Thomas Richter-Trummer

Instant Mixed Reality Lighting from Casual Scanning

MASTER'S THESIS

to achieve the university degree of

Diplom-Ingenieur

submitted to

Graz University of Technology

Supervisor

Univ.-Prof. Dipl.-Ing. Dr.techn Dieter Schmalstieg
Institute for Computer Graphics and Vision

Graz, Austria, June. 2016

Abstract

We present a method for recovering both incident lighting and surface materials from casually scanned geometry. By casual, we mean a rapid and potentially noisy scanning procedure of unmodified and uninstrumented scenes with a commodity RGB-D sensor. In other words, unlike reconstruction procedures which require careful preparations in a laboratory environment, our method works with input that can be obtained by consumer users. To ensure a robust procedure, we segment the reconstructed geometry into surfaces with homogeneous material properties and compute the radiance transfer on these segments. With this input, we solve the inverse rendering problem of factorization into lighting and material properties using an iterative optimization in spherical harmonics form. This allows us to account for self-shadowing and recover specular properties of Phong-like materials. The resulting data can be used to generate a wide range of mixed reality applications, including the rendering of synthetic objects with matching lighting into a given scene, but also re-rendering the scene (or a part of it) with new lighting. We show the robustness of our approach with real and synthetic examples under a variety of lighting conditions and compare them with ground truth data.

Kurzfassung

Diese Arbeit präsentiert eine Methode die Beleuchtung und Material aus gescannter Geometrie zurückzurechnen. Die gescannten Daten können relativ fehlerbehaftet und verrauscht sein wie das üblicherweise bei kommerziell erhältlichen RGB-D Sensoren und Structure-From-Motion Systemen der Fall ist. Die Methode ist folglich nicht auf kalibrierte Beleuchtungssysteme oder sonstiges professionelles Equipment angewiesen und kann von einem typischen Endverbraucher bedient werden. Die Methode segmentiert dazu die rekonstruierte Geometrie in Teile die einheitliche Materialeigenschaften aufweisen und berechnen die spezifische Lichtausstrahlung dieser Segmente. Diese Eingabedaten werden dann mit einem Lineargleichungslöser iterativ in Spherical Harmonics basierende Beleuchtung und Phong-Material Eigenschaftsparameter zerlegt. Diese Herangehensweise erlaubt es auch höher komplexe Licht-Transferfunktionen wie Selbstabschattung und Global Illumination zu verarbeiten welche die Lösung im vergleich zu einfacheren lokalen Modellen verbessern kann. Die Beleuchtungsattribute und errechneten Materialeigenschaften können in weiterer Folge für das Einfügen von synthetischen Objekten oder das Neuzeichnen der Szene(oder Teile davon) mit veränderter Beleuchtung verwendet werden. Die Robustheit des Ansatzes wird mit Hilfe von echten und synthetischen Tests in unterschiedlichen Beleuchtungssituationen gezeigt und mit Ground-Truth Daten verglichen.

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used.

The text document uploaded to TUGRAZonline is identical to the present master's thesis dissertation.

______________________ Date

______________________ Signature

Acknowledgments

First and foremost I would like to thank Prof. Dieter Schmalstieg for his passionate help, great input and motivation to write down and publish the proposed methods. He was the main driving factor to create this thesis and was there from the first spark of the idea when I sat down together with Lukas Gruber, and had the idea of using a linear solver over higher complex spherical harmonics based linear equation system, that led to a first publication in 2012. Together with Dieter we were able to expand this single material method to arbitrary numbers of materials and even found a solution for specular properties, where I thought might not be possible with casual input data. Fortunately he convinced me to try this out and proved me (and some famous papers) wrong resulting in a, I think, novel approach to recover a wide variety of material properties. Additionally I would like to thank Denis Kalkofen for enlightening discussions and his great creativity with data acquisition and processing. Furthermore I want to thank Angelique (I love you) and friends, family and work mates for supporting me to get through the late hours of data acquisition, equation solving and rendering. Thanks to my computer to process test data trough dark nights and days without crashing, all though its hard disk sticks half out of the case and its CPU runs overclocked at sometimes alarming temperatures, but never failed. This thesis made me work together with amazingly talented people that brought me a great leap forward in life and motivated me to invest more time in research and development in the future.

Contents

1 Introduction

Recent advances in sensing technologies, in particular inexpensive RGB-D cameras, have lowered the cost and effort for 3D scanning. Scanning detailed object geometry is now possible in real time, providing essential input for many applications of Mixed Reality (MR). Thus, we can safely say that *geometric* reconstruction is now sufficiently "casual" for MR. However, *photometric* reconstruction proves to be more difficult. To determine *both* material and illumination from the same set of input images, we must solve an ill-posed inverse rendering problem. Just from an image, we can not be sure which combination of light color and surface material caused a particular color observation in an image.

For this reason, most photometric reconstruction methods put additional constraints on the input. These constraints make sense for professional studio work, but cannot be afforded by casual users. For instance, consider the capabilities of a user wanting to play a game or shop for furniture at home with MR:

- The user must be able to capture unmodified environments of up to room size. Special setups, such as a green-screens, are not available.
- Capturing must work rapidly (under 5 minutes) and in one pass. The user cannot be expected to work for a long time or repeat the capturing procedure.
- Input must be obtained from a single, inexpensive camera. We used a Kinect V1, which is currently the most widespread and economical device for RGB-D sensing. Such cameras set exposure and white-balancing automatically and do not even allow to read out these settings.
- Multiple cameras cannot be used. A suitable rig for mounting the cameras or a trigger for camera synchronization are not consumer items.
- Controlled camera motion cannot be used. Turntables and other capturing contraptions are not consumer equipment. Even when operated by an experienced user, data collected with a handheld camera will be corrupted by noise and cover the scene with varying density.

- Controlled lighting cannot be used. This rules out calibrated light stages. Moreover, the restriction to a single capturing pass also rules out repeated capturing under multiple (uncalibrated) illumination conditions.

The main contribution of our work is a method for obtaining estimations for illumination and material using input from casual scanning. As we will point out in the discussion of related work below, most existing methods are not suitable for such a scenario. We demonstrate that even with the severe restrictions enumerated above, it is possible to deliver high-quality results. We achieve this by first segmenting the reconstructed geometry into surfaces with homogeneous material properties and computing the radiance transfer on these segments in spherical harmonics form. The inverse rendering problem is solved with an iterative least squares optimization, recovering both incident lighting and Phong-like materials with diffuse and specular properties.

We demonstrate the versatility of our approach with results across the whole spectrum of photo-realistic MR applications, spanning from Augmented Reality (inserting virtual objects into real scenes) to Augmented Virtuality (inserting scanned objects into virtual scenes).

2 Related Work

A large body of work on coherent rendering for MR exists [17]. Here, we focus on methods which provide an estimate of the current lighting in the user's environment, and on methods which recover information about the material of the objects around the user.

2.1 Direct estimation of light

The seminal work of Debevec demonstrated how to make use of a light probe in order to directly measure the incident light [4]. Light probes are placed inside an environment and provide light measures as high-dynamic range images, which are subsequently used to compute an environment map. Active light probes obtain the environment map directly. A camera with fish-eye lens or an omni-directional camera is placed directly in the scene to acquire images of all directions in one step [9, 15, 24]. Passive light probes make use of a reflective object, commonly a sphere, which is placed within the scene and observed by a camera [5].

While light probes deliver environment maps at full frame rates, they are invasive and not suitable for casual use. If the lighting is assumed to be static, the environment map can be acquired with an offline scanning pass. Meilland [22] introduced a dense real-time 3D tracking and mapping approach and uses an RGB-D camera to capture a 3D HDR light field for reflection rendering of artificial objects into the video stream. It also detects light sources for rendering reflective objects with shadows. However, obtaining a full environment map with a handheld camera is significantly more work than just scanning the relevant portion of the environment. Without an artificial light probe or a full, directly scanned environment map, the incident light must be recovered by analyzing reflections in the scene, as explained in the following.

2.2 Light estimation from specular reflections

Specular reflections observed on a known object allow direct estimation of the incident light from the reflected direction. This principle can not only be applied to a dedicated light probe, but also to any specular object with known shape in the scene. For example, Lagger and Fua [18] detect specular highlights on small moving objects, and Mashita et al. [21] infer the real-world lighting by detecting specular highlights on planar objects.

Jachnik et al. [13] capture a 4D light field over a small planar surface, such as a glossy book cover. They heuristically separate the observations into diffuse and specular reflection by assuming that the diffuse reflection varies only with position, but not with the viewing direction, while specular reflection varies only with viewing direction, but not with position. Our work uses a similar factorization, but applied to a general scene with arbitrary non-planar objects having unknown material properties. Unlike Jachnik et al., we do not know in advance which objects exhibit specularities, and we must also detect self-shadowing effects.

2.3 Light estimation from diffuse reflections

A lack of specular objects can render the search for specular reflections unsuccessful. Alternatively, one can attempt to estimate illumination from diffuse reflections. However, recovering incident light from diffuse surfaces is a more difficult problem, since one must separate the contributions of incoming light from many directions. A mathematically consistent approach for computing diffuse light transport in an efficient manner is based on spherical harmonics (SH). Ramamoorthi and Hanrahan [26] introduce a general framework for spherical harmonics based lighting estimation. It is limited to convex single material geometry, although they propose a more general method for recovering surface parameters from textured geometry in a known lighting environment. This thesis extends the basic principles with adding self shadow and material estimation within an unknown lighting environment.

Gruber et al. [10] demonstrated that an SH framework is able to recover real-world lighting in real time from an RGB-D camera. They reconstruct the scene from depth images and solve for incident directional lighting in SH form from selected sample points on the reconstructed surfaces. However, the sample points must have a good distribution of surface normals. Since diffuse reflection aggregates light from all directions, shadows from other objects in the scene must be computed for every sample point. With image-space optimizations [11], their system can estimate both, dynamic incident light and shadows cast from dynamic real objects, at 20 frames per second on a desktop GPU. However, the lack of specular effects and the sparse sampling owed to the real-time constraints impose restrictions on the visual quality.

Boom et al. [1] presented a system which estimates a single point light source from arbitrary scene geometry. It uses RGB-D input to estimate a single light direction and color image-based object segmentation with no depth or other geometry information for segmentation. They then try to find a light direction by a downhill simplex method to minimize the error between the input image and the re-lit image. They assume a Lambertian reflectance model; specularities and ambient lights are treated as noise. They render synthetic objects into the scene casting shadows. The approach is based on segmenting the image by color into superpixels, for which constant diffuse albedo is assumed. Knorr and Kurz [16] estimate incident light from human faces. Their method uses offline machine learning from a variety of faces under different illumination conditions. The online method applies a face tracker and matches distinctive observations points of the detected face to the training database to estimate the real-world light in SH. However, the accuracy of the face tracking is a limiting factor on the quality of the recovered illumination. Moreover, the method is limited to applications where a face is seen by the camera.

While previous work on light estimation from diffuse reflections either assumes a constant albedo color over the entire scene or requires information about the lighting, our method recovers unknown incident lighting together with unknown Phong-like material properties.

2.4 Light estimation from shadows

Another method for estimating light sources is to observe the shadows in an image. The method is based on full or partial knowledge of the shadow caster's geometry and the correct classification and measurement of the shadow appearance in the image. For example, Wang et al. [33] estimate multiple directional light sources based on critical boundaries and shadow information while the shadow receiving object need to have the same homogenous material. Ikeda et a. [12] estimate illumination from shadows of possible incomplete object shapes captured by an RGB-D camera. However, the observed objects have to be placed on a flat Lambertian reflective plane which receives all shadows. In addition, this system is able to recover only white light and requires expensive computations.

2.5 Retinex-based Albedo estimation

Conventional Intrinsic Image Decomposition [32] [37] decomposes an image into its reflectance and shading parts and creates a diffuse albedo representation, suitable for relighting. The approach operates on 2D images with no information about geometric features like shape and normals and is therefore not suitable to solve the ambiguity between object color and lighting including self occlusion and light bleeding. Recent developments [19] use depth data from a depth sensor to improve the results but they are not building up a full light transfer function. Others [2] use full RGB-D data sets with impressive results, but only produce a single shading color for decomposing the diffuse albedo, which results in errors, if the scene is lit by multiple light colors.

2.6 Material estimation

Separating light and material properties without any prior information is an ill-formed problem, because the observed surface color depends on both, the surface materials and the lighting. If we can make the simplifying assumption that surface material is diffuse and constant over the entire scene, we can estimate the incident lighting by using a linear solver based on color and geometry properties [10, 26, 27]. However, in realistic scenes, surfaces have different material properties, leading to wrong illumination estimates, since albedo color variations are incorrectly compensated by the incident lighting modification. For example, a red surface may lead to a solution involving red light coming from the direction of the plane normal.

In material estimation, a popular approach to resolve the ambiguity between lighting and reflectance is to use changing lighting, for example, using a light stage [6]. Weber et al. [35] reconstruct shape and illumination from an image stream, but require calibrated, switchable light sources. With known lighting, inverse global illumination can accurate recover materials [36].

Alternatively, objects can be moved relative to the light source. For example, a turning plate can be used to rotate small objects relative to a static light source, allowing controlled variation of the angle of incident light. Sato [28] calculates diffuse and specular properties based on a per-frame input stream, but needs a known calibrated point light and a rotating setup. Dong et al. [7] recover a highly detailed data-driven BRDF on the whole surface by rotating the object relative to the lighting, but use convex objects shapes and ignore inter-reflections and self-occlusions.

Li et al. [20] capture a scene of a moving human with multiple cameras under unknown, static lighting in a green room. By tracking with a human shape as a prior, they can use the motion of the human relative to the light sources to obtain multiple observations for every surface point over time. They segment the body into uniform materials, for which they recover a Phong BRDF. Our approach combines a similar material segmentation, but uses radiance transfer functions [25] based on self-occlusion to determine materials with only a single relationship between scene and lighting.

3 Method

Our method simultaneously estimates incident lighting and Phong-like materials of the objects in the scene. As input to this inverse rendering process, we require a mesh representing the scene geometry and a set of keyframes, consisting of images of the scene taken from various known view points. Both can be obtained with a handheld camera. No other knowledge about the lighting or the object's material is needed. Our method is illustrated in Figure 3.1. Denoting vectors in lowercase boldface, matrices in uppercase boldface, scalars in italics, and compound structures in uppercase italics, our method can be outlined as follows:

1. **Data association** (section 3.1): Reconstruct a per-vertex lit texture $\mathbf{f}(\mathbf{v}_i)$ by assigning pixels from keyframes K_j to mesh vertices $\mathbf{v}_i$, and iteratively compensate for relative exposure changes e_j and errors in the camera position $\mathbf{pos}(K_j)$ and camera orientation $\mathbf{rot}(K_j)$ associated with the keyframes K_j.

2. **Material segmentation** (section 3.2): Segment the mesh $M = \{\mathbf{v}_i\}$ into patches $M_u \subseteq M$ of vertices sharing the same material.

3. **Inverse rendering** (section 3.3): Iteratively determine the incident lighting $\mathbf{i}$ as well as diffuse albedo color $\mathbf{a}$ and specular coefficients from $\mathbf{f}$ using the radiance transfer of M.

4. **Specular coefficients** (section 3.4): Determine estimates for the diffuse part $w_d(M_u)$ and the specular part $w_s(M_u)$ of the material for each segment M_u.

3.1 Data association

Dense reconstructions can not only be acquired with multi-camera setups [3], but, increasingly, with dense simultaneous localization and mapping (SLAM). Kinect Fusion [23], a

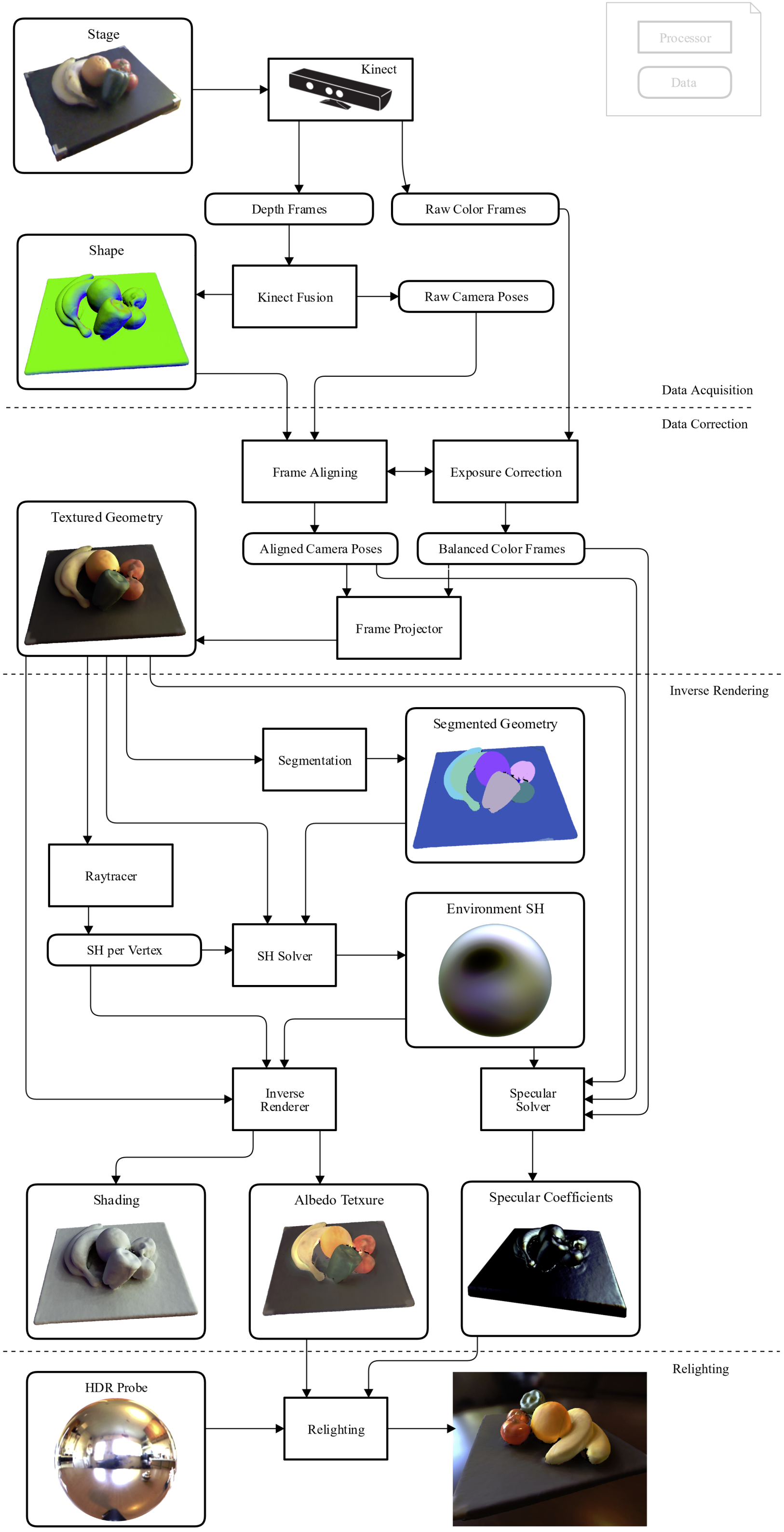
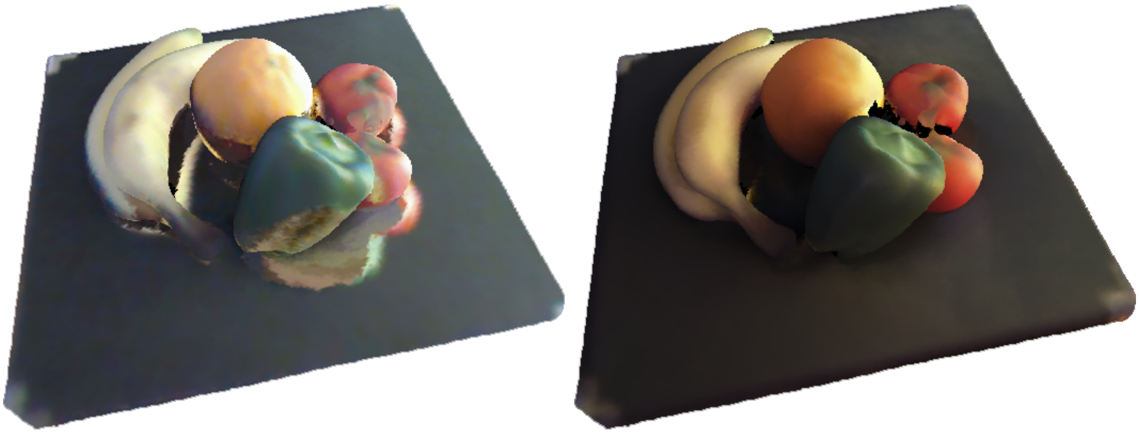


Figure 3.1: Process and Data Stages Overview.



(a) Kinect Fusion

(b) Iterative Vertex Coloring

Figure 3.2: Comparison of per vertex colors obtained from Kinect Fusion and our approach.

popular SLAM variant, turns RGB-D input into a volumetric model, from which a surface mesh is extracted using isosurface raycasting. We rely on the implementation in the Microsoft Kinect SDK [1] to obtain an initial mesh geometry. We regularize the mesh to ensure distances between neighboring vertices are within a tight interval. Excessive detail is removed by iterative edge collapses, followed by local Delauney re-triangulation. Areas with large triangles are refined using Catmull-Clark subdivision. After regularization, normals must be re-estimated.

Since our method depends on the quality of the color information, we do not make use of the color information from Kinect Fusion. Instead, we greedily select a minimal set of keyframes K_j from the input sequence covering the entire scene, such that the camera centers are spaced sufficiently far apart:

$$|\mathbf{pos}(K_a) - \mathbf{pos}(K_b)| \geq c_1 \ \forall a \neq b \tag{3.1}$$

Initially, pixels from all keyframes are projected to the mesh vertices $\mathbf{v}_i$ using the perspective projection defined by the keyframes camera pose and collected in a per-vertex surface light field $\tilde{\mathbf{f}}(\mathbf{v}_i, K_j)$.

Before we can factor the observations into incident lighting and material properties, we must refine the data association. We do this by estimating a lit-color texture $\mathbf{f}(\mathbf{v}_i)$ from all corresponding samples $\tilde{\mathbf{f}}(\mathbf{v}_i, K_j)$. Using a robust estimator ρ, we can compute $\mathbf{f}(\mathbf{v}_i)$ as follows:

$$\mathbf{f}(\mathbf{v}_i) = \arg\min_y \rho(\mathbf{1} \cdot y - \tilde{\mathbf{f}}(\mathbf{v}_i, K_j) \cdot e_j) \tag{3.2}$$

The samples are corrected for different exposure between keyframes with per-keyframe

[1] https://dev.windows.com/en-us/kinect

weights e_j. In the first iteration, we assume $e_j = 1\ \forall j$. Given enough samples, it suffices to use the median or even the mean for ρ.

In the initial estimation of $\mathbf{f}(\mathbf{v}_i)$, we only consider samples for which the angular difference between projection direction and surface normal $\mathbf{n}(\mathbf{v}_i)$ exceeds a threshold:

$$\frac{\mathbf{v}_i - \mathbf{pos}(K_j)}{|\mathbf{v}_i - \mathbf{pos}(K_j)|} \cdot \mathbf{n}(\mathbf{v}_i) > c_2 \tag{3.3}$$

Moreover, vertices close to a geometric edge are initially not considered, because they can cause color bleeding. This problem can be best inspected in the image space of the keyframe K_j from which a sample $\tilde{\mathbf{f}}(\mathbf{v}_i, K_j)$ originates. For a sample from K_j corresponding to a pixel position $\mathbf{x}$, we trace rays through neighboring pixels and determine a set of intersection points M_s on the mesh. If a point in M_s is too far from $\mathbf{v}_i$ or the normals deviate too much, we do not consider $\mathbf{v}_i$:

$$\exists \mathbf{v}_s \in M_s(\mathbf{v}_i) \wedge (|\mathbf{v}_s - \mathbf{v}_i| > c_3 \;\vee\; \mathbf{n}(\mathbf{v}_s) \cdot \mathbf{n}(\mathbf{v}_i) > c_4) \tag{3.4}$$

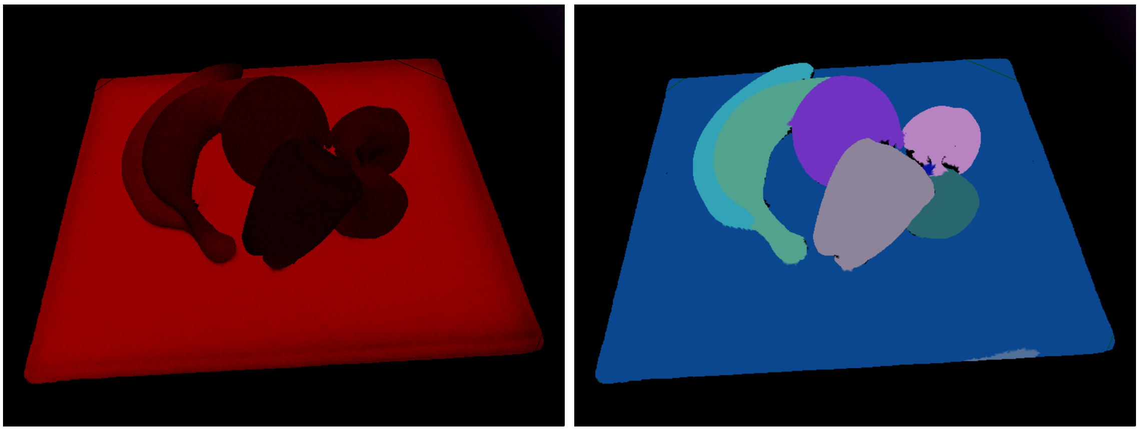
For every pixel $pix(K_j, \mathbf{x})$ at location $\mathbf{x}$ in keyframe K_j, we obtain the corresponding $sample(\mathbf{f}, K_j, \mathbf{x})$ by rendering the mesh and compute the pixel-wise ratio of the pixel and the sample. We use the ratios to robustly estimate an exposure correction factor e_j for keyframe K_j with respect to the brightness represented by $\mathbf{f}$.

$$e_j = \arg\min_y \rho\left(\mathbf{1} \cdot y - \tfrac{pix(K_j,\mathbf{x})}{sample(\mathbf{f},K_j,\mathbf{x})}\right) \tag{3.5}$$

After updating $\mathbf{f}$ using the e_j as weights (Equation 3.2), we correct for small errors in the camera poses. We make small changes to the external camera parameters $\mathbf{pos}(K_j)$ and $\mathbf{rot}(K_j)$, re-render the images, and search for a local minimum along the image gradient using a Lucas-Kanade method.

We repeat the cycle consisting of exposure compensation and pose correction. In later iterations, the restrictions on eligible vertices can be stepwise relaxed to incorporate more data. The iteration terminates, if the overall error is small enough, or no more improvements are achieved.

The result of the data association is demonstrated in Figure 3.2(b). Note that the image reconstructed from the lit-color texture represent averaged colors and include averaged camera-dependent lighting effects, such as specular highlights. We remove these artifacts later, after estimating specular reflection properties (see section 3.4). For comparison, Figure 3.2(a) shows the color reconstruction produced by Kinect Fusion, which heavily suffer from color bleeding, unstable exposures and erroneous camera poses.



(a) SDF Values

(b) Segmented Mesh Parts

Figure 3.3: Mesh segmentation results based on Color, Normal Gradient and SDF data.

3.2 Material-based segmentation

To guide the inverse rendering, we compute a material prior by segmenting the mesh M into disjoint clusters M_u, each consisting of vertices with a similar material. We begin by analyzing the mesh connectivity and isolate unconnected surface components. For every mesh component, we apply a density-based scan (DBSCAN), as proposed by Ester et al. [8]. The algorithm searches for connected clusters of samples that have a certain density with respect to a user-defined distance function *dist*. We used a distance function that is a weighted combination of differences in the lit-color texture $\tilde{\mathbf{f}}$, the negative normal gradient distance $nngd(\mathbf{v}_a, \mathbf{v}_b)$ and the shape diameter function $sdf(\mathbf{v}_i)$.

$$dist(\mathbf{v}_a, \mathbf{v}_b) = w_1 \cdot |\mathbf{f}(\mathbf{v}_a) - \mathbf{f}(\mathbf{v}_b)| + w_2 \cdot nngd(\mathbf{v}_a, \mathbf{v}_b) + w_3 \cdot (sdf(\mathbf{v}_a) - sdf(\mathbf{v}_b)) \quad (3.6)$$

The lit-color texture differences describe material discontinuities. We assume low-frequency distance lighting, which does not produce hard shadows. Therefore, high-frequency color changes can only result from surface texture.

The negative normal gradient distance describes how two nearby surface points are oriented towards one another. If the surface to which the points belongs is planar or convex, we heuristically assume that they belong to the same object. If their arrangement is concave, we assume they may belong to two different objects touching each other. This measure can be computed from the angle between the normal of the first surface point and the vector to the second surface point:

$$nngd(\mathbf{v}_a, \mathbf{v}_b) = \max(0, \cos^{-1}((\mathbf{v}_a - \mathbf{v}_b) \cdot (\mathbf{n}(\mathbf{v}_a)) - \frac{\pi}{2})) \quad (3.7)$$

The shape diameter function models the local thickness of the object underneath a mesh vertex. It has been shown to be a reliable indicator for detecting object parts [29]. We sample it stochastically by shooting rays into the negative half-space underneath a vertex.

After clustering all mesh vertices, we merge small segments, which consist only of a few points, with their best matching neighboring cluster. An example of the resulting segmentation is shown Figure 3.3. Note that each segment combines surface points sharing the same material.

3.3 Inverse rendering

In this section, we introduce an inverse rendering method to factor the lit-color texture into incident lighting and albedo colors.

Let us first consider forward rendering using radiance transfer [31], which describes how light is reflected at a surface point $\mathbf{v}_i$. We write $\mathbf{illum}(w)$ for the light intensity from direction w. For a purely diffuse material and only directional illumination, the radiance transfer $\mathbf{t}$ is essentially a sum over the illumination from all incoming directions, weighted by a Lambertian term and the visibility $visib$, which tells us if the light in a given direction w is occluded or not. We sample the visibility at each vertex by tracing rays into the scene, including a few bounces. The result is scaled with the diffuse albedo color $\mathbf{a}$:

$$\mathbf{t}(\mathbf{v}_i, \mathbf{illum}) = \frac{\mathbf{a}(\mathbf{v}_i)}{\pi} \sum_{w} visib(\mathbf{v}_i, w) \cdot \mathbf{illum}(w) \max(\mathbf{n}(\mathbf{v}_i) \cdot w, 0) \tag{3.8}$$

For the inverse rendering, we express both the radiance transfer and the incident lighting in SH form. The radiance transfer becomes a matrix $\mathbf{T} = [\mathbf{t}_{i,k}]$ (see Figure 5.11(a) for an example) storing the k^{th} SH-coefficient at vertex $\mathbf{v}_i$ in $\mathbf{t}_{i,k}$. The lit color $\mathbf{f}$ can be determined from the dot product of the incident lighting $\mathbf{i}$ in SH form and the radiance transfer.

$$\mathbf{f}(\mathbf{v}_i) = \mathbf{a}(\mathbf{v}_i) \sum_{k} \mathbf{t}_{i,k} \cdot \mathbf{i}_k \tag{3.9}$$

Many real world environments contain area light sources, such as ceiling lights or windows, which can be characterized as distant and low frequency. This has two important implications. First, distant lighting implies that color variations are related to surface texture. If two diffuse surfaces face in the same direction, distant lighting has the same lighting effect on them. Therefore, different lit colors can only result from different diffuse albedo colors. Second, under low-frequency distant lighting and in the absence of occlusions, high-frequency color changes can only result from albedo colors, because the lighting does not produce hard shadows.

Taking these considerations into account, we could compute the incident lighting together with albedo colors using a non-linear solver in just one pass. However, we found that this approach tends to deliver rather unstable estimates, especially on noisy input data. Instead, we solve the resulting linear equation system in the least squares sense [10]:

$$\mathbf{i} = \arg\min_{\mathbf{i}} |\mathbf{f} - \mathbf{a} \cdot \mathbf{1}^{\mathrm{T}} \cdot \mathbf{T} \cdot \mathbf{i}|_2 \tag{3.10}$$

We iterate the linear solver and regularize it using the material-based segmentation. In the first iteration, the diffuse albedo colors **a** are unknown and are initialized to a constant setting (medium-intensity white). Consequently, the solver tries to derive all lit-colors from colored lighting, which leads to an error proportional to the color intensity incorrectly attributed to the lighting and not the albedo color.

The error can be estimated by comparing forward rendering and inverse rendering. We re-evaluate Equation 3.9 with the newly found **i** and subtract the result from **f** to obtain individual per-vertex differences $\Delta\mathbf{f}(\mathbf{v}_i)$. The differences are averaged for all vertices in one material segment M_u, yielding a per-segment error $\Delta\mathbf{f}_u$.

$$\Delta\mathbf{f}_u = \frac{1}{|M_u|} \sum_{\mathbf{v}_i \in M_u} \Delta\mathbf{f}(\mathbf{v}_i) \tag{3.11}$$

The per-segment error is subtracted from all vertices in the segment and added to all albedo colors in the segment:

$$\mathbf{v}_i \in M_u \rightarrow \mathbf{f}(\mathbf{v}_i) = \mathbf{f}(\mathbf{v}_i) - \Delta\mathbf{f}_u,\ \mathbf{a}(\mathbf{v}_i) = \mathbf{a}(\mathbf{v}_i) + \Delta\mathbf{f}_u \tag{3.12}$$

The iteration terminates, if all segment errors falls below a certain threshold. An example of the resulting incident lighting is shown as an environment map in Figure 5.11(b)).

After convergence, we obtain an estimate of the incident illumination **i**, which is free of bias from the albedo, and an average unlit albedo color **a** per material segment.

We can derive the high-frequency parts of the albedo color by dividing the remaining lit-color texture **f** by the corresponding lighting effect. We obtain the lighting effect by forward-rendering the scene once again (Equation 3.9) with **i** and the constant albedo color used in the first iteration. The high-frequency part is added to the average per-segment albedo color to obtain the complete albedo color.

Finally, we improve the radiance transfer by using the reconstructed albedo color. With meaningful albedo colors, radiance transfer can take color bleeding into account.

3.4 Recovering specular coefficients

In order to recover specular material properties, we return to the original keyframes K_j. Our method is able to detect all specular reflections, if a complete light field exist for each point. However, in practice, we usually have only measurements from a fraction of all possible light directions for each point. To compensate for this sparse information, we use an approach similar to Jachnik et al. [14] and compute the specular values per material segment, rather than per vertex.

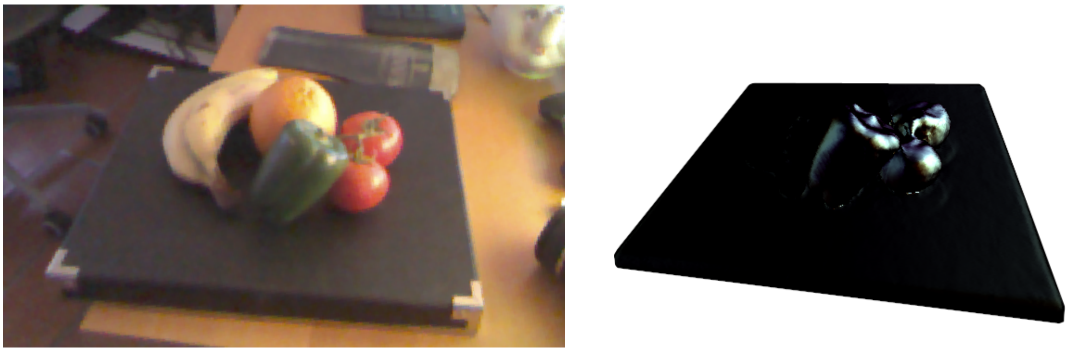


Figure 3.4: Recovered Specularities for a single input frame.

We use a Phong-like material, which assumes that the the observed lit colors $\tilde{\mathbf{f}}$ are the result of a weighted sum of diffuse and specular reflection. At a given vertex, the diffuse lit color is given by $\mathbf{f}$. The purely specular color $\mathbf{f}_s$ can be determined for a given viewpoint $\mathbf{pos}(K_j)$ by sampling $\mathbf{i}$ in the direction of the reflection, unless the reflected ray is blocked by the scene. This leads to the following equation:

$$\tilde{\mathbf{f}}(\mathbf{v}_i, K_j) = w_d \cdot \mathbf{f}(\mathbf{v}_i) + w_s \cdot \mathbf{f}_s(\mathbf{v}_i, K_j) \tag{3.13}$$

For every segment M_u, we build and solve an overdetermined linear equation system in two unknowns $w_d(M_u)$ and $w_s(M_u)$ for all observations of a vertex $\mathbf{v}_i \in M_u$. We empirically observed that w_d corresponds well to subjective roughness of the surface material. More importantly, equation 3.13 can be used directly in a shader. Figure 3.2(d) demonstrates the resulting specularity map of a bowl of fruits.

4 Non-Iterative Reverse Lighting

As a comparison, we tried a non-linear Levenberg-Marquardt solver, to solve for material segment colors and incident lighting directly by just a single iteration. To do that, we reformulated the SH lighting formula.

If we allow only one material $\mathbf{a}'_u$ per segment M_u and write $\mathbf{a}'(\mathbf{v}_i) = \mathbf{a}'_u$ iff $\mathbf{v}_i \in M_u$, we can write an SH lighting equation as follows:

$$\mathbf{f}(\mathbf{v}_i) = (\sum_k \mathbf{i}_k \cdot \mathbf{t}_{i,k}) \cdot \mathbf{a}'(\mathbf{v}_i) \tag{4.1}$$

We can solve for both $\mathbf{i}$ and $\mathbf{a}'$ in a non-linear manner using the Levenberg-Marquart method:

$$\underset{\mathbf{i},\mathbf{a}'}{\text{argmin}}||(\sum_k \mathbf{i}_k \cdot \mathbf{t}_{i,k}) \cdot \mathbf{a}'(\mathbf{v}_i) - \mathbf{f}(\mathbf{v}_i)||_2 \tag{4.2}$$

While generic tests gets solved accurately, noisy real life tests show that the solver is very sensible to errors and tend to be less stable by creating results with high negative and positive spherical function parameters. We use spherical segments, which are more stable, because we can limit the coefficients to positive values but we need to apply a non linear solver that is typically slower.

4.1 Spherical Segments

Using spherical harmonics to represent the lighting has the advantage of very fast calculation with acceptable results. However, they are not suited for solving high-frequency lighting, because the minimum size of a light region is bound to the number of bands. When using a high number of bands, spherical harmonics tend to create unpleasant artifacts like the Gibbs ringing.

If high-frequency lighting is desired, we replace the SH basis with a more suitable basis with spherical support. We divide the spherical domain into a set of discrete segments,

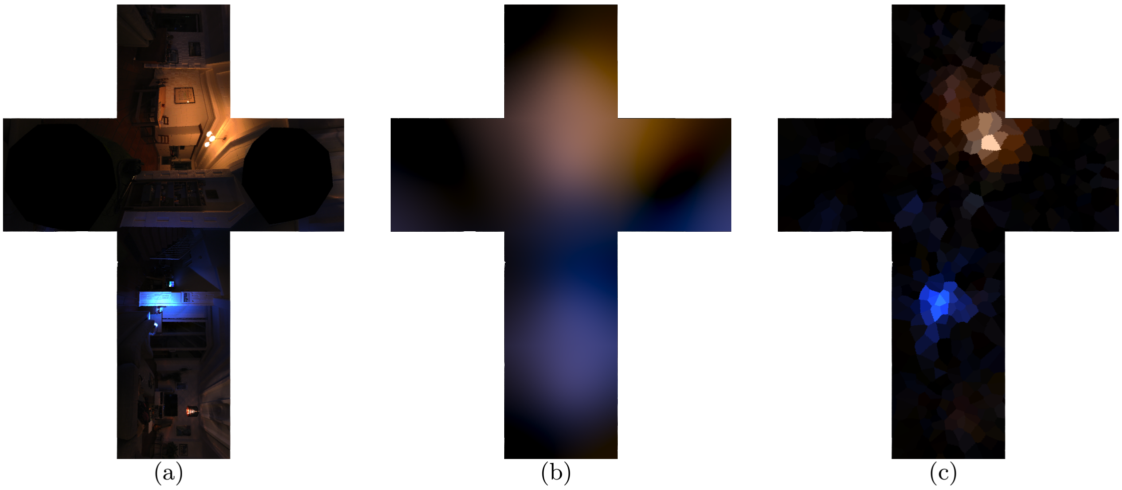


Figure 4.1: Comparison of discrete spherical segments with traditional spherical harmonics. (a) Input lighting captured with a HDR digital camera. (b) Spherical harmonics representation. (c) Spherical segment.

with as many basis function as segments. Each basis function is 1 inside area of support and 0 everywhere else. Solving the lighting with this set of functions is essentially equivalent to defining a light point per segment and solving for the color of this point light.

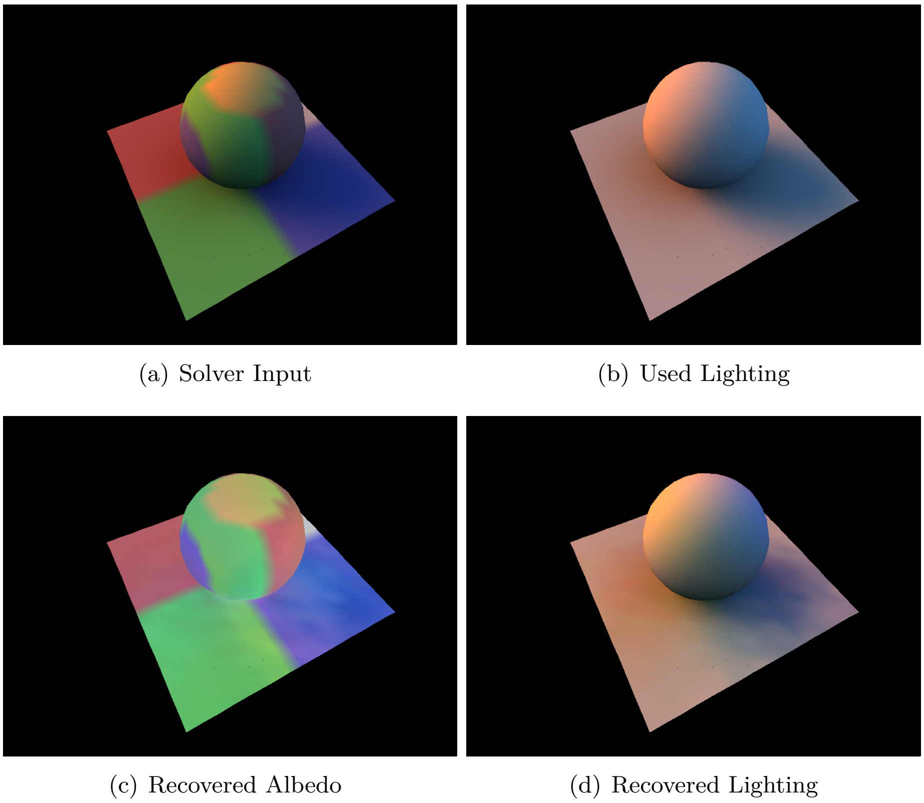
Despite its simplicity, the spherical segment basis is very suitable for inverse rendering. Radiance transfer can directly be built from per-segment visibility information. Unlike SH, we can postulate a positive bound for the solution, because a segment cannot emit negative light. This makes the optimization more stable on noisy input data.

Note that Wang et al. [34] deduce light directions from shadow information with a similar approach, but only with a binary value per segment. On the downside, computation times and memory consumption increase with the number of basis functions, in a way comparable to SH. Because of their robustness, we find spherical segments preferable for a basis function count above 50.

Figure 4.1 shows a comparison of a reverse rendering based on SH and based on a spherical segments. Note how discrete segments are sharper and correlate more with the input lighting.

4.2 Solving texture and lighting without segmentation

We can use the equation defined in equation 4.2 to recover lighting and diffuse albedo color without creating any segments if we have very accurate input data. The solver solves for an individual material per per vertex. In other words, there are as many segments as registration points. In generic tests this produces acceptable results, however it gets less stable with noisy input data. Further research needs to be done to stabilize this solution,



(a) Solver Input

(b) Used Lighting

(c) Recovered Albedo

(d) Recovered Lighting

Figure 4.2: Albedo and lighting reconstruction with non-linear solver without segmentation using discrete spherical light segments.

since it is very promising to deduce lighting and albedo colors from an arbitrary input scene without any pre-processing steps like segmentation.

Figure 4.2 shows the result based on a synthetic image-based lit scene. The solver is able to separate diffuse albedo and lighting accurately from solving colors from 4.2(a) without any preprocessing like segmentation. The quality of the recovered albedo color 4.2(c) and the recovered lighting 4.2(d) is therefore not dependent on the accuracy of a segmentation process. Due to the complexity of the solving process and the high memory demand of the used non-linear solver the number of registration points has been limited, which results in a lower number of vertices and a lower albedo texture resolution.

5

Results and Applications

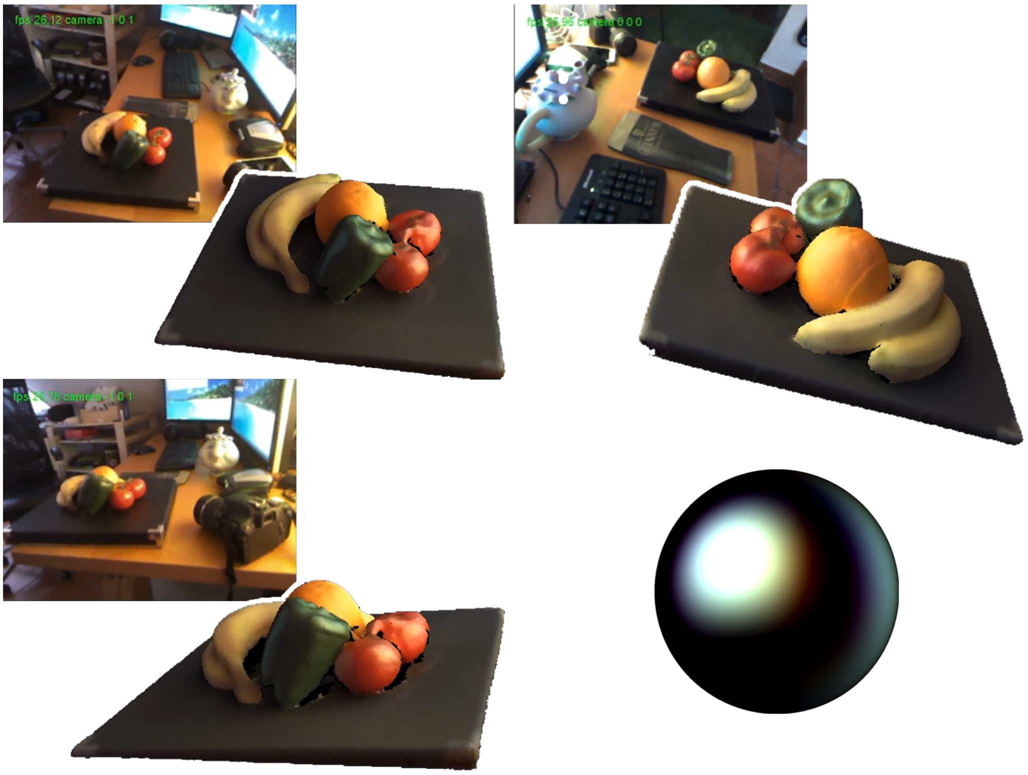


Figure 5.1: Visual comparison between video frame and our rendering which uses our estimations.

Our reconstruction enables many different applications of Mixed Reality. For example, Figure 5.7 shows a visually coherent rendering. The scene consists of a real action figure (The Hulk), which we used to estimate the current lighting. In addition, we placed a virtual characters (Stanford Bunny), which were rendered using our estimates. Note the two windows in the top row, which The Hulk is facing and where most of the incident

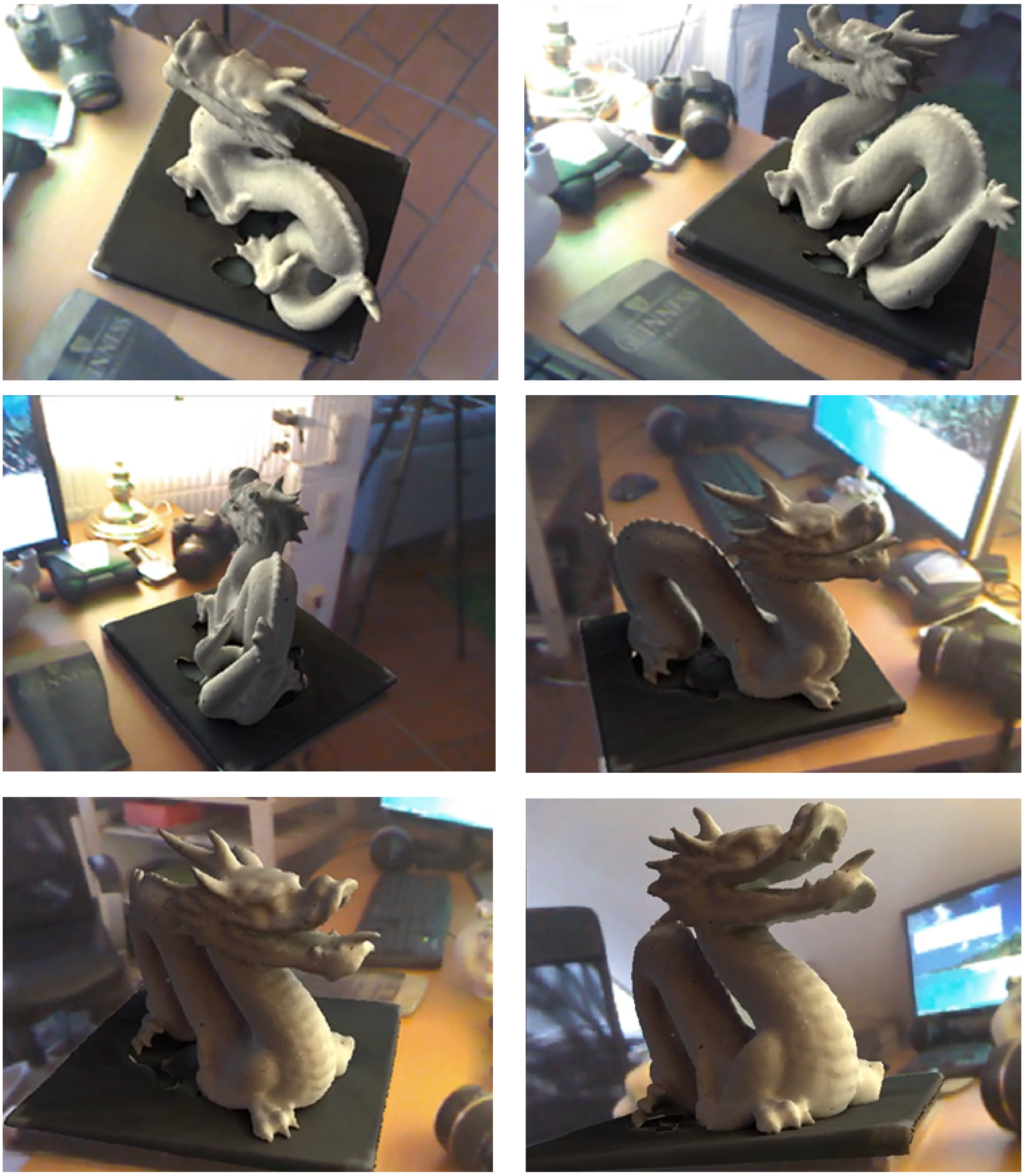


Figure 5.2: Mediated Reality. The fruits have been replaced by a dragon which has been rendered using the light estimation demonstrated before.

lighting is coming from. This has been correctly estimated and used to create the specular reflections on the virtual characters.

Visually coherent rendering of virtual objects in the user's real environment requires a quick estimation, since, otherwise, the lighting may change during processing. Our method scales with the number of vertices in the mesh. The fruit scene consists of an instant geometric reconstruction of medium size received from the Kinect Fusion SDK. Estimating all parameters for this scene took approximately 1-2 minutes on a mid-range notebook computer.

For comparison, we also used a structure-from-motion system, which is able to create a high quality 3D mesh reconstruction within a couple of minutes. We used this system on the Hulk model to create a large 3D mesh consisting of 470K vertices. On this large data set, our method ran for under 1 hour to derive all parameter.

Figure 5.1 demonstrates the quality of our reconstruction by visual comparison. It shows renderings which use our light and reflection estimates, next to the corresponding camera frame. While the rendering and the video frame share the same camera pose, the renderings have been zoomed to improve the presentation. The sphere represents the reconstructed lighting. Note the computer monitors in the front part of the scene, which are the main source of light.

Figure 5.3 depicts a set of generic samples. All samples have been rendered with the same HDR lighting probe. A Monte-Carlo ray-tracer was used, which produces noise that depends on the number of casted rays. Note the noise artifacts in Sample 5.3(b) and in the self-shadow area in sample 5.3(c). While geometry, quality and segment distribution vary greatly, the recovered lighting is very similar and very close to the input lighting. Sample 5.3(a) and 5.3(b) have no ground plane, hence the solving process results in slightly blurred light distribution below the horizon. Some noise artifacts are present in the recovered albedo texture, including lighting artefacts due to different ray-tracer light distribution.

Figure 5.4 depicts a set of real life samples. In order to demonstrate the quality of our method, we have reconstructed several different scenes under different light conditions. In particular, we have scanned a fruit table shown in Figure 5.1 and 5.4(a), an action figure shown in Figure 5.7 and 5.4(b) and a toy car shown in Figure 5.10 and 5.4(c). The action figure is mostly diffusely reflecting, while the toy car and the fruits consist of diffusely and specular reflective materials.

The fruits scene 5.4(a) was reconstructed directly from Kinect Fusion data using depth and color frames. The action figure (Figure 5.4(b)) and the toy car (Figure 5.4(c)) was reconstructed by Structure from Motion with a sparse set of photographs as input. The toy car (Figure 5.4(c)) was not segmented improperly due to noisy input geometry. Note that the ground plane has a small gap. However, all three samples, while significantly different in material distribution, geometry and reconstruction quality, are successfully deconstructed into its lighting and shading parts, with the same segmentation parameters and solving thresholds.

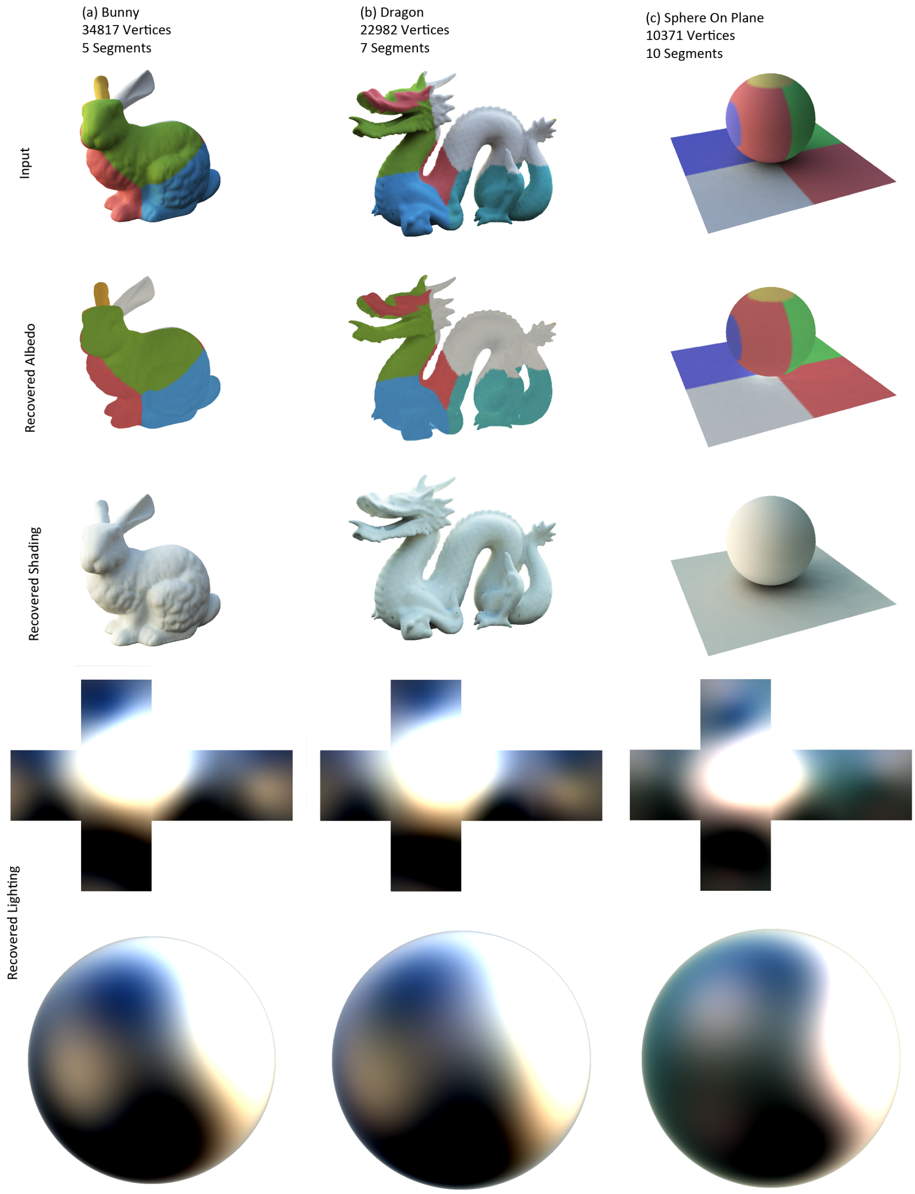


Figure 5.3: Generic samples.

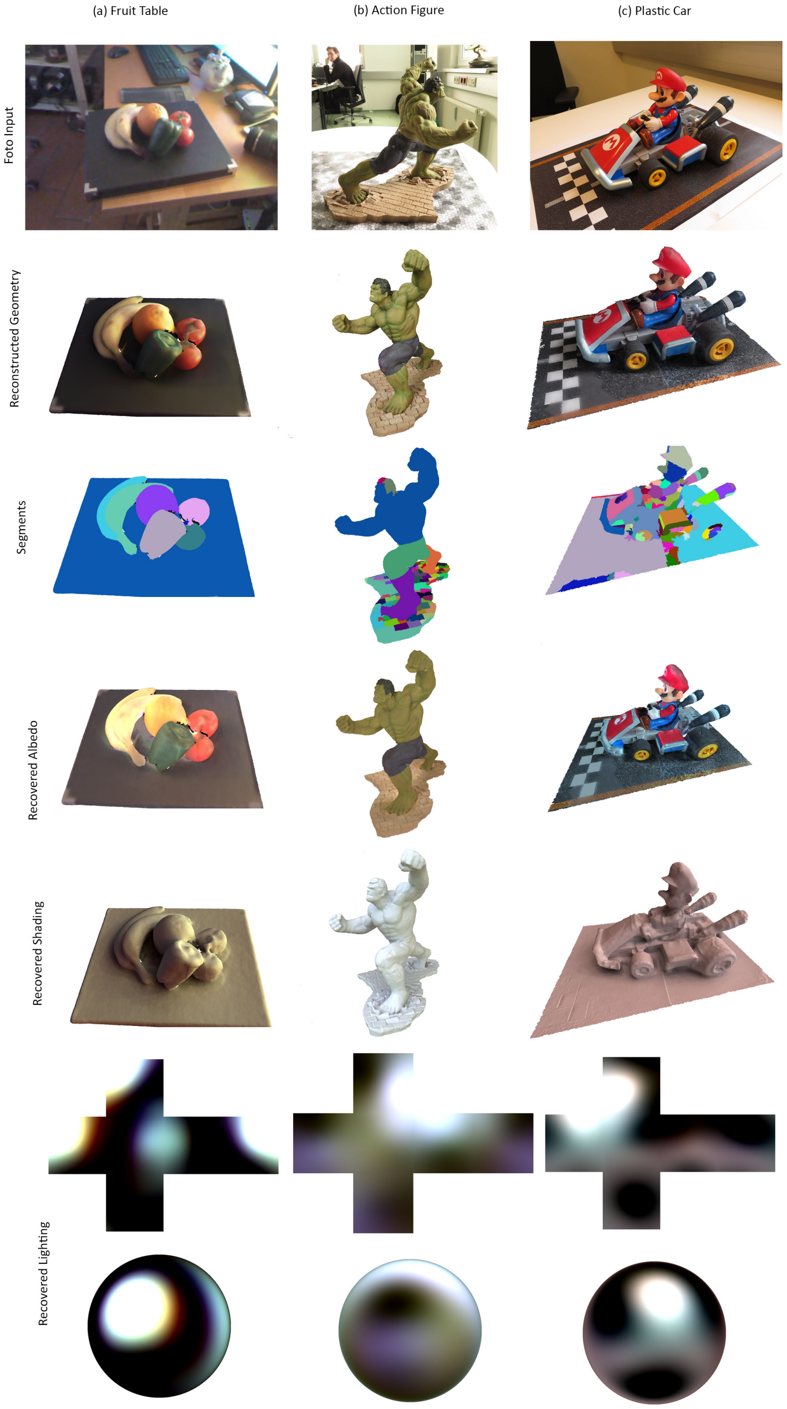


Figure 5.4: Set of real life samples.

Stage	Vertices	SH (Dif)	SH (Shad)	SH (GI)	Segmenting	Solving	Total (GI)
Ball	10371	0.029	0.780	10.669	0.048	6.333	17.050
Dragon	22982	0.099	4.260	76.990	0.113	10.090	87.193
Bunny	34817	0.100	3.140	22.410	0.145	10.970	33.525
Fruits	72119	0.212	7.550	112.603	0.314	15.993	128.910
Hulk	229059	0.656	32.887	470.782	1.109	48.341	520.232

Table 5.1: Processing time (in seconds) for all steps ordered by stage complexity.

Table 5.1 lists the processing time (in seconds) for all steps ordered by stage complexity. Spherical Harmonics are calculated in three different ways. SH(Dif) only takes vertex normals into account (no scene ray-casting necessary), hence only dependents on the number of vertices. SH(Shad) checks for occlusion, hence the performance is depending on the concavity of the scene. SH(GI) calculates a multi-bounce transfer function and is therefore highly dependent on the scene concavity and complexity.

Figure 5.5 shows the results using different sets of light transfer functions applied on a single material mesh. In addition, the error compared to the reference rendering is illustrated using color coding. Figure 5.5(a) shows the reference scene, which has been rendered by a multi-bounce ray-tracer and lit by an HDR environment map. Figure 5.5(b) uses a light transfer function which is only taking local geometry information like normals into account and cannot correctly separate albedo from lighting. The reconstructed albedo color in Figure 5.5(b) contains artifacts from self-shadows and light bleeding, since this information is not part of the transfer function used during reconstruction. Figure 5.5(c) is taking self-shadow information into account. Because the light transfer function does not include light/surface reflection, the error in the albedo reconstruction is high in concave areas, where light bouncing and bleeding have a noticeable impact on lighting. The recovered lighting shows less error than Figure 5.5(b), but still contains strong ringing and blurred highlights. The albedo texture would lead to artifacts when used for relighting. Figure 5.5(d) shows that GI has the least amount of error. Note that the reference was conventionally ray-traced without lossy SH compression; we cannot expect to match it exactly. We expect that using more SH bands would further reduce the error.

Figure 5.6 shows the results of solving a single material mesh (Figure 5.6(a)) and a multi-material mesh (Figure 5.6(b)) without segmentation and compares it to the results of segmented solving (Figure 5.6(c)) to illustrate the importance of segmentation. Unsegmented solving explains albedo colors by varying the lighting results. The recovered shading in Figure 5.6(b) contains green from top, red from left and blue from right as present in the input mesh. The recovered lighting contains similar artifacts. Segmented solving is able to separate the lighting from albedo very closely to the unsegmented solver applied to a single material mesh. The recovered shading in Figure 5.6(c) shows a slight offset to green, since this material is dominant. The recovered lighting gets white-balanced, resulting in a reduced bias for the recovered SH lighting.

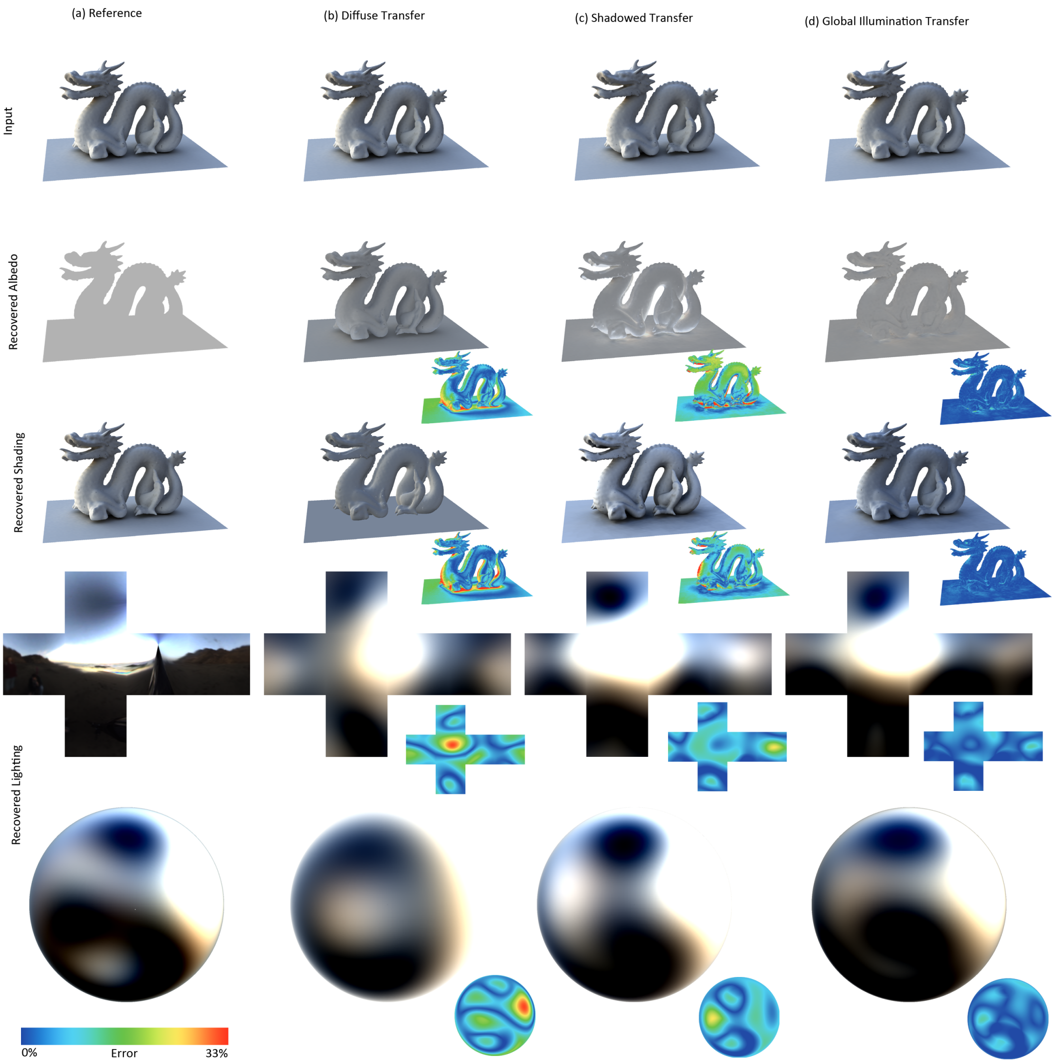


Figure 5.5: Quality comparison based on the type of light transfer functions used for light reconstruction.

Figure 5.7 shows an example for Differential Rendering. A generic mesh (Standford Bunny) assigned with recovered material properties of one segment of the input geometry is inserted into the scene. The image on the left shows the original frame with known camera pose. The image in the middle shows generic geometry, rendered together with the recovered geometry and materials to support light inter-reflection. The image on the right shows the generic mesh with ground shadow inserted into the original photograph. Recovered geometry has been removed but is available for depth culling.

Since our method supports instant light estimation without any special laboratory setup, it can also be applied to enable coherent lighting for Diminished or Mediated Reality applications. Figure 5.2 shows an example in which we replace the fruits shown before

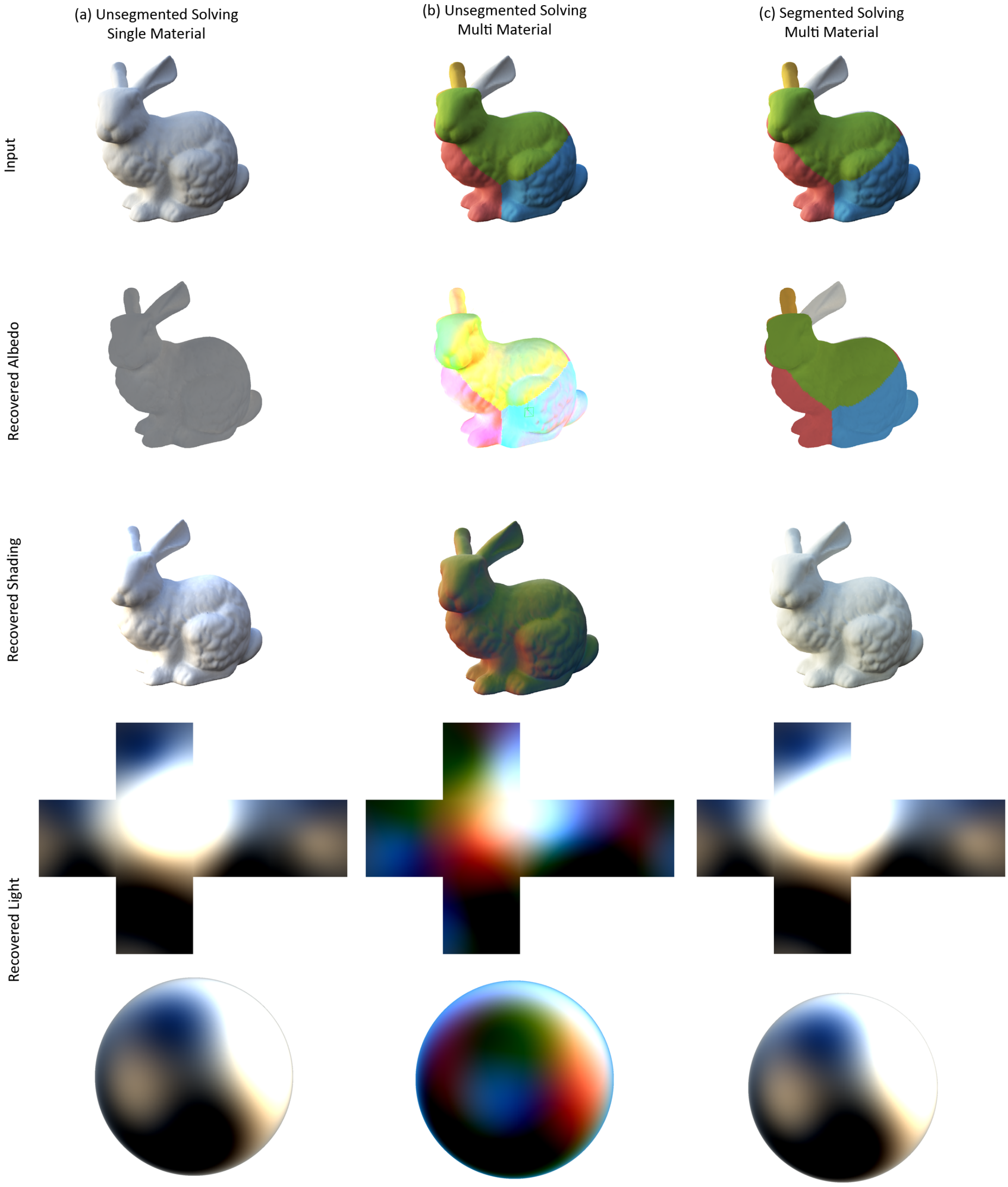


Figure 5.6: Comparing reconstructions based on segmentation..

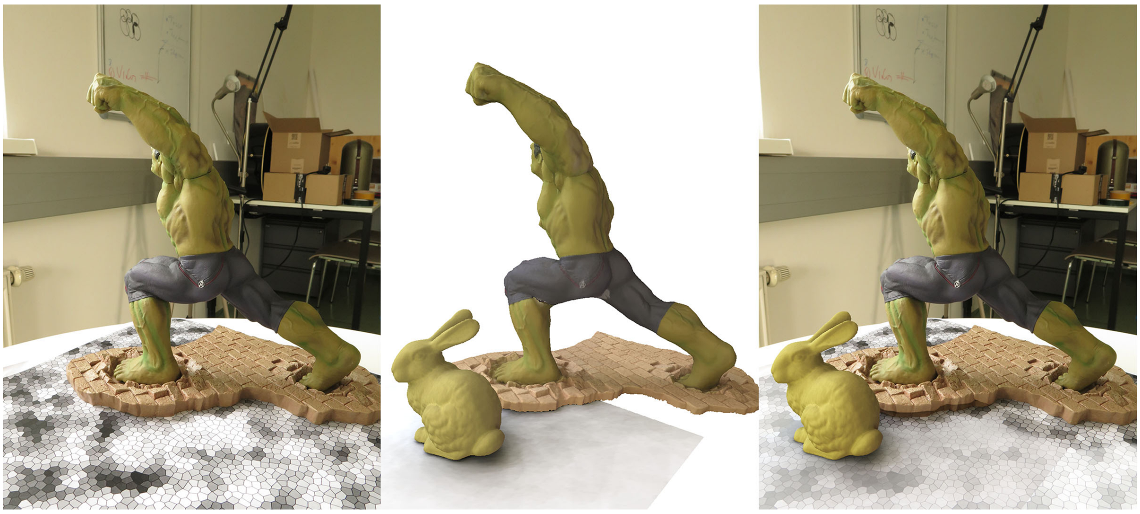


Figure 5.7: Differential Rendering.

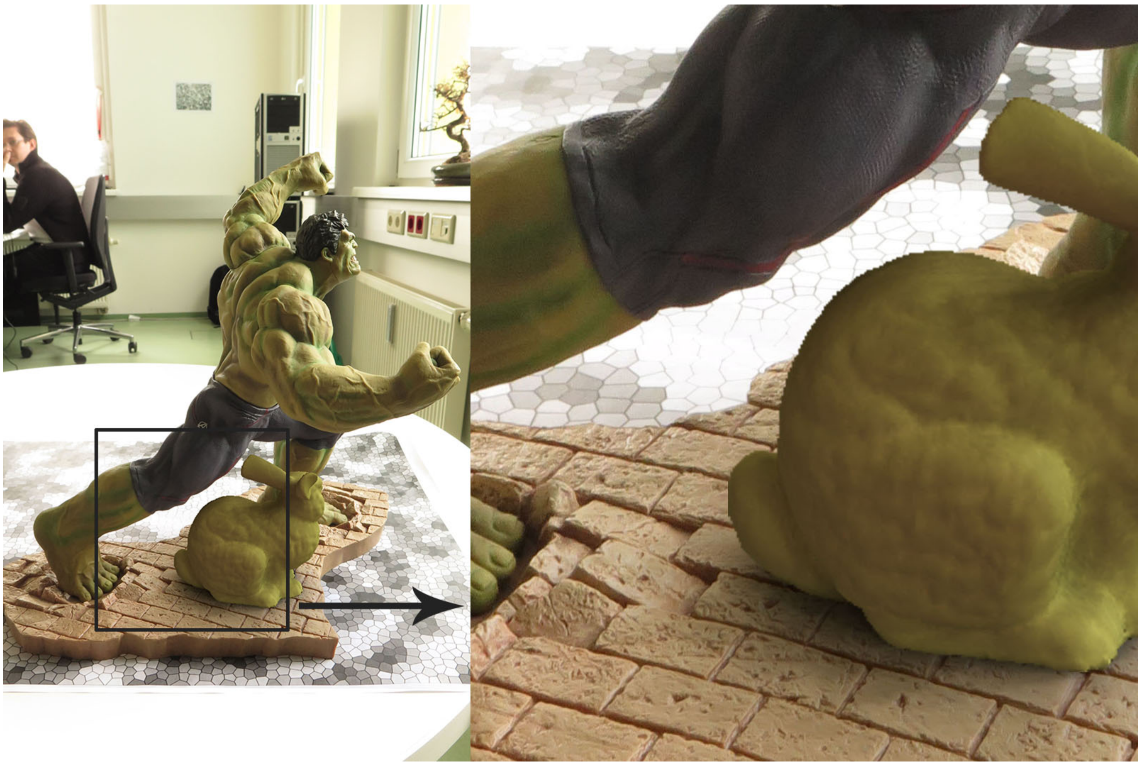


Figure 5.8: Differential Rendering. Image shows strong self shadow effects from recovered geometry to virtual geometry and vice versa.

with a virtual dragon. We use the same scene as in Figure 5.1, so the virtual dragon is mainly lit by the computer monitors in front of the removed fruits.

Furthermore, our reconstruction of material parameters allows to relight 3D scans of real objects. Thus, we can instantly add scanned real objects into virtual environments. For example, Figure 5.10, left, shows our reconstruction of a toy car, which we subsequently placed into a 3D video downloaded from a video website (Figure 5.10, right). In order to coherently render the toy car within the 3D video, we estimate the lighting in every video frame and apply it to the car.

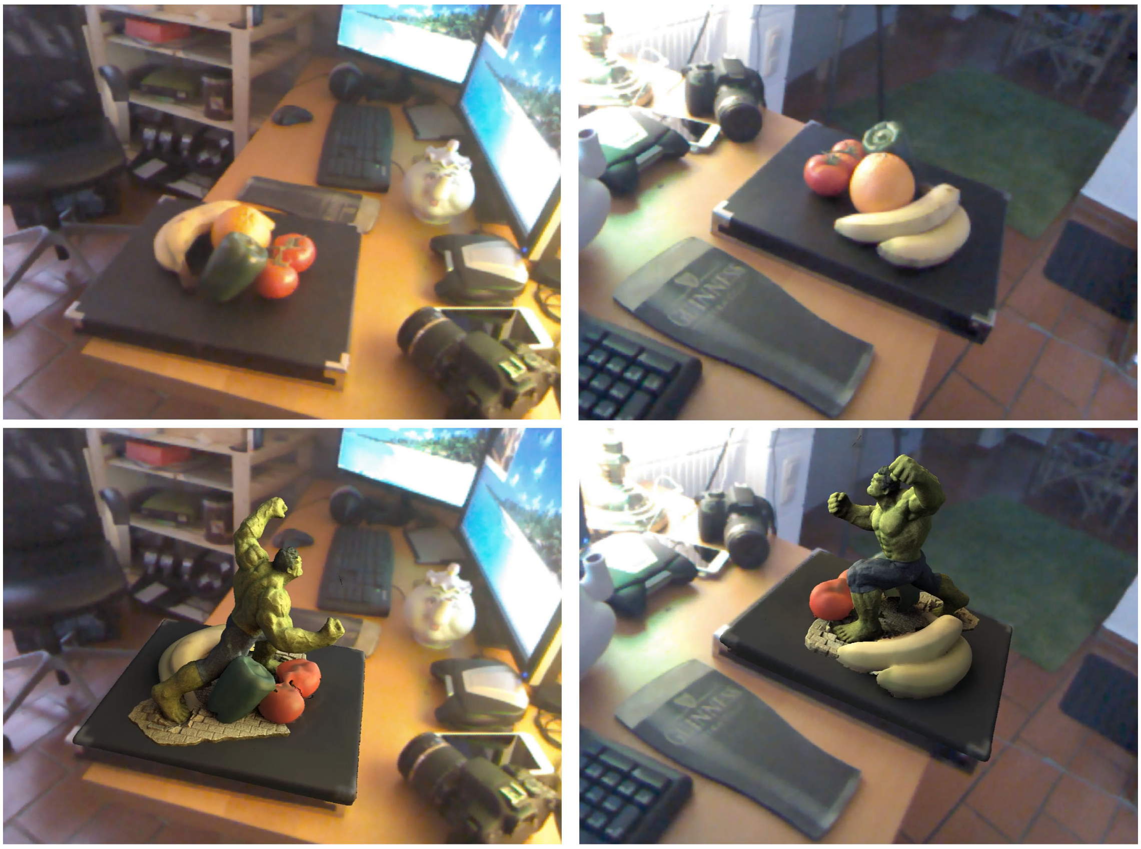


Figure 5.9: Interchanged Augmented Virtuality. Geometry reconstructed from a real scene shown in Figure 5.8 gets inserted into another real scene matching the lighting. To support inter-reflections like self-shadow the target geometry is rendered in addition.

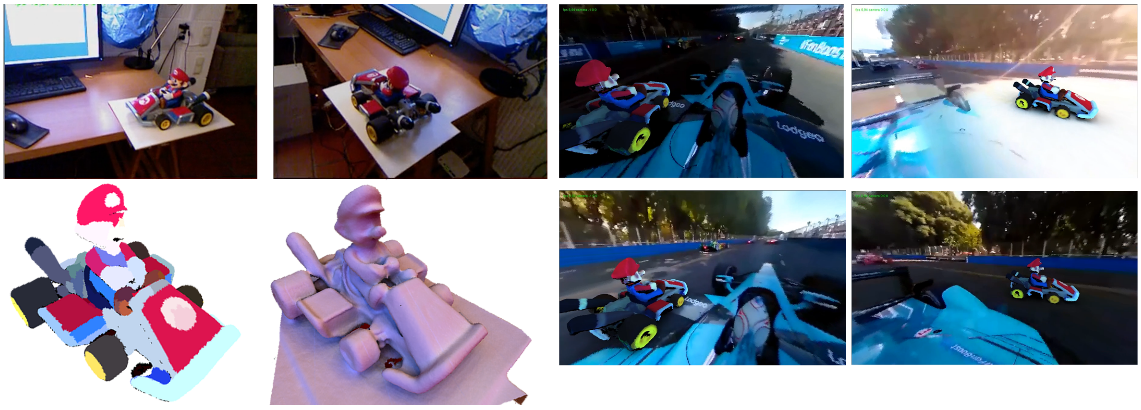


Figure 5.10: Augmented Virtuality. (left) Input data and estimated albedo color and lighting. (right) We place the scan inside a 360 video and relight the geometry using our material estimated and an estimation of lighting in each video frame video.

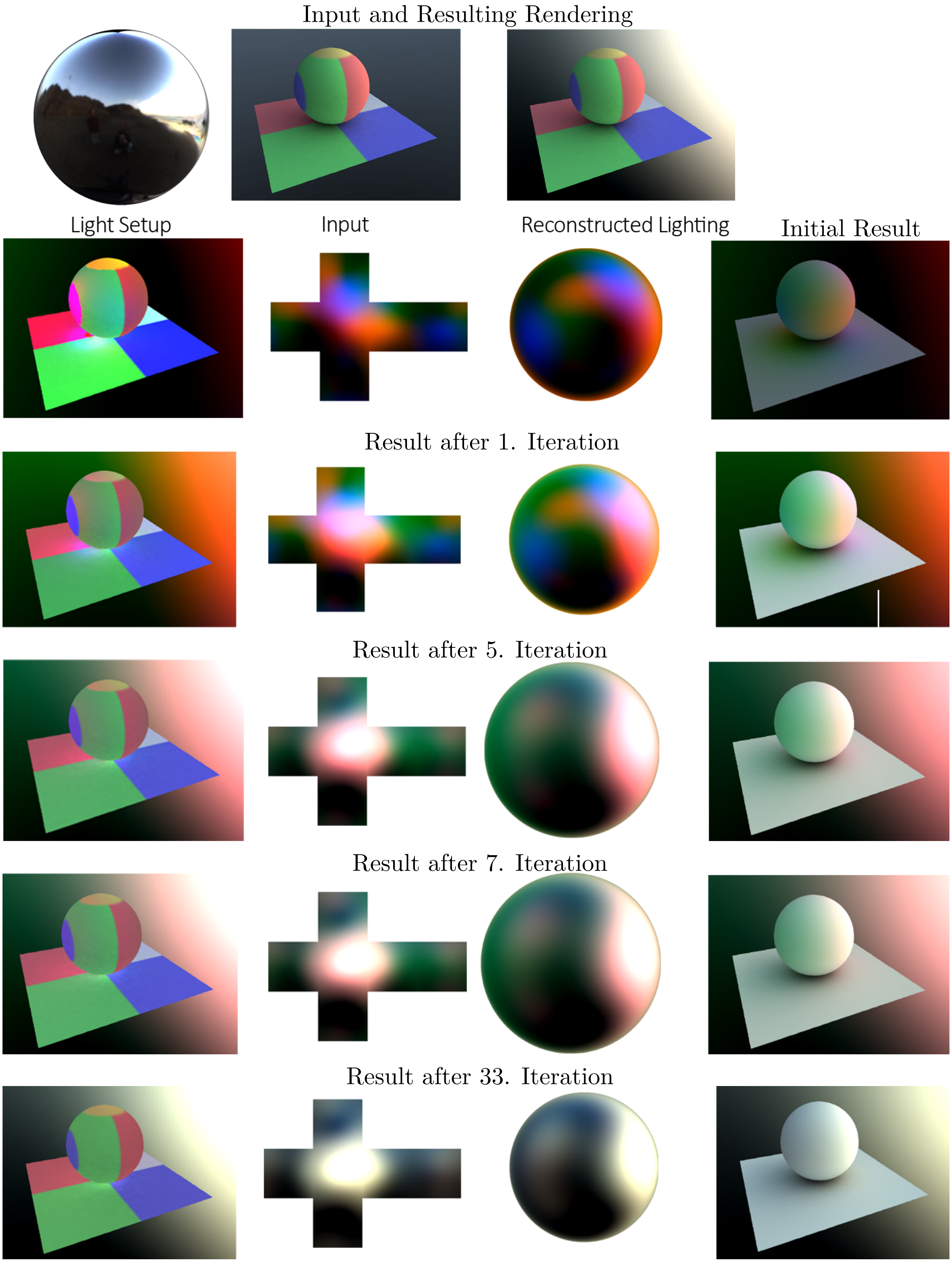

Input and Resulting Rendering
Light Setup
Input
Reconstructed Lighting
Initial Result
Result after 1. Iteration
Result after 5. Iteration
Result after 7. Iteration
Result after 33. Iteration

Figure 5.11: Reconstructing Light. (left) Recovered Albedo Color (middle) Recovered Lighting (right) Geometry lit with recovered lighting.

6 Conclusion and future work

We have presented a method for obtaining estimations for illumination and material using input from casual scanning and demonstrated that, even with the severe restrictions of casual scanning, it is possible to deliver high-quality results. Moreover, we have demonstrated the versatility of our approach with results across the whole spectrum of photo-realistic MR applications, spanning from Augmented Reality (inserting virtual objects into real scenes) to Augmented Virtuality (inserting scanned objects into virtual scenes).

While our method enables many applications, it suggests several directions for future work based on the assumptions we have made. For example, we assume directional lighting with only low- to mid-range frequencies and piecewise constant specularity, which limits specular effects. Furthermore, since we assume a static scene, dynamic elements, such as moving or deformable objects require a dynamic geometric reconstruction as well as information about the light transfer for each point in time. Furthermore, we plan improved color management using LAB instead of RGB color space as proposed by Sheng et al. [30].

A List of Publications

My work at the Institute for Computer Graphics and Vision led to the following peer-reviewed publications. For the sake of completeness of this Thesis, they are listed in chronological order along with the respective abstracts.

A.1 2012

Real-time Photometric Registration from Arbitrary Geometry

Lukas Gruber, Thomas Richter-Trummer, Dieter Schmalstieg
In: *International Symposium on Mixed and Augmented Reality (ISMAR), 5-8 Nov. 2012*

Abstract: Visually coherent rendering for augmented reality is concerned with seamlessly blending the virtual world and the real world in real-time. One challenge in achieving this is the correct handling of lighting. We are interested in applying real-world light to virtual objects, and compute the interaction of light between virtual and real. This implies the measurement of the real-world lighting, also known as photometric registration. So far, photometric registration has mainly been done through capturing images with artificial light probes, such as mirror balls or planar markers, or by using high dynamic range cameras with fish-eye lenses. In this paper, we present a novel non-invasive system, using arbitrary scene geometry as a light probe for photometric registration, and a general AR rendering pipeline supporting real-time global illumination techniques. Based on state of the art real-time geometric reconstruction, we show how to robustly extract data for photometric registration to compute a realistic representation of the real-world diffuse lighting. Our approach estimates the light from observations of the reconstructed model and is based on spherical harmonics, enabling plausible illumination such as soft shadows, in a mixed virtual-real rendering pipeline.

A.2 2016

Instant Mixed Reality Lighting from Casual Scanning

Thomas Richter-Trummer, Denis Kalkofen, Jinwoo Park, Dieter Schmalstieg
In: *International Symposium on Mixed and Augmented Reality (ISMAR), 19-23 Sep. 2016*

Abstract: We present a method for recovering both incident lighting and surface materials from casually scanned geometry. By casual, we mean a rapid and potentially noisy scanning procedure of unmodified and uninstrumented scenes with a commodity RGB-D sensor. In other words, unlike reconstruction procedures which require careful preparations in a laboratory environment, our method works with input that can be obtained by consumer users. To ensure a robust procedure, we segment the reconstructed geometry into surfaces with homogeneous material properties and compute the radiance transfer on these segments. With this input, we solve the inverse rendering problem of factorization into lighting and material properties using an iterative optimization in spherical harmonics form. This allows us to account for self-shadowing and recover specular properties. The resulting data can be used to generate a wide range of mixed reality applications, including the rendering of synthetic objects with matching lighting into a given scene, but also re-rendering the scene (or a part of it) with new lighting. We show the robustness of our approach with real and synthetic examples under a variety of lighting conditions and compare them with ground truth data.

Bibliography

[1] Boom, B., Ning, X., McDonagh, S., Sandilands, P., Fisher, R., and Orts-Escolano, S. (2013). Point light source estimation based on scenes recorded by a rgb-d camera. In *Proceedings of the British Machine Vision Conference*. BMVA Press. (page 4)

[2] Chen, Q. and Koltun, V. (2013). A simple model for intrinsic image decomposition with depth cues. *International Conference on Computer Vision (ICCV)*. (page 5)

[3] Collet, A., Chuang, M., Sweeney, P., Gillett, D., Evseev, D., Calabrese, D., Hoppe, H., Kirk, A., and Sullivan, S. (2015). High-quality streamable free-viewpoint video. *ACM Trans. Graph.*, 34(4):69:1–69:13. (page 7)

[4] Debevec, P. (1998). Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 189–198, New York, NY, USA. ACM. (page 3)

[5] Debevec, P. (2003). Light probe image gallery. *SIGGRAPH Course*. (page 3)

[6] Debevec, P., Hawkins, T., Tchou, C., Duiker, H.-P., Sarokin, W., and Sagar, M. (2000). Acquiring the reflectance field of a human face. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 145–156. (page 6)

[7] Dong, Y., Chen, G., Peers, P., Zhang, J., and Tong, X. (2014). Appearance-from-motion: recovering spatially varying surface reflectance under unknown lighting. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH Asia*, 33(193). (page 6)

[8] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. (page 11)

[9] Grosch, T., Eble, T., and Mueller, S. (2007). Consistent interactive augmentation of live camera images with correct near-field illumination. In *Proceedings of the 2007 ACM Symposium on Virtual Reality Software and Technology*, VRST '07, pages 125–132, New York, NY, USA. ACM. (page 3)

[10] Gruber, L., Richter-Trummer, T., and Schmalstieg, D. (2012). Real-time photometric registration from arbitrary geometry. In *Proceedings of the 2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, ISMAR '12, pages 119–128, Washington, DC, USA. IEEE Computer Society. (page 4, 6, 13)

[11] Gruber, L., Ventura, J., and Schmalstieg, D. (2015). Image-space illumination for augmented reality in dynamic environments. *IEEE Virtual Reality*. (page 4)

[12] Ikeda, T., Oyamada, Y., Sugimoto, M., and Saito, H. (2012). Illumination estimation from shadow and incomplete object shape captured by an rgb-d camera. In *ICPR12*, pages 165–169. (page 5)

[13] Jachnik, J., Newcombe, R. A., and Davison, A. J. (2012). Real-time surface light-field capture for augmentation of planar specular surfaces. In *Proceedings of the 2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, ISMAR '12, pages 91–97, Washington, DC, USA. IEEE Computer Society. (page 4)

[14] Jan Jachnik, Richard A. Newcombe, A. J. D. (2012). Real-time surface light-field capture for augmentation of planar specular surfaces. *Proc. IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. (page 13)

[15] Knecht, M., Traxler, C., Mattausch, O., Purgathofer, W., and Wimmer, M. (2010). Differential instant radiosity for mixed reality. In *ISMAR*, pages 99–107. IEEE Computer Society. (page 3)

[16] Knorr, S. B. and Kurz, D. (2014). Real-time illumination estimation from faces for coherent rendering. In *IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2014, Munich, Germany, September 10-12, 2014*, pages 349–350. (page 5)

[17] Kronander, J., Banterle, F., Gardner, A., Miandji, E., and Unger, J. (2015). Photo-realistic Rendering of Mixed Reality Scenes. *Computer Graphics Forum*. (page 3)

[18] Lagger, P. and Fua, P. (2006). Using specularities to recover multiple light sources in the presence of texture. In *ICPR (1)*, pages 587–590. IEEE Computer Society. (page 4)

[19] Lee, K. J., Zhao, Q., Tong, X., Gong, M., Izadi, S., Lee, S. U., Tan, P., and Lin, S. (2012). Estimation of intrinsic image sequences from image+depth video. *Computer Vision ECCV 2012 - 12th European Conference on Computer Vision*. (page 5)

[20] Li, G., Wu, C., Stoll, C., Liu, Y., Varanasi, K., Dai, Q., and Theobalt, C. (2013). Capturing relightable human performances under general uncontrolled illumination. *Eurographics 2013*, 32(2). (page 6)

[21] Mashita, T., Yasuhara, H., Plopski, A., Kiyokawa, K., and Takemura, H. (2013). In-situ lighting and reflectance estimations for indoor AR systems. In *IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2013, Adelaide, Australia, October 1-4, 2013*, pages 275–276. (page 4)

[22] Maxime Meilland, Christian Barat, A. C. (2013). 3d high dynamic range dense visual slam and its application to real-time object re-lighting. *Proc. IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. (page 3)

[23] Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohli, P., Shotton, J., Hodges, S., and Fitzgibbon, A. (2011). Kinectfusion: Real-time dense surface mapping and tracking. In *IEEE ISMAR*. IEEE. (page 7)

[24] P. Supan, I. S. and Haller, M. (2006). Image based shadowing in real-time augmented reality. *IJVR, 3*, 5(3):1–7. (page 3)

[25] R. Ramamoorthi, P. H. (2001). An efficient representation for irradiance environment maps. *SIGGRAPH*. (page 6)

[26] Ramamoorthi, R. and Hanrahan, P. (2001). A signal-processing framework for inverse rendering. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 117–128, New York, NY, USA. ACM. (page 4, 6)

[27] Richardt, C., Stoll, C., Dodgson, N. A., Seidel, H.-P., and Theobalt, C. (2012). Coherent spatiotemporal filtering, upsampling and rendering of RGBZ videos. *Computer Graphics Forum (Proceedings of Eurographics)*, 31(2). (page 6)

[28] Sato, Y. (1997). Object shape and reflectance modeling from observation. *SIGGRAPH*. (page 6)

[29] Shapira, L., Shamir, A., and Cohen-Or, D. (2008). Consistent mesh partitioning and skeletonisation using the shape diameter function. *Vis. Comput.*, 24(4):249–259. (page 12)

[30] Sheng, Y., Cutler, B., Chen, C., and Nasman, J. (2011). Perception: Perceptual global illumination cancellation in complex projection environments. *Computer Graphics Forum*, 30(4):1261–1268. (page 31)

[31] Sloan, P.-P., Kautz, J., and Snyder, J. (2002). Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '02, pages 527–536. (page 12)

[32] Tappen, M. (2005). Recovering intrinsic images from a single image. *Pattern Analysis and Machine Intelligence*, 27. (page 5)

[33] Wang, Y. and Samaras, D. (2003a). Estimation of multiple directional light sources for synthesis of augmented reality images. *Graph. Models*, 65(4):185–205. (page 5)

[34] Wang, Y. and Samaras, D. (2003b). Estimation of multiple directional light sources for synthesis of augmented reality images. *Graphical Models*, pages 185–205. (page 16)

[35] Weber, M., Blake, A., and Cipolla, R. (2004). Towards a complete dense geometric and photometric reconstruction under varying pose and illumination. *Image and Vision Computing*, pages 787–793. (page 6)

[36] Yu, Y., Debevec, P., Malik, J., and Hawkins, T. (1999). Inverse global illumination: Recovering reflectance models of real scenes from photographs. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 215–224, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co. (page 6)

[37] Zhao, Q. (2012). A closed-form solution to retinex with nonlocal texture constraints. *Pattern Analysis and Machine Intelligence*, 34. (page 5)