EUROGRAPHICS Workshop on Graphics and Cultural Heritage (2016)
C. E. Catalano and L. De Luca (Editors)

Using Flash Photography and Image-Based Rendering to Document Cultural Heritage Artifacts

M. Tetzlaff[1] and G. Meyer[1]

[1]University of Minnesota

Figure 1: Left: A photograph of an ancient Chinese *you* vessel illuminated by an on-camera-axis flash. Center/Right: Synthetic images of the same vessel under different illumination conditions, based on 116 such flash photographs.

Abstract

A novel image-based rendering system is proposed for documenting cultural heritage artifacts. The system utilizes backscattering photography to acquire the initial pictures and derives estimates for the object's diffuse albedo, surface normals, and the specular reflectivity from the images. A projective texture mapping technique is used to create a novel view of the artifact by blending the original photographs and projecting them onto a mesh that is also derived from the photos. By weighting the images according to how they best depict the manner in which a virtual light source illuminates the artifact's surface, object relighting is also achieved.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Viewing algorithms I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture I.4.1 [Image Processing and Computer Vision]: Digitization and Image Capture—Reflectance I.4.7 [Image Processing and Computer Vision]: Feature Measurement—Texture

1. Introduction

Image-based rendering methods have the potential to forever change the way that cultural heritage institutions use photographs to document the objects in their collections. Instead of employing a carefully constructed lighting and viewing setup to take a single image of an artifact, museum staff will photograph the object using a more comprehensive set of simpler pictures and rely upon computer graphic interpolation techniques to construct the final image. This approach creates a more complete initial record of the artifact and allows new viewing and lighting variations of the original image to be created whenever it is necessary. The method remains true to the original photographs because it is directly derived from them. This paper shows how this result can be achieved using a simple photographic setup to take the initial images: flash photography.

DOI: 10.2312/gch.20161397

Appearance has traditionally been represented in computer graphics as a *bidirectional reflectance distribution function*, or BRDF, which gives the *reflectance*, that is, the percentage of light reflected, as a function of two directions in 3D space: the incident light direction and the direction the surface is viewed from. For a complex object, a single BRDF is usually not sufficient to capture all of the object's visual properties, but instead, each sample point on the object's surface has a separate BRDF associated with it.

This is a very complex function to model with six degrees of freedom – two from the light direction, two from the viewing direction, and two from the surface position. Because of this, two separate, simpler appearance modeling problems have been considered. The first simplification has traditionally been refered to as *photometric stereo* when it has been applied in computer vision to acquire detailed surface geometry from photographs. It addresses the problem of measuring reflectance using a moving light source, but with a fixed camera. The alternate simplification, conversely, is where the lighting is fixed and the camera moves, and is the domain of *light field* representation and rendering. Other solutions have also been proposed that avoid either of these simplifications, including the use of specialized hardware to accelerate the process of taking all the photographs and the use of representations based on basis functions that help to resolve undersampling issues.

However, an alternative to all of the aforementioned methods which has thus far received fairly little attention is the *backscattering* configuration in which the light source is attached to the camera and the object moves with respect to both of them. In this work, it will be argued that this is not only a straightforward method for acquiring photographs, but it can be used to produce high-quality synthetic images using an extension of image-based rendering.

The primary contribution of this work is a demonstration of the applicability of backscattering photography within the domain of image-based rendering. In order to do this, a rendering algorithm is presented, based heavily on existing surface light field rendering methods, but permitting re-illumination based on the assumption that the object's reflectance can be represented by an unknown spatially varying microfacet-based BRDF. We use off-the-shelf photogrammetry software for camera calibration and mesh reconstruction, but also provide several novel preprocessing steps, the most important of which is one in the spirit of photometric stereo for determining a diffuse albedo texture for glossy two-layer materials and improving the geometric details in the form of a normal map.

2. Related Work

2.1. Photometric Stereo

Traditionally, one of the most effective ways to increase the realism of a photographically acquired virtual object has been to have an accurate model of the object's geometry. Woodham [Woo80] was the first to describe *photometric stereo*, the problem of determining an object's geometric surface properties from several photographs taken from a fixed viewpoint under varying illumination conditions. A number of photometric stereo methods have explicitly used shading cues to compute surface normal maps that enhance the surface appearance. For instance, Bernardini et al. [BMR01] demonstrated how to use photometric stereo from multiple viewpoints to create a set of surface normal maps that can be merged to obtain complete 3D models of large sculptures for cultural heritage curation purposes. Nehab et al. [NRDR05] showed that normal maps can even be integrated back into the original geometry in order to render parallax effects more accurately.

An alternative to traditional photometric stereo is the *polynomial texture map* representation originally proposed by Malzbender et al. [MGW01], in which the linear Lambertian lighting equation is replaced with either a quadratic or cubic [ZD14] polynomial to better model effects like self shadowing, subsurface scattering, and interreflections. Drew et al. [DHOMH12] showed that since the polynomial representation is generally optimal for diffuse reflectance only, the specular reflectance can be modeled separately.

2.2. Light Field Rendering

The converse of photometric techniques, which capture reflectance from a fixed viewpoint under varying illumination, are the methods which have attempted to capture the outgoing radiance of a scene under fixed incident illumination. The most general form of this is the 5D plenoptic function proposed by Adelson and Bergen [AB91], which determines the radiance at any point in 3D space (three dimensions), in any direction (two dimensions). In 1996, Levoy and Hanrahan [LH96] and Gortler et al. [GGSC96] simultaneously proposed that the 5D plenoptic function could be reduced to a 4D *light field* or *lumigraph* by assuming the scene is relatively free of occluders. This assumption implies that radiance is constant along a ray, which results in a dimensional reduction.

One way to represent light fields is by using known geometry, which is sometimes called a *surface light field* representation. Surface light fields do not require as many views as the earlier light field methods because the presence of accurate geometry can reduce artifacts like ghosting, where features disappear in one place and reappear in another instead of smoothly moving from one to the other. Although the phrase *image-based rendering* is often used to describe techniques that do not directly use geometric information, there is early precedent for using the term for methods which directly project photographs onto a geometric proxy; for instance Debevec et al. [DYB98] developed an image-based rendering algorithm which they called *view-dependent texture mapping* or VDTM. Buehler et al. [BBM*01] generalized view-dependent texture mapping and light field rendering with *unstructured lumigraph rendering*, which behaves like a light field when a large number of images are available but the geometry may be inaccurate, and behaves like VDTM when the number of images is small but the geometry is accurate.

Since light fields contain view-dependent specular reflections, it is in theory possible to obtain a general reflectance model from a light field dataset. Yu et al. [YDMH99] acquired spatially varying diffuse albedo and piecewise-uniform specular BRDFs in a manner that is sensitive to global illumination effects. Nishino et al. [NZI01, NIZ05] similarly showed that an estimate of the illumination could be used to reduce a surface light field to a parameterized homogeneous Torrance-Sparrow reflectance model [TS67] without geometric or Fresnel factors. Obtaining a spatially varying specular model from a surface light field dataset is also possible and

was demonstrated first by Yu et al. [YWAC06] and later by Palma et al. [PCDS12] and Dong et al. [DCP*14] who both exploited temporal coherency in video streams. Haber et al. [HFB*09] demonstrated how to obtain a spatially varying specular BRDF in the case when images come from an online photo collection and therefore do not necessarily have equivalent illumination conditions.

2.3. Reflectance Field Acquisition

While photometric stereo is effective at capturing the illumination dependent properties of an object, and light fields are effective at capturing the view-dependent properties, in order to fully model an arbitrary object's appearance it is necessary to consider the case when either the illumination conditions or the view may be changed. This is fundamentally a six-dimensional function, depending on illumination direction (two angles), viewing direction (two angles), and surface position (two dimensions in an appropriate surface parameterization). The sort of function being described is often called a *reflectance field* [DHT*00], particularly when thought of as a generalization of a light field.

One of the first breakthroughs in acquiring reflectance fields was accomplished by Debevec et al. [DHT*00], who built the *light stage* at Berkeley for the purpose of efficiently acquiring reflectance fields of human faces. They also considered fitting raw BTFs to parameterized reflectance models to save storage space. Similar light stages were used the following year by Hawkins et al. [HCD01] for acquiring cultural heritage artifacts. The light stage is an example of specialized hardware that is designed to aid in densely sampling an object's reflectance. It is very effective once the hardware is in place, but requires a substantial initial investment in time and money in order to get it up and running, which has generally made it impractical for cultural heritage institutions.

Several methods have been devised for acquiring the reflectance of low-relief objects. Gardner et al. [GTHD03] and more recently, Chen et al. [CDP*14], built mechanical systems where a light source is automatically scanned over a flat sample to be captured. Since these kinds of systems, like the light stage, require a good deal of effort to build and calibrate, Aittala et al. [AWL13] showed that the mechanical system can be replaced by an LCD screen to emulate a moving linear light source. Several other methods involve a moving camera but are still limited to flat objects, such as the *pocket reflectometry* method proposed by Ren et al. [RWS*11] which requires only a portable linear light source, a pocket sized BRDF chart, and the camera on a mobile phone, and the work of Filip and Vavra [FV13] who used nothing but a point light source and a consumer camera.

2.4. Backscattering

Since reflectance fields for full 3D objects are still cumbersome to acquire, photometric stereo is limited to a fixed viewpoint, and light fields cannot easily be relit, several works have used a fourth configuration, often called *backscattering* [DHI*15], in which the light source is attached to the camera. A common example of a backscattering image is a photograph taken with a typical consumer camera with an on-camera flash. In addition to its convenience, Petcschnigg et al. [PSA*04] observed that flash photography has several key advantages over conventional photography, such as its high signal to noise ratio and the ease with which it may be accurately white-balanced. Because flash photography has many desirable properties, it would be highly desirable to be able to peform image-based rendering with photographs taken under this configuration.

Several early works used what was essentially a backscattering configuration, where the camera and light source are fixed and the object turns to expose multiple viewpoints. Sato et al. [SWI97] mounted their object on a robotic arm and were able to obtain a high-resolution diffuse albedo color and a low-resolution, per-vertex, Phong specular lobe. Marschner [Mar98] placed objects on a turntable and used an on-camera flash to acquire spatially-varying diffuse albedo. Dellepiane et al. [DCC*10] projected color-corrected photographs onto known geometry to determine diffuse albedo, but filtered out specular highlights rather than attempting to model them. Wu et al. [WZ15] recently used the Microsoft Kinect to measure infrared backscattering which when combined with the built-in RGB camera could be used to infer a specular lobe. All of the above methods attempt to use the backscattering images to a parameterized model, and thus are limited by the reflectance model that they use, which tends to cause objects to lose their photorealistic quality.

In the context of photometric stereo, acquiring images using backscattering is convenient because established methods for calibrating camera positions can be leveraged to determine the position of the light source. Paterson et al. [PCF05] and, more recently, Riviere et al. [RPG14] used backscattering as their photographic technique for performing what was essentially photometric stereo. Both works demonstrated how to acquire not only a normal map but also a spatially varying microfacet-based BRDF for flat surfaces. However, both of these works limited their study to flat 2D objects and did not attempt to generalize it to arbitrary 3D surfaces.

3. Preprocessing

A dataset in this work begins as a set of color images taken under a single point light source that is nearly colocated with the camera in each view. We must first calibrate the camera poses so that the position and orientation of the camera is known in each view, and also produce a geometric model of the object. Although Westoby et al. [WBG*12] have provided a method for this, we used the off-the-shelf tool Agisoft PhotoScan instead, which implements a similar pipeline, is extremely robust, and is currently the tool of choice among most cultural heritage institutions. In addition to aligning the photographs, PhotoScan can produce the geometric model using the same flash photographs that will be used for rendering.

The ultimate method to be presented in Section 4 is in theory a standalone method; however, several additional preprocessing steps can improve the rendering quality. Each of these preprocessing steps may also have standalone value in future work. These steps are:

- Light offset estimation – the light can be assumed to be perfectly colocated with the camera, but the offset of the light could introduce bias and undesirable shadows.
- Light intensity estimation – for accurate results, the colors in the

image should be interpreted not just as relative radiance values but as absolute reflectance measurements grounded by a reference chart.
- Diffuse albedo estimation – for metallic or directional diffuse materials, this step is unnecessary; however, for glossy materials, fitting a diffuse albedo that will be treated separately from the glossy directional layer can lead to more realistic rendering.
- Surface normal estimation – for bumpy surfaces where only approximate geometry is available, a normal map can be fit to increase the apparent geometric detail.
- Specular reflectivity estimation – for best quality at grazing angles, an absolute specular reflectivity can be determined which controls the scale of the Fresnel effect.

3.1. Sampling Procedure

In each of the following preprocessing steps, the images can be considered to be a collection of reflectance samples for the geometry. We use *projective texture mapping*, first proposed by Segal et al. [SKVW*92] for the purpose of efficiently rendering shadows, to map pixels from perspective images onto the geometric surface.

Projective texture mapping over the set of views provides a set of reflectance samples $S(\mathbf{p})$ where each element is the color of the pixel in a particular view at the surface position $\mathbf{p}$. We use the notation that for each sample s in $S(\mathbf{p})$, the color itself is denoted by $L_s(\lambda)$, the corresponding viewing direction vector $\mathbf{v}_s$, and the light direction vector $\mathbf{l}_s$. In practice, each sample in the set $S(\mathbf{p})$ is processed as it is generated from projective texture mapping, rather than storing the entire set at once. If the geometry is back-facing for a particular view, the corresponding sample is discarded.

In some images, the surface position $\mathbf{p}$ will be occluded by another surface that is closer to the camera, or may be shadowed by another surface that is closer to the light source. These issues are resolved by using *shadow mapping* [SKVW*92] to cast shadows both from the light source and from the camera (which are close but not perfectly co-located) and masking out the pixels that are in "shadow."

3.2. Diffuse Reflectance Model

The model assumed throughout this work for diffuse reflectance is the ubiquitous Lambertian diffuse lobe defined by:

$$L_o(\lambda,\mathbf{l}) = L_i(\lambda)R_D(\lambda)(\mathbf{n}\cdot\mathbf{l}) \tag{1}$$

$L_o(\lambda,\mathbf{l})$ is the reflected color which is assumed to be diffuse, or independent of the observer's viewpoint. $R_D(\lambda)$ is the diffuse albedo color, the color that would be reflected when the light hits the surface at a right angle; $L(\lambda)$ is the radiance of the incident light at the surface; and $\mathbf{n}\cdot\mathbf{l}$ is a cosine factor that accounts for how the surface appears darker when the light hits the surface at a sharper angle. $\mathbf{n}$ is the unit surface normal vector which comes from the geometry, and $\mathbf{l}$ is the a unit vector in the direction toward the light source from the surface position.

3.3. Light Offset Estimation

In order to eliminate the problem of determining where the light source is positioned, one option is to assume that the light source is effectively co-located with the camera. This assumption is of course never perfectly satisfied, but if the camera is far enough from the object and the light source is close enough to the camera this approximation is acceptable. In general, however, the light direction should be considered distinct from the camera direction.

Initially, we produce many estimates of the light offset, in each case using only the samples at a particular surface position. For a given surface position, let $\mathbf{n}$, $\mathbf{l}$, and $\mathbf{v}$ be transformed into camera space so that $\mathbf{v} = -\mathbf{p}$, where $\mathbf{p}$ is the surface position at a given pixel in camera space, and $\mathbf{l} = \mathbf{l_0} - \mathbf{p}$, where $\mathbf{l_0} = \langle l_x, l_y, 0\rangle$ is the displacement of the light source from the camera, an invariant across all views. L_o, $\mathbf{p}$, and $\mathbf{n}$ are known quantities, while R_D, L_i, and $\mathbf{l}$ are unknown. The light displacement ought not to be wavelength-dependent, so each pixel color should be converted into luminance by transforming it into the CIE-XYZ color space and selecting the Y component.

Assuming that the diffuse reflectance is dominant across the collection of views:

$$L_o = L_i R_D \mathbf{n}\cdot\frac{\mathbf{l_0}-\mathbf{p}}{||\mathbf{l_0}-\mathbf{p}||} \tag{2}$$

The incident radiance L_i is equal to the radiant intensity of the light source I divided by the square of the distance from the light source to the surface:

$$L_o = IR_D\mathbf{n}\cdot\frac{\mathbf{l_0}-\mathbf{p}}{||\mathbf{l_0}-\mathbf{p}||^3} \tag{3}$$

Rearranging terms:

$$L_o = \frac{\mathbf{n}}{||\mathbf{l_0}-\mathbf{p}||^3}\cdot IR_D\mathbf{l_0} + \frac{\mathbf{n}\cdot(-\mathbf{p})}{||\mathbf{l_0}-\mathbf{p}||^3}IR_D \tag{4}$$

The distance from the light source to the surface, $||\mathbf{l_0}-\mathbf{p}||$ could vary slightly between views – but in practice, the only effect of light attenuation on the equation is a scaling factor that will end up being absorbed into the unknown diffuse albedo and light intensity. Because of this, and the fact that $||\mathbf{l_0}||$ is much smaller than $||\mathbf{p}||$, it is permissible to make the approximation: $||\mathbf{l_0}-\mathbf{p}|| \approx ||-\mathbf{p}||$ This causes equation 4 to simplify to:

$$L_o = \frac{1}{||-\mathbf{p}||^3}\langle n_x, n_y, \mathbf{n}\cdot(-\mathbf{p})\rangle^T IR_D\langle l_x, l_y, 1\rangle \tag{5}$$

Let $\alpha = IR_D l_x$, $\beta = IR_D l_y$, and $\gamma = IR_D$.

Then it can be seen that the preceding equation is linear and can be solved using least squares, weighted by $\mathbf{n}\cdot\mathbf{v}$ to favor images with more effective resolution, for α, β, and γ. Then the light offset estimate for each surface position is:

$$\mathbf{l_0} = \langle l_x, l_y, 0\rangle = \left\langle \frac{\alpha}{\gamma}, \frac{\beta}{\gamma}, 0\right\rangle \tag{6}$$

Once all the light offset estimates have been calculated, they are themselves averaged together for a final light offset estimate. This average should be weighted towards the surface positions that are likely to have yielded a more accurate estimate. The heuristic weighting function used is:

$$w(\mathbf{p}) = (IR_D)_\mathbf{p}\det(M_\mathbf{p})/\overline{(\mathbf{n_p}\cdot\mathbf{v})^2} \tag{7}$$

In the preceding equation, for a surface position $\mathbf{p}$, $\det(M_\mathbf{p})$ is the determinant of the 3x3 matrix of the least squares system, $\overline{(\mathbf{n_p} \cdot \mathbf{v})^2}$ is the sum of the squares of the inner products over all samples, and $(IR_D)_\mathbf{p}$ is just the solution for the third variable (γ) in the least squares system.

3.4. Light Intensity Estimation

Another essential problem to be solved before beginning reflectance calculation is to determine the absolute intensity of the light source and the effect that tonemapping and post-processing have had on the colors in the images, and correspondingly interpret the colors as correct reflectivity measurements. A tool that can be exploited, which is already prevalent in cultural heritage photography, is the ColorChecker chart [MMD76]. The colors on the chart correspond to absolute diffuse reflectivity measurements. The chart can be photographed next to the object in order to provide an absolute reflectance reference.

In practice, inverse-square light falloff may cause a surface behind the ColorChecker chart to appear darker than a square on the chart that has the same reflectance. This inverse-square falloff can easily be modelled, but the distance to the chart must then be known. This could be measured explicitly and provided to the method, but in this work it is just assumed to be the distance from the camera to the nearest surface position on the object (taken from the depth buffer that is already used for visibility testing), within the view that the ColorChecker chart was placed in.

3.5. Diffuse Albedo Estimation

For glossy two-layer materials, it is important to be able to distinguish between the glossy microfacet layer, which participates in Fresnel reflectance, and the diffuse layer, which does not participate. Before determining the diffuse albedo, L_o, $L_i(\lambda)$, $\mathbf{l}$, and $\mathbf{n}$ are known quantities, and R_D is unknown. In theory, the diffuse albedo is just the radiance observed in each sample divided by the irradiance received – that is, the product of the radiance received $L_i(\lambda)$ and the Lambertian factor $\mathbf{n} \cdot \mathbf{l}$. The most significant challenge for computing an accurate diffuse albedo color is preventing it from being excessively biased by samples dominated by a specular highlight. The primary way this challenge is addressed is by using iteratively-reweighted averages, where in each iteration the previous result influences the weights used for the next average. After a few iterations, this process converges to a point where subsequent iterations have an insignificant effect on the model. The weights also take into account the effect of surface orientation on effective image resolution by including a factor of $(\mathbf{n} \cdot \mathbf{v}_s)$. The diffuse albedo is computed by dividing the weighted average of the observed radiances by the weighted average of the irradiances E_i, taking into account both the incident angle of the light and inverse-square light dispersion.

The following is the Gaussian weighting equation to be used in each iteration:

$$w(s) = (\mathbf{n} \cdot \mathbf{v}_s) \exp\left(-\frac{\sum_{\lambda \in \Lambda} (L_s(\lambda) - L_o(\lambda, \mathbf{l}_s))^2}{2\delta^2}\right) \tag{8}$$

L_s is the measured radiance of the sample s from the photograph, and $L_o(\lambda, \mathbf{l}_s)$ is the radiance based on the reflectance estimate from the previous iteration. The parameter δ is configurable and affects how quickly a weight decreases as its sample strays from the previous regression. For all of the examples presented in this work, no more than 16 iterations were required using $\delta = 0.1$.

3.6. Surface Normal Estimation

In modern rendering algorithms, the normal vector is most commonly used for shading, and techniques such as *normal mapping* [COM98] can be used to "fake" a higher level of geometric detail than is actually present in the triangle mesh by storing normal vectors in a high-resolution texture image. These normals are usually relatively close to the normal vectors defined by the geometry, but are not necessarily orthogonal to the surface defined by the triangle mesh.

While computing diffuse albedo as in the previous section, detailed surface normals can also be computed from the shading present in the images by augmenting the iteratively reweighted least squares fit to include three degrees of freedom instead of just one. The vector that the algorithm produces will be called the *computed normal*, notated by $\mathbf{n}'$. Although this is in spirit the same as photometric stereo, the method shown here additionally stands out against prior photometric stereo work in that it is demonstrated for arbitary 3D objects using photographs captured with only a single camera-mounted light source.

This method, when used, will extend and override the diffuse albedo estimation presented in the previous section. The hue and saturation of the diffuse albedo should be determined independently of the brighness by computing the weighted average $\overline{L}(\lambda)$ of the color samples $\{L_s(\lambda) : s \in S\}$ for the given surface position, using the same iteratively reweighted method as before. Mathematically speaking, $R_D(\lambda)$ is assumed to be proportional to $\overline{L}(\lambda)$ with respect to wavelengh λ.

To find the actual diffuse albedo color and normal vector, let:

$$\mathbf{N}(\lambda) = R_D(\lambda)\mathbf{n}' \tag{9}$$

and then solve the following linear equation for $\mathbf{N}(\lambda)$:

$$L_o(\lambda, \mathbf{l}) = (L_i(\lambda)\mathbf{l}) \cdot \mathbf{N}(\lambda) \tag{10}$$

This is solved as a weighted least squares problem at the same time as the weighted average, using the same weights, for each required wavelength $\lambda \in \Lambda$ (typically $\Lambda = \{\text{red}, \text{green}, \text{blue}\}$).

The following equations are used to convert $\mathbf{N}(\lambda)$ back to R_D and $\mathbf{n}'$:

$$\mathbf{n}' = \sum_{\lambda \in \Lambda} \overline{L}(\lambda) \frac{\mathbf{N}(\lambda)}{||\mathbf{N}(\lambda)||} \tag{11}$$

$$R_D(\lambda) = \min_{\lambda^* \in \Lambda} \left(\frac{||\mathbf{N}(\lambda^*)||}{\overline{L}(\lambda^*)} \right) \overline{L}(\lambda) \tag{12}$$

These equations have several desirable properties. First, $\mathbf{n}'$ is a weighted average of the values of $\mathbf{N}(\lambda)$ over each $\lambda \in \Lambda$. Secondly, $R_D(\lambda)$ is proportional to $\overline{L}(\lambda)$, with a brightness based on the magnitude of $\mathbf{N}(\lambda)$.

Note that this procedure is different than many existing photometric stereo methods. Many previous methods used a minimal number of samples and computed an exact solution for the surface normal [BMR01], whereas the method presented here uses all the samples available. It is also explicitly designed for 3D surfaces captured from many viewpoints using a camera-mounted light source; whereas most other photometric methods were either restricted to the case of a 2D object or a 3D object from a single viewpoint. Those that do handle general 3D objects usually require multiple illumination conditions for each viewpoint, which prevents them from being applicable for a backscattering configuration that produces only one illumination condition per viewpoint.

3.7. Specular Reflectivity Estimation

Once the diffuse albedo is known, the specular reflectivity can be computed from the residual after subtracting the diffuse contribution from each input image. The specular reflectivity does not need to be precise as the specular term will ultimately be rendered by blending the original images in a view-dependent manner, but it needs to be on the correct order of magnitude for correctly scaling the Fresnel effect at grazing angles. For this reason, the reflectivity is assumed here to be mostly uniform over the object's surface, and is not represented using a spatially-varying texture. The reflectivity can also be taken to be non-wavelength dependent, since the Fresnel effect naturally eliminates wavelength dependency – Fresnel highlights are always the same color as the light source being reflected.

According to the Cook-Torrance model [CT82], specular reflectance can be represented in terms of the Fresnel reflectance F, a microfacet distribution D describing the probability of the surface normal facing in a direction in which mirror reflectance would occur (namely, the halfway direction), and a geometric attenuation factor G compensating for microfacet occlusion:

$$R_S(\mathbf{l},\mathbf{v}) = \frac{F(\lambda,\mathbf{v})D(\mathbf{h})G(\mathbf{l},\mathbf{v})}{4(\mathbf{n}\cdot\mathbf{l})(\mathbf{n}\cdot\mathbf{v})} \tag{13}$$

For backscattering images, $F(\lambda,\mathbf{v})$ is constant and equal to the reflectivity we want to find, F_0. When $(\mathbf{n}\cdot\mathbf{v})$ and $(\mathbf{n}\cdot\mathbf{l})$ are sufficiently large (in this work, greater than $\cos 45^\circ$), geometric attenuation can be assumed to be essentially constant (this can be verified with the equation used by Cook and Torrance, shown later as equation 18). Then the equation can be solved for the product of the microfacet distribution function times the reflectivity:

$$4[R_S(\mathbf{l},\mathbf{v})(\mathbf{n}\cdot\mathbf{l})](\mathbf{n}\cdot\mathbf{v}) = F_0 D(\mathbf{h}) \tag{14}$$

$[R_S(\mathbf{l},\mathbf{v})(\mathbf{n}\cdot\mathbf{l})]$ is taken to be the measured radiance in the photograph divided by the incident radiance (the implicit conversion from irradiance, in the definition of reflectance, to radiance cancels out the $(\mathbf{n}\cdot\mathbf{l})$ factor). Since the integral of the microfacet distribution over the hemisphere is required to be equal to 1, the reflectivity can be found by numerically computing the following integral:

$$F_0 = 8\pi \int_0^1 [R_S(\mathbf{l},\mathbf{v})(\mathbf{n}\cdot\mathbf{l})](\mathbf{n}\cdot\mathbf{v})(\mathbf{n}\cdot\mathbf{h})d(\mathbf{n}\cdot\mathbf{h}) \tag{15}$$

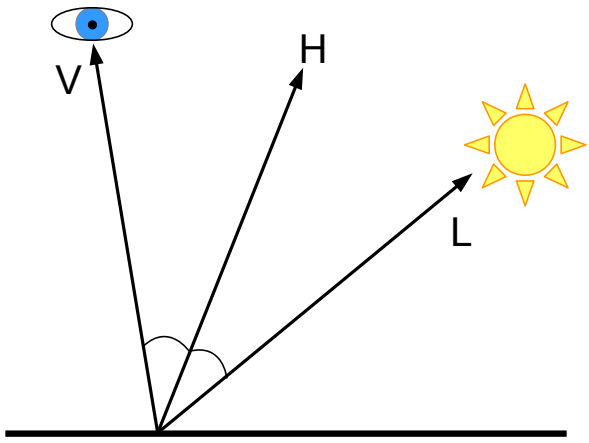

Figure 2: An illustration of the relationship between the light direction L, the view direction V, and the halfway direction H. H is defined such that the angle between L and H equals the angle between H and V.

4. Image-Based Rendering

The ultimate result of this work is an image-based rendering method utilizing data from a camera-mounted light source. Whereas much of the preceding section focused on extracting diffuse reflectance, this section will now focus exclusively on rendering specular reflectance using the same backscattering photographs. If a diffuse texture map is available, the diffuse contribution should be taken out of the photographs before applying the method presented here.

This algorithm is similar in spirit to the light field rendering algorithm proposed by Buehler et al. [BBM*01], which was improved on by Berrier et al. [BTLM15] in a recent publication at Digital Heritage. The implementation details of this algorithm will not be discussed here; instead, refer to the paper by Berrier et al. This method uses projective texture mapping to map images onto a geometric mesh of the object, blending them as they are being projected. It permits view-dependent rendering by weighting each image based on proximity to the viewing direction of the original camera position.

In order to permit relighting, two key changes need to be made to the light field rendering algorithm. The first change is to modify the weighting function so that the highest-weighted photographs are the ones where the reflectance behavior is most similar, rather than where the viewpoint is most similar. For instance, if the virtual camera is oriented in the direction of maximum specular reflectance for a particular light source, the highest weighted photographs should be those that are expected to exhibit similar specular reflectance. Plausible images could be created by doing nothing more than changing the weighting function. However, more realistic images can be created by modeling grazing angle effects that occur when the light source and the camera are nearly opposite each other, which are not captured with a backscattering configuration.

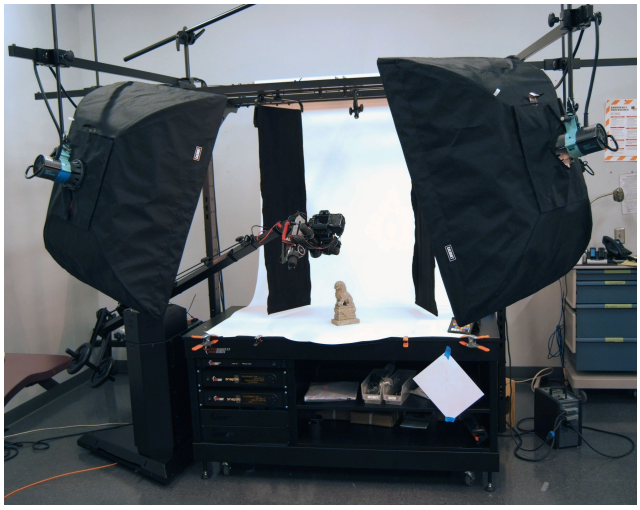

Figure 3: An image of the robot used for 3D photography at the Minneapolis Institute of Arts.

In addition to these two changes, shadow mapping can also be used to synthesize shadows for each light source.

4.1. Changes to the Weighting Function

In order to permit relighting, the images should be weighted based on the similarity of the actual *halfway vector* to the virtual halfway vector, not based on viewpoint, as with light field rendering algorithms. The halfway vector is defined to be the vector that points in the direction which bisects the viewing direction and the lighting direction (see Figure 2). That is, for a viewing direction $\mathbf{v}$ and an illumination direction $\mathbf{l}$, let:

$$\mathbf{h} = \frac{\mathbf{l}+\mathbf{v}}{||\mathbf{l}+\mathbf{v}||} \tag{16}$$

This is justified by the fact that most established parametric reflectance models exhibit a primary dependence on the halfway vector, including the ubiquitous Blinn-Phong model [Bli77], as well as microfacet-based models such as the Cook-Torrance [CT82] model. So if $\mathbf{h}$ is the virtual halfway vector for a particular viewpoint and light source, and $\mathbf{h}_i$ is the halfway vector for the viewpoint and light source of image i, then the weight for each image can be defined as:

$$w(i) = \frac{1}{1-(\mathbf{h}_i \cdot \mathbf{h})^{\alpha}} - 1 \tag{17}$$

4.2. Factoring out Grazing Angle Effects

Changing the weighting formula is in theory all that is necessary to extend a light field rendering algorithm to one that permits reillumination using backscattering images under the assumption that reflectance is primarily dependent on the halfway vector. However, it should be observed that according to microfacet reflectance models such as the Cook-Torrance [CT82] model (refer back to equation 13), there are factors other than the microfacet distribution which are not only dependent on the halfway vector.

It is possible to easily factor out the geometric attenuation term assuming that it can be represented by the V-cavity model used by Cook and Torrance, which due to symmetry is always the same regardless of the microfacet distribution. (See [Hei14] for a good overview of the V-cavity model and its alternatives.) Under the V-cavity model, using the notation that $\mathbf{n}$ is the known surface normal and $\mathbf{h}$ is the halfway vector as defined in equation 16, the geometric attenuation is defined as:

$$G(\mathbf{l},\mathbf{v}) = \min\left(1, \frac{2(\mathbf{h}\cdot\mathbf{n})(\mathbf{v}\cdot\mathbf{n})}{\mathbf{v}\cdot\mathbf{h}}, \frac{2(\mathbf{h}\cdot\mathbf{n})(\mathbf{l}\cdot\mathbf{n})}{\mathbf{v}\cdot\mathbf{h}}\right) \tag{18}$$

In order to remove the effects of geometric attenuation as well as the $1/(4(\mathbf{n}\cdot\mathbf{v}))$ factor in the Cook-Torrance model, the color of each sample i should be multiplied by $4(\mathbf{n}\cdot\mathbf{v}_i)/G(\mathbf{l}_i,\mathbf{v}_i)$, where $\mathbf{l}_i$ is the light vector for sample i and $\mathbf{v}_i$ is the view vector for sample i. The $(\mathbf{n}\cdot\mathbf{l}_i)$ factor is implicitly cancelled out by the cosine factor in the incident irradiance. After multiplying a pixel containing reflected radiance by $4(\mathbf{n}\cdot\mathbf{v}_i)/G(\mathbf{l}_i,\mathbf{v}_i)$ and then dividing by the incident radiance, all that remains is the product of the microfacet distribution and the Fresnel reflectance (which is just the specular reflectivity $F_0(\lambda)$ for backscattering images). The weighting function in equation 17 should also be multiplied by $G(\mathbf{l}_i,\mathbf{v}_i)$ to avoid the singularities that occur at perfect grazing angles. These computations can be performed using the surface normals from the photometric technique introduced in the previous section, if available.

Finally, for realistic rendering, grazing angle effects must be synthesized. Fresnel reflectance can be simulated similarly to the approximate version of the Cook-Torrance model [CT82]. A wavelength independent reflectivity $\overline{F_0}$ can be precomputed as described in Section 3.7. Then the wavelength-independent Fresnel reflectance $\overline{F_\theta}$ can be calculated using the Fresnel equation as given by Cook and Torrance. The peak reflectance $F_{\max}(\mathbf{h})$ can also be estimated by dividing $F_0(\lambda)D(\mathbf{h})$ by the precomputed reflectivity estimate $F_0(\lambda)$ and then similarly converting to CIE-XYZ. The general

Figure 4: The results of the surface normal estimation method for the *you* dataset. Left: the input geometry with per-vertex normals. Right: the same geometry rendered with the computed normal map. See Figure 1 for final results.

Figure 5: Synthetic images of the child's tiger hat (67 views) and *Kuan Yu* (72 views).

product $F(\lambda,\mathbf{v})D(\mathbf{h})$ can ultimately be computed as follows:

$$F(\lambda,\mathbf{v})D(\mathbf{h}) = F_0(\lambda)D(\mathbf{h}) + (F_{\max}(\mathbf{h}) - F_0(\lambda)D(\mathbf{h}))\frac{\max(0,\overline{F_\theta} - \overline{F_0})}{1-\overline{F_0}} \tag{19}$$

Finally, in order to re-introduce geometric attenuation in the final rendering, the result can be multiplied by $G(\mathbf{l},\mathbf{v})/(\mathbf{n}\cdot\mathbf{v})$ to complete the full Cook-Torrance reflectance model.

4.3. Significance

It is important to highlight the distinctions between this image-based rendering method and previous works based on backscattering photography. Several prior works including Paterson et al. [PCF05] and Riviere et al. [RPG14] only considered flat 2D objects. In contrast, the method shown here has been demonstrated for general 3D objects. The recent work of Wu et al. [WZ15] measuring backscattering with the Microsoft Kinect is noteworthy, but it requires specialized hardware (namely the Kinect) and does not apply for photographs taken with a normal camera.

Other methods imposed limitations on the specular reflectance, such as the work of Sato et al. [SWI97], who limited it to a low-resolution, fitted Phong lobe, and Marschner [Mar98], who did not consider spatially-varying specular reflectance at all. The method shown here, however, is image-based and thus can effectively render any reflectance that can be represented by a non-spectral microfacet distribution model and thereby retain the photorealistic image quality that is absent from many parameterized methods. Additionally, none of the aforementioned works considered grazing angle effects.

5. Results

As part of the continued cooperation between the University of Minnesota and the visual resources team at the Minneapolis Institute of Arts, several cultural heritage datasets have been captured with a light source mounted on a camera as described above. For convenience, with most datasets the camera was mounted on a mechanical arm and the object placed on a turntable to automate capturing, but this is not a requirement for the algorithm. Robotics such as this are becoming common at many cultural heritage institutions. Figure 3 shows the robot, while Figure 1 shows a sample image taken by the robot. Figure 7 shows a sample image from a dataset acquired without using the robot.

It typically takes about one or two hours to capture the required images either with or without a robot, although larger objects or objects which require more views can take longer. The objects shown in this work were captured from anywhere between 149 views (for the *Celestial Horse*) and 536 views (for the *ding*). Processing with Agisoft PhotoScan can then take anywhere from a few hours to several days, depending on the desired quality of the mesh, and requires periodic user interaction. The novel preprocessing steps presented here are much shorter in comparison to the initial task of processing with PhotoScan: the *you* example can be processed in about fifteen minutes using an NVidia GTX 780M, the majority of which is spent reading image files. Figure 4 shows the results of the surface normal estimation algorithm when run on the *you* dataset, using a delta value of 0.1. By visual inspection, it can be seen that the photometric normals clearly have more detail than the normals computed from the triangle mesh. This result in itself could be used to improve rendering quality in a traditional graphics engine, or be applied to the image-based rendering algorithm presented here.

Finally, Figures 5, 6, 7, and 8, as well as Figure 1 on the first page, show the results of image-based rendering. Typically an α exponent value of 16 was used in most examples, although a higher value of 100 was required for the *ding* example due to its specularity. Each image was lit by three distinct virtual point light sources shining at the object from different directions. Improved surface normals were used for the *you* and *ding* examples. The other three datasets (*Kuan Yu*, *Celestial Horse*, and the child's tiger hat), do not have a Lambertian diffuse component, so neither a diffuse albedo nor a normal map are used for those examples.

The *Kuan Yu* example (Figure 5, right, and Figure 6) in particular is a very shiny bronze object, demonstrating that this technique is

Figure 6: Synthetic images of *Kuan Yu* under different illumination conditions.

Figure 7: Left: A sample photograph of the *Celestial Horse* taken with a camera-mounted light source in a gallery at the Minneapolis Institute of Arts. Right: Synthetic image of the *Celestial Horse*, using 149 such photographs.

not limited to diffuse objects. The child's tiger hat (Figure 5, left) is made of silk, meaning that the assumption of a microfacet-based BRDF may begin to break down; nonetheless it yields an aesthetically pleasing result that is superior to the diffuse texture mapping that has been most commonly used in cultural heritage. The *Celestial Horse* (Figure 7) is very large (over a meter high) and was captured using a handheld camera, rather than a robot, demonstrating that this technique is applicable both for large objects and for completely unstructured collections of images.

For the *ding* shown in Figure 8, a photograph is included for comparison. The differences in the specular highlights are caused by the image interpolation method (based on the unstructured lumigraph technique). However, it should be emphasized that the diffuse reflectance is reproduced faithfully and the specular highlights are placed correctly, if not being a perfect match.

For image-based rendering, the collection of views should usually be pruned down to between 50 and 150 images to achieve interactive framerates. The *ding* is an exception; due to its specularity, more images were used to reduce interpolation artifacts. Even with such pruning, the framerate is limited; for instance the rendering runs at about 10 frames per second for the *you* example on an NVidia GTX 780M. Improving this benchmark remains an area for future work.

Figure 8: Left: a photograph of the *ding* illuminated by three small lights, one of which is the on-axis light used to capture the object. Right: A synthetic image of the *ding* under the same viewpoint and similar illumination conditions, using 536 views.

6. Summary

The three paradigms which have primarily been used in the past to acquire image-based datasets are photometric stereo (camera fixed with respect to object), light fields (light fixed with respect to object) and reflectance fields (light, camera, and object all independent). However, the backscattering configuration in which the light is fixed to the camera, with neither fixed with respect to the object, has been shown here to be just as effective as photometric stereo and light fields. This approach can be used to produce detailed photometric normal maps to augment the appearance of geometric detail. In a manner similar to surface light fields, the method can also be used for image-based rendering while permitting novel illumination conditions. Additionally, the capturing process is additionally much simpler than scanning a full reflectance field. This work is therefore an important step forward in making image-based rendering practical for organizations such as cultural heritage institutions.

Acknowledgments

This work was partially funded by DARPA (contract HR0011-16-C-0024) and a seed grant from the UM Digital Design Consortium. Special thanks to the Minneapolis Institute of Arts and Charles Walbridge, who photographed all of the datasets shown in this paper.

References

[AB91] ADELSON E. H., BERGEN J. R.: *The plenoptic function and the elements of early vision*. Vision and Modeling Group, Media Laboratory, Massachusetts Institute of Technology, 1991. 2

[AWL13] AITTALA M., WEYRICH T., LEHTINEN J.: Practical svbrdf capture in the frequency domain. *ACM Trans. Graph. 32*, 4 (2013), 110. 3

[BBM*01] BUEHLER C., BOSSE M., MCMILLAN L., GORTLER S., COHEN M.: Unstructured lumigraph rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM, pp. 425–432. 2, 6

[Bli77] BLINN J. F.: Models of light reflection for computer synthesized pictures. In *ACM SIGGRAPH Computer Graphics* (1977), vol. 11, ACM, pp. 192–198. 7

[BMR01] BERNARDINI F., MARTIN I. M., RUSHMEIER H.: High-quality texture reconstruction from multiple scans. *Visualization and Computer Graphics, IEEE Transactions on 7*, 4 (2001), 318–332. 2, 6

[BTLM15] BERRIER S., TETZLAFF M., LUDWIG M., MEYER G.: Improved appearance rendering for photogrammetrically acquired 3d models. In *Digital Heritage International Congress (DigitalHeritage), 2015* (2015), Springer. 6

[CDP*14] CHEN G., DONG Y., PEERS P., ZHANG J., TONG X.: Reflectance scanning: estimating shading frame and brdf with generalized linear light sources. *ACM Transactions on Graphics (TOG) 33*, 4 (2014), 117. 3

[COM98] COHEN J., OLANO M., MANOCHA D.: Appearance-preserving simplification. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (1998), ACM, pp. 115–122. 5

[CT82] COOK R. L., TORRANCE K. E.: A reflectance model for computer graphics. *ACM Transactions on Graphics (TOG) 1*, 1 (1982), 7–24. 6, 7

[DCC*10] DELLEPIANE M., CALLIERI M., CORSINI M., CIGNONI P., SCOPIGNO R.: Improved color acquisition and mapping on 3d models via flash-based photography. *Journal on Computing and Cultural Heritage (JOCCH) 2*, 4 (2010), 9. 3

[DCP*14] DONG Y., CHEN G., PEERS P., ZHANG J., TONG X.: Appearance-from-motion: recovering spatially varying surface reflectance under unknown lighting. *ACM Transactions on Graphics (TOG) 33*, 6 (2014), 193. 3

[DHI*15] DUPUY J., HEITZ E., IEHL J.-C., POULIN P., OSTROMOUKHOV V.: Extracting microfacet-based brdf parameters from arbitrary materials with power iterations. In *Computer Graphics Forum* (2015), vol. 34, Wiley Online Library, pp. 21–30. 3

[DHOMH12] DREW M. S., HEL-OR Y., MALZBENDER T., HAJARI N.: Robust estimation of surface properties and interpolation of shadow/specularity components. *Image and Vision Computing 30*, 4 (2012), 317–331. 2

[DHT*00] DEBEVEC P., HAWKINS T., TCHOU C., DUIKER H.-P., SAROKIN W., SAGAR M.: Acquiring the reflectance field of a human face. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (2000), ACM Press/Addison-Wesley Publishing Co., pp. 145–156. 3

[DYB98] DEBEVEC P., YU Y., BORSHUKOV G.: *Efficient view-dependent image-based rendering with projective texture-mapping*. Springer, 1998. 2

[FV13] FILIP J., VÁVRA R.: Fast method of sparse acquisition and reconstruction of view and illumination dependent datasets. *Computers & Graphics 37*, 5 (2013), 376–388. 3

[GGSC96] GORTLER S. J., GRZESZCZUK R., SZELISKI R., COHEN M. F.: The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), ACM, pp. 43–54. 2

[GTHD03] GARDNER A., TCHOU C., HAWKINS T., DEBEVEC P.: Linear light source reflectometry. In *ACM Transactions on Graphics (TOG)* (2003), vol. 22, ACM, pp. 749–758. 3

[HCD01] HAWKINS T., COHEN J., DEBEVEC P.: A photometric approach to digitizing cultural artifacts. In *Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage* (2001), ACM, pp. 333–342. 3

[Hei14] HEITZ E.: Understanding the masking-shadowing function in microfacet-based brdfs. *Journal of Computer Graphics Techniques 3*, 2 (2014), 32–91. 7

[HFB*09] HABER T., FUCHS C., BEKAER P., SEIDEL H.-P., GOESELE M., LENSCH H. P., ET AL.: Relighting objects from image collections. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (2009), IEEE, pp. 627–634. 3

[LH96] LEVOY M., HANRAHAN P.: Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), ACM, pp. 31–42. 2

[Mar98] MARSCHNER S. R.: *Inverse rendering for computer graphics*. PhD thesis, Citeseer, 1998. 3, 8

[MGW01] MALZBENDER T., GELB D., WOLTERS H.: Polynomial texture maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM, pp. 519–528. 2

[MMD76] MCCAMY C. S., MARCUS H., DAVIDSON J.: A color-rendition chart. *J. App. Photog. Eng 2*, 3 (1976), 95–99. 5

[NIZ05] NISHINO K., IKEUCHI K., ZHANG Z.: Re-rendering from a sparse set of images. *Department of Computer Science, Drexel University* (2005). 2

[NRDR05] NEHAB D., RUSINKIEWICZ S., DAVIS J., RAMAMOORTHI R.: Efficiently combining positions and normals for precise 3d geometry. In *ACM transactions on graphics (TOG)* (2005), vol. 24, ACM, pp. 536–543. 2

[NZI01] NISHINO K., ZHANG Z., IKEUCHI K.: Determining reflectance parameters and illumination distribution from a sparse set of images for view-dependent image synthesis. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on* (2001), vol. 1, IEEE, pp. 599–606. 2

[PCDS12] PALMA G., CALLIERI M., DELLEPIANE M., SCOPIGNO R.: A statistical method for svbrdf approximation from video sequences in general lighting conditions. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 1491–1500. 3

[PCF05] PATERSON J. A., CLAUS D., FITZGIBBON A. W.: Brdf and geometry capture from extended inhomogeneous samples using flash photography. In *Computer Graphics Forum* (2005), vol. 24, Wiley Online Library, pp. 383–391. 3, 8

[PSA*04] PETSCHNIGG G., SZELISKI R., AGRAWALA M., COHEN M., HOPPE H., TOYAMA K.: Digital photography with flash and no-flash image pairs. *ACM transactions on graphics (TOG) 23*, 3 (2004), 664–672. 3

[RPG14] RIVIERE J., PEERS P., GHOSH A.: Mobile surface reflectometry. In *ACM SIGGRAPH 2014 Posters* (2014), ACM, p. 68. 3, 8

[RWS*11] REN P., WANG J., SNYDER J., TONG X., GUO B.: Pocket reflectometry. In *ACM Transactions on Graphics (TOG)* (2011), vol. 30, ACM, p. 45. 3

[SKVW*92] SEGAL M., KOROBKIN C., VAN WIDENFELT R., FORAN J., HAEBERLI P.: Fast shadows and lighting effects using texture mapping. In *ACM SIGGRAPH Computer Graphics* (1992), vol. 26, ACM, pp. 249–252. 4

[SWI97] SATO Y., WHEELER M. D., IKEUCHI K.: Object shape and reflectance modeling from observation. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (1997), ACM Press/Addison-Wesley Publishing Co., pp. 379–387. 3, 8

[TS67] TORRANCE K. E., SPARROW E. M.: Theory for off-specular reflection from roughened surfaces. *JOSA 57*, 9 (1967), 1105–1112. 2

[WBG*12] WESTOBY M., BRASINGTON J., GLASSER N., HAMBREY M., REYNOLDS J.: 'Structure-from-motion' photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology 179* (2012), 300–314. 3

[Woo80] WOODHAM R. J.: Photometric method for determining surface orientation from multiple images. *Optical engineering 19*, 1 (1980), 191139–191139. 2

[WZ15] WU H., ZHOU K.: Appfusion: Interactive appearance acquisition using a kinect sensor. In *Computer Graphics Forum* (2015), Wiley Online Library. 3, 8

[YDMH99] YU Y., DEBEVEC P., MALIK J., HAWKINS T.: Inverse global illumination: Recovering reflectance models of real scenes from photographs. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), ACM Press/Addison-Wesley Publishing Co., pp. 215–224. 2

[YWAC06] YU T., WANG H., AHUJA N., CHEN W.-C.: Sparse lumigraph relighting by illumination and reflectance estimation from multi-view images. In *ACM SIGGRAPH 2006 Sketches* (2006), ACM, p. 175. 3

[ZD14] ZHANG M., DREW M. S.: Efficient robust image interpolation and surface properties using polynomial texture mapping. *EURASIP Journal on Image and Video Processing 2014*, 1 (2014), 1–19. 2