Image-Based Material Editing

Erum Arif Khan[1] Erik Reinhard[2,1] Roland W. Fleming[3] Heinrich H. Bülthoff[3]
[1] University of Central Florida, [2] University of Bristol, [3] Max Planck Institute for Biological Cybernetics

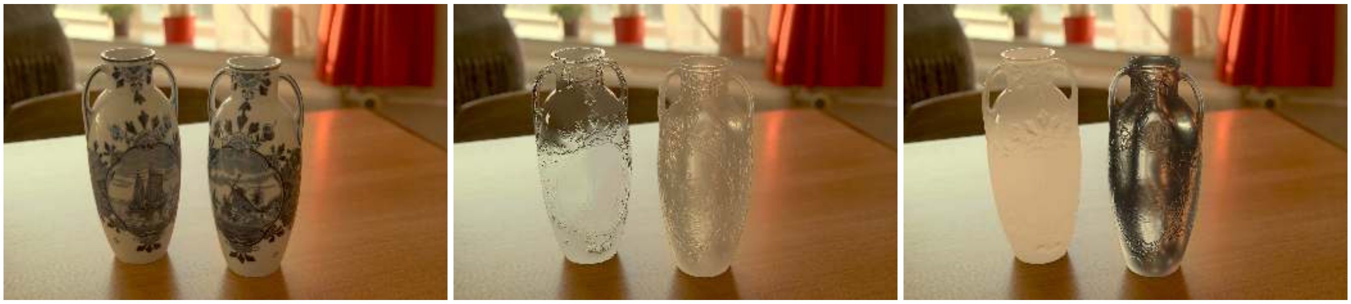


Figure 1: *Given a high dynamic range image such as shown on the left, we present techniques to make objects transparent and translucent (left vases in middle and right images), as well as apply arbitrary surface materials such as aluminium-bronze (middle) and nickel (right).*

Abstract

Photo editing software allows digital images to be blurred, warped or re-colored at the touch of a button. However, it is not currently possible to change the material appearance of an object except by painstakingly painting over the appropriate pixels. Here we present a method for automatically replacing one material with another, completely different material, starting with only a single high dynamic range image as input. Our approach exploits the fact that human vision is surprisingly tolerant of certain (sometimes enormous) physical inaccuracies, while being sensitive to others. By adjusting our simulations to be careful about those aspects to which the human visual system is sensitive, we are for the first time able to demonstrate significant material changes on the basis of a single photograph as input.

CR Categories: I.4.10 [Computing Methodologies]: Image Processing and Computer Vision—Image Representation

Keywords: Image-based material editing, Image-based lighting, High dynamic range imaging

1 Introduction

Suppose you have a photograph of some object — say a bronze sculpture or a porcelain vase — and you want to know how it would look if it were made out of some completely different material, such as glass, plastic, or wax. Given nothing more than a photograph of the object, is it possible to modify the appearance of the material in a visually plausible way?

To do this in a physically correct way would involve accurately estimating the lighting, and 3D geometry of the object from the image; a task that not even state-of-the-art computer vision can achieve to a high degree of precision. However, we have set as our goal not a physically accurate reconstruction, but a visually satisfying illusion of material transformation. Our approach relies on the fact that human vision tolerates certain deviations from reality. Indeed, sometimes even enormous inconsistencies go un-noticed by observers [Ostrovsky et al. 2005]. An example is change-blindness, the inability to detect changes in the environment [Simons and Levin 1997].

Recent research has suggested the possibility of changing material properties in an image using simple image processing operations [Adelson 2001; Fleming and Bülthoff 2005; Motoyoshi et al. 2005]. One of the key contributions of this paper is to show how we may exploit limitations of human vision regarding shape perception [Belhumeur et al. 1999; Koenderink et al. 2001], as well as its inability to reliably estimate illumination [te Pas and Pont 2005a; te Pas and Pont 2005b] for the purpose of transforming materials in images.

Because of the inherent ambiguity of visual cues, the human visual system makes certain assumptions when reconstructing a scene from an image. When these assumptions are violated, perception breaks down and either a visual illusion occurs, or scene recognition is negatively affected [Metzger 1975; Tarr et al. 1999; Lawson et al. 2003].

With this in mind, we present our work which aims to take as input a photograph of an object as well as an alpha matte outlining this object, and produce as output a new photograph where the object has been given an entirely new material. The transformations presented in this paper range from applying a texture to the surface of an object, to the application of any arbitrary bi-directional reflectance distribution function (BRDF). In addition, we present techniques to handle light-matter interactions that are not well modeled with BRDFs, such as transparency and translucency. Examples of some of our results are shown in Figure 1. The use of this work lies in the facilitation of digital content creation for art and design, as well as visual effects.

1.1 Previous Work

Several related image editing techniques exist [Oliveira 2002]. For instance, techniques have been demonstrated to change the shape, color and illumination of objects depicted in images [Oh et al. 2001]. The lighting in a scene can also be manipulated [Eisemann and Durand 2004]. Further, images can be used to light 3D geometry [Debevec 1998; Debevec 2002]. Perhaps closest to our approach is Fang and Hart's object re-texturing technique, which allows objects' macro structure to be altered [Fang and Hart 2004; Zelinka et al. 2005]. However, our work focuses on changes to objects' micro structure that would normally be modeled with functions such as bi-directional reflectance distribution functions (BRDFs) and related models (BSSRDFs for instance).

All other techniques require more than a single image to make any material changes. For instance, by using precomputed information, objects with arbitrary materials (such as glass) may be placed in an image [Zongker et al. 1999; Chuang et al. 2000].

1.2 Problem Definition

For the purpose of manipulating images, insight into image formation may be gained from studying the rendering equation [Kajiya 1986]:

$$L(x,y) = L_e + \int_\Omega f_r\left(\Theta_i,\Theta_o\right)\, L_i\left(\Theta_i\right)\, \cos\Theta_i d\omega_i \qquad (1)$$

For non-emitting objects, we can set $L_e = 0$. Thus, the light L focused onto a pixel (x,y) is formed by taking the corresponding nearest surface point in the scene, and constructing a hemisphere Ω around the surface normal associated with this point. Then, light from all directions Θ_i covered by Ω is reflected toward the pixel by an amount governed by the reflectance function f_r.

We are interested in manipulating pixel values $L(x,y)$ such that objects appear to be made of different materials. Thus, we aim to take an existing image L and transform it into a different image L' such that:

$$L'(x,y) = \int_\Omega f'_r\left(\Theta_i,\Theta_o\right)\, L_i\left(\Theta_i\right)\, \cos\Theta_i d\omega_i \qquad (2)$$

However, in the above two equations there are several unknowns: the orientation of the hemisphere Ω for each point on the object, the reflectance function f_r, nor the incident lighting L_i are available. All we know are the pixel values L and the desired reflectance function f'_r. Thus, computing L' is not possible without making further assumptions and simplifications. The key point of this paper is that we will show how to simplify the problem such that although the physics of light is not strictly followed, the results remain plausible. We achieve this by exploiting limitations of human vision in a consistent manner.

First, for each pixel belonging to the object we wish to alter, we require an approximation of Ω, or equivalently the surface normal $\mathbf{n}$. Details of both the problem and our solution are presented in Section 2.

It is possible to approximate the incident lighting for all directions on the basis of the partial knowledge we have given background in the input image. Dependent on the nature of the desired reflectance function f'_r, the approximation needs to be more or less precise. In particular, if f'_r is relatively diffuse, the approximation can be relatively coarse [Ramamoorthi and Hanrahan 2001]. In addition, human perception is such that relatively coarse approximations are sometimes tolerated. For instance, Ostrovsky et al [2005] have found that deliberate inconsistencies in the direction of illumination are very difficult to detect. Other experiments have confirmed that humans cannot in general estimate illumination reliably [te Pas and Pont 2005a; te Pas and Pont 2005b]. This implies that inaccuracies in illumination are likely to go unnoticed — a feature we actively exploit in our work.

The unknown values of L_i can be classified into two groups: values that are outside the image, for instance behind the photographer, and values coming from behind the object we wish to alter. For these two cases we adopt different solutions because the number of missing values is different in each case, and therefore the level of approximation can be different. Together these yield a set of incident lighting values L'_i, as discussed further in Section 3.

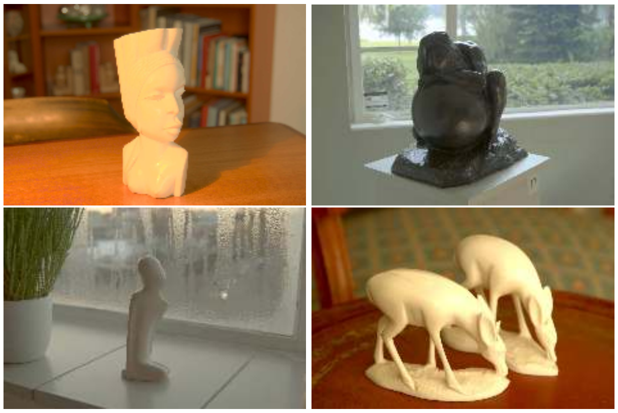


Figure 2: *Test images used in this paper.*

After computing approximate values for the object's surface shape as well as the incident lighting, a new reflectance function f'_r can be chosen and a new image may be rendered:

$$L'(x,y) = \int_{\Omega'} f'_r\left(\Theta_i,\Theta_o\right)\, L'_i\left(\Theta_i\right)\, \cos\Theta_i d\omega_i \qquad (3)$$

Finally, the types of material changes we are currently interested in involve re-texturing of objects, increasing or decreasing specularity, adding transparency and translucency, as well as replacing an object's BRDF. Our work on transforming materials on the basis of a single input image is novel, and constitutes a significant generalization to object texturing [Fang and Hart 2004].

1.3 Overview

The general approach we adopt is first to acquire a high dynamic range (HDR) photograph of the object of interest by concatenating multiple exposures [Debevec and Malik 1997]. Most of the techniques we present also work to some extent with standard (low dynamic range) images. However, HDR input yields more robust results. The gradients used to estimate Ω' are smoother if derived from HDR images; highlight identification is also facilitated. Importantly, estimates of L_i can be directly used to light the object, i.e. the background pixels are suitable for techniques akin to image-based lighting [Debevec 2002; Reinhard et al. 2005b].

Second, we create an alpha matte to separate the object from the background. Although we have created our mattes by hand, sophisticated techniques have recently become available to assist the user in this task [Li et al. 2004; Rother et al. 2004], or go toward automatic alpha matte extraction [McGuire et al. 2005; Reinhard and Khan 2005]. The pixels forming the object will be altered to fit the object with a new material. By aligning our image processing with the object's boundaries, the illusion of a different material may be created.

In the remainder of this paper, we first discuss the approaches we have found useful for estimating Ω' and L'_i (Sections 2 and 3), followed by the various material transforms we have achieved (Section 4). Some caveats and limitations are discussed in Section 5, while general conclusions are drawn in Section 6. Edited images are shown throughout this paper, using input images depicted in Figure 2.

2 Object Shape (Ω')

A hemisphere of directions Ω' can be estimated from pixels belonging to the object by first estimating the distance between the viewer and the object, i.e. the depth. Gradients in depth are then used to compute a surface normal $\mathbf{n}$ for each pixel. Each of these steps are discussed in the following subsections.

2.1 Depth Recovery

Computing a depth map from a single image is the classic shape-from-shading problem. Unfortunately, as this problem is severely under-constrained, good solutions for arbitrary images do not exist [Zhang et al. 1999]. Solutions have been developed for highly constrained cases [Kang 1998; Igarashi et al. 1999; Oh et al. 2001]. However, to date no general solution has been developed.

Fortunately, by carefully exploiting certain assumptions of human vision, we can get away with highly approximate depth reconstructions. A surprisingly effective starting point is to assume that surface depth is inversely related to image intensity. There is experimental evidence that human observers treat lower luminance values as more distant surface locations [Langer and Bülthoff 2000], and indeed, this is loosely what algorithms for shape from shading under diffuse illumination rely on [Langer and Zucker 1994]. Although the resulting surface reconstructions are far from perfect, there are two main reasons why the approximations are sufficient for our application.

First, certain errors in the depth estimates are visually masked by subsequent processing. For example, highlights in the original image appear as protrusions in the depth map. However, in the final output, highlights are generally pasted back onto these surface locations, effectively masking them. Further, because we do not dramatically alter the illumination of the object, spurious hills and valleys do not generate tell-tale shadows, which would undermine the effect.

Second, visual cues to 3D shape are invariant across a wide class of geometrical transformations. This means that as long as the distortions of the reconstructed shape belong to these transformations, they will be undetectable by the visual system. Probably the most important in this context is the bas-relief ambiguity, which makes it difficult or impossible to distinguish surfaces that are related by an affine transformation along the line of sight [Belhumeur et al. 1999; Koenderink et al. 2001]. The bas-relief ambiguity allows shearing in the depth map due to oblique illumination to go unnoticed, provided the line of sight is not altered, as in our application.

Further, the generic view principle helps the illusion: the visual system assumes by default that the object is not viewed from special viewpoints, and that there is nothing peculiar about the geometry [Koenderink and van Doorn 1979; Freeman 1994; Freeman 1996].

We therefore aim to derive a locally-consistent depth-map from the object's luminance distribution, with higher luminance values specifying the parts of the object closer to the observer (i.e. following the dark-is-deep paradigm). Our approach consists of several steps. First, we compute the luminance of each pixel [ITU 1990]:

$$L(x,y) = 0.213\, R(x,y) + 0.715\, G(x,y) + 0.072\, B(x,y) \quad (4)$$

An initial crude depth map $d(x,y)$ may be computed by simply setting $d(x,y) = L(x,y)$. However, there are several problems with this approach. First, an object may be textured, leading to errors in the depth map. Second, the shape implied by the depth values may bulge in the direction of any light sources, as argued above. Third, highlights may cause significant distortions in the depth map — a problem addressed in Section 4.1.

The luminance variation due to shading is useful as an indicator of shape, whereas treating textured areas as valid depth information is only useful for special effects (such as making an object appear embossed). The discontinuities in luminance due to silhouettes ought to be preserved. We therefore need some means to isolate shading from other classes of intensity variation.

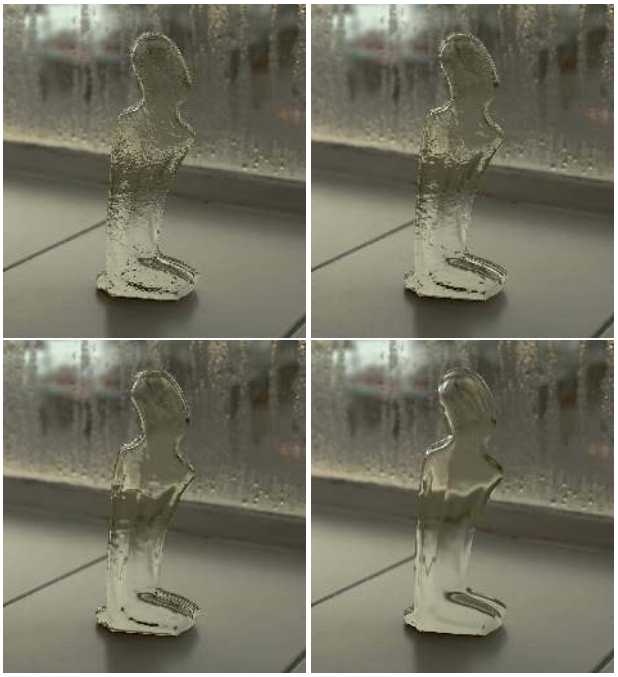


Figure 3: *The value of $\sigma_{spatial}$ controls the smoothness of the depth-map. Here, its value is doubled from 0.001 to 0.002, 0.004 and 0.008 times the width of the image.*

While disentangling luminance variations due to several different processes is once more an under-constrained problem, in practice shading, textures and patterns tend to occur at different frequencies and/or luminance levels. We therefore require a filter that is tunable to spatial scales as well as luminances. The bilateral filter directly fits these requirements [Tomasi and Manduchi 1998]:

$$d(x,y) = \frac{\sum_u \sum_v b(x,y,u,v)\; L_s(x-u, y-v)}{\sum_u \sum_v b(x,y,u,v)} \quad (5)$$

$$\begin{aligned} b(x,y,u,v) = {} & G_{\text{spatial}} \left(\sqrt{(x-u)^2 + (y-v)^2},\; \sigma_{\text{spatial}} \right) \times \\ & G_{\text{L}} \left(L_s(x-u, y-v) - L_s(x,y),\; \sigma_{\text{L}} \right) \end{aligned} \quad (6)$$

The bilateral filter kernel b is composed of two Gaussian filter kernels, G_{spatial} and G_{L} with kernel widths of σ_{spatial} and σ_{L}, affording user control over the recovered object shape. For larger values of this parameter, small details are increasingly suppressed while the overall round shape is emphasized. Further, control over surface indentations is afforded by σ_{spatial}, as shown in Figure 3. Dependent on the specific shape of the object different parameter settings may thus yield satisfactory depth maps.

The bilateral filter is typically used in the logarithmic domain [Durand and Dorsey 2002]. While $\log L$ is a better behaved quantity than L, the range of values produced by taking the logarithm is unconstrained. A closer model of early human vision involves employing sigmoidal compression, which produces behavior similar to log luminance in the mid-range of values, but additionally has a range of output values that is strictly limited between 0 and 1:

$$L_s(x,y) = \frac{L^n(x,y)}{L^n(x,y) + \sigma^n} \quad (7)$$

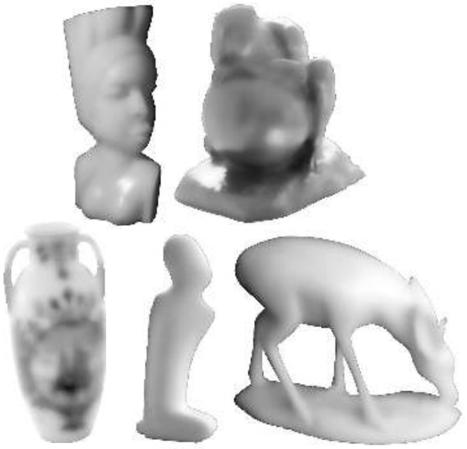


Figure 4: *Depth maps for some of the images used in this paper.*

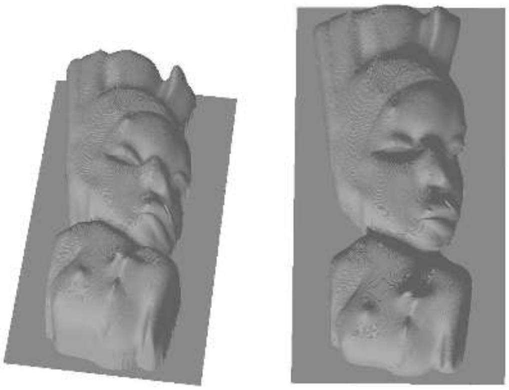


Figure 5: *Our approximate depth recovery technique is helped by the bas-relief ambiguity, provided the viewpoint is not altered.*

The exponent n is set to 0.73 by default, a value common in modeling for instance photoreceptor responses [Naka and Rushton 1966]. The semi-saturation constant σ determines which luminance values are mapped in near-logarithmic fashion. A common and sensible value is to choose σ equal to the log average luminance of the object's pixel values:

$$\sigma = \exp\left(\frac{1}{N}\sum_{x,y}\log\left(L(x,y)\right)\right) \tag{8}$$

Compressing the input L as above helps make the results more robust, and additionally facilitates the choice of values for σ_{L}.

Some depth maps are shown in Figure 4, showing that the depth is generally a smooth function following the luminance distribution. As a consequence, for objects lit from the side, the depths are noticeably skewed, as seen in the top left and bottom right images. These maps deviate substantially from the true object geometry, and would not be satisfactory for general use. However, subsequent treatment of the depth maps, combined with the masking effects occurring in the final results mean that for our application, the gross inaccuracies in shape reconstruction do not interfere with the visual effect; indeed, they are largely undetectable. Finally, Figure 5 shows that distortions in depth maps thus obtained are not perceived as long as the viewpoint is not altered, as predicted by the bas-relief ambiguity.

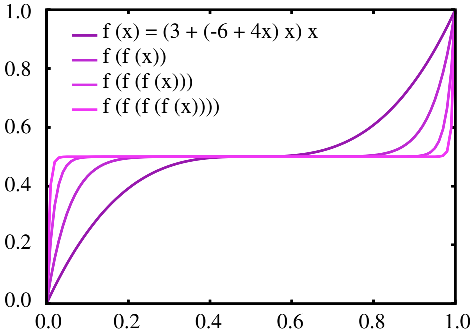


Figure 6: *A family of gradient reshaping functions.*

2.2 Gradients and Surface Normals

Due to the sigmoidal compression applied earlier, the range of values of d is strictly limited. This means that gradients near silhouette edges are too small. We therefore invert the sigmoidal compression function (7) first:

$$d'(x,y) = \left(-\frac{\sigma^n\, d(x,y)}{d(x,y)-1}\right)^{1/n} \tag{9}$$

The recovered depth $d'(x,y)$ can now be used to estimate local gradients and surface normals. In two dimensions, the gradient field $\nabla d'(x,y)$ is defined in terms of neighboring depth values:

$$\nabla d'(x,y) = \big(d'(x+1,y) - d'(x,y),\ d'(x,y+1) - d'(x,y)\big) \tag{10}$$

As luminance values were used to derive these gradients, we can not expect that the range of gradient values is directly commensurate with the shape of the object. In addition, there is no reason to believe that the conversion from luminance values to gradient values should be linear. We have empirically determined a non-linear spline function which reshapes the gradient field to boost small and large gradients, attenuating intermediate gradients, as shown in Figure 6. In addition the amount of reshaping can be adjusted by recursively applying this function (also shown in this figure). Typically, we apply this function between one and four times, dependent on the input data.

The resulting gradient field can thus be used to compute 3D surface normals, or may be applied directly to warp textures, as shown in Section 4. We compute 3D surface normals using the following gradient vectors:

$$\mathbf{g}_x = \left[1,\ 0,\ \nabla_x d'\right]^T \tag{11}$$

$$\mathbf{g}_y = \left[0,\ 1,\ \nabla_y d'\right]^T \tag{12}$$

The surface normal is given by the cross product $\mathbf{n} = \mathbf{g}_x \times \mathbf{g}_y$, which must be normalized if used in a ray tracing environment.

3 Incident Light (L_i)

Given an (approximated) representation of 3D shape, from (3) we see that a successful manipulation of an object's material requires knowledge of the incident lighting L_i[1]. Given a single photograph,

[1] In this and subsequent sections, we will retain the notation L_i when discussing incident lighting. However, it should be noted that while this symbol is commonly used to denote luminance values, we will use the same symbol to indicate RGB triplets where-ever this does not lead to confusion.

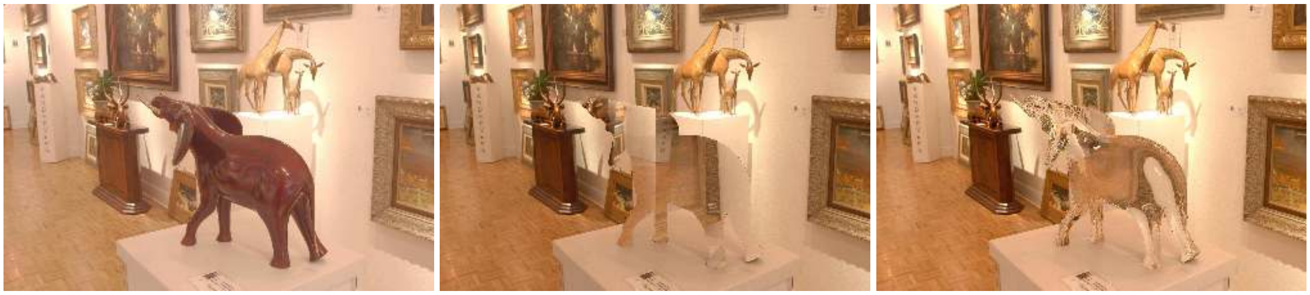


Figure 7: *Removing the elephant from the original image (left) is shown in the middle. The masking effect of further processing is demonstrated with the simulation of transparency (right).*

all the background pixels together provide information for a subset of the directions we are interested in. For all other directions, namely those behind the object, and those outside the image, we will have to substitute an approximation L_i'.

Our material edits can be grouped into full 3D simulations, as well as 2D remapping functions. For the 3D simulations, we need a value L_i' for every possible direction. For the 2D remapping schemes, we may restrict ourselves to a set of directions that correspond to existing image pixels. However, in both cases an impression of what lies behind the object is desirable. Solutions for both problems are outlined in the following subsections.

3.1 Light from Behind the Object

To determine what light may come from behind the object, we may employ an image inpainting algorithm [Bertalmio et al. 2000; Drori et al. 2003; Sun et al. 2005]. The input to such algorithms is typically the image itself as well as an alpha matte which defines the hole that needs to be replaced with a new background. The result is an image with the gap filled with plausible background pixels.

While the more advanced algorithms may yield more plausible results, after applying them to (3) the quality of the inpainting algorithm is largely masked due to further processing, contorting and warping. In the following, we demonstrate that for this particular application the inpainted region only needs to capture the correct image statistics, which is a weaker requirement than those met by full inpainting algorithms. This affords a much simpler algorithm, while still yielding satisfactory results.

An important observation is that for the background to be useful as a complex light source, its precise configuration is unimportant. However, the statistical properties of the inpainted pixels ought to be similar to the remainder of the environment[2]. In particular the image's power spectral slope should be preserved [Ruderman 1997; van der Schaaf 1998; Torralba and Oliva 2003; Reinhard et al. 2004], as well as its color composition.

The most straightforward approach that would achieve these goals is to copy large blocks of pixels from the background into the gap formed by removing the object. This approach would minimize the introduction of new edges, and thus preserve the aforementioned image statistics. To minimize the introduction of artifacts along the gap's boundary, we copy pixels both from the left and from the right parts of the image, and blend them according to how much they have been displaced.

To begin, we find the left-most and right-most extent of the gap's boundary, and call the corresponding x-coordinates x_{min} and x_{max}. The value assigned to all pixels inside the gap is then L_i':

$$L_i'(x,y) = w_1\, L(2\, x_{\text{min}} - x, y) + w_2\, L(2\, x_{\text{max}} - x, y) \tag{13}$$

$$w_1 = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \tag{14}$$

$$w_2 = \frac{x_{\text{max}} - x}{x_{\text{max}} - x_{\text{min}}} \tag{15}$$

In case the x-coordinates index outside the image's boundaries, we use wrap-around within the regions $[0, x_{\text{min}}]$ and $[x_{\text{max}}, X_{\text{max}}]$ (with X_{max} being the image's horizontal resolution). The effect of this operation is given in Figure 7, showing the original image (left), the result (middle), and a simulation of transparency to demonstrate the masking effect of further processing.

3.2 Reconstructing a 3D Environment

The image with the filled-in background can be used directly in 2D mapping operations as outlined in Section 4. For mapping a full BRDF onto an object, it is necessary to reconstruct a full spherical 3D environment. In traditional image-based lighting approaches, photographs of mirrored spheres or the output of scanning panoramic cameras are mapped to a spherical environment [Debevec 2002]. In our case, an approximate mapping can be constructed between the hole-filled photograph and the desired environment.

The plane of the image in three-dimensional space may be thought of as partitioning the environment into two half spaces. Our environment map is created by extruding a circular part of the background image to form a hemisphere behind the image plane, and again to form a hemisphere in front of the plane, as illustrated in Figure 8. This leads to the following computation of direction vectors for given pixel coordinates.

The center of the circle $[x_c\ y_c\ 0]^T$ is given by the middle of the image, and this circle is made as large as possible, i.e. the radius R is $\min(x_c, y_c)$. Next, we construct for each pixel within this circle, a normalized coordinate pair $[x_n\ y_n]$:

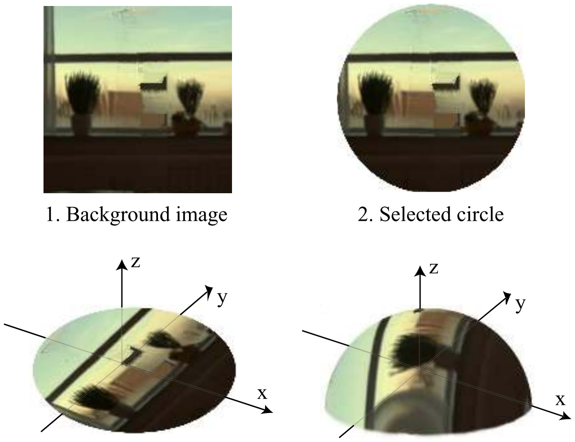
$$[x_n,\ y_n]^T = \left[\frac{x - x_c}{R},\ \frac{y - y_c}{R}\right]^T \tag{16}$$

A pixel $[x_n\ y_n]^T$ can now be mapped to a direction vector $\mathbf{d}$ with:

$$\mathbf{d} = \left[x_n,\ y_n,\ 1 - \sqrt{(x_n)^2 + (y_n)^2}\right]^T \tag{17}$$

These direction vectors are equivalent to the incoming and outgoing directions Θ_i and Θ_o in (3). This means that for every desired direction, a corresponding index into the hole-filled image can be computed for the purpose of solving (3).

[2] This is not generally the case for inpainting algorithms, but merely a consequence of our specific application.



1. Background image

2. Selected circle

3. Placed in image plane

4. Extruded to form half the environment map

Figure 8: *The background image with the object removed is used to create an HDR environment map. First, a circle is cut from the middle of the image. This is then placed in the image plane, and then extruded to become half the environment. The image is also pulled out in the opposite direction to form the other half of the environment.*

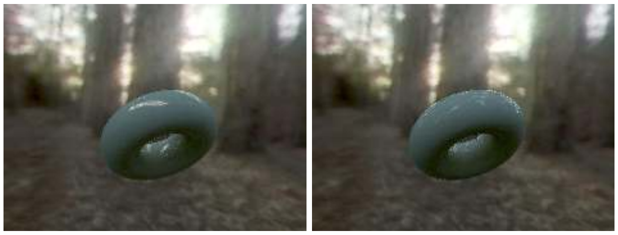


Figure 9: *Plastic torus rendered using techniques described in Section 3.2 (left), and standard image-based lighting (right).*

While this mapping distorts the environment leading to physically inaccurate lighting, the distortion is locally consistent, and therefore largely unobjectionable to the human visual system. This is demonstrated in Figure 9, where we used this mapping to render a torus. This figure also shows the same torus rendered with standard image-based lighting techniques using a light probe. While the differences are clear, each image on its own looks plausible.

As such, we have recovered both plausible surface normals for the object itself, as well as plausible lighting to be used to relight the object. We therefore have all the preliminaries necessary to perform material transformations.

4 Material Transforms

Editing an object's material properties can take several different forms, depending on the desired target material. For instance, increasing or decreasing the gloss on an object would require only a straightforward luminance remapping, whereas replacing the BRDF of an object would mean a full evaluation of (3). Retexturing an object as well as creating transparent and translucent objects require techniques that sit somewhere in-between these extremes in terms of computational complexity. Hence, we discuss each of these material transforms separately in the following sections.

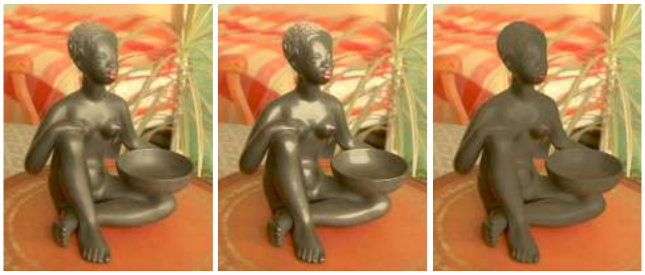


Figure 10: *The luminance values of the statue (left) were remapped to create a more specular (middle) and more diffuse (right) image.*

4.1 Glossy Surfaces

For objects that already exhibit some amount of glossiness, an exponential remapping of luminance values may increase or decrease the perception of gloss [Fleming and Bülthoff 2005]. To increase specularity, the following mapping yields acceptable results:

$$L' = \begin{cases} L_{\text{min}} + (L_{\text{max}} - L_{\text{min}}) \left(\alpha \, \dfrac{L - L_{\text{min}}}{L_{\text{max}} - L_{\text{min}}} \right)^{\beta} & L > L_{\text{min}} \\ L & L \leq L_{\text{min}} \end{cases} \tag{18}$$

where L_{max} is the maximum luminance of all object pixels, and L_{min} is the minimum luminance that can still be considered part of the highlight. This is a user parameter, which may alternatively be estimated with the highlight detection algorithm presented in Section 4.4. To steer the amount of specularity, the user parameters α and β are available (default values: $\alpha = 0.05$ and $\beta = 20$).

To make an object appear more diffuse, we use the following remapping:

$$L' = \begin{cases} L_{\text{hmax}} + (L_{\text{min}} - L_{\text{hmax}}) \left(\dfrac{L - L_{\text{hmax}}}{L_{\text{max}} - L_{\text{hmax}}} \right)^{\beta} & L > L_{\text{hmax}} \\ L & L \leq L_{\text{hmax}} \end{cases} \tag{19}$$

with β typically set to 0.05. The value of L_{hmax} is computed by taking the luminance associated with the peak of the histogram.

Examples of both remappings are shown in Figure 10. For high-gloss objects, the latter remapping may also be useful as a preprocessing step before recovering Ω' (Section 2). This would permit the shape recovery to be less sensitive to highlights.

4.2 BRDF Replacement

With estimates of surface normals $\mathbf{n}$ available for each point on the object, and having constructed a mapping from a rectangular image to a full 3D environment to provide values for L'_i, we can evaluate (3) directly. While our approach follows common practice in image-based lighting [Debevec 2002], the remainder of this section is devoted to detailing our specific implementation.

To sample the environment for each point on the surface of the object, a fast and efficient sampling scheme is required. One of the most efficient ways to generate sampling points for our environment is afforded by Ostromoukhov's importance sampling technique [Ostromoukhov et al. 2004]. Using this algorithm as a preprocess, we typically find up to 400 sampling coordinates $[x_n \; y_n]^T$ in our background image, leading to an equal number of values of L'_i. Each of these sampling points has an associated 3D direction vector $\mathbf{d}$, computed with (17). For each point on the object, we sample only those importance samples for which $\mathbf{d} \cdot \mathbf{n} > 0$, i.e. those samples that lie within the hemisphere Ω'.

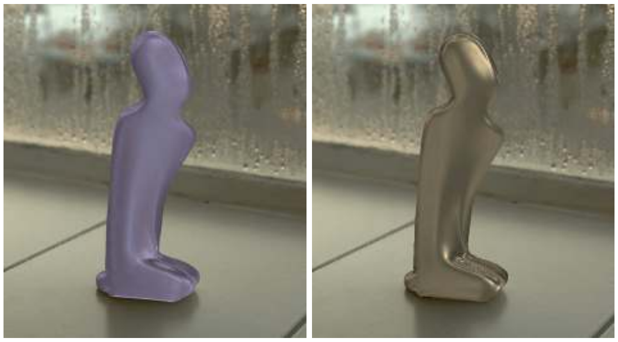


Figure 11: *Examples of mapping arbitrary BRDFs onto an object.*

A common convention is to evaluate BRDFs in a coordinate system which is aligned with the surface normal. This is achieved by rotating the view vector over an angle $\alpha = \mathbf{n} \cdot \mathbf{v}$ around an axis $\mathbf{a} = \mathbf{n} \times \mathbf{v}$. For each sample point in the environment we wish to evaluate, its associated direction vector $\mathbf{d}$ is rotated over the same angle, giving $\mathbf{d}_\alpha$. Quaternion rotation should be used to avoid gimbal lock [Hart et al. 1994].

Although our application will work with any BRDF model, we have chosen Matusik et al's measured BRDFs [Matusik et al. 2003] since these were captured in high dynamic range, and are therefore directly compatible with our system. Thus, the equation we evaluate to fit an object with a new material is:

$$L'(x,y) = \sum_{\{\mathbf{d}|\mathbf{d}\cdot\mathbf{n}>0\}} f_{\text{Matusik}}\left(\mathbf{d}_\alpha,\, \mathbf{v}_\alpha\right)\; L'_i\left(\mathbf{d}\right)\; \mathbf{d}\cdot\mathbf{n} \qquad (20)$$

Some results of this approach are shown in Figure 11.

4.3 Re-texturing

The machinery outlined in Section 2 is sufficient to allow objects to be re-textured. In particular, the gradient field $\nabla d'$ can be used to estimate a warping of an arbitrary texture $\mathbf{T}$, which can then be applied to the object. To steer the amount of apparent curvature, we introduce a linear scale factor s, and compute texture indices $[t_x,\ t_y]^T$ as follows:

$$t_x = x + s\,\nabla_x d' \qquad (21)$$

$$t_y = y + s\,\nabla_y d' \qquad (22)$$

The indices are taken modulo the texture size and are used to compute the new pixel colors of the object. Assuming a pixel $[x\ y]$ belonging to the object has an RGB color triplet denoted $\mathbf{C}(x,y)$, the new color $\mathbf{C}'(x,y)$ can be derived from its original color $\mathbf{C}(x,y)$ and the texture look-up value $\mathbf{T}(t_x,t_y)$. For extra flexibility, we follow Smith and Blinn [1996] and add two user parameters, namely a scalar $f \in [0,1]$, and a new object color $\mathbf{G}$, leading to the matting equation:

$$\mathbf{C}'(x,y) = (1-f)\;\mathbf{G}\;\mathbf{T}(t_x,t_y) + f\;\mathbf{C}(x,y) \qquad (23)$$

The parameter f linearly interpolates between the original object color and the texture mapped color. Results for $f = 0$ and $\mathbf{G} = [1\ 1\ 1]^T$ are shown in Figure 12. This result is included to show that our approach is capable of reproducing Fang and Hart's [2004]. It also forms the basis for modeling transparency and translucency.

4.4 Transparency and Translucency

A key factor affecting the appearance of a transparent object, is its geometric shape. It determines how light incident from the background is refracted, leading to the characteristic distortions associated with transparent objects. With an appropriate estimation of the object's shape, such as Ω' (Section 2), it should therefore be possible to warp the background image L'_i such that it gives the appearance of a transparent object.

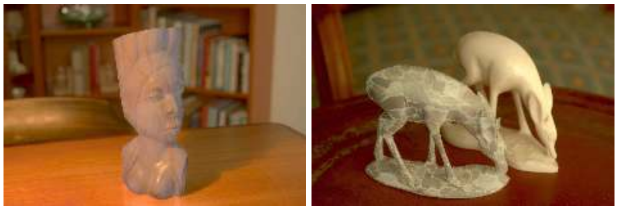


Figure 12: *A marble texture is applied to the left image, whereas a stone texture is applied to the right image.*

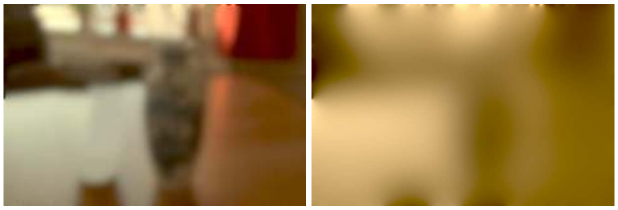


Figure 13: *A texture suitable for simulating transparency (left, see also middle panel of Figure 1, and translucency (right, as well as the right panel of Figure 1).*

Thus, rather than apply an arbitrary texture to an object, we may use the approach of Section 4.3 to create the illusion of transparency. We have found that as long as several conditions are met, convincing results may be obtained. First, the warp needs to be consistent with Ω', which in turn needs to be at least locally consistent with the actual geometry Ω. Note that this is a weaker requirement than having access to the full volumetric geometry, which would have made the problem intractable. Second, the color composition and natural image statistics of the background image should be as outlined in Section 3.1. Our basic approach to create transparent and translucent objects is then given by (23), whereby $\mathbf{T} = L'_i$ is called the background map (see footnote 1 on page 4).

There are various extensions to this basic solution, leading to different types of transparent materials, from clear glass to frosted glass, to translucency. Each of these involve pre-processing of $\mathbf{T}$. In addition, we have found value in detecting the highlights in the original object, and preserving them. Both topics are discussed in the remainder of this section.

To simulate objects that are not perfectly clear, we convolve the background map $\mathbf{T}(t_x,t_y)$ with a Gaussian filter, as for example shown in Figure 13 (left). A much larger filter kernel is used to simulate sub-surface scattering in translucent objects.

The latter type of object additionally receives a color adjustment to simulate wavelength-dependent absorption of light by the material. This adjustment may be achieved by employing a color transfer algorithm [Reinhard et al. 2001] whereby the background map is given the same color statistical properties as the object's pixels. A typical treatment to simulate translucent materials is shown in Figure 13 (right).

In addition to manipulating $\mathbf{T}(t_x,t_y)$, dark glass may be simulated using an exponential a after normalizing the matting function (23):

$$C'_{\text{dark}} = \left(\mathbf{C}'(x,y)/C_{\max}\right)^a \qquad (24)$$

An exponent a of around $1.2 - 1.4$ produces realistic attenuation. The user parameter f in (23) controls the degree to which the object

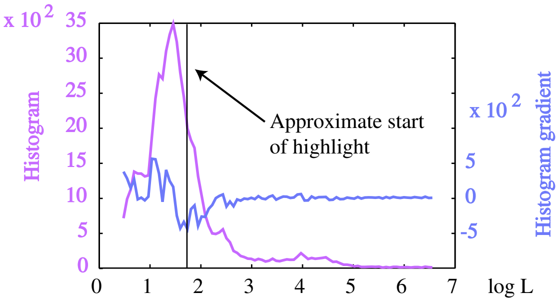


Figure 14: *Histogram of an HDR image and its derivative.*

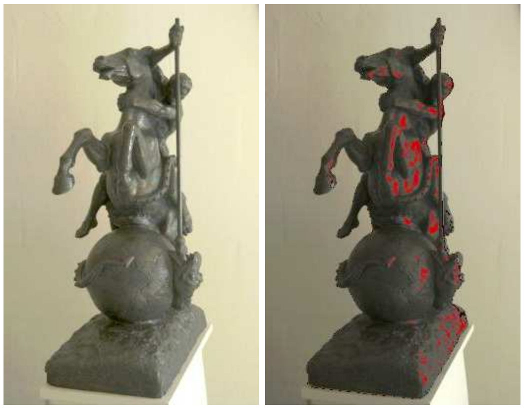


Figure 15: *Using the minimum in the derivative of the histogram, highlight pixels of the left image are detected, and for demonstration purposes painted red in the right image.*

is transformed — $f = 0$ produces a completely transparent object, whereas $f = 1$ yields the original object. Inbetween values model semi-transparent objects.

A final refinement of our transparency/translucency algorithm involves detection and placement of highlights. For objects with relatively sharp highlights, it may be beneficial to retain these in the transparent and translucent versions of the same object, since highlights generally enhance realism [Fleming and Bülthoff 2005] and aid the perception of 3D shape [Todd and Mingolla 1983; Blake and Bülthoff 1990; Blake and Bülthoff 1991].

As we only require detection of highlights on pixels belonging to the object, our highlight detection task is relatively simple. In particular, while highlights are not necessarily the brightest part of a scene, they do tend to be the brightest part of a single object. Furthermore, in conventional imaging, highlights are frequently burned out ("clipped") due to limitations of the capturing device. By using HDR imaging, highlights may be captured accurately, which also simplifies their detection. Rather than resort to one of several sophisticated schemes [Klinker et al. 1988; Drew 1994], we therefore propose a simple algorithm.

We will assume that the brightest pixels of the object form the highlights. The histogram of object pixels is analyzed to determine the intensity of the darkest pixel that may still be considered part of the highlight. Intensity histograms of real objects typically show a large peak for the darker pixels, and a long tail forming the highlight pixels (Figure 14). We have found that for many objects, the minimum of the histogram's derivative is a reasonable approximation of the start of the highlight, also shown in Figures 14 and 15. This method is useful as a starting point, although on occasion the user may wish to manually reduce the size of the highlight somewhat.

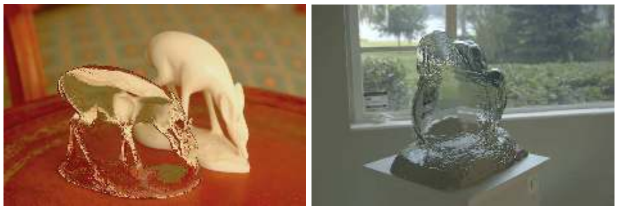


Figure 16: *Transparency results demonstrated on different objects.*

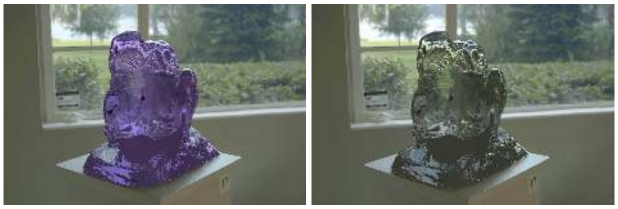


Figure 17: *Colored glass (left) and darkened glass (right) simulated with a linear attenuation to achieve color (***G** *in 23), and non-linear scaling to simulate dark glass (24). The refractions in these images are different from the ones shown in Figure 16 due to a different choice of filter kernel sizes of the bilateral filter, as well as a different choice of scale factor s.*

Once we have detected highlights in the original object, we replace the corresponding pixels in the new image by the pixel values in the original image.

The results of these manipulations are shown in Figure 16. Various types of glass may be simulated, including colored and dark glass, as demonstrated in Figure 17. Translucency is demonstrated in Figure 18.

5 Limitations

As our techniques necessarily employ approximations, and rely on the human visual system filling in missing pieces of information, there are some limitations. Each of these limitations afford interesting further research.

First, we assume that the input image is given as a high dynamic range image. As the method involves the computation of gradients, it is assumed that quantized low dynamic range data will not be amenable to the recovery of suitable gradients for further processing. Figure 19 shows a comparison, where the middle exposure of a sequence of nine images was processed separately. The high dynamic range result shows appropriate curvature in the object, whereas the shape has become flat in the low dynamic range image, and loses the appearance of transparency. Skilled photography, in combination with more forgiving scenes, may yield better results with LDR input.

The input is further limited to objects with clearly defined boundaries, with a limited amount of texture. It is currently not possible to transform transparent objects to non-transparent ones.

The parameters for the bilateral filter typically need to be chosen such that the smoothness of the depth map, and subsequent gradient field, is matched to the type of material that is applied. Metallic BRDFs, for instance, suggest that at the microscopic level the surface is relatively smooth. Usually, this means that at the level of geometry, the surface also needs to be relatively smooth. A similar effect can be observed for transparency in Figure 3, where some depth maps are more appropriate for the simulation of glass than others.

We consider our reshaping functions of both the depth map and

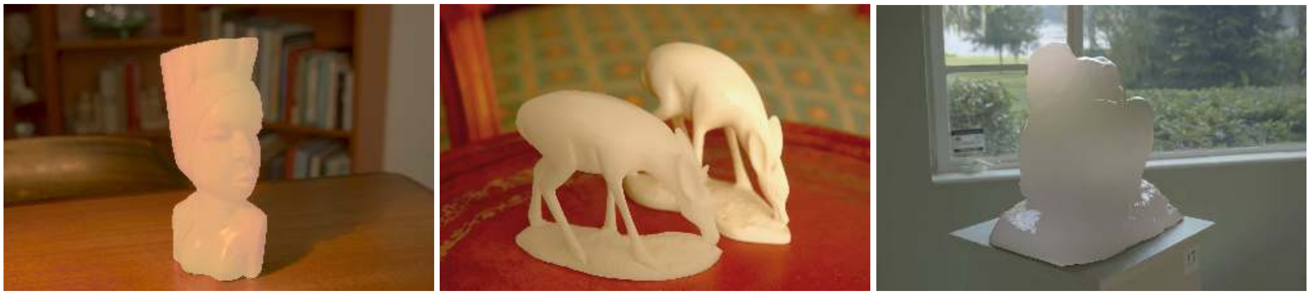


Figure 18: *Translucency applied to various different objects.*

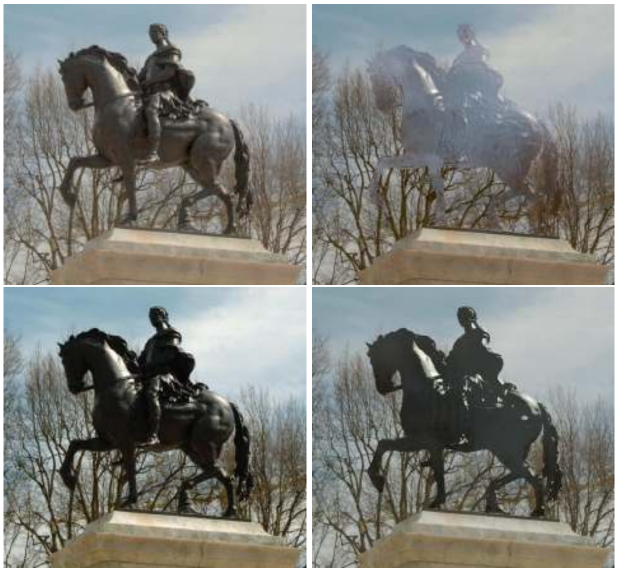


Figure 19: *Comparison between HDR and LDR input for a transformation to a glassy material. The top row shows input and output in HDR (tonemapped for display), and the bottom row shows the low dynamic range equivalent. The HDR images were tonemapped with the photographic operator [Reinhard et al. 2002], whereas the LDR images were linearly scaled to fit the display range.*

the gradient fields to be temporary solutions. While in general, plausible results can be obtained, further improvements are possible. In particular, near the silhouettes the current reshaping functions allow the gradients to become too large, leading to unnatural compression, as for instance seen in the stone texture applied to the deer in Figure 12.

Finally, changing the material of an object will also alter its interaction with the environment. For instance, making an object transparent will likely introduce caustics elsewhere in the environment. An interesting avenue of further research would be to model these interactions with the environment.

6 Conclusions

Given a single high dynamic range photograph, we have presented tools and techniques to change the material properties of objects found in such an image. A key observation enabling dramatic transformations of object appearance is that certain aspects of the interaction between light and matter need to be accurately reproduced, whereas other aspects can be approximated without harm. For instance, we have found that accurate retrieval of the object's surface normals is much more important than the precise refraction of a background through an object. The environment used to light the object needs to have the correct range of colors and intensity, but the reflections and refractions of an object do not have to be positioned with high accuracy.

These observations allow us to produce compelling imagery useful for special effects and digital content creation. We achieve this by carefully reconstructing surface normals based on the luminance distribution of the object. Since this aspect of image-based material editing requires high accuracy, the use of high dynamic range imagery is recommended as this reduces quantization errors.

Our approach thus enables extensive material manipulations using images alone, and does so without the need for expensive equipment, light probes or 3D modeling. This approach does not introduce significant temporal artifacts if applied to individual frames in a sequence [Reinhard et al. 2005a]. In addition to being useful in the special effects community, this work affords exciting opportunities in the study of the human visual system and its perception of materials.

Acknowledgments

We'd like to thank Wojciech Matusik for his measured BRDF data, Peter Shirley, Kanad Biswas, Mark Colbert, Jaroslav Křivánek, and Ahmet Oğuz Akyüz for providing valuable feedback. We thank Native Visions art gallery in Winter Park, Florida for allowing us to photograph the Loet Vanderveen bronze shown in Figure 7. Finally, we thank Karen Louden and the Albin Polasek Museum of Winter Park, Florida, for letting us take HDR images of their beautiful statues, depicted in Figure 2 (top right) and Figure 15.

References

ADELSON, E. 2001. On seeing stuff: The perception of materials by humans. In *Proc. of the SPIE, Human Vision and Electronic Imaging VI*, vol. 4299, 1–12.

BELHUMEUR, P., KRIEGMAN, D., AND YUILLE, A. 1999. The bas-relief ambiguity. *International Journal of Computer Vision 1*, 33–44.

BERTALMIO, M., SAPIRO, G., CASELLES, V., AND BALLESTER, C. 2000. Image inpainting. In *SIGGRAPH '00: Proceedings of the 27th annual conference on computer graphics and interactive techniques*, 417–424.

BLAKE, A., AND BÜLTHOFF, H. H. 1990. Does the brain know the physics of specular reflection? *Nature 343*, 6254, 165–168.

BLAKE, A., AND BÜLTHOFF, H. H. 1991. Shape from specularities: Computation and psychophysics. *Philosophical Transactions of the Royal Society (London) Series B 331*, 237–252.

CHUANG, Y., ZONGKER, D. E., HINDORFF, J., CURLESS, B., SALESIN, D., AND SZELISKI, R. 2000. Environment matting extensions: Towards higher accuracy and real-time capture. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 121–130.

Debevec, P. E., and Malik, J. 1997. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH 97 Conference Proceedings*, 369–378.

Debevec, P. E. 1998. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with illumination and high dynamic range photography. In *SIGGRAPH 98 Conference Proceedings*, 45–50.

Debevec, P. 2002. A tutorial on image-based lighting. *IEEE Computer Graphics and Applications 22*, 2, 26–34.

Drew, M. S. 1994. Robust specularity detection from a single multi-illuminant color image. *Computer Vision, Graphics, and Image Processing: Image Understanding 59*, 3, 320–327.

Drori, I., Cohen-Or, D., and Yeshurun, H. 2003. Fragment-based image completion. *ACM Transactions on Graphics 22*, 3, 303–312.

Durand, F., and Dorsey, J. 2002. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Transactions on Graphics 21*, 3, 257–266.

Eisemann, E., and Durand, F. 2004. Flash photography enhancement via intrinsic relighting. *ACM Transactions on Graphics 23*, 3, 673–678.

Fang, H., and Hart, J. C. 2004. Textureshop: Texture synthesis as a photographic editing tool. *ACM Transactions on Graphics 23*, 3, 354–358.

Fleming, R. W., and Bülthoff, H. H. 2005. Low-level image cues in the perception of translucent materials. *ACM Trans. on Applied Perception 2*, 3, 346–382.

Freeman, W. T. 1994. The generic viewpoint assumption in a framework for visual perception. *Nature 368*, 542–545.

Freeman, W. T. 1996. Exploiting the generic viewpoint assumption. *International Journal of Computer Vision 20*, 3, 243–261.

Hart, J. C., Francis, G. K., and Kauffman, L. H. 1994. Visualizing quaternion rotation. *ACM Transactions on Graphics 13*, 3, 256–276.

Igarashi, T., Matsuoka, S., and Tanaka, H. 1999. Teddy: a sketching interface for 3d freeform design. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 409–416.

ITU. 1990. *International Telecommunication Union ITU-R Recommendation BT.709, Basic Parameter Values for the HDTV Standard for the Studio and for International Programme Exchange*. Geneva. Formerly CCIR Rec. 709.

Kajiya, J. T. 1986. The rendering equation. In *SIGGRAPH '86: Proc. of the 13th annual conference on Computer graphics and interactive techniques*, 143–150.

Kang, S. 1998. Depth-painting for image based rendering applications. Tech. rep., Microsoft.

Klinker, G. J., Shafer, S. A., and Kanade, T. 1988. The measurement of highlights in color images. *International Journal of Computer Vision 2*, 1, 7–32.

Koenderink, J. J., and van Doorn, A. J. 1979. The internal representation of solid shape with respect to vision. *Biological Cybernetics 32*, 211–216.

Koenderink, J. J., van Doorn, A. J., Kappers, A., and Todd, J. 2001. Ambiguity and the 'mental eye' in pictorial relief. *Perception 30*, 4, 431–448.

Langer, M. S., and Bülthoff, H. H. 2000. Depth discrimination from shading under diffuse lighting. *Perception 29*, 6, 649–660.

Langer, M. S., and Zucker, S. W. 1994. Shape-from-shading on a cloudy day. *Journal of the Optical Society of America A 11*, 467–478.

Lawson, R., Bülthoff, H. H., and Dumbell, S. 2003. Interactions between view changes and shape changes in picture - picture matching. *Perception 32*, 1465–1498.

Li, Y., Sun, J., Tang, C.-K., and Shum, H.-Y. 2004. Lazy snapping. *ACM Transactions on Graphics 23*, 3, 303–308.

Matusik, W., Pfister, H., Brand, M., and McMillan, L. 2003. A data-driven reflectance model. *ACM Transactions on Graphics 22*, 3, 759–769.

McGuire, M., Matusik, W., Pfister, H., Hughes, J. F., and Durand, F. 2005. Defocus video matting. *ACM Transactions on Graphics 24*, 3, 567–576.

Metzger, W. 1975. *Gesetze des Sehens (Laws of Seeing)*. Waldemar Kramer, Frankfurt.

Motoyoshi, I., Nishida, S., and Adelson, E. H. 2005. Luminance re-mapping for the control of apparent material. In *Proceedings of the 2nd ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization*, 165.

Naka, K. I., and Rushton, W. A. H. 1966. S-potentials from luminosity units in the retina of fish (cyprinidae). *Journal of Physiology 185*, 587–599.

Oh, B. M., Chen, M., Dorsey, J., and Durand, F. 2001. Image-based modeling and photo editing. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 433–442.

Oliveira, M. M. 2002. Image-based modeling and rendering techniques: A survey. *RITA - Revista de Informática Teórica e Aplicada IX* (October), 37–66.

Ostromoukhov, V., Donohue, C., and Jodoin, P. 2004. Fast hierarchical importance sampling with blue noise properties. *ACM Transactions on Graphics 23*, 3, 488–495.

Ostrovsky, Y., Cavanagh, P., and Sinha, P. 2005. Perceiving illumination inconsistencies. *Perception 34*, 1301–1314.

te Pas, S. F., and Pont, S. C. 2005. A comparison of material and illumination discrimination performance for real rough, real smooth and computer generated smooth spheres. In *APGV '05: Proceedings of the* 2^{nd} *ACM Symposium on Appied Perception in Graphics and Visualization*, 75–81.

te Pas, S. F., and Pont, S. C. 2005. Estimations of light source direction depend critically on material brdfs. *Perception 34*, 212. Supplement ECVP05, A Coruña.

Ramamoorthi, R., and Hanrahan, P. 2001. A signal-processing framework for reflection. *ACM Transactions on Graphics 23*, 4, 1004–1042.

Reinhard, E., and Khan, E. A. 2005. Depth-of-field-based alpha-matte-extraction. In *APGV '05: Proceedings of the* 2^{nd} *ACM Symposium on Appied Perception in Graphics and Visualization*, 95–102.

Reinhard, E., Ashikhmin, M., Gooch, B., and Shirley, P. 2001. Color transfer between images. *IEEE Computer Graphics and Applications 21* (September/October), 34–41.

Reinhard, E., Stark, M., Shirley, P., and Ferwerda, J. 2002. Photographic tone reproduction for digital images. *ACM Trans. on Graphics 21*, 3, 267–276.

Reinhard, E., Shirley, P., Ashikhmin, M., and Troscianko, T. 2004. Second order image statistics for computer graphics. In *APGV '04: Proc. of the* 1^{st} *ACM Symposium on Applied Perception in Graphics and Visualization (APGV)*, 99–106.

Reinhard, E., Khan, E. A., Akyüz, A. O., Fleming, R. W., and Bülthoff, H. H. 2005. Image-based material editing. *ACM SIGGRAPH Computer Animation Festival / Video Review*, 152.

Reinhard, E., Ward, G., Pattanaik, S., and Debevec, P. 2005. *High Dynamic Range Imaging: Acquisition, Display and Image-Based Lighting*. Morgan Kaufmann Publishers, San Francisco.

Rother, C., Kolmogorov, V., and Blake, A. 2004. "grabcut" — interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics 23*, 3, 309–314.

Ruderman, D. L. 1997. The statistics of natural images. *Network: Computation in Neural Systems 5*, 4, 517–548.

van der Schaaf, A. 1998. *Natural image statistics and visual processing*. PhD thesis, Rijksuniversiteit Groningen, The Netherlands.

Simons, D. J., and Levin, D. T. 1997. Change blindness. *Trends in Cognitive Science 1*, 261–267.

Smith, A. R., and Blinn, J. F. 1996. Blue screen matting. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 259–268.

Sun, J., Yuan, L., Jia, J., and Shum, H.-Y. 2005. Image completion with structure propagation. *ACM Transactions on Graphics 24*, 3, 861–868.

Tarr, M. J., Kersten, D., and Bülthoff, H. H. 1999. Why the visual recognition system might encode the effects of illumination. *Vision Research 39*, 2259–2275.

Todd, J. T., and Mingolla, E. 1983. Perception of surface curvature and direction of illuminant from patterns of shading. *Journal of Experimental Psychology: Human Perception and Performance 9*, 583–595.

Tomasi, C., and Manduchi, R. 1998. Bilateral filtering for gray and color images. In *Proc. IEEE International Conference on Computer Vision*, 836–846.

Torralba, A., and Oliva, A. 2003. Statistics of natural image categories. *Network: Computation in Neural Systems 14*, 391–412.

Zelinka, S., Fang, H., Garland, M., and Hart, J. C. 2005. Interactive material replacement in photographs. In *GI '05: Proceedings of the 2005 conference on Graphics interface*, 227–232.

Zhang, R., Tsai, P., Cryer, J., and Shah, M. 1999. Shape from shading: a survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence 21*, 8, 690–706.

Zongker, D. E., Werner, D. M., Curless, B., and Salesin, D. H. 1999. Environment matting and compositing. In *SIGGRAPH '99: Proc. of the 26th annual conference on Computer graphics and interactive techniques*, 205–214.