by checking each diffuse albedo in the set and finding whether its  $k = 10$  nearest diffuse albedos are in the same material set. If more than 50% of them come from material sets other than  $M_i$ , we remove this diffuse color value from  $M_i$ . After that, we



**Figure 9:** Comparisons of different normal reconstruction methods. (a) is the input shading map rendered from a ground truth normal map. (b) is a reference image rendered from the ground truth under a novel lighting direction. (c) to (g) are images rendered from the results generated by different normal reconstruction methods. The lighting directions used in rendering are the same as the one used in (b). Previous shape from shading methods either generate biased results (c), or smooth out the detailed normal variations (d). (e) Simply taking the shading as depth [Khan et al. 2006] does not generate a reasonable result. The photometric stereo method [Goldman et al. 2010] (f) can accurately reconstruct the normal map, but requires much more input data. Our normal reconstruction algorithm can well preserve normal details and generate a convincing result (g) from a single input image.

compute the probability  $p_i(x)$  that each pixel  $x$  belongs to the  $i$ -th material type by using Shepard's method [1968] as

$$p_i(x) = \frac{d_i(x)^{-p}}{\sum_{j=1}^m d_j(x)^{-p}}, \quad (17)$$

where  $m$  is the total number of material types and  $d_i(x)$  is the distance from pixel  $x$ 's diffuse color to the  $i$ -th material type's material set, which is computed by

$$d_i(x) = \frac{1}{10} \sum_{j=0}^{10} \|\rho_d(m_j) - \rho_d(x)\|, \quad (18)$$

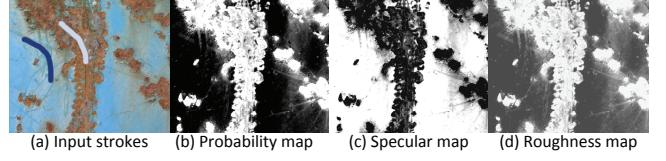
where  $\rho_d(m_j)$  is the 10 diffuse albedos in  $M_i$  that are closest to  $\rho_d(x)$ . In practice, we pick  $p = 1$  for all the results. Although it is possible to apply other edit propagation methods [An and Pellacini 2008; Xu et al. 2009] for determining the  $d_i$  for each pixel, we apply Shepard's method in our current implementation because of its simplicity.

**Specular coefficient assignment.** After material classification, we assign the specular coefficient  $\rho_s(x)$  and specular roughness  $\alpha(x)$  to each pixel by

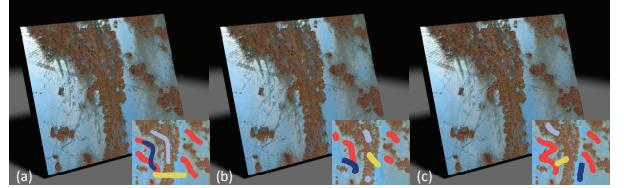
$$\rho_s(x) = \sum_{i=1}^M \rho_i p_i(x), \quad \alpha(x) = \sum_{i=1}^M \alpha_i p_i(x), \quad (19)$$

where  $p_i(x)$  is the probability that pixel  $x$  belongs to the  $i$ -th material, and  $\rho_i$  and  $\alpha_i$  are the specular coefficient and roughness of the  $i$ -th material respectively.

In our implementation, the specular BRDFs are selected from pre-defined 120 specular BRDFs extracted from measured materials. We represent the specular lobe shapes  $f_r$  by using the Ward model controlled by the roughness parameter  $\alpha(x)$  ranging from 0.005 to 0.5. When necessary, the user can fine-tune the parameters as well as the specular color of the selected BRDF. We use the Ward model as it can be easily adjusted by the user to fine tune the lobe shape. Our method itself is not constrained to a parametric specular model.



**Figure 10:** Specular assignment. We ask the user to assign single BRDFs to pixels that belong to the same material type by sparse strokes (a). Our algorithm then classifies the material type of pixels and determines the probability map (b). Finally, the specular (c) and roughness (d) coefficient of each pixel are calculated based on the probability map (b) and assigned BRDFs.



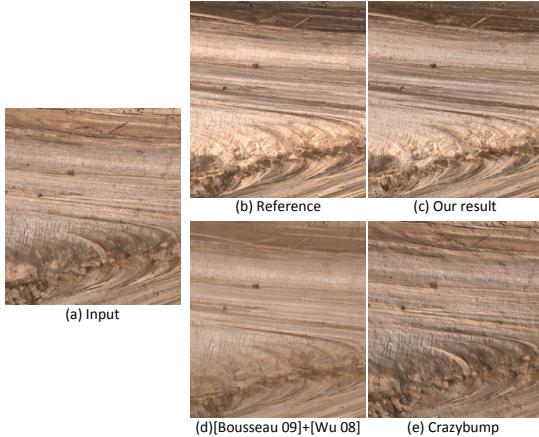
**Figure 11:** Images rendered from the results generated with different stroke inputs. The user stroke inputs are shown in the bottom right. Although the stroke inputs are different, the results generated by our method are similar.

## 7 Experimental Results

**Performance** We performed our tests on a PC with an Intel Xeon 2.83GHz CPU and 4GB RAM. For a typical input image of size  $768 \times 768$ , the albedo and shading separation runs within 1.8 seconds, the two-step normal map reconstruction converges within 1.5 seconds, depending on the input shading map, and the specular assignment step takes less than 0.2 seconds for material classification and reflectance coefficient blending. The fast computations in each step provide responsive feedback for user interaction.

**User Input** All results shown in the paper are generated by an artist in one to five minutes. Depending on the complexity of the input image, up to 9 strokes were used for diffuse shading separation, while 1 to 4 strokes were drawn for specular assignment. For all results shown in Figs. 14 to 18, we display all the strokes used for modeling in the input image. Different types of strokes are rendered in different colors. Similar to other stroke based methods, our method does not require accurate user strokes. Fig. 11 illustrates the results generated from the same input but with different input strokes. Provided that the user intention was the same, these different user strokes generate similar results.

**Comparison with Standard Toolsets** Fig. 12 compares a ground truth rendering (12.b) with the result generated by our method (12.c), a combination of [Bousseau et al. 2009] and [Wu et al. 2008] (12.d), and CrazyBump (12.e). We use an RGBN dataset from [Fattal et al. 2007] to render the input image (12.a) under directional lighting. We then compare the rendering of the ground truth data and the data reconstructed by the different methods under a new lighting direction. In generating these images, we assume that the surface presented by the RGBN data is diffuse and take the RGB value as the albedo. To generate the result shown in Fig. 12(d), we applied the algorithm in [Bousseau et al. 2009] to separate the input image and then computed the normals from the resulting shading map with [Wu et al. 2008]. As shown in Fig. 12, both CrazyBump and the combination of [Bousseau et al. 2009] and [Wu et al. 2008] fail to recover the reflectance and normals from the input. On the contrary, our method recovers the reflectance and normal details well. We include the shading and reflectance maps generated by these methods in the supplemental material.



**Figure 12:** Comparison with prior work. (a) Input image rendered from ground truth data [Fattal et al. 2007]. (b) Reference image rendered from ground truth data lit from different lighting direction. (c) Rendering results of our method. (d) Results generated by a combination of [Bousseau et al. 2009] and [Wu et al. 2008]. (e) Results modeled by an artist using Crazybump. (b)-(e) are rendered under the same light direction, but different from (a). The result generated by our method is closer to the ground truth, while other methods fail to recover the shading and reflectance presented in the input image.

Fig. 3 illustrates two results generated by an experienced artist, in about one hour each, using standard image manipulation tools, including Photoshop and CrazyBump. With our method, the user generates similar results using few user strokes within five minutes. Although a more systematic user study would be needed to derive formal results, we feel that this comparison is typical of our experience and shows the efficiency of our approach.

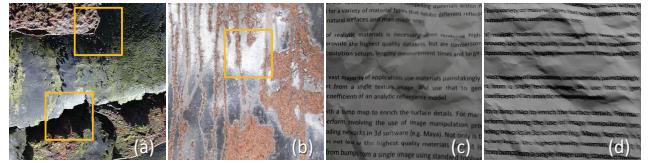
**Results** We tested our system with a wide variety of input images, shown in Fig. 14-18 together with users strokes and the final materials, rendered under different viewing and lighting. We show user input as yellow strokes for *shading correction strokes*, red strokes for *albedo correction strokes*, and blue ones with different intensity to indicate the strokes that mark the placement of the different specular BRDFs. In the supplemental material, we include shading map, albedo map and additional renderings of these results, as well as additional test cases.

We chose input images that correspond to a wide range of materials, ranging from man-made materials such as paper (Fig. 15) and wood carving (Fig. 18-a), to natural materials like wood (Fig. 14-c, 16-a), stone (Fig. 14-a), asphalt (Fig. 17-c), and rusted metals (Fig. 16-c, 17-a). These input images and corresponding materials show a wide range of spatial distributions of geometry and reflectance, demonstrating the wide applicability of our approach.

Fig. 14 shows results of two natural materials with complex geometric details. Note how the large scale geometry variations of the rock and the sharp cracks in the raw wood are modeled well by our method.

Fig. 15 shows two results of wrinkled papers, which consist of sharp color variations and rich geometric details at multiple scales. Our method captures well these reflectance and normal variations and generates convincing rendering results.

In many natural materials, such as wood and rust, complex reflectance and normal variations are often combined. Fig. 16 shows two typical images of these material types. Note that with minimal interaction, our method models well the spatially-varying high-



**Figure 13:** Failure cases of our method. (a) Image with strong geometric structures. (b) Image containing large regions of highlights. (c) Grayscale image, where the albedo only has grayscale variations. (d) Our method fails to separate the black text from shading variations and generates artifacts in the normal map.

lights of the finished wood and the rich color and reflectance details in the rust.

Figs. 17 and 18 demonstrate examples of surfaces composed of multiple material types, each of which exhibits its own spatially-varying appearance. The rusted metal in Fig. 17 has shiny specular highlights in the metallic regions, while the rusted parts are nearly diffuse with strong normal variations. Our result reproduces realistic appearance for both the metallic and rusted regions. Note the natural variations of the highlights in the final rendered images. For the asphalt in Fig. 17, our method successfully reveals the detailed bump variations of the surface while preserving the global structure of the yellow paint.

Fig. 18 shows two materials that have man-made global structure and detailed natural textures. Our system reproduces the geometry variations for both man-made carvings and natural textures. In the wood carving, the specular reflectance of the polished red wood and matte reflectance of the dusted part are well preserved by our specular assignment. For the rock wall example, the rich albedo variations and sharp boundaries between rock and cement are well modeled in our result.

**Limitations** Although our system works well for a wide variety of texture images, we made several assumptions that might be violated by some input, as shown in Fig. 13. First, we assume that the input comes from a nearly planar surface lit by directional lighting. For images that violate this assumption, our algorithm may fail to produce convincing results (e.g. Fig. 13.a). Second, our method mostly uses the diffuse component of the input image for material modeling. For images that contain large regions of highlight and shadows (e.g. Fig. 13.b), that are often clamped, the material variations in these regions cannot be reconstructed well by our method. Finally, our interactive shading separation method relies on chroma variations during initialization. For materials with grayscale albedo, our separation method fails to generate the initial separation result, thus potentially requires too much user input for diffuse shading separation. Fig. 13.c illustrates a typical case: an image of a wrinkled paper with greyscale text on it, where our method generates artifacts in the resulting normal map and albedo map.

## 8 Conclusion

In this paper, we present *AppGen*, an interactive system for modeling materials from a single image. Given an input texture image of a nearly-planar surface lit by directional lighting, our system lets artists efficiently model the spatially-varying reflectance and normals, requiring only a few minutes of interaction to place a few rough strokes. A diffuse shading separation algorithm and a two-step normal reconstruction method are presented for deriving the normal and reflectance details from the texture input. We illustrate the capability of our method by modeling a wide variety of materials generated from images with different reflectance and normal variations. We believe that our system can greatly speed up the

workflow of expert artists and might allow less technically-trained ones to start modeling materials.

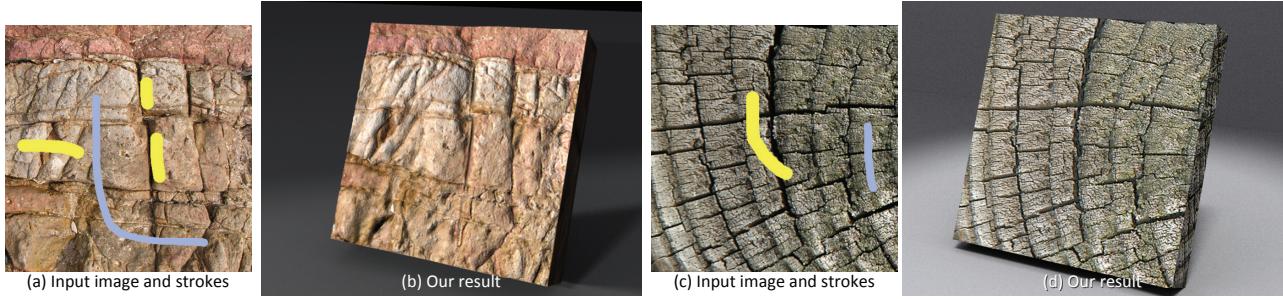
Our method assumes that the input image is lit with directional lighting. One direction for future work is to extend our system to model materials from images taken under environmental lighting. We are also interested in investigating the modeling of materials with complex meso-structure, that produce images with complex shadowing and inter-reflection. Finally, we are interested in developing methods that leverage measured material datasets when available in order to speed up the material modeling process.

## Acknowledgments

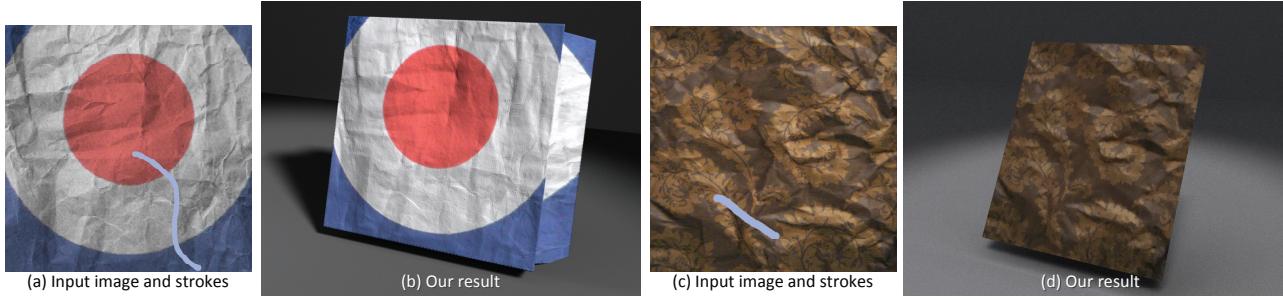
The authors would like to thank Stephen Lin for paper proofreading and Matt Callcut for video dubbing. The authors also thank GrungeTexture, Leo Reynolds, ka2rina from Flickr and Mayang.com for providing texture inputs. The authors are grateful to the anonymous reviewers for their helpful suggestions and comments. Fabio Pellacini was supported by the NSF (CNS-070820, CCF-0746117), Intel and the Sloan Foundation.

## References

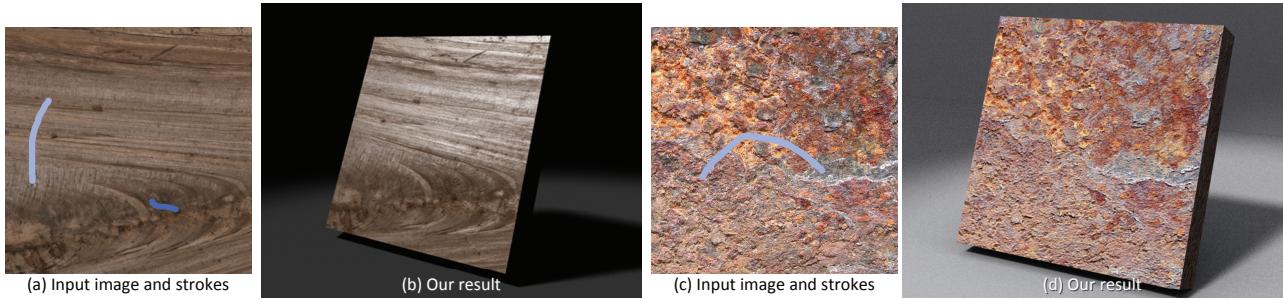
- AN, X., AND PELLACINI, F. 2008. Appprop: all-pairs appearance-space edit propagation. *ACM Trans. Graph.* 27 (August), 40:1–40:9.
- BERTALMIO, M., SAPIRO, G., CASELLES, V., AND BALLESTER, C. 2000. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, SIGGRAPH '00, 417–424.
- BOUSSEAU, A., PARIS, S., AND DURAND, F. 2009. User-assisted intrinsic images. *ACM Trans. Graph.* 28 (December), 130:1–130:10.
- CLARK, R., 2010. Crazybump. <http://www.crazybump.com/>.
- DISCHLER, J.-M., MARITAUD, K., AND GHAZANFARPOUR, D. 2002. Coherent bump map recovery from a single texture image. In *Graphics Interface*, 201–208.
- DUROU, J., FALCONE, M., AND SAGONA, M. 2008. Numerical methods for shape-from-shading: A new survey with benchmarks. *Computer Vision and Image Understanding* 109, 1, 22–43.
- FANG, H., AND HART, J. C. 2004. Textureshop: texture synthesis as a photograph editing tool. In *ACM SIGGRAPH 2004 Papers*, ACM, New York, NY, USA, SIGGRAPH '04, 354–359.
- FATTAL, R., AGRAWALA, M., AND RUSINKIEWICZ, S. 2007. Multiscale shape and detail enhancement from multi-light image collections. In *ACM SIGGRAPH 2007 papers*, ACM, New York, NY, USA, SIGGRAPH '07.
- GILET, G., AND DISCHLER, J.-M. 2010. An image-based approach for stochastic volumetric and procedural details. *Comput. Graph. Forum* 29, 4, 1411–1419.
- GLENCROSS, M., WARD, G. J., MELENDEZ, F., JAY, C., LIU, J., AND HUBBOLD, R. 2008. A perceptually validated model for surface depth hallucination. *ACM Trans. Graph.* 27 (August), 59:1–59:8.
- GOLDMAN, D., CURLESS, B., HERTZMANN, A., AND SEITZ, S. 2010. Shape and spatially-varying brdfs from photometric stereo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32, 6 (june), 1060 –1071.
- GROSSE, R., JOHNSON, M., ADELSON, E., AND FREEMAN, W. 2009. Ground truth dataset and baseline evaluations for intrinsic image algorithms. In *Computer Vision, 2009 IEEE 12th International Conference on*, 2335 –2342.
- HORN, B. K. P., AND BROOKS, M. J. 1989. *Shape From Shading*. The MIT Press.
- HORN, B. K. P. 1986. *Robot Vision (MIT Electrical Engineering and Computer Science)*, mit press ed ed. The MIT Press, March.
- KHAN, E. A., REINHARD, E., FLEMING, R. W., AND BÜLTHOFF, H. H. 2006. Image-based material editing. *ACM Trans. Graph.* 25 (July), 654–663.
- KIMMEL, R., ELAD, M., SHAKED, D., KESHET, R., AND SOBEL, I. 2003. A variational framework for retinex. vol. 52, 7–23.
- OH, B. M., CHEN, M., DORSEY, J., AND DURAND, F. 2001. Image-based modeling and photo editing. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, SIGGRAPH '01, 433–442.
- SHEN, L., TAN, P., AND LIN, S. 2008. Intrinsic image decomposition with non-local texture cues. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 1 –7.
- SHEPARD, D. 1968. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, ACM, New York, NY, USA, ACM '68, 517–524.
- TAPPEN, M. F., ADELSON, E. H., AND FREEMAN, W. T. 2006. Estimating intrinsic component images using non-linear regression. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, 1992 – 1999.
- WANG, X., WANG, L., LIU, L., HU, S., AND GUO, B. 2003. Interactive modeling of tree bark. *Computer Graphics and Applications, Pacific Conference on* 0, 83.
- WEYRICH, T., LAWRENCE, J., LENSCHE, H. P. A., RUSINKIEWICZ, S., AND ZICKLER, T. 2009. Principles of appearance acquisition and representation. *Foundations and Trends in Computer Graphics and Vision* 4, 2, 75–191.
- WU, T.-P., SUN, J., TANG, C.-K., AND SHUM, H.-Y. 2008. Interactive normal reconstruction from a single image. *ACM Trans. Graph.* 27, 5, 1–9.
- XU, K., LI, Y., JU, T., HU, S.-M., AND LIU, T.-Q. 2009. Efficient affinity-based edit propagation using k-d tree. *ACM Transactions on Graphics* 28, 5, 118:1–118:6.
- XUE, S., WANG, J., TONG, X., DAI, Q., AND GUO, B. 2008. Image-based material weathering. *Comput. Graph. Forum* 27, 2, 617–626.
- ZELINKA, S., FANG, H., GARLAND, M., AND HART, J. C. 2005. Interactive material replacement in photographs. In *Proceedings of Graphics Interface 2005*, Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, GI '05, 227–232.
- ZHANG, R., TSAI, P.-S., CRYER, J. E., AND SHAH, M. 1999. Shape from shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21, 690–706.



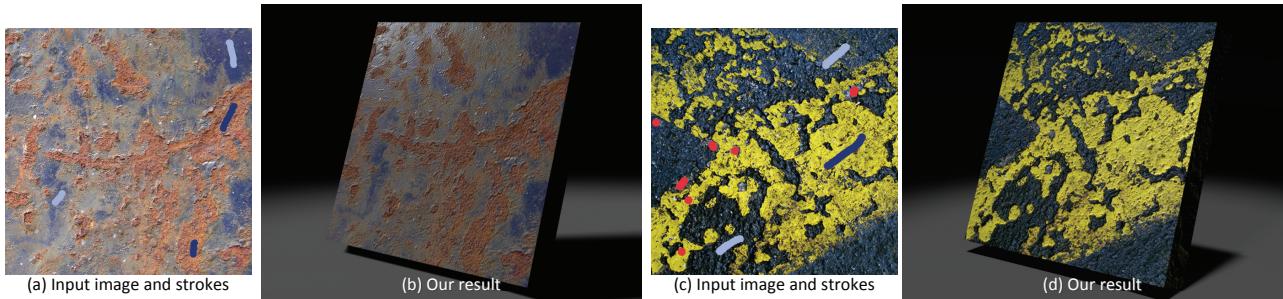
**Figure 14:** Natural materials with complex geometric details generated by our system. (a)(b) Rock, (c)(d) raw wood.



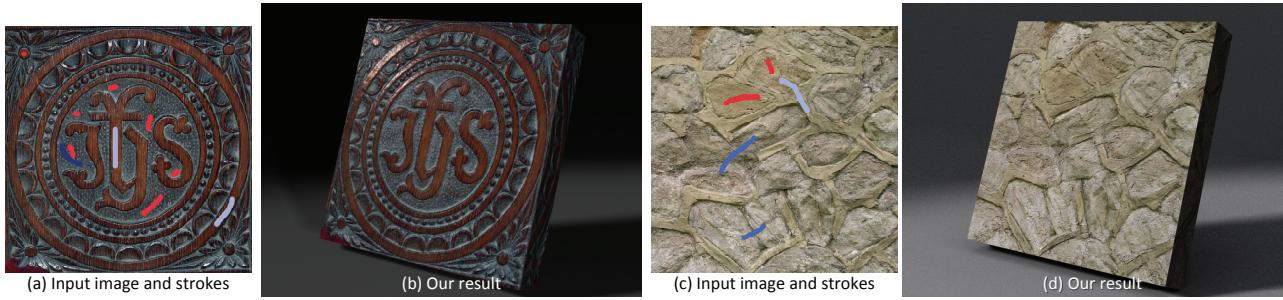
**Figure 15:** Wrinkled paper results generated by our system.



**Figure 16:** Finished wood (a)(b) and rusted metal (c)(d) generated by our system.



**Figure 17:** Rusted metal (a)(b) and asphalt with yellow paint (c)(d) generated by our system.



**Figure 18:** Carved wood plate (a)(b) and concrete rock wall (c)(d) generated by our system.