BRDF approximation and estimation for Augmented Reality

Patrick Kühtreiber

Martin Knecht
Institute of Computer Graphics and Algorithms
Vienna University of Technology

Christoph Traxler
Institute of Computer Graphics and Algorithms
Vienna University of Technology

Abstract— **In Augmented Reality applications it is important to have a good description of the surfaces of real objects if a consistent shading between real and virtual object is required. If such a description of a surface is not available it has to be estimated or approximated.**
In this paper several methods are presented that deal with the bi-directional reflectance distribution function (BRDF) approximation in Augmented Reality. Of course an important thing to discuss is whether the applications we present work in real-time and compute real looking results.
Different methods can be used to achieve these goals. All of the methods presented work via image based lighting. Some require a 3D polygonal mesh representation of the object, for which the BRDF shall be approximated. Some methods estimate the BRDF parameters via error values and provide results at each iteration.

I. INTRODUCTION

This paper provides a state of the art report about BRDF approximation and estimation in respect to Augmented Reality applications. We looked at several different approaches and emphasized whether the presented methods are applicable in AR environments.
Several papers deal with the problem of BRDF approximation. If BRDFs for certain objects are not known they have to be approximated as closely as possible. In Augmented Reality applications it would be desirable to do this during run-time.
We need a good representation of the reflection behaviours of the surfaces and, what makes it even more difficult, we need them in real-time in order to avoid a pre-modeling step.
The most of the presented methods are rather similar, as all use image based lighting. Still there are differences in performance and photorealism which we will point out at the end of each section. A complete summary and conclusion are given in sections 7 and 8.

The papers presented are:

- Image-Based Rendering of Diffuse, Specular and Glossy Surfaces from a Single Image, [1]
- Recovery of material under complex illumination conditions, [2]
- A Framework for Automatically Recovering Object Shape, Reflectance and Light Sources from Calibrated Images, [3]
- Recovering surface reflectance and multiple light locations and intensities from image data, [4]
- On-line estimation of diffuse materials, [5]

patrick.kuehtreiber@chello.at
martin.knecht@cg.tuwien.ac.at
traxler@cg.tuwien.ac.at

II. Image-Based Rendering of Diffuse, Specular and Glossy Surfaces from a Single Image

This approach, developed by Boivin and Gagalowicz, addresses the problem in an iterative way [1]. Every iteration step adds another level of detail to the rendered scene until the result - rendered with Ward's BRDF reflectance model [6] - looks good enough.
The input for this algorithm is a simple photograph and a geometric model of the scene which includes camera position and light sources that are not exact but approximated. For the Ward reflectance model five parameters are needed: the diffuse reflectance ρ_d, the specular reflectance ρ_s, the anisotropy direction $\vec{x}$ and the anisotropic roughness parameters α_x and α_y [1].

As stated before the algorithm works iteratively and each iteration step is refined several times. Each subsequent state refines the previous one based on an error picture between the (offscreen) rendered picture and the original photograph. The authors used a global illumination method to compute the rendered picture. The algorithm runs through a couple of assumptions where the next step is only computed if the error value from the previous picture is over a certain threshold.

A. *Perfect diffuse*

The error ε is computed as the ratio between the average of the radiances from an object (more exactly its a group of objects) in the real picture and in the synthetic one. The diffuse reflectance ρ_d can be corrected iteratively now with help of the error value. The steady refinement of ρ_d can be seen in Fig. 1.

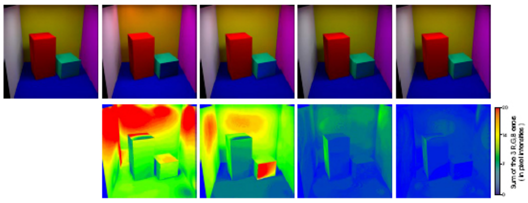

Fig. 1: In the top row are from left to right the refined images based on the error pictures below. Picture from Boivin and Gagalowicz [1].

If the perfectly diffuse (i.e. no highlights in the image) assumption fails the second assumption is tried (perfectly specular = all objects are mirrors). This is easy to accomplish as you simply have to set the diffuse reflectance $\rho_d = 0$ and the specular reflectance $\rho_s = 1$ and replace all the ρ_d with ρ_s.

If there are still objects that have a high error value, they are now considered to have diffuse and specular reflectance properties. To get good approximations to these surface reflectances the diffuse reflectance ρ_d and the specular ρ_s are computed together in a two dimensional system of linear equations.

B. Isotropic surfaces

In this step the surfaces are assumed to have a certain roughness factor α. This roughness factor is considered in Ward's BRDF Model [6]. Now ρ_d, ρ_s and α have to be minimized. Boivin and Gagalowicz did this with the downhill simplex method [7]. As the accuracy does not have to be that high (10^{-2} for ρ_d and ρ_s and 10^{-4} for α respectively), because the visual difference would become imperceptible, the parameters can be found within two minutes [1].

C. Anisotropic surfaces

In this section five parameters of Ward's BRDF model have to be taken into account: The diffuse and specular reflectances ρ_d and ρ_s, the anisotropy direction $\vec{x}$ and the roughness factors α_x and α_y. As the rather independent ρ_d and ρ_s have already been computed, they do not have to be calculated anew.
$\vec{x}$ is determined from the original picture as follows. First the authors consider the anisotropic surfaces as perfect mirrors, compute a synthetic image and estimate the difference between this image and the real one. Next they compute the index buffer for this mirror of all surfaces visible through it, then they look at a surface which has the biggest reflection area and which is closest to the mirror. Then they compute a number of vectors where each one determines a direction to traverse the error image and then they compute the average of the standard error derivation. Finally they select the direction for which the average error becomes smallest [1].

D. Textured surfaces

This is their final assumption. As the textures of the objects in the real image are already illuminated by the light sources they can not simply take those textures. So they introduce the notion *radiosity texture* that balances the extracted texture with an intermediate texture in order to minimize the error between the real and the synthetic image [1].
To increase the speed of this method they propose several alterations of the algorithm. For example, if the error after the specular assumption is higher than 50% the algorithm goes directly to the textured case as the isotropic and anisotropic cases last the longest (almost 4 hours for the anisotropic case).

Fig. 2: Result of the algorithm presented by Boivin and Gagalowicz. These images show the usability of their approach for Augmented Reality applications as there have been made some changes from the original image. The image on the upper left side shows the original image with some removed objects. The other ones show the same scene under novel lighting conditions, new viewpoints or with new objects. Picture from Boivin and Gagalowicz [1].

E. Discussion

The discussions at the end of each section is divided into discussion about running time, input data and illumination model.

1) Running time: The recovery of the image in the anisotropic case took more than four hours. With the enhancements they made the algorithm does not have to try the isotropic and anisotropic cases as they do not promise to deliver better results. So the two most time consuming parts of their algorithm can be skipped if the error after the specular assumption is too high.
The rendering of Fig. 2 took about half an hour. The preprocessing was of course more time consuming, the inverse algorithm took them 4 hours and 40 minutes, where 4 hours alone have been spent to recover the anisotropy parameters for the aluminium surface [1].

2) Input data: The input data for this algorithm was a simple photograph of the scene, where the position of the light sources have to be known and a geometric model of the scene. If both are available then the approach from Boivin and Gagalowicz provides good looking results and is suitable for AR applications.

3) Illumination model: The rendering was done via a global illumination algorithm, because they had to compare the whole scene with the photograph. The BRDF parameters were retrieved for Ward's BRDF model because they took more advanced parameters like roughness into account.

III. Recovery of material under complex illumination conditions

Wu et al. presented a method on how to recover the BRDF for RADIANCE's low parameter reflectance model [8], which also uses Ward's model, for a real homogenous object under complex lighting conditions from a high dynamic range (HDR) photograph of the object and one of the environment to find the light sources [2].
Again their aim is to recover the BRDF ρ, where all other variables are known. The parameters they need for their modified RADIANCE reflectance model are specular, diffuse and directional diffuse reflactance and transmission.
The BRDF model they used can be expressed as

$$f_r = \max(0, \vec{q}\cdot\vec{n}_p)\left[\frac{\rho_d}{\pi}+\rho_s\right]+\max(0,-\vec{q}\cdot\vec{n}_p)\left[\frac{\tau_d}{\pi}+\tau_s\right] \tag{1}$$

where

$$\begin{aligned}
\rho_d &= pC(1-a_4)\\
\rho_s &= r_s\frac{f_s(\vec{q})}{\sqrt{(\vec{q}\cdot\vec{n}_p)(-\vec{v}\cdot\vec{n}_p)}}\\
\tau_d &= a_6(1-r_s)(1-a_7)pC\\
\tau_s &= a_6a_7(1-r_s)\frac{g_s(\vec{q})}{\sqrt{(-\vec{q}\cdot\vec{n}_p)(-\vec{v}\cdot\vec{n}_p)}}\\
r_s &= \begin{cases} a_4 & \text{plastic}\\ \{a_1a_4, a_2a_4, a_3a_4\} & \text{metal}\end{cases}
\end{aligned}$$

where $\vec{q}$ is the direction from the surface point to a light source sample, $\vec{v}$ is the viewing vector, $\vec{n}_p$ is the surface normal at the point p, C is the surface colour, p is the material pattern and $a_i (i = 1,\ldots,7)$ are parameters. The ρ-parameters define the reflection, the τ-parameters the transmission coefficients [2].

After the acquisition of the illumination maps they now proceed to recover the wanted materials for the object which is illuminated by known lighting that is represented as an illumination field. The recovery of the material parameters is done, similarily to the method in the section before, via the minimization of a difference value between the real (I_r) and a synthetic rendered (I_o) image of the object. The difference in the mean of least squares is

$$\chi^2 = \|I_o - I_r\|^2 \tag{2}$$

This is a nonlinear optimization problem that Wu et al. solved with a simulated annealing algorithm. The algorithm works with a set of initial parameter values to optimize them and to reduce χ^2 step by step until a global minimum is found. So we see that they also used a global illumination algorithm to detect the differences between the original photograph and the rendered scene. These calculated parameters are then used to render the object with ray tracing. A result of their work can be seen in Fig. 3.

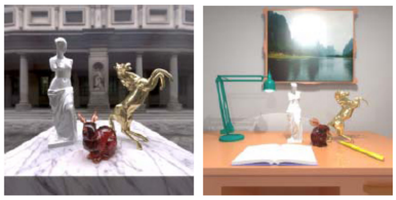

Fig. 3: Left: Virtual objects rendered into a real (Augmented Reality) scene. Right: Virtual objects rendered into a virtual scene. Picture by Wu et al. [2].

A. Discussion

1) Running time: The recovery of the materials took them about 2 to 3 hours on a Dell Dimension 4100 with a 667 MHz CPU and with 128 MB of working memory [2].

2) Input data: The method is similar to the method by Boivin and Gagalowicz [1] but they do not need a geometrical model of the scene which makes the pre-computations more efficient. On the other hand they needed several low dynamic range (LDR) pictures to derive a high dynamic range image and high dynamic range environment maps. Whereas the light sources - other than in the approach described before - do not have to be known, the estimation of the positions of the light sources takes some time.

3) Illumination model: Again the whole scene had to be rendered with a global illumination algorithm in order to be able to compare the images. The object's surface was reconstructed using a model which is similar to Ward's reflectance model.

IV. A Framework for Automatically Recovering Object Shape, Reflectance and Light Sources from Calibrated Images

Mercier et al. [3] present a method for recovering object shapes, reflectance properties (for the modified Phong model [9]) and the position of light sources from a set of images. We will only focus on the surface and reflectance recovery. The modified Phong model is expressed as

$$L_r = \frac{L_s k_d}{\pi r^2}\cos\theta + \frac{(n+2)L_s k_s}{2\pi r^2}\cos\theta\cos^n\phi \tag{3}$$

where L_r is the radiance reflected, L_s is the radiance emitted by a light source S arriving at P, r is the distance between

S and P, θ is the angle between the surface normal and the direction of the light source, $\phi$is the angle between the mirror reflection direction and the actual reflection direction and again k_s and k_d are the specular and diffuse parameters and n is the specular exponent.

For each of the images in the image-set the position and orientation of the camera have to be known.

They made different images for different purposes. They made an overexposed image for segmentation of the surfaces and a second image for reflectance and for the estimation of the light sources.

Their first step is to acquire the object shape from these images using a shape from silhouette approach. They used a marching cubes algorithm that considers image pixels to extract the polygonal surface and the surface normals that are later needed to estimate BRDF parameters.
After they have acquired the polygonal mesh of the object they now proceed to the estimation of the light source directions [3].
During the estimation of the light sources they also present an identification algorithm to provide the BRDF coefficients again using an error function E_a.
They split the estimation of light sources into two classes. The first one applies to diffuse - the second to glossy surfaces (i.e. surfaces that reflect highlights). To find the appropriate class of surface (diffuse or glossy) a variation coefficient V^{class} is computed from the radiance samples. Now for a perfectly diffuse surface the variation coefficient equals zero. V^{class} increases directly proportional to the specular aspect of the surface i.e., it is higher the glossier a surface is. Mercier et al. applied an identification algorithm with the help of a gradient descent method in order to minimize the error E_a and hereby find the BRDF parameters. The parameters are chosen so that E_a becomes as low as possible and the type of surface is known.

A. *Discussion*

1) Running time: They used a Dual Intel Xeon 2.4 GHz processor with 1 GB of working memory. The BRDF estimation of a small object took 6 minutes and 30 seconds precomputation time. The approach by Mercier et al. has certain limitations. For example is it hard to acquire the surface properties if the object possesses a variety of textures [3].

2) Input data: As input they needed an overexposed image for the estimation of the position of the light sources and several images from the scene. In all of these images the camera orientation and position have to be known. From these images they estimate the mesh of the object.

3) Illumination model: Mercier et al. do not apply a global illumination algorithm, therefore their method works faster but only recovers a BRDF of a single object. The BRDF parameters are retrieved for the modified Phong model.

V. Recovering surface reflectance and multiple light locations and intensities from image data

Xu and Wallace presented a method to recover reflectance properties from multiple objects using two intensity images and one depth image. Their approach provides the diffuse and constant specular reflectance parameters from object images [4]. They also recover the light source parameters.

To get the surface geometry they used an active 3D scanner and a stereo pair of CCD cameras. They split their approach into two steps. The first step is to get the light source parameters and the specular reflection for the Phong-Blinn reflection model with the simplified formula for the specular irradiance which also considers the light intensity.

$$I_{spec} = k_s L(P)(H \cdot N)^n \tag{4}$$

where $L(P)$ is the light intensity at point P, N is the normal at that point, n is the specular exponent and H is the halfway vector which calculates as

$$H = \frac{l+V}{|l+V|},$$

where l is the normalized vector pointing to the lightsource and V is the viewing vector [4], [10].

A. *Obtaining the specular reflectance*

As said before they used two cameras. They assume that both cameras have the same radiometric constant. The first step is to calculate the difference between both camera images. This way the part in the formula of the image brightness for the diffuse reflectance disappears as the diffuse reflectance is the same for each viewpoint and only the specular reflectance differs. Now Xu and Wallace compute the difference ε_r between the measured and the predicted values.

To solve this minimization function **f** Xu and Wallace used a gradient descent least-square optimization procedure on **f** using the Levenberg-Marquardt method. To compute the initial values for the specular coefficient k_{s_j} (the j parameter is for multiple light sources) they solved a linear system which is obtained by approximating the measured image brightness difference ΔI_m [4].

B. *Obtaining the diffuse reflectance*

Following the method to gain the specular reflectance Xu and Wallace now use the calculated difference between the two camera pictures to estimate the diffuse parameter for each point. They again calculated the difference between the two camera images and erased the points which are visible to the relevant camera or light.

A result of Xu and Wallace's work can be seen in Fig. 4. The difference between the estimated specular parameters and the ground truth was (at 1% additive noise) 4.76% for the specular coefficient and 1.1% for the specular exponent. This image had only one point light source. The error goes up as the number of light sources increases.

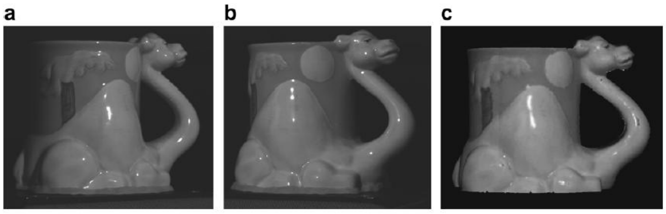

Fig. 4: a) and b) are the two stereo images of the camel cup, c) is the rendered image with the estimated parameters from a new viewpoint [4].

C. Discussion

1) Input data: As input they took several images to estimate the positions of the light sources and hereby detecting the specular and diffuse attributes. They do not need a polygonal mesh of the object as they used a 3D scanner and a pair of CCD cameras to get the surface geometry.

2) Illumination model: Like the method presented by Mercier et al., discussed in section 4, they do not use a global illumination algorithm as their algorithm just detects BRDF parameters for one single object. As illumination model they used the Phong-Blinn reflection model.

VI. Online Estimation of Diffuse Materials

Ritschel and Grosch presented a way to get diffuse parts of BRDFs at run-time from digital photographs [5]. They do not really work in real-time environments but expect their approach to also work in real-time. The only difference really is the parametrization of the model (which has to be done automatically at real-time). They used two HDR video cameras to get the diffuse materials. The equation for the outgoing radiance on a photograph at a surface point is

$$L_0 = \int_{2\pi} f_r(\omega_i, \omega_0) L_i(\omega_i) cos(\theta_i) d\omega_i \tag{5}$$

where f_r is the BRDF of the surface, L_i is the incoming radiance from direction ω_i and $cos(\theta_i)$ is the cosine of the angle between incoming direction and surface normal.

The previously mentioned HDR cameras are positioned as follows. One is observing the object whose BRDF shall be approximated and the second is filming the light source. This *light-camera* is at a fixed position and records the whole environment illumination with a fisheye lens. The *object-camera* should be moved as close to the object as possible so we get the same illumination on the virtual object as there is on the real object.

A marker is placed besides the object to track camera position and orientation. This happens via optical tracking with ARToolkit [11].

The images captured with this camera run through a couple of processes which are out of the scope of this paper. The software side of their procedure is divided into two steps: *Inverse Texturing* (storing the camera images to a texture) and *Inverse Lighting* (processing these textures to one final reflectance texture).

As their approach only works for diffuse lighting, they propose to factorize the software steps into orthogonal components to get the specular part as well. A result of their work can be seen in Fig. 5), where a real object gets duplicated. The BRDF of the duplicated object is approximated so that it matches the different allignment.

Fig. 5: A cloned donkey from Ritschel et al. [5]

A. Discussion

1) Running time: The algorithm presented by Ritschel and Grosch has a performance of 5 fps with a model that has 100 facets and a resolution of 320x240. The most time consuming process was the inverse lighting (70 ms). Of course the performance always depends on the number of pixels and texels.

2) Input data: As input data they used a polygonal mesh of the object whose BRDF shall be approximated but they state that an algorithm which approximates a surface model of a given object on-line would also work.

3) Illumination model: Ritschel and Grosch approximate a BRDF of one single object and give as output BRDF parameters for the equation mentioned earlier (5).

VII. Comparison of the presented methods

A. Running time

If a fast solution is needed, the approach from Boivin and Gagalowicz [1] is the best choice even though a fast approximation provides an image which is not the correct one.

If you change the threshold and the error value accordingly you get a rough approximation after a very short time (if you skip the isotropic and anisotropic assumptions it will always be fast).
The only method which estimates a BRDF of a single object on-line is the one presented by Ritschel and Grosch [5]

B. Input data

Several presented papers used image based methods to estimate the BRDF parameters. Most need an HDR, or couple of LDR pictures of the scene and a 3D model of the scene/of an object. Some papers need a polygonal mesh as input [1]. The other ones include or reference to a parametrization method which is of course also costly.
So if a 3D polygonal mesh is known the Method from Boivin and Gagalowicz [1] can be applied to approximate or estimate the BRDF. Additionally the method from Boivin and Gagalowicz [1] provides very good looking results with an algorithm that is not hard to implement.
If the 3D polygonal mesh is not known one of the more modern approaches should be used as some of them find the shape of the objects just from images [3], [2], [5]. Xu and Wallace [4] used an active 3D scanner and a pair of CCD cameras which does not appear to be a low budget solution.

C. Illumination model

Some papers ([3], [4]) used the Phong illumination model or the Phong-Blinn model respectively. While Phong's model is widely used and simple it is still not physically plausible. Other papers try to make their objects look more realistic and adapt different illumination models. Two papers approximated and estimated parameters for Ward's BRDF model [12], [1], others for RADIANCE [2] or a completely different illumination algorithm [5].
The realism of the papers was generally rather good.

D. Suitable for Augmented Reality

The best solution for Augmented Reality applications was presented by Boivin and Gagalowicz who also presented some ways to deal with their method in Augmented Reality (setting of a novel viewpoint, changing illumination conditions, adding and removing objects). Most of the other papers dealt only with the recovery of BRDF (or shape) of a certain model as they assumed that the geometry is not known.
A summary of the comparison can be seen in table 1.

TABLE I: Comparison of the methods

Method	BRDF	Input data	Real-time
[1]	Ward	Mesh and image	No
[2]	RADIANCE	LDR images	No
[3]	Modified Phong	Images of the scene Camera, light sources	No
[4]	Phong-Blinn	Images of the scene	No
[5]	Own BRDF model	Mesh	Yes

VIII. Conclusion

In our paper we presented 5 papers that deal with BRDF approximation or estimation. All mention the possibility to use their work in Augmented Reality applications which was of course the topic and an important point of our paper. In each section we summarized the parts of the papers that dealt with BRDF approximation or estimation with respect to their usage in Augmented Reality. We presented results of the different papers and concluded each section with a short paragraph about the performance (if given in the original work) and problems that occurred or might occur when using the corresponding algorithm. At the end we compared all the relevant papers in respect to input data, running time, the illumination method used (Phong, Phong-Blinn, Ward,...), photorealism, whether or not a rough approximation of an image before the actual computation of the end result is possible, their usage in Augmented Reality applications and whether the methods are costly or low budget solutions.

References

[1] S. Boivin and A. Gagalowicz. Image-based rendering of diffuse, specular and glossy surfaces from a single image. In *SIGGraph-01*, pages 107–116, 2001.

[2] Enhua Wu, Qimin Sun, and Xuehui Liu. Recovery of material under complex illumination conditions. In Yong Tsui Lee, Stephen N. Spencer, Alan Chalmers, and Seah Hock Soon, editors, *GRAPHITE*, pages 39–45. ACM, 2004.

[3] B. Mercier, D. Meneveaux, and A. Fournier. A framework for automatically recovering object shape, reflectance and light sources from calibrated images. *International Journal of Computer Vision*, 73(1):77–93, June 2007.

[4] S. Xu and A. M. Wallace. Recovering surface reflectance and multiple light locations and intensities from image data. *Pattern Recognition Letters*, 29(11):1639–1647, August 2008.

[5] Tobias Ritschel and Thorsten Grosch. On-line estimation of diffuse materials, April 01 2008.

[6] Gregory J. Ward. Measuring and modeling anisotropic reflection. In Edwin E. Catmull, editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 265–272, July 1992.

[7] J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[8] Gregory J. Ward. The RADIANCE Lighting Simulation and Rendering System. In *Computer Graphics Proceedings, Annual Conference Series, 1994 (ACM SIGGRAPH '94 Proceedings)*, pages 459–472, 1994.

[9] R. R. Lewis. Making shaders more physically plausible. *Computer Graphics Forum*, 13(2):109–120, January 1994.

[10] Donald Hearn and M. Baker. *Computer Graphics with OpenGL*. Prentice-Hall PTR, pub-PHPTR:adr, third edition, 2004.

[11] Shared-Space-ARToolKit. www.hitl.washington.edu/projects/shared space.

[12] Y. Yu, P. E. Debevec, J. Malik, and T. Hawkins. Inverse global illumination: Recovering reflectance models of real scenes from photographs. In *SIGGraph-99*, pages 215–224, 1999.