Technical Report, State Key Lab of CAD&CG, Zhejiang University

AppFusion: Interactive Appearance Acquisition using a Kinect Sensor

Hongzhi Wu Kun Zhou

State Key Lab of CAD&CG, Zhejiang University

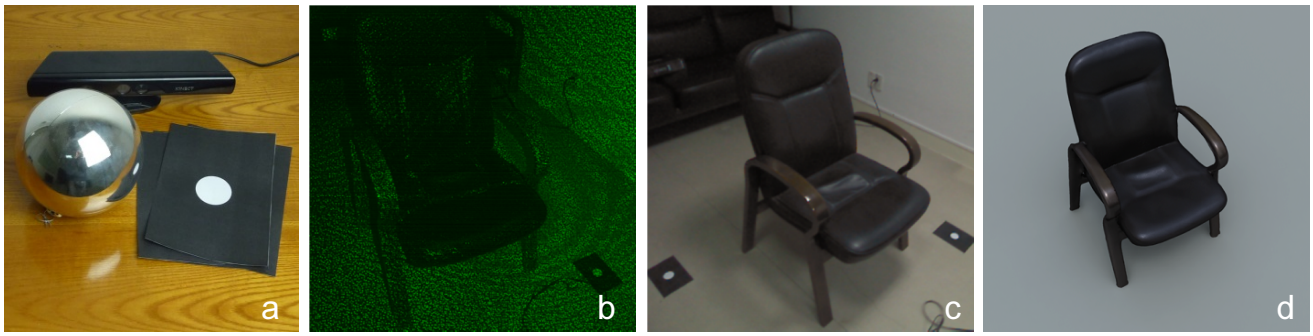

Figure 1: *Using a Kinect, a mirror ball and printed markers (a), we compute the appearance of an object from the images captured by the Kinect infra-red camera (b) and the RGB camera (c). A rendered result using novel lighting and view configurations is shown on the right (d).*

Abstract

We present an interactive material acquisition system for average users to capture the spatially-varying appearance of daily objects. While an object is being scanned, our system computes its appearance on-the-fly and provides quick visual feedback. We build the system entirely on low-end, off-the-shelf components: a Kinect, a mirror ball and printed markers. We exploit the Kinect infra-red emitter/receiver, originally designed for depth computation, as an active hand-held reflectometer, to segment the object into clusters of similar specular materials and estimate the roughness parameters of BRDFs simultaneously. Next, the diffuse albedo and specular intensity of the spatially-varying materials are rapidly computed in an inverse rendering framework, using data from the Kinect RGB camera. We demonstrate captured results of a range of materials, and physically validate our system.

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

Keywords: appearance acquisition, BRDF fusion, Kinect

1 Introduction

With the prevalence of low-end geometry acquisition devices, it is becoming increasingly easier for ordinary users to digitalize everyday objects. For example, using a Kinect sensor, a user can quickly and intuitively create a geometric model of an object at home, by moving and pointing the device towards the object from different points of view. The real-time visualization of the result is provided to the user on-the-fly, to help guide the scanning process.

However, no such ordinary-user-friendly system exists to acquire the material appearance of an object. One fundamental issue is the sheer complexity of the appearance: it can be modeled as a 6D function, which varies in space, and with lighting and view conditions. Traditional methods rely on expensive custom-built devices and/or professional expertise, to sample the 6D domain and measure the appearance function [Weyrich et al. 2009]. Recently, researchers have proposed approaches for casual users to capture the high-quality appearance, based on relatively simple set-ups [Dong et al. 2010; Ren et al. 2011]. But they are limited to planar samples, and require active lighting. Moreover, existing work typically needs minutes or even hours of processing, before presenting the appearance results to the user. This is in contrast to Kinect-based geometry scan with real-time feedback [Izadi et al. 2011], which greatly facilitates average users in the acquisition process.

In this paper, we present AppFusion, a novel system for ordinary users to capture the 6D appearance approximation of an object with interactive visual feedback. Our system is entirely built on low-end, off-the-shelf components: a Kinect, a mirror ball and printed paper markers. We present a unified acquisition framework, similar to KinectFusion [Izadi et al. 2011]: the user simply moves the Kinect around the object, then our system quickly computes the updated appearance and displays the result interactively. In our experiments, it takes on average 10 minutes to scan the geometry and the appearance of a variety of objects, ranging from a pepper to an armchair.

AppFusion is a hybrid system that both uses active lighting in the infra-red (IR) spectrum, and processes passive lighting in the visible spectrum. Specifically, we employ the Kinect IR emitter/receiver, originally designed for depth computation, as an active hand-held reflectometer. The reflected IR light is used to segment the object into clusters of similar specular materials, and estimate the roughness parameters of BRDFs at the same time. We reduce the noise in the IR spectrum by a process called BRDF fusion, which integrates the received IR data over both the spatial and temporal domain. Next, the diffuse albedo and specular intensity of the spatially-varying materials are rapidly computed in an inverse rendering framework, using data from the Kinect RGB camera.

In summary, the major contributions of our paper are:

- We present an interactive appearance acquisition system for ordinary users to digitize objects at home easily. The system is built entirely on low-end, off-the-shelf components and has the unique feature of displaying the captured appearance on-the-fly.
- We present a novel form of hybrid appearance acquisition. The Kinect IR emitter/receiver, originally designed for depth computation, are used as *an active hand-held reflectometer* in the IR spectrum. We also handle passive lighting in the visible spectrum.
- We essentially extend the idea of reflectance sharing [Zickler et al. 2006] to both the spatial and temporal domain, using a process called BRDF fusion.

2 Related Work

In this section, we review previous work on estimating (spatially-varying) BRDFs from measuring real-world objects. An excellent recent survey can be found in [Weyrich et al. 2009].

Methods based on Professional Devices. SVBRDFs (spatially-varying BRDFs) can be directly measured by professional set-ups, such as spatial gonioreflectometers [Dana et al. 1999; Mcallister 2002], which densely sample the lighting and view directions, as well as the spatial domain. The light-stage [Debevec et al. 2000] installs a large number of directional light sources, and rapidly changes their intensities in the acquisition phase, for efficient capturing of the appearance of human faces. Other dedicated devices are also proposed, such as a rotating LED arm with five cameras [Tunwattanapong et al. 2013], to acquire highly specular materials using spherical harmonic illumination. Professional devices can measure SVBRDFs with impressively high quality, but they are not targeted for ordinary users. It typically takes considerable amount of financial budget, time and expertise to custom-build, calibrate, and use such systems for appearance acquisition.

Image-based Methods. These approaches mainly use cameras to capture images for estimating SVBRDFs. Marschner et al. [1999] acquires a 4D BRDF from a convex object, using a camera and a light source. Lensch et al. [2003] cluster and fit Lafortune models to SVBRDFs over an object of known geometry, using a sparse set of images taken with controlled lighting. Gardner et al. [2003] scan the SVBRDF of a planar sample, with the help of a linear light source. Alldrin et al. [2008] use a bivariate model for isotropic BRDFs, to recover the SVBRDF and normals from images with varying illumination conditions. Recently, researchers propose novel appearance acquisition methods, with a camera and an LCD monitor as a programmable light source [Ghosh et al. 2009; Aittala et al. 2013].

Another class of image-based methods do not perform active lighting control. Romeiro et al. [2008] also adopt the bivariate model, and estimate a BRDF from a single image, with captured environment lighting. Haber et al. [2009] optimize both the lighting and SVBRDFs in an inverse rendering framework, from photographs of an object with no control over the lighting or the camera. The system in [Palma et al. 2012] takes video frames and a known geometry as input. It estimates environment lighting via points with specular reflections, and then optimize for the SVBRDFs. Li et al. [2013] reconstruct a dynamic geometry from a multi-view video, and optimize both the lighting and SVBRDFs.

While our approach can be viewed as an image-based method, one key difference is that we provide interactive feedback at the time of scanning, which is important for average users. Previous work typically requires minutes or even hours of processing, in addition to the acquisition time. More time would be needed, if the user is not satisfied with the result and has to restart the process again. Note that for the application of mixed reality, Knecht et al. [2012] quickly estimate BRDFs using inverse rendering, for a fixed-view depth map captured by a Kinect. Their system cannot produce a complete 3D model for viewing at different angles by average users. In comparison, we resolve the lighting-material ambiguity, by measuring from multi-views with active IR lighting (Sec. 7).

Example-based Methods. Hertzmann and Seitz [2003] reconstruct normals and reflectance, by capturing the object of interest alongside with a reference object of known geometry with similar materials. No geometric or radiometric calibration of the camera or the light source is needed, while the presence of a reference object is required. Matusik et al. [2003] analyze measured isotropic BRDFs, and represent them using a linear combination of representatives. The idea is applied to recover human skin SVBRDFs in [Weyrich et al. 2006]. Dong et al. [2010] develop a specialized device for quickly capturing representative BRDFs, and presents a two-pass algorithm that captures a SVBRDF as linear combinations of the representatives. Ren et al. [2011] uses a linear light source, and a BRDF chart, which contains tiles of a variety of known BRDFs. They image a planar sample along with the chart, and reconstruct the SVBRDF by aligning the reflectance sequences of the object and the chart, via dynamic time warping.

In comparison, our acquisition system is built with all off-the-shelf components, which are easily accessible to casual users. We do not require reference materials with known BRDFs that are close to the appearance of interest. Only standard white A4 paper is used for exposure calibration and white balancing of the Kinect RGB camera (Sec. 4.2).

3 Preliminaries

We represent the material appearance as SVBRDFs, defined over the surfaces of an object. The BRDF at a point, f_r, is modeled as a Lambertian term plus a specular BRDF term:

$$f_r(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \frac{\rho_d}{\pi} + \rho_s f(\boldsymbol{\alpha}; \boldsymbol{\omega}_i, \boldsymbol{\omega}_o). \tag{1}$$

Here $\boldsymbol{\omega}_i$ is the lighting direction; $\boldsymbol{\omega}_o$ is the view direction; ρ_d/ρ_s are the diffuse albedo/specular intensity in RGB channels. $f(\boldsymbol{\alpha}; \cdot)$ is a parametric specular BRDF, with $\boldsymbol{\alpha}$ as its parameters. In this paper, we employ the Ward model [Ward 1992] as f. Thus, α is the roughness parameter.

Similar to previous work (e.g., [Lensch et al. 2003; Palma et al. 2012]), we model different ρ_d for each point on the geometry, and different ρ_s and α for a user-specified number of clusters, based on the observation that common objects have slowly-varying specular BRDFs in space [Weyrich et al. 2009].

Next, the reflected radiance L at a point along a view direction $\boldsymbol{\omega}_o$ is modeled according to [Kajiya 1986]:

$$L(\boldsymbol{\omega}_o) = \int_\Omega L_i(\boldsymbol{\omega}_i) f_r(\boldsymbol{\omega}_i', \boldsymbol{\omega}_o')(n \cdot \boldsymbol{\omega}_i) d\boldsymbol{\omega}_i, \tag{2}$$

where L_i describes the incoming light distribution; $\boldsymbol{\omega}_i'$ and $\boldsymbol{\omega}_o'$ are the lighting and view directions, expressed in the local frame at the point; Ω is the upper hemisphere, and n is the surface normal. In the appearance acquisition context, we measure L and L_i, acquire the geometry of an object, and then solve for f_r using the above equation.

Assumptions. We omit indirect lighting and assume isotropic f_r in appearance computation based on Eq. (2), which is common in

previous material acquisition techniques. Due to performance concerns, we do not compute and process visibility information in our pipeline. We assume that $\boldsymbol{\omega}_i' \approx \boldsymbol{\omega}_o'$ in IR acquisition, as the baseline between the IR emitter and receiver is small. Detailed justifications are in Sec. 5.1.

We model the incident light as distant illumination, recorded in an environment map. In addition, we assume that for most part of the object, the specular component is negligible from at least one view in the acquisition (i.e., no bright light subtending a large solid angle in the environment map). Following [Ren et al. 2011], we also assume that the scene can be observed with sufficient dynamic range without clipping, by setting an appropriate auto-exposure brightness on the RGB camera.

Moreover, we assume that the roughness estimated from the IR data is close to that of the visible spectrum. Note that this assumption is reasonable for dielectrics. This is because the specular reflectance is determined by the index of refraction [Dorsey et al. 2007], and the index of refraction of measured dielectrics varies little with respect to wavelength, in the range of visible and IR spectrum. The assumption also works for two types of glossy paint, which are non-dielectric materials measured in Sec. 7.

4 Acquisition Pipeline

Our interactive appearance acquisition system consists of three components: a Kinect sensor, a mirror sphere and printed markers (Fig. 1). The markers are four white circles over a black background, generated using a laser printer and subsequently placed on a supporting plane.

We now briefly describe our acquisition pipeline (illustrated in Fig. 2). The first step is to capture the environment lighting. We scan the geometry of the mirror ball using KinectFusion (Sec. 4.1), fit a sphere to the geometry, and process the frames taken by the RGB camera to generate a environment map (Sec. 4.3). Next, the geometry of the object of interest is also acquired via KinectFusion. We then segment the object into a manually-specified number of clusters of different specular BRDFs, and compute the corresponding roughness parameter α, based on the IR signals reflected off the object (Sec. 5.1 and 5.2). After that, we rapidly optimize for spatially-varying ρ_d and ρ_s in an inverse rendering framework, using data from the RGB camera (Sec. 5.3). Finally, post-processing (Sec. 6) is performed to produce the output of our system: a triangular mesh representing the geometry, along with a few texture maps storing ρ_d, ρ_s and α.

Note that except for post-processing, all computation is done on-the-fly to give prompt visual feedback throughout the scanning process. Please refer to the accompanying video for a live demo of our system.

4.1 Geometry Acquisition

The geometry of an object can be directly acquired using KinectFusion. Foreground / background geometry separation is achieved with the help of the white circles placed around the object of interest: we use the four circles to define a virtual box with a predefined size on top of the supporting plane; anything outside of the box is considered part of the background. After the separation, we discretize the surfaces of the object mesh, by randomly sampling points with a probability proportional to the surface area. These points $\{p_i\}$ are then stored in a spatial grid based on their positions, for fast point-wise nearest neighbor query in later appearance acquisition stage (Sec. 5). In addition, we apply bilateral filtering [Fleishman et al. 2003] to smooth the noisy normals directly obtained from KinectFusion, denoted as $\{n_i\}$.

4.2 RGB Data Pre-processing

We use Kinect SDK API to get image data in the linear space, and thus do not explicitly calibrate the response curves of the RGB camera. For each RGB image frame, we pre-process the data to compensate for potential color shift and time-varying exposures, based on the markers (similar to previous work like [Ren et al. 2011]). Specifically, we first map the calibrated positions of white circles from the 3D space to the image. If one or more centers are found in the image, the corresponding pixels are averaged to obtain a reference white pixel. We then divide every pixel in the image by the reference white pixel on a per-channel basis. The calibrated High-Dynamic-Range (HDR) result is sent to our system for further processing. An appropriate auto-exposure brightness is set manually for the RGB camera to avoid saturations.

4.3 Lighting Acquisition

We acquire the environment lighting from the mirror ball, for inverse rendering in the appearance acquisition stage (Sec. 5). The first step is to scan the geometry of the ball using KinectFusion. However, the resulting geometry is typically noisy, due to the highly-specular nature of the ball material: the reflected IR light is weak at most part of the ball where there is no mirror reflection; therefore, the corresponding depth estimate is less accurate, when compared to a material with a non-negligible Lambertian component. To alleviate this problem, we exploit the prior information that the object is a ball, and fit a perfect sphere to the captured geometry using least squares, right after the automatic object/background geometry separation. Next, image frames obtained from the RGB camera are mapped onto the sphere; for each pixel, we find the reflection direction with respect to the current view direction, and update the corresponding texel in the environment map. In each update, we calculate the weighted average of existing texel values and the new pixel sample. Following [Lensch et al. 2003], the weight of a sample is computed as $\boldsymbol{\omega}_o \cdot n$, which penalizes grazing views. The final lighting results are mapped onto squares via [Praun and Hoppe 2003].

According to Sec. 4.2, the environment lighting we capture is of high dynamic range. Note that possible lighting blocking in acquisition is alleviated, by using the multi-view weighted average in determining texels of the environment map. One additional benefit of using the markers is that it implicitly aligns the environment map with the object of interest, if both geometries are transformed to the local coordinate system defined by markers. Moreover, we can capture the environment lighting only once, and share it for subsequent appearance scanning based on the same set of markers.

5 Appearance Acquisition

From previous stages, we have already obtained the lighting, and the geometry discretized as points $\{p_i\}$ and corresponding normals $\{n_i\}$. In this stage, we first group points into a user-specified number of clusters depending on the specular BRDF f, and compute the roughness α for each cluster, using the Kinect IR emitter and receiver as an active reflectometer. Next, we switch to the RGB camera and perform inverse rendering, to compute per-point ρ_d and per-cluster ρ_s. It is worth mentioning that Kinect IR images has been used for geometry refinement in [Choe et al. 2014].

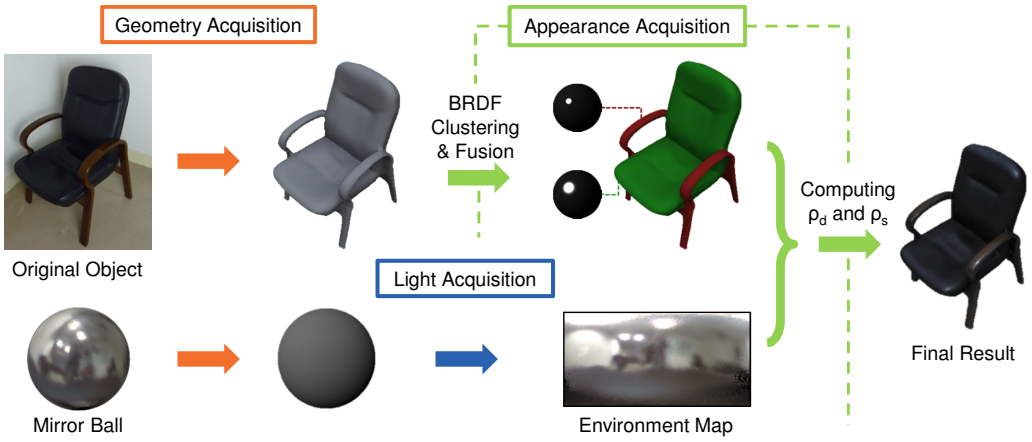

Figure 2: *Our system pipeline: given an object, we first scan its geometry via KinectFusion; next, we segment the object into clusters of similar specular BRDFs, and estimate the roughness of the BRDF in each cluster, using the Kinect as an active IR reflectometer; with a previously captured environment map, we compute ρ_d and ρ_s in an inverse rendering framework; after that, we obtain both the appearance and geometry as the final result.*

5.1 BRDF Fusion

Before describing details about BRDF clustering, we introduce the idea of BRDF fusion, which is used to compute the roughness parameter α, given a cluster of points that share the same specular BRDF f. We apply Eq. 2 to the IR spectrum, which requires known L and L_i. L can be directly measured using the depth camera. But L_i, the IR light emitted by the Kinect and originally designed for depth estimation only, varies considerably in space [Khoshelham and Elberink 2012]. The high-frequency nature makes it challenging and demanding, for average users to recover the exact spatial intensity distribution of the IR light. Instead, we model L_i as a noisy source, and average reflected IR light, when determining the BRDF parameter. The exact value of L_i is not needed, as only the roughness α is computed. The high-level idea is similar to KinectFusion [Izadi et al. 2011], where individual noisy depth maps are integrated to produce results with less noise.

Specifically, for each pixel in an image frame captured by the depth camera, we register it to the geometry, by finding the nearest neighbor p_i in the spatial grid structure we built in the geometry acquisition stage (Sec. 4.1), with respect to the corresponding 3D location of the pixel. Then, the lighting and view directions, estimated from the KinectFusion camera position, are transformed into the local frame at p_i. If either $\boldsymbol{\omega}_i'$ or $\boldsymbol{\omega}_o'$ is close to the grazing angle, we discard the pixel, as the measurements are unreliable. Otherwise, the local $\boldsymbol{\omega}_i'$, $\boldsymbol{\omega}_o'$ along with the IR pixel value are stored in a data structure for further processing. Fig. 3 illustrates the case.

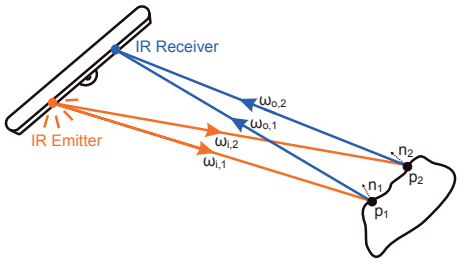

Figure 3: *The Kinect as an active reflectometer in the IR spectrum. Light from the IR emitter may hit a surface point, and get reflected to the IR receiver.*

For the data structure that integrates IR samples, we expect that it should take all samples into consideration for efficient noise reduction, and at the same time consume moderate memory footprint. To meet these requirements, We propose a Half-angle BRDF Sample Accumulation Buffer (HBSAB), essentially representing a discretized version of a BRDF. An HBSAB stores an average IR value v and a weight w for each discretized half vector, parameterized over a square according to [Praun and Hoppe 2003]. Suppose that we receive a new IR pixel valued $\hat{v}$, with a weight $\hat{w}$ of $\boldsymbol{\omega}_o \cdot n$ (similar to Sec. 4.3). To update the buffer, we compute the half vector between the local $\boldsymbol{\omega}_i'$ and $\boldsymbol{\omega}_o'$, locate the corresponding entry in the buffer, and update it as:

$$v_{new} = \frac{v \cdot w + \hat{v} \cdot \hat{w}}{w + \hat{w}}, \quad w_{new} = w + \hat{w}. \tag{3}$$

The final step is to compute the roughness α from the accumulation buffer: for each entry with its weight exceeding a threshold, we treat it as a BRDF sample by setting both $\boldsymbol{\omega}_i'$ and $\boldsymbol{\omega}_o'$ to their half vector, and the BRDF value to v; then we separate the specular reflectance from the diffuse reflectance, which is computed as the minimum over all measured reflectance samples; Ward BRDFs are finally fitted to the specular reflectance using non-linear optimizations [Ngan et al. 2005]. As the user scans the object, we update the HBSAB and perform BRDF fitting from time to time. Please refer to Fig. 4 for HBSAB examples and the influence of increasing number of IR samples.

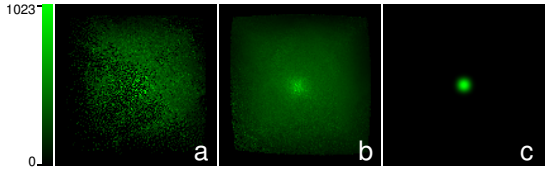

Figure 4: *The IR noise is considerably reduced with increasing number of IR samples. Average IR values in HBSABs of 10K (a) and 100K samples (b) are visualized. A specular BRDF fitting result on (b) is shown in (c).*

We now justify the implicit assumption in the aforementioned HBSAB processing, that both $\boldsymbol{\omega}_i'$ and $\boldsymbol{\omega}_o'$ are equal to their half vector. In fact, $\boldsymbol{\omega}_i'$ and $\boldsymbol{\omega}_o'$ are close, when capturing with Kinect. Consider that the IR emitter and receiver are approximately 7.5cm apart on

the Kinect, and the minimum view distance between the Kinect and an object is 40cm. Then, the maximum angle between $\boldsymbol{\omega}_i'$ and $\boldsymbol{\omega}_o'$ is $2 \times \arctan(0.5 \times \frac{7.5}{40}) = 10.7°$. The angle decreases to $7.2/5.4°$ at more typical 60/80cm view distances. Moreover, we conduct simulation experiments, using synthetic Ward models (α ranging from 0.05 to 0.75). Assuming a 40cm view distance, we randomly generate $\boldsymbol{\omega}_i'$ and $\boldsymbol{\omega}_o'$, sample the BRDF at these directional pairs, and update the HBSABs. Next, we compare the α computed from BRDF fitting with the ground truth. For all experiments, the maximum relative error is 0.26%, which is sufficiently small for our applications.

Note that 1D buffers can be used in place of 2D HBSABs for isotropic BRDFs. We employ the 2D buffers, as they are efficiently processed by our system already, and can directly handle anisotropic BRDFs in a future extension.

5.2 BRDF Clustering

In this subsection, we describe how to partition the object into a user-specified number of clusters while scanning, as well as a subsequent manual refinement process. First of all, we would like to exploit the reflected IR data, which essentially represents partial SV-BRDFs over the object, to facilitate clustering. However, the IR data reflected from one point p_i is usually noisy (Sec. 5.1), which is not reliable for clustering directly. Inspired by reflectance sharing [Zickler et al. 2006], we could aggregate the IR data from all points of the same specular material, to obtain a result with less noise. However, determining points of the same specular BRDF is our problem in the first place.

To tackle this challenge, we propose a data-structure called BRDF cut. The key idea to decouple the determination of points with reliable IR measurements and specular BRDF similarities, into two separate processes: we first use a BRDF cut to identify groups of points with reliable IR measurements, then perform clustering over these groups to find points that have similar specular BRDFs. Specifically, we build a binary tree that partitions all points $\{p_i\}$ of an object, based on the vector concatenating a position and its normal: we start with all $\{p_i\}$, recursively divide them into smaller sets, until a minimum number of points in a set is reached. Please see Fig. 5 for an example. In this tree, every node represents a subset of points $\{p_i\}$, and every cut corresponds to a partition of all points. In addition, an HBSAB is stored in each node.

When new IR data at a point p_i is received, we look up its corresponding leaf node in the tree, and update the HBSAB using BRDF fusion. To determine whether the points in a node have reliable IR measurements, we weighted-average the HBSABs of its children, and test if there are enough samples and a sufficient coverage in a potential highlight region, based on a synthetic BRDF of a broad highlight. Note that this test is similar to [Lensch et al. 2003].

BRDF clustering is performed based on nodes in the cut. Initially, the cut contains the root only. It is then updated as we traverse its nodes: whenever a node has at least one child with reliable measurements, we delete the node from the cut, and insert its two children instead. Next, we perform k-means clustering, using HBSABs of the nodes in the cut with reliable measurements. For the remaining unreliable cut nodes, we assign them to one of the cluster centers based on the L2 distance between corresponding HBSABs, after subtracting the diffuse components. As more IR data is received, the cut / clusters are gradually refined in space. In our user interface, we visualize the clustering by rendering the points of each cluster with a different color. In the end, we obtain both the clustering of specular materials, and the corresponding BRDF roughness parameters by fitting aggregated HBSABs, in a unified process.

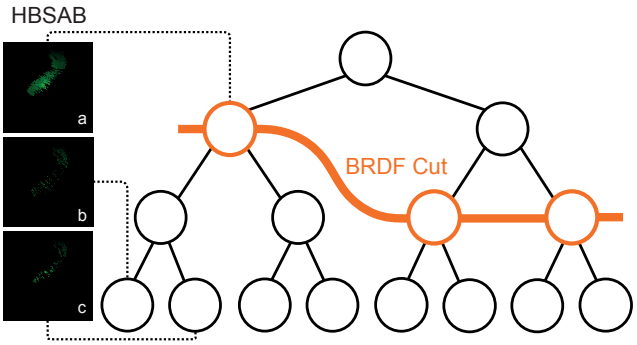

Figure 5: *A BRDF cut example. By averaging the HBSABs at leaf nodes (b & c), the computed HBSAB at the cut node (a) is more reliable than that of any of its leaf node.*

It is worth mentioning that the general 4D BRDF cut was first introduced in [Cheslack-Postava et al. 2008] to accelerate rendering, but not implemented due to its prohibitive size. In comparison, our 2D HBSAB-based BRDF cut requires much less memory footprint and can be efficiently implemented.

Manual Clustering Refinement. Even though our BRDF-cut-based method can quickly cluster specular BRDFs, there might be small regions on the object assigned to wrong clusters, mainly due to unreliable IR measurements (e.g., parts with significant occlusions). We provide a simple, optional 3D painting interface to fix this issue. The user can easily and quickly paint the correct cluster assignment to parts of the 3D object, using a brush with an adjustable size.

5.3 Computation of ρ_d and ρ_s

We use the image frames captured by the RGB camera to compute ρ_d and ρ_s in the visible spectrum. Similar to Sec. 5.1, for each calibrated pixel L in an image frame, we register it to the nearest point p_i. Next, the view direction $\boldsymbol{\omega}_o$ estimated from the KinectFusion camera position, is transformed into the local frame, denoted as $\boldsymbol{\omega}_o'$. L and $\boldsymbol{\omega}_o'$ are then buffered for subsequent processing.

To rapidly compute ρ_d and ρ_s for prompt visual feedback, we use an inverse rendering framework based on precomputed double-product wavelet integrals [Ng et al. 2003], similar to [Haber et al. 2009]. We first derive the related formulation. Substituting Eq. 1 into Eq. 2, we have:

$$\begin{aligned} L(\boldsymbol{\omega}_o) &= \int_\Omega L_i(\boldsymbol{\omega}_i)(\frac{\rho_d}{\pi} + \rho_s f(\boldsymbol{\alpha}; \boldsymbol{\omega}_i', \boldsymbol{\omega}_o'))(n \cdot \boldsymbol{\omega}_i) d\boldsymbol{\omega}_i, \\ &= \rho_d \int \frac{1}{\pi} L_i(\boldsymbol{\omega}_i)(n \cdot \boldsymbol{\omega}_i) d\boldsymbol{\omega}_i + \\ &\quad \rho_s \int L_i(\boldsymbol{\omega}_i) f(\boldsymbol{\alpha}; \boldsymbol{\omega}_i', \boldsymbol{\omega}_o'))(n \cdot \boldsymbol{\omega}_i) d\boldsymbol{\omega}_i. \end{aligned} \quad (4)$$

For efficient evaluation of Eq. 4, we precompute the first integral as $D(n) = \int \frac{1}{\pi} L_i(\boldsymbol{\omega}_i)(n \cdot \boldsymbol{\omega}_i) d\boldsymbol{\omega}_i$, for different n sampled from the unit sphere. The second integral in the equation can be viewed as a convolution of two hemispherical functions, the local lighting $\tilde{L}(n; \boldsymbol{\omega}_i') = L_i(\boldsymbol{\omega}_i)(n \cdot \boldsymbol{\omega}_i)$, and the BRDF slice $\tilde{f}(\boldsymbol{\alpha}, \boldsymbol{\omega}_o'; \boldsymbol{\omega}_i') = f(\boldsymbol{\alpha}; \boldsymbol{\omega}_i', \boldsymbol{\omega}_o')$, where the variable is $\boldsymbol{\omega}_i'$. By expressing both functions using Haar wavelet basis, this integral can be quickly computed as the sum of the product of common non-zero coefficients [Ng et al. 2003]. In practice, we precompute the wavelet transforms for $\tilde{f}(\boldsymbol{\alpha}, \boldsymbol{\omega}_o'; \cdot)$ with respect to a variety of α

and discretized $\boldsymbol{\omega}'_o$. For $\tilde{L}(n;\cdot)$, we compute its Haar wavelet coefficients after the lighting acquisition, essentially pre-rotating the environment map.

Now we can express Eq. 4 as a linear equation:

$$L = D\rho_d + S\rho_s, \tag{5}$$

where L is the calibrated RGB pixel value, and S is the double-product term.

Following previous work such as [Wang et al. 2008], we compute $\rho_{d,i}$, the diffuse albedo at p_i, as:

$$\rho_{d,i} = \min_j \frac{L_{i,j}}{D_{i,j}}, \tag{6}$$

where $L_{i,j}$ is the j-th RGB sample reflected from p_i, $D_{i,j}$ is the corresponding diffuse term. In practice, we maintain a histogram at each point p_i, and typically use the 10th percentile, rather than the minimum among all $\frac{L_{i,j}}{D_{i,j}}$, as $\rho_{d,i}$. In our experiments, this approach is more robust against errors like mis-registrations.

Once $\rho_{d,i}$ is computed for all points of the object, we fix them and derive the following equation for ρ_s from Eq. 5:

$$S_{i,j}\rho_s = L_{i,j} - D_{i,j}\rho_{d,i}. \tag{7}$$

Here p_i is a point that belongs to the current cluster of specular material, and $S_{i,j}$ is the corresponding double-product terms. Since the only unknown factor is ρ_s, we compute it as the analytical least squares solution to Eq. 7:

$$\rho_s = \frac{\sum S_{i,j}(L_{i,j} - D_{i,j}\rho_{d,i})}{\sum S_{i,j}S_{i,j}}. \tag{8}$$

In our implementation, we only store the nominator and the denominator on the right hand side of the equation, and update them as new RGB samples arrive. There is no need to store individual $L_{i,j}$, $S_{i,j}$ or $D_{i,j}$. Essentially, we use very small constant-sized memory to solve a global optimization, taking into account all samples.

6 Additional Details

Calibrations. We estimate the IR background noise by switching off the IR emitter, capturing and averaging a number of IR frames, and finding in a histogram the bin with most pixels as the noise level.

We calibrate the IR/depth camera response curve by viewing a sheet of white paper from different distances. Assuming that the IR light intensity distribution is spatially and temporally stationary, we compute both unknown IR intensity distribution, modeled as a shifted Γ distribution, and the response curve using the reflected IR light from the paper at different distances via a brute-force optimization. Our estimated γ is 0.9. The computed probability density function for IR intensity x is $f(x) = \frac{t^{k-1}e^{-\frac{t}{\theta}}}{\theta^k\Gamma(k)}$, where $k = 1.6$, $\theta = 2.8$ and $t = -3.4 + 57.7d^2x$. Note that d is the view distance.

Post-processing. Once the acquisition using the Kinect is finished, we perform post-processing to generate the final result. A new mesh is created by the ball-pivoting algorithm [Bernardini et al. 1999] from the discretized points of the object. We then clean holes in the mesh, create a uv parameterization, and export textures representing ρ_d, ρ_s and α.

7 Results and Discussions

We capture the appearance of objects, using the IR and the RGB cameras, both at a resolution of 640×480 with a rate of 30 frames per second. We set up different platforms for scanning objects of different sizes. For small objects, we place the white circle markers on top of a cabinet, which serves as the supporting plane. We also cover the rest surface area with printed black paper, to reduce potential inter-reflections between the top of the cabinet and the object. For large objects, we directly place on the floor four sheets of paper, each one with a white circle in the center and a black background. We conduct our acquisition experiments in a small office with four area light sources on the ceiling, which is visualized in Fig. 2. In the geometry acquisition stage, we randomly sample 100K points over the surfaces of the mesh obtained from KinectFusion. We represent the environment light at a resolution of 2×128^2. The prerotated environment maps, expressed in Haar wavelets, occupies 145MB. A BRDF slice of fixed $\boldsymbol{\omega}_o$ is represented at a resolution of 128^2. We precompute the wavelet transforms of 500 Ward BRDFs of varying roughness parameters. The size of the result is 9.1GB.

We demonstrate in Fig. 6 the appearance acquisition results on five objects with a range of different materials, an armchair made of leather and lacquered wood, a plastic trashbin, a ceramic pot, an eggplant and a pepper. Here visual comparisons are shown, between the calibrated image from the RGB camera and the rendering result using our reconstructed appearance. Our results qualitatively match the corresponding images, and can be used to generate realistic rendering of the objects under novel lighting and view conditions. Please also refer to the accompanying video for animated rendering results.

We further test our system on a more challenging object, which varies continuously in specular roughness, as shown in Fig. 7. A silver metallic paint, composed of tiny metallic flakes, is manually sprayed on a plaster ball (Fig. 7-a). We spray the paint mainly towards one region as indicated by the white arrow in Fig. 7-d, in order to create a varying density of metallic flakes, which result in changing specular roughness, that increases from that region to distant ones. Our reconstruction result (Fig. 7-b & e) roughly approximates the original appearance, using four clusters of specular materials with no manual refinement (Fig. 7-d). The estimated α are 0.11 / 0.12 / 0.14 / 0.22 for the cluster visualized in red / green / blue / yellow, respectively. Overall, the clustering result is in accordance with the spray paint density over the ball. In comparison, a reconstruction result using a single specular material (Fig. 7-c & f) approximates the original appearance poorly. For example, in Fig. 7-a, the highlight on the top part of the ball is blurred out due to a large specular roughness; but a narrower and brighter highlight appears in the same region in Fig. 7-c. Note that due to the IR measurement noise (Sec. 5.1 & 5.2), we cannot use more clusters or obtain a more spatially-refined clustering of materials, which limits the quality of our result.

We would like to emphasize that our results are not high-precision SVBRDF measurements. Instead, the focus of our work is to allow average users to intuitively and quickly model an approximation to the appearance of an object, which can later be used for realistic rendering/editing, solely using low-end, off-the-shelf components.

Validation on Planar Samples. We physically validate our approach on three different planar material samples: diffuse blue paper, glossy gold paint and mirror-like red metallic foil (Fig. 8). Their appearance is first captured using a small light stage, with 80 lighting directions by 7 view directions. Then, the acquired ground-truth BRDFs are fitted using Ward models, and compared with results from our system in Fig. 8. For the diffuse paper and the glossy paint, our results approximate the ground-truth BRDFs

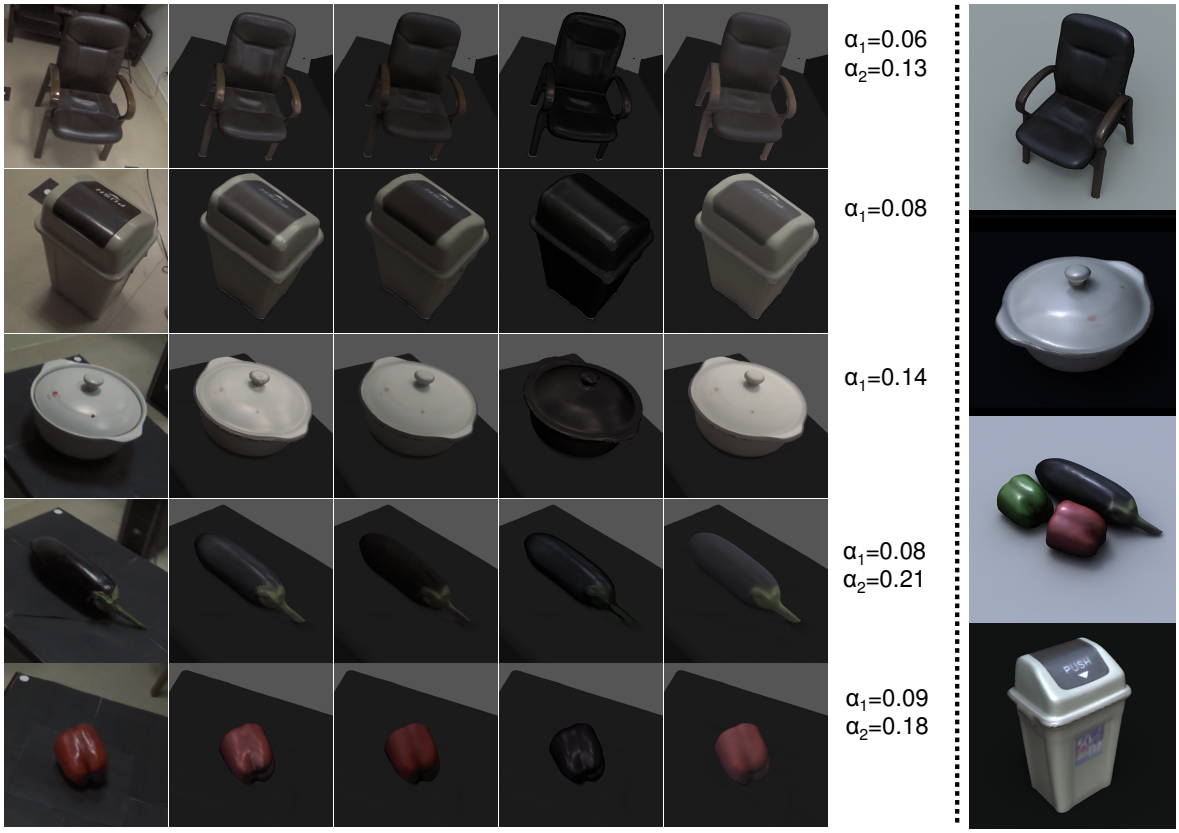

Figure 6: *Comparisons between our results and images captured from the RGB camera (left), and rendering results under novel lighting and view conditions (right). From the left column to right, calibrated RGB images, our rendered results, rendering using diffuse components, rendering using specular components, rendering using a Lambertian model with $\rho_d + \rho_s$ as albedos, estimated roughness parameters, novel rendering results (we change the hue of a red pepper to green).*

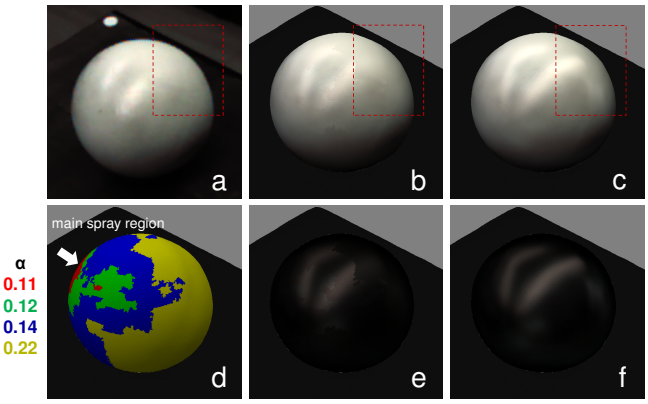

Figure 7: *Capturing a ball that varies in specular roughness: a photograph of the ball (a); our reconstruction (b) and estimated specular components (e), using four specular clusters with no manual refinement, which are visualized in (d) with cluster-color-coded roughness α listed on the left; a reconstruction using one specular cluster only (c) and the corresponding specular component (f). Note the appearance differences in red dashed rectangles in (a, b & c).*

reasonably well in terms of visual perception and the roughness α. However, for the mirror-like foil, our estimate of α is considerably larger than the ground-truth. The main reason is that the Kinect cannot estimate the camera position and orientation with enough precision to capture highly specular materials.

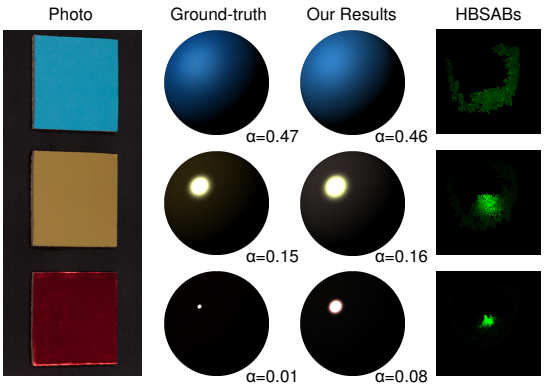

Figure 8: *Validation experiments. From left to right: a photograph of three planar material samples; BRDFs fitted using measurements from a light stage, rendered with a directional light; our results; corresponding HBSABs.*

Error Analysis on IR Roughness Computation. A number of factors could affect the roughness computed from IR measurements. We would like to identify the major source of error and suggest possible improvements. To do so, a series of experiments are conducted on a glossy ball, whose ground-truth roughness ($\alpha_{truth} = 0.20$) is computed via an image-based approach [Marschner et al. 1999], using an LED point light source and a Canon EOS 50D DSLR (Fig. 9-a).

First, using our system we obtain a roughness of $\alpha_{ours} = 0.24$ (Fig. 9-b), which is larger than α_{truth}. Note that α_{ours} is smaller than $\alpha_{single} = 0.28$, which is obtained from a single IR image using [Marschner et al. 1999]. To investigate the impact of inaccurate geometry / normals, we fit a perfect sphere to the scanned geometry (similar to Sec. 4.3), and get $\alpha_{fit} = 0.24$. So the measured geometry / normal is not the main error source. Finally, we cover the IR emitter with a thin layer of paper, to make the IR light distribution close to being uniform (Fig. 9-c). Since the depth computation in Kinect does not work in this case, we again employ [Marschner et al. 1999] to estimate the roughness $\alpha_{uniform} = 0.20$, which is equal to α_{truth}. This suggests that the major source of error in IR roughness computation is the large spatial variation in the IR light source. The accuracy of our system on general objects is expected to improve, using future devices that emit more uniform light, such as Kinect v2.

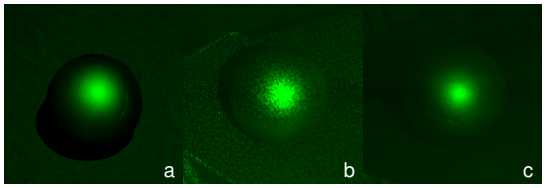

Figure 9: *Computing the IR roughness by different methods: using [Marschner et al. 1999] from an HDR image (a) captured by a DSLR; using our system based on IR measurements (b); using [Marschner et al. 1999] from a single IR image (c), while covering the IR emitter with paper. Original images are shown in green for a better visualization.*

Performance. We implement our system and measure its performance on a PC, with a quad-core Intel i7 3770K and 32GB of memory. While the user is scanning the object, our system provides interactive visual feedback, at 7∼16 frames per second. In our experiments, acquiring one novel environment lighting takes 3∼5 minutes. A detailed breakdown of the timing results for appearance acquisition can be found in Tab. 1. On average, it takes 10 minutes for a user to scan the geometry and the appearance of an object. The majority of time is spent on the geometry acquisition via KinectFusion and capturing ρ_d, to get a good coverage of view points, which is important for rendering/relighting of the object.

	Geo. Acq.	Clus. & Fus.	Ref.	ρ_d	ρ_s	Total
Armchair	4.2	3.2	1.1	3.7	2.0	14.2
Trashbin	3.9	0.9	0.0	3.6	2.0	10.4
Pot	2.2	0.6	0.0	5.1	1.3	9.2
Eggplant	1.6	1.1	0.6	3.0	1.7	8.0
Pepper	1.9	1.2	0.5	3.8	2.0	9.4

Table 1: *Appearance acquisition timing results (in minutes) using our system. From left to right, geometry acquisition, BRDF clustering and fusion, manual clustering refinement, scanning and computing ρ_d/ρ_s, and the total time.*

Limitations. The quality of our results is negatively affected, if the assumptions introduced in Sec. 3 are not satisfied. For example, interreflections could be incorrectly computed as part of the diffuse albedo ρ_d. Moreover, ignoring the visibility factor makes our approach less accurate for objects with significant occlusions. One possible way to alleviate this problem, is to record all input data from the Kinect through an on-the-fly scan. In a subsequent post-processing step, one can compute a more accurate result, by calculating visibility functions of all points over the geometry.

Our acquisition quality is also limited by the Kinect sensor, including inaccurate camera position/orientation estimate and the highly noisy IR light source. It would interesting to develop new devices or tailor existing ones to improve on these aspects.

8 Conclusions and Future Work

We present AppFusion, an interactive appearance acquisition system for ordinary users, which is built entirely on low-end, off-the-shelf components and has the unique feature of displaying captured results on-the-fly. We test our system by acquiring the appearance of a range of objects with various materials. Our work is a first step towards interactive, realistic appearance acquisition for home users. We hope that more work will be stimulated, to make high-quality appearance acquisition as easy as using KinectFusion for scanning geometry today, using tens of millions of Kinect sensors that have already been shipped.

For future work, we would like to address the limitations of our system, as mentioned in Sec. 7. In addition, due to the device limitation [Microsoft 2013], we are unable to access data simultaneously from the Kinect RGB and IR cameras. We hope that future hardware/SDK design could implement such a functionality, so that BRDF clustering and fusion and the computation of ρ_d can be performed in one pass to further reduce the acquisition time.

References

AITTALA, M., WEYRICH, T., AND LEHTINEN, J. 2013. Practical SVBRDF capture in the frequency domain. *ACM Trans. Graph. 32*, 4 (July), 110:1–110:12.

ALLDRIN, N., ZICKLER, T., AND KRIEGMAN, D. 2008. Photometric stereo with non-parametric and spatially-varying reflectance. In *Proc. of CVPR 2008*.

BERNARDINI, F., MITTLEMAN, J., RUSHMEIER, H., SILVA, C., AND TAUBIN, G. 1999. The ball-pivoting algorithm for surface reconstruction. *IEEE Trans. Vis. Comp. Graph. 5*, 4, 349–359.

CHESLACK-POSTAVA, E., WANG, R., AKERLUND, O., AND PELLACINI, F. 2008. Fast, realistic lighting and material design using nonlinear cut approximation. *ACM Trans. Graph. 27*, 5 (Dec.), 128:1–128:10.

CHOE, G., PARK, J., TAI, Y.-W., AND KWEON, I. S. 2014. Exploiting shading cues in kinect ir images for geometry refinement. In *Proc. of CVPR 2014*.

DANA, K. J., VAN GINNEKEN, B., NAYAR, S. K., AND KOENDERINK, J. J. 1999. Reflectance and texture of real-world surfaces. *ACM Trans. Graph. 18*, 1 (Jan.), 1–34.

DEBEVEC, P., HAWKINS, T., TCHOU, C., DUIKER, H.-P., SAROKIN, W., AND SAGAR, M. 2000. Acquiring the reflectance field of a human face. In *Proc. of SIGGRAPH '00*, 145–156.

DONG, Y., WANG, J., TONG, X., SNYDER, J., LAN, Y., BEN-EZRA, M., AND GUO, B. 2010. Manifold bootstrapping for SVBRDF capture. *ACM Trans. Graph. 29*, 4 (July), 98:1–98:10.

DORSEY, J., RUSHMEIER, H., AND SILLION, F. 2007. *Digital Modeling of Material Appearance*. Morgan Kaufmann Publishers Inc.

FLEISHMAN, S., DRORI, I., AND COHEN-OR, D. 2003. Bilateral mesh denoising. *ACM Trans. Graph. 22*, 3 (July), 950–953.

GARDNER, A., TCHOU, C., HAWKINS, T., AND DEBEVEC, P. 2003. Linear light source reflectometry. *ACM Trans. Graph. 22*, 3 (July), 749–758.

GHOSH, A., CHEN, T., PEERS, P., WILSON, C. A., AND DEBEVEC, P. 2009. Estimating specular roughness and anisotropy from second order spherical gradient illumination. In *Proc. of EGSR 2009*, 1161–1170.

HABER, T., FUCHS, C., BEKAERT, P., SEIDEL, H.-P., GOESELE, M., AND LENSCH, H. P. A. 2009. Relighting objects from image collections. *Proc. of CVPR 2009*.

HERTZMANN, A., AND SEITZ, S. M. 2003. Shape and materials by example: A photometric stereo approach. In *Proc. of CVPR 2003*, 533–540.

IZADI, S., KIM, D., HILLIGES, O., MOLYNEAUX, D., NEWCOMBE, R., KOHLI, P., SHOTTON, J., HODGES, S., FREEMAN, D., DAVISON, A., AND FITZGIBBON, A. 2011. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *Proc. of UIST 2011*, 559–568.

KAJIYA, J. T. 1986. The rendering equation. In *Proc. of SIGGRAPH 86*, 143–150.

KHOSHELHAM, K., AND ELBERINK, S. O. 2012. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors 12*, 2, 1437–1454.

KNECHT, M., TANZMEISTER, G., TRAXLER, C., AND WIMMER, M. 2012. Interactive BRDF estimation for mixed-reality applications. *Journal of WSCG 20*, 1 (June), 47–56.

LENSCH, H. P. A., KAUTZ, J., GOESELE, M., HEIDRICH, W., AND SEIDEL, H.-P. 2003. Image-based reconstruction of spatial appearance and geometric detail. *ACM Trans. Graph. 22*, 2 (Apr.), 234–257.

LI, G., WU, C., STOLL, C., LIU, Y., VARANASI, K., DAI, Q., AND THEOBALT, C. 2013. Capturing relightable human performances under general uncontrolled illumination. *Comput. Graph. Forum (Proc. of EG 2013) 32*, 2, 275–284.

MARSCHNER, S. R., WESTIN, S. H., LAFORTUNE, E. P. F., TORRANCE, K. E., AND GREENBERG, D. P. 1999. Image-based BRDF measurement including human skin. In *Proc. of EGWR '99*, 131–144.

MATUSIK, W., PFISTER, H., BRAND, M., AND MCMILLAN, L. 2003. Efficient isotropic BRDF measurement. In *Proc. of EGWR 2003*, 241–247.

MCALLISTER, D. K. 2002. Ph.D. Thesis. A generalized surface appearance representation for computer graphics. University of North Carolina at Chapel Hill.

MICROSOFT. 2013. Kinect SDK `http://www.microsoft.com/en-us/kinectforwindowsdev/`.

NG, R., RAMAMOORTHI, R., AND HANRAHAN, P. 2003. All-frequency shadows using non-linear wavelet lighting approximation. *ACM Trans. Graph. 22*, 3 (July), 376–381.

NGAN, A., DURAND, F., AND MATUSIK, W. 2005. Experimental analysis of BRDF models. In *Proc. of EGSR 2005*, 117–126.

PALMA, G., CALLIERI, M., DELLEPIANE, M., AND SCOPIGNO, R. 2012. A statistical method for SVBRDF approximation from video sequences in general lighting conditions. *Comput. Graph. Forum (Proc. of EGSR 2012) 31*, 4, 1491–1500.

PRAUN, E., AND HOPPE, H. 2003. Spherical parametrization and remeshing. *ACM Trans. Graph. 22*, 3 (July), 340–349.

REN, P., WANG, J., SNYDER, J., TONG, X., AND GUO, B. 2011. Pocket reflectometry. *ACM Trans. Graph. 30*, 4 (July), 45:1–45:10.

ROMEIRO, F., VASILYEV, Y., AND ZICKLER, T. 2008. Passive reflectometry. In *Proc. of ECCV 2008*, 859–872.

TUNWATTANAPONG, B., FYFFE, G., GRAHAM, P., BUSCH, J., YU, X., GHOSH, A., AND DEBEVEC, P. 2013. Acquiring reflectance and shape from continuous spherical harmonic illumination. *ACM Trans. Graph. 32*, 4 (July), 109:1–109:12.

WANG, J., ZHAO, S., TONG, X., SNYDER, J., AND GUO, B. 2008. Modeling anisotropic surface reflectance with example-based microfacet synthesis. *ACM Trans. Graph. 27*, 3 (Aug.), 41:1–41:9.

WARD, G. J. 1992. Measuring and modeling anisotropic reflection. *SIGGRAPH Comput. Graph. 26*, 2 (July), 265–272.

WEYRICH, T., MATUSIK, W., PFISTER, H., BICKEL, B., DONNER, C., TU, C., MCANDLESS, J., LEE, J., NGAN, A., JENSEN, H. W., AND GROSS, M. 2006. Analysis of human faces using a measurement-based skin reflectance model. *ACM Trans. Graph. 25*, 3 (July), 1013–1024.

WEYRICH, T., LAWRENCE, J., LENSCH, H. P. A., RUSINKIEWICZ, S., AND ZICKLER, T. 2009. Principles of appearance acquisition and representation. *Found. Trends. Comput. Graph. Vis. 4*, 2 (Feb.), 75–191.

ZICKLER, T., RAMAMOORTHI, R., ENRIQUE, S., AND BELHUMEUR, P. N. 2006. Reflectance sharing: Predicting appearance from a sparse set of images of a known shape. *IEEE Trans. Pattern Anal. Mach. Intell. 28*, 8 (Aug.), 1287–1302.