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

Samuel Boivin* André Gagalowicz*

Mirages Project
INRIA-Rocquencourt

Abstract

In this paper, we present a new method to recover an approximation of the bidirectional reflectance distribution function (BRDF) of the surfaces present in a real scene. This is done from a single photograph and a 3D geometric model of the scene. The result is a full model of the reflectance properties of all surfaces, which can be rendered under novel illumination conditions with, for example, viewpoint modification and the addition of new synthetic objects. Our technique produces a reflectance model using a small number of parameters. These parameters nevertheless approximate the BRDF and allow the recovery of the photometric properties of diffuse, specular, isotropic or anisotropic textured objects. The input data are a geometric model of the scene including the light source positions and the camera properties, and a single image captured using this camera. Our algorithm generates a new synthetic image using classic rendering techniques, and a lambertian hypothesis about the reflectance model of the surfaces. Then, it iteratively compares the original image to the new one, and chooses a more complex reflectance model if the difference between the two images is greater than a user-defined threshold.
We present several synthetic images that are compared to the original ones, and some possible applications in augmented reality.

CR Categories: I.2.10 [Artificial Intelligence]: Vision and Scene Understanding—modeling and recovery of physical attributes; I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Radiosity, Ray Tracing; I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Color, Photometry, Shading;

Keywords: Image-Based Rendering, Reflectance Recovery, BRDF Models, Radiance, Radiosity, Rendering, Inverse Rendering, Rerendering, Global Illumination

*email: {Samuel.Boivin|Andre.Gagalowicz}@inria.fr

ACM SIGGRAPH 2001, 12-17 August 2001, Los Angeles, CA, USA

1 Introduction

1.1 Overview of the problem

Since its origin, Computer Graphics has aimed at depicting reality. Rendering algorithms have been developed specifically to generate near-perfect images under realistic illumination conditions. It is often difficult to say if such images are realistic or not because there is no real reference such as a photograph. Moreover, the application may need to create novel viewpoints and/or novel illumination conditions from a sparse set of photographs. This is difficult to achieve without using image-based modeling and rendering algorithms. For example, suppose we want to insert a new synthetic object on top of a real anisotropic mirror inside a real scene. This operation clearly requires taking into account the interaction between the new object and its environment (especially this mirror). This is impossible to do, if we do not have an approximation of the reflectance properties of the real surfaces in the image. Therefore specific algorithms are necessary to recover these reflectance properties from the real images.

Many authors have contributed to the resolution of this problem [21, 25, 32, 31, 33, 26, 27, 34, 7, 41, 23, 24, 30, 29, 14, 11, 28]. The algorithms that they have produced vary greatly and not all can be re-used for our applications. Considerable work has been done for the reflectance estimation of an isolated object in particular illumination conditions [21, 25, 32, 31, 33, 26, 27] . Although these techniques often bring very detailed reflectance information (i.e. a full BRDF sometimes), their goal is more to replace the use of an expensive gonioreflectometer rather than to be able to change the viewpoint and/or the illumination. Recently, several methods have been developed to take into account the interaction between objects inside a real scene, from a sparse set of photographs [7, 41, 23, 24]. Fournier [14] proposed a different approach but with the use of a single image. However, his technique was limited to perfectly diffuse environments and was not be able to take into account specular surfaces. Our method has the similar ambition to recover an approximation of the BRDF of the surfaces from a single image, including the processing of specular, isotropic or anisotropic surfaces. This is extremely difficult to achieve because it is not possible to compute a full BRDF correctly without having several images, except for trivial cases.

We propose a hierarchical and iterative technique that computes the best possible approximation of a real image, using the error computed between the rerendered image and the real one. Each of the new images is generated by making more and more complex assumptions about the reflectance properties of the real surfaces. It is rendered by a global illumination software that takes into account these reflectance changes (see figure 1). The main advantages of our approach are: it does not need any special device to capture the real image (a classical camera is enough), and it estimates the reflectances of all types of surfaces (including anisotropic mirrors) from a single image without any particular constraint for the view-

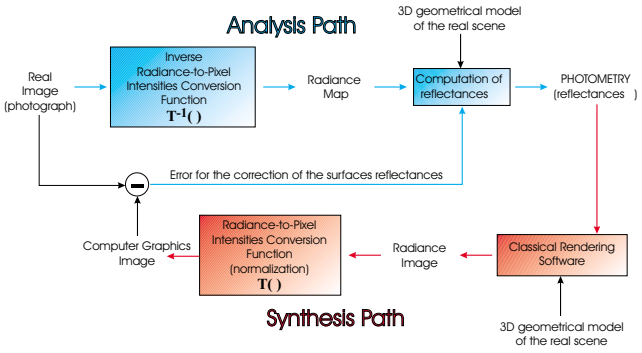

Figure 1: **General Principle of our Method** this figure shows the global scheme of the inverse rendering process. Initial data are: one real image and a 3D geometrical model of the scene.

point position, the light sources[1] or the objects orientation. The goal of our method is to recover an approximation of the BRDF of the surfaces, and to compute the best synthetic image preserving the real properties of the scene (a real mirror has to be simulated as a specular surface and not as a textured surface for example).

1.2 Organization of the paper

The paper is organized as follows. In the next section, we discuss previous work related to image-based rendering. Section 3 describes the bases and the tools necessary to our algorithm. In particular we introduce the notion of group which solves the problem of objects that are not directly seen in the image, and the BRDF model that we use. We also give a short description of the input data and the rendering software that we have developed. Section 4 describes the algorithm in full detail using the previously discussed tools. In particular, we explain the methods to process each case of reflectance property separately. In section 5, we describe one of the advantages inherent in our methodology: the possibility of analyzing some surfaces that are not directly seen in the real image, but indirectly through a mirror. Section 6 completes the technical discussion by explaining the optimizations that we have implemented to accelarate the rerendering process. Section 7 shows several results of rerendering, including images containing many kinds of photometric properties. Some applications are given in the domain of augmented reality, including rendering of new images under novel viewpoints, novel illumination conditions and the insertion/removal of objects. The last section gives some conclusions and future research directions.

2 Background and Related Work

All the techniques and ideas in this paper have been inspired by works about photorealistic rendering including global illumination and ray tracing, image-based modeling and BRDF modeling. However, the most relevant domains deal with *inverse rendering*, *image-based rendering* and *reflectance recovery*. We can split the *reflectance recovery* algorithms into three parts: direct measure of reflectances on the object using a specific device [37, 20, 2, 6], the extraction of reflectances from a set of images [21, 25, 32, 31, 33, 26, 27, 34, 7, 41, 23, 24], and the extraction of reflectances from a single image [30, 29, 14, 11, 28]. The last two parts may be subdivided into two categories, depending on whether the method takes into account energetic interreflections (using a global illumination algorithm for example) or not.

[1] In fact, the emittances of the light sources are supposed to be known. However, if it is not the case Fournier et al. [14] propose a method to recover them automatically.

2.1 Reflectance Recovery using a Specific Device

Ward [37] proposed to directly measure the reflectances of an object, using a low-cost device. Ward introduced a device to estimate the five parameters of his anisotropic BRDF model, that he developed for these purposes. Karner et al. [20] presented another device using the Ward's BRDF model.

Baribeau et al. [2] described a method for measuring three reflectance parameters of several objects inside a scene. The diffuse reflectance, the Fresnel term and the roughness of the objects are estimated using a polychromatic laser range sensor. However, this method is limited to uniform reflectance properties over each object.

Dana et al. [6] suggest using a device containing a robotic manipulator and CCD camera to allow simultaneous measurement of the BTF (Bidirectional Texture Function) and the BRDF of large samples (about 10cm $\times$ 10cm).

2.2 Reflectance Recovery from Several Images

2.2.1 Methods without Global Illumination

Kay et al. [21] described a method to compute the surface reflectances using the Torrance-Sparrow light reflection model [35]. They used a depth map and four or eight images obtained with different point light sources. By increasing the number of intensity images, they estimated the parameters of the Torrance-Sparrow's model, reduced to three terms: the diffuse reflection coefficient k_d, the specular reflection coefficient k_s and the roughness factor c. Lu et al. [25] did not use any reflection model, but directly estimated the reflectances from the pixel intensities. Nineteen black and white images were captured using a custom device that turns around the object. For each incident angle of light, they built a reflection function, depending on the maximum pixel intensity in the image.

Y. Sato et al. [32, 31] proposed to register a range map (to get a 3D geometric model) and a set of color images of an object, using a 360 degrees rotation device. Next, they extracted the pixel intensities from the images and from the 3D model of the object reprojected onto the images by a Z-buffer algorithm. These parameters were used to separate and then compute the diffuse component and the specular component, i.e. the k_d term of the Lambert's model and the k_s and c terms of a simplified Torrance-Sparrow reflection model.

Y. Sato et al. [33] needed 120 color images and 12 range maps to compute the Torrance-Sparrow's parameters, separating the diffuse and the specular component. They recovered the BRDF of highly textured objects (this was impossible to do with previous techniques presented in [21, 25, 32]), and proposed the creation of new images under novel viewpoints and novel illumination conditions.

Marschner et al. [26, 27] directly estimated the Lafortune's et al. BRDF [22] of an object from a set of images (30). To obtain the BRDF, the radiance received by the pixels from the object is divided by the irradiance received by this object from the light source. He applied this computation to the rerendering of objects under novel illumination conditions.

Finally, Wong et al. [34] described a method that recovers the reflectance of each pixel of an image, considered as a set of small facets, each one having its own BRDF. The BRDFs are estimated from a set of images taken under different viewpoint and illumination conditions, as the ratio of the pixel intensity divided by the light source intensity. Wong et al. applied their method to the re-illumination of the scene with new light sources.

2.2.2 Methods with Global Illumination

Debevec [7] used global illumination for augmented reality applications. To insert new objects inside a real image, he needed to take into account interreflections and compute the reflectances of

the surfaces in the part of the scene influenced by this insertion. He created a geometrical 3D model of this part of the scene (called the *local scene*), and calculated manually the reflectance parameters of all the modeled objects. Each of the non-diffuse BRDF parameters are changed by the user iteratively until the rerendered image becomes close enough to the original one. The perfectly diffuse parameters are set by an automatic procedure.

Yu et al. [41] proposed a complete solution for the recovery of the surfaces BRDF from a sparse set of images captured with a camera (twelve of the 150 images were taken specifically to get specular highlights on surfaces). They built 40 radiance maps for the estimation of the reflectance parameters and the computation of the radiance-to-pixel intensities conversion function (camera transfer function) [8]. Using an image-based modeling software such as *Facade* [9], a 3D geometrical model of the scene was built from the set of images. All these data were then utilized to recover the BRDF of the modeled surfaces. Their method minimized the error on the parameters of the Ward's anisotropic BRDF model [37] to estimate the best possible BRDF for each object. This work was applied to the insertion of new objects in the scene, to the modification of the illumination conditions and to the rendering of a new scene under novel viewpoints. However, this method only works if at least one specular highlight is visible on an object. Otherwise this object is simulated as perfectly diffuse.

Loscos et al. [23] proposed a method based on an original idea from Fournier et al. [14]. Their algorithm recovers the diffuse reflectances of the surfaces inside a set of photographs of a scene, taking into account the textures of the objects (each surface has to be unshadowed in at least one image of the set). They applied their technique, to insert/remove objects and to modify the lighting conditions of the original scene (insertion of a new light source for example). More recently, Loscos et al. [24] extended this technique by removing the constraint of the unshadowed surfaces. To improve the results, they transformed their reflectance recovery algorithm into an iterative process. However, the method remained limited to perfectly diffuse surfaces (the mirrors are considered to be diffuse textured objects for example).

2.3 Reflectance Recovery from a Single Image

2.3.1 Methods without Global Illumination

K. Sato et al. [30] described an algorithm for the reflectance recovery of an isolated object from a single image and a 3D geometrical model of this object. They applied some constraints on the light source position and the camera parameters. In addition, they simplified the Torrance-Sparrow reflection model. This way, they estimated separately the diffuse component and the specular component to recover the uniform reflectance of the surface.

More recently, I. Sato et al. [29] proposed to recover the BRDF of an object, using the shadows generated by the surfaces of the scene. They used a single omnidirectional image of the environment and a 3D geometrical description of the surfaces. They developed a 6-step iterative algorithm to minimize the error between the real and the synthetic image with respect to the BRDF parameters of the surfaces.

2.3.2 Methods with Global Illumination

A pioneering work in this domain was completed by Fournier et al. [14] in 1993. He proposed to rerender an original image using a 3D representation of the scene (including the positions of the light source and the camera parameters) and a single image of this scene. All the surfaces are considered as perfectly diffuse, and they used their reprojection on the real image to estimate their reflectances. A radiosity-based algorithm then computes an image applying these reflectances to a progressive radiosity technique [4] to obtain a new synthetic image.

An extension of the previous method was developed by Drettakis et al. [11]. They proposed an interactive version of the initial paper and added a vision algorithm for the camera calibration and the 3D geometrical model automatic positioning. They described a slightly different technique for the estimation of the reflectances of the surfaces and they used a hierarchical radiosity algorithm [18] to compute a new synthetic image close to the real one.

An approach similar to Fournier et al.'s was chosen by Gagalowicz [28]. It included a feedback that compares the real image to the synthetic one. He described a technique to generate a new synthetic image from a single one (except the 3D geometrical model, which was built from two stereo images) using an iterative method that minimizes the error between the real image and the synthetic one. However, this technique is limited to a pure lambertian approximation of the surface reflectances.

3 Elements of Reflectance Recovery

3.1 The Notion of Group

The inputs of our reflectance recovery algorithm are separated into two categories, the 3D geometrical model of the scene and a single image of this scene captured with a standard camera. This method is based on the extraction of the object reflectances from the pixels covered by the projection of these objects in the image (as described later in section 4).

Using a single image to recover all the surface reflectances of the scene raises several problems related to the geometrical model and the size of the projection of the objects in the image. First of all, there are generally many surfaces that are not directly visible in the real image. It is then extremely difficult (sometimes impossible) to compute their reflectances because no information is available about them. This is not important if the position of the observer is never changed. However, it is usual to modify this position especially in augmented reality applications. Therefore, we introduce the notion of *group* of objects and surfaces. These *groups* specify the objects and the surfaces which have the same reflectance properties. This is a very fast manual operation left to the user after or during the geometrical modeling process. For example, in figure 2, the 'red cube' was modeled as a *group* containing six planar objects which have the same reflectance properties. Our reflectance algorithm will then use this description to propagate the estimated reflectance from the three visible faces of the cube to the three other ones.

This *group* notion often solves the second modeling problem which could happen during the reflectance estimation. Indeed, the area covered by the projection of some objects in the real image could be too small to give a good approximation to the reflectance of these objects. Therefore, if the user joins these objects with others which have the same reflectance and a bigger projection area in the real image, it becomes possible to obtain a better approximation of their reflectance. However, if there are no other bigger objects, a very rough approximation of the reflectance will be computed for these small objects, and the resulting image may be biased. This problem is inherent in all image-based rendering methods [7, 41, 23, 24, 14, 11, 28] which use the area covered by the projection of an object in the real image to determine its reflectance. Nevertheless, as our method uses a feedback through the comparison between the real and synthetic image, bias is considerably reduced.

3.2 Reflectance Model and Data Description

For the past several years, the construction of a 3D geometrical model from a single image or a set of images has been widely investigated and is known as *image-based modeling* (see [9] for

an overview of these methods). In our paper, the 3D geometrical model is built interactively using *Alias|Wavefront*'s *Maya* modeler. The positioning of the full 3D geometrical model of figure 2 took around six hours to be complete, including the recovery of the camera parameters and the light sources positions. Typically, for the camera parameters we use the Dementhon and Davis [10] technique combined with a *downhill simplex* minimization method [17, 19]. The light sources have been modeled approximately (because of their complex geometry) and they have been placed manually with a precision of $\pm$ 5cm[2]. Our photometric recovery method is based on the use of Ward's reflectance model [37]. We chose the same BRDF model as Yu et al. [41] because of its small number of parameters and its ability to simulate anisotropic surfaces. This model only requires the knowledge of five parameters for a complex BRDF: ρ_d the diffuse reflectance, ρ_s the specular reflectance, $\vec{x}$ the anisotropy direction (called the *brushed direction*) and the anisotropic roughness parameters α_x and α_y (see [37] for a detailed description of this BRDF model). Furthermore, this model avoids the costly computation of the Fresnel term which has been replaced by a normalization factor.

Figure 2: Example of a real image with the superposition of its 3D reconstructed geometrical model (in white)

When the 3D geometrical model (objects, camera and light sources positions) and the photometric model (reflectances and light sources intensity) are determined, it is possible to render a synthetic image using a classical rendering software such as *Radiance* [38]. We developed our own rendering software called *Phoenix* to obtain a high-performance computing power and to take advantage of the specific architecture of the Silicon Graphics workstations used[3]. *Phoenix* is a global illumination software. It computes the form factors of a progressive radiosity system [4] using a 64 bit A-Buffer [3, 13] mapped on each face of the hemicube [5]. This increases the resolution of each face of the hemicube by a factor of 64 with a negligible increase in computation time, with respect to a classical Z-Buffer software.

Moreover, *Phoenix* uses advanced *OpenGL* programming techniques called *offscreen rendering* to compute the index buffers (or *item buffers* [39]) necessary for the extraction of the pixel intensities from the original image and the synthetic one. Each number in the index buffer indicates either a group number, or an object number, depending on whether we need to compute the reflectance of a group or of an object.

4 Inverse Rendering from a Single Image

4.1 Overview of the Algorithm

The core of our technique is incremental and hierarchical (see figure 3). It is incremental because the surface reflectances evolve to their optimum value. It is hierarchical because the general algorithm forces the surface BRDFs to be more and more complex if the error between the real and the synthetic image does not decrease for these surfaces. This algorithm is iterative and will proceed to successive corrections of the surface reflectances by minimizing the error between the real and the synthetic image. Indeed, each computed error for a group of objects having the same photometric properties drives the correction of their reflectance. Our technique successively applies the selected assumption on the group reflectances until the error became smaller than a user-defined threshold. The notion of threshold and how to fix its value to give them will be discussed in the section 6.

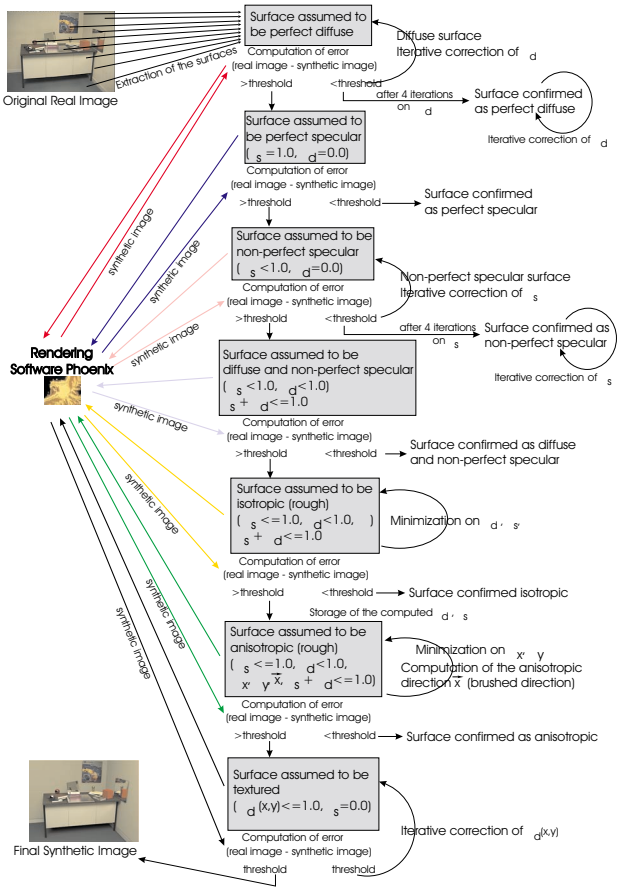

Figure 3: General iterative and hierarchical algorithm for reflectance recovery. Each surface of the scene is analyzed separately, depending on the assumption about its reflectance (perfectly diffuse, perfectly specular, etc.). If the assumption is false (the error between the real and the synthetic image is big), then the surface reflectance is assumed to be more complex (hierarchical principle). If the assumption is correct then the surface reflectance is modified accordingly in order to minimize the error between the two images (iterative principle). During each global rerendering iteration, the reflectances of all surfaces are then continuously updated, to take into account the incident energy coming from any surface for which the BRDF has changed (a diffuse surface which became *perfectly specular* for example).

We start the algorithm with the *perfectly diffuse* case without considering texture (the diffuse reflectance of a group is computed averaging the radiances covered by its projection in the real image). All the surfaces are then considered as perfectly lambertian, and the rendering software (*Phoenix* in this case[4]) computes a new approximation of the image. If the difference between the real and the synthetic image for a group is greater than a fixed threshold on all the group projection, then the reflectance of this group is considered as *perfectly specular* for the next rerendering iteration. If, after *Phoenix* has recomputed a new image using the new assump-

[2] Our technique can be used regardless of how the geometry is acquired.

[3] This work was carried out on a SGI Octane SI 2$\times$ R12000 300Mhz.

[4] It is possible to use any other global illumination rendering software, such as *Radiance* [38] for example.

tion, the error for this group remains large, then its reflectance is simulated as *non-perfectly specular*. We apply the same principle to change again the group reflectance to a *both diffuse and specular* one. Until then, all the surfaces were considered with no roughness term (only a ρ_d and a ρ_s were estimated). In the next assumption, if the difference between the two images still produces big errors, they are considered as isotropic and a roughness factor (α) has to be evaluated. This assumption is extended to anisotropic properties if the user-defined threshold for the error has not been reached. If all assumptions have failed, the group is presumed to be highly textured. Since only a single image is available, it is extremely difficult and sometimes impossible to create a combination between this texture and other reflectance properties (a glossy textured surface for example). This situation is discussed in paragraph 4.7.

4.2 The case of perfectly diffuse surfaces

One of the simplest cases of reflectances is the *perfectly diffuse* one. During the first inverse rendering iteration, all the objects of the scene are simulated as perfectly diffuse. A diffuse reflectance (ρ_d) is then computed for each group, as the average of radiances covered by the projection of the groups in the original image. This technique is different from Drettakis et al. [11, 14] because we do not pay attention to the texture of the surfaces. It is interesting to note that some textured surface may be simulated using a pure diffuse reflectance (as shown in figure 14), to create a good visual approximation. This method is very different from [11, 14] because it is not limited to the computation of the average reflectance to produce the new final synthetic image. We correct this reflectance iteratively until the error between the original and the rerendered image becomes small. For an object, this error is computed as the ratio between the average of the radiances[5] covered by the projection of the groups in the original image, and the average of the radiances covered by the projection of the groups in the synthetic image (see equation 1).

$$\widehat{\varepsilon_j} = \frac{\widehat{B_{o_j}}}{\widehat{B_{n_j}}} = \frac{\widehat{T^{-1}(P_{o_j})}}{\widehat{T^{-1}(P_{n_j})}} \quad (1)$$

$\widehat{B_{o_j}}$ and $\widehat{P_{o_j}}$ are respectively the average of the radiances and the pixels covered by the projection of object j in the original image.
$\widehat{B_{n_j}}$ and $\widehat{P_{n_j}}$ are respectively the average of the radiances and the pixels covered by the projection of object j in the synthetic image.
$T()$ is the camera transfer function (a γ correction function here).

Since the average radiance $\widehat{B_j}$ of object j is proportional to the diffuse reflectance ρ_{dj}, the iterative correction of the ρ_{dj} can be written for each rerendering iteration k as:

$$\rho_{di_{k+1}} = \rho_{di_k} \times \widehat{\varepsilon_i} \quad (2)$$

$$\rho_{di_{k+1}} = \rho_{di_k} \times \frac{\sum_{j=1}^{n_i} f(\widehat{\varepsilon_j}) \cdot (\widehat{\varepsilon_j} \times m_j)}{\underbrace{\sum_{j=1}^{n_i} f(\widehat{\varepsilon_j}) \cdot m_j}_{\neq 0}} \quad (3)$$

$$\text{and } f(\widehat{\varepsilon_j}) = \begin{cases} 0 & \text{if } \widehat{\varepsilon_j} \geq (1+\lambda) \cdot md \\ 1 & \text{else} \end{cases}$$

[5]These radiances have been obtained using the inverse of the camera transfer function that was simulated as a γ correction function with a γ value of 2.2 according to Tumblin et al. [36]. However a more powerful algorithm could be applied if we had more than one photograph of our scene [8].

$\widehat{\varepsilon_i}$ and $\widehat{\varepsilon_j}$ are respectively the total error between the original and the synthetic image for group i and object j.
n_i is the number of objects for group i.
md is the median of the errors (selects the middle value of the sorted samples).
λ is the authorized dispersion criteria.
m_j is the number of pixels covered by the projection of object j.

The function $f()$ eliminates problems generated by smaller objects for which the error is very important, because they are more sensitive to the image noise (their projection in the image cover a small amount of pixels). An example of iterative correction of ρ_d is provided by figure 4 on a very simple synthetic scene, nevertheless containing high color bleeding effects (see how the green cube is influenced by the blue floor for example).

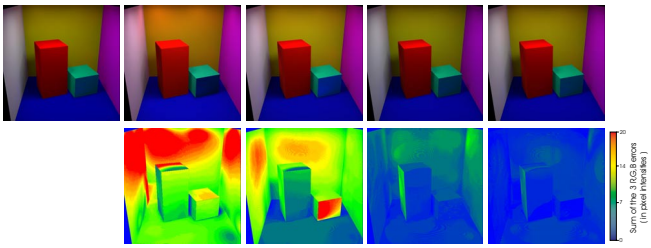

Figure 4: In the top row, from left to right: the original synthetic image (top left) generated using a rendering software was rerendered for 4 iterations (the next four images). The differences between this original image and the regenerated images are shown in the bottom row and displayed using a specific error colormap (at the bottom right). We observe a regular decrease of the error from left to right.

As textures are not taken into account in this section, we only consider a diffuse reflectance parameter ρ_d. It could be interesting and maybe faster to directly inverse the radiosity equation as suggested by Yu et al. [41]. If we know the radiances, the emittances and the full geometry (i.e. the form factors), it is possible to directly solve the radiosity equation [16] for the reflectances. However, this is not so simple, because we work with a single image. Because of this, there may be some surfaces that are not directly visible in the original image. Therefore, their radiosities are unknown and it is impossible to guess their values. Thus, we can not inverse the radiosity equation.

4.3 The case of perfectly and non-perfectly specular surfaces

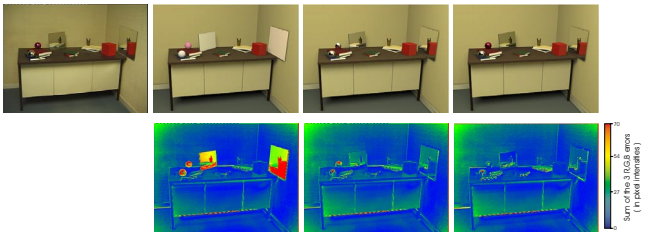

Figure 5: Simulation of hierarchical inverse rendering, where the top row from left to right consists of the real image captured with a camera, the synthetic image with a pure diffuse assumption (first iteration), the synthetic image with perfectly diffuse and perfectly specular assumptions (fifth iteration) and the synthetic image with pure diffuse and non-perfectly specular surfaces (seventh iteration). On the bottom row, we can see the error images corresponding to the difference between the real and the synthetic image.

If the previous diffuse hypothesis about the surface reflectance failed, it is now considered as a *perfect mirror*. It is the easiest case

to solve because the diffuse reflectance of a perfect mirror has a null value ($\rho_d = 0$) and its specular reflectance is equal to 1 ($\rho_s = 1$). It is worth noting that there is no need to iterate on the specular reflectance and a new synthetic image can be directly rendered. On the other hand, the reflectance for a non-perfectly specular object has to be iteratively modified to obtain an optimum ρ_s. The iterative correction of ρ_s is similar to equation 3, except ρ_d has to be replaced by ρ_s. An example of the use of the hierarchical algorithm on a scene containing both diffuse, non-perfectly specular surfaces is shown in figure 5.

4.4 The case of both diffuse and specular surfaces with no roughness factor

At this point of the algorithm, surfaces with big errors are now considered as both diffuse and specular ($\rho_d \neq 0$ and $\rho_s \neq 0$) but still with no roughness.
The differences between the real image and the synthetic are minimized as a function of ρ_d and ρ_s (in the Ward's BRDF model [37]):

$$(T^{-1}(I_{synth}) - T^{-1}(I_o))^2 = \sum_{i=1}^{nbg} (\rho_d \cdot B_d + \rho_s \cdot B_s - T^{-1}(I_o))^2$$

with nbg, the number of pixels covered by the group projection.

This minimization has an analytical solution for each wavelength R, G, B:

$$\begin{pmatrix} \rho_d \\ \rho_s \end{pmatrix} = \begin{pmatrix} \sum_{nbg} B_d T^{-1}(I_o) \\ \sum_{nbg} B_s T^{-1}(I_o) \end{pmatrix} \begin{pmatrix} \sum_{nbg} B_d^2 & \sum_{nbg} B_d B_s \\ \sum_{nbg} B_d B_s & \sum_{nbg} B_s^2 \end{pmatrix}^{-1}$$

In practice, such surfaces in real cases are very rare but not impossible. For example, the top face of the desk in the figure 14 presents some photometric properties very close to this approximation.

4.5 The case of isotropic surfaces

Until now, all the surfaces were supposed to be without roughness. In the case of an isotropic surface, the diffuse reflectance ρ_d, the specular reflectance ρ_s and a roughness coefficient α have to be recovered according to Ward's BRDF model.

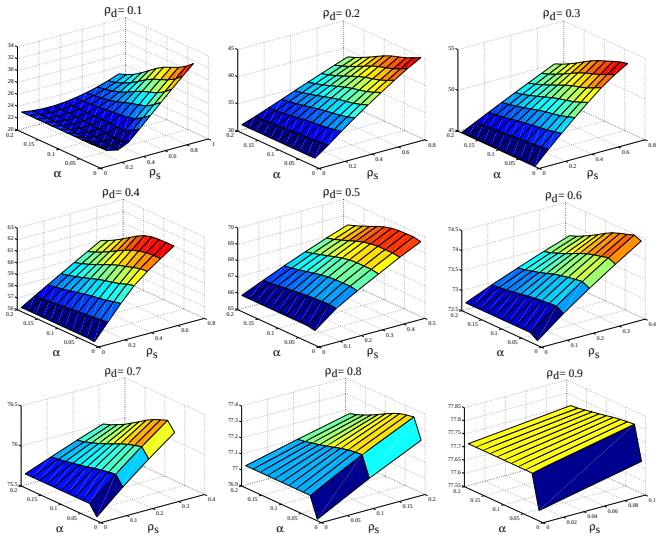

Figure 6: Error function (synthetic image - real image), for a fixed diffuse reflectance, with respect to variations of the isotropic values α, and ρ_s specular reflectance. The evolving steps are 0.018 for α and 0.1 for ρ_s.

A first idea is to use a classical minimization algorithm to solve for these three paremeters. However, the error function (difference between the real and the synthetic image) for an anisotropic surface is radically different if ρ_d is varying in $]0.0; 1.0[$ (figure 4.5) or if ρ_d has a null value (figure 7). Directly minimizing the error function for ρ_d, ρ_s and α in the interval $[0.0; 1.0[$ is thus not possible. We propose to miminize the error function using two separate error functions: one for the interval $]0.0; 1.0[$ and the other for the $\rho_d = 0$ particular case. The minimization algorithm (we use the *downhill simplex* method [17, 19] for the two minimizations) that provides the smallest error will determine the final value of ρ_d, ρ_s and α. One of the disadvantages of the method is that it could take a lot of time minimizing such functions. Indeed, these isotropic surfaces use ray-tracing [1] techniques for their correct simulation. Even if optimization techniques greatly accelerate the rendering [15, 12], it still could take around one hour and fifty minutes to recover the ρ_d, ρ_s and α values (using ten bounced rays for each primary ray (nine per pixel) that reached a glossy surface). In fact, the optimum values of ρ_d and ρ_s are found in only two minutes because the resulting value does not need to be obtained with a precision better than $1 \cdot 10^{-2}$ (the visual difference became imperceptible). On the other hand, α requires a determination with a $1 \cdot 10^{-4}$ precision (according to Ward [37], the α parameters may vary between 0.001 for a perfectly specular surface to 0.2 for a mostly diffuse surface).

Figure 8 shows the result of these minimizations: the aluminium surface (in the center of image) has been simulated as isotropic, and an optimum value of $\rho_d = 0.0$ and $\rho_s = 1.0$ has been found. However the error image shows that maybe a better approximation seems to be possible for this particular surface. The error remains important in the extent of the specular reflection area of the two books on this surface. Therefore a more complex BRDF is needed and the algorithms tries now to simulate the surface as an anisotropic one.

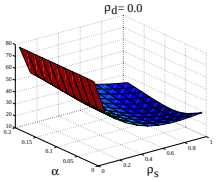

Figure 7: Error function (synthetic image - real image), for $\rho_d = 0$ with respect to variations of α (isotropy value), and ρ_s (specular reflectance).

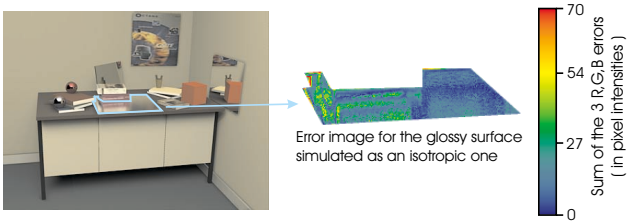

Figure 8: Approximation of the aluminium surface (anisotropic) of the real image (left) by an isotropic surface in the synthetic image (center). The error between these two images for the aluminium surface is visible in the right image. We remark that the error is still important in the area of the specular reflection of the books. The red pixels correspond to a high error but they are not significant because they are coming from an approximative positioning of the 3D geometrical model ont the image, especially on the edges of the objects.

4.6 The case of anisotropic surfaces

Working with anisotropic surfaces is clearly the most complicated case of our algorithm because the anisotropic model of Ward re-

quires minimizing a function of five parameters: the diffuse reflectance ρ_d, the specular reflectance ρ_s, the anisotropy direction $\vec{x}$ (or *brushed direction* [37]) and the roughness factors α_x, α_y. However, it is possible to keep the previous ρ_d and ρ_s values computed for the isotropic case: the error functions (see figure 4.5 and 7) show that the ρ_s parameter is not correlated to the α parameter, because these functions are quite constant with respect to α. We may then suppose that the ρ_d and ρ_s do no differ from the isotropic case to the anisotropic one.

The error function to minimize has now three parameters left (see figure 9). We remark on this figure that for a given rotating angle θ of the vector $\vec{x}$ and varying values of α_x and α_y, this error function presents several minima on all the curves, and they are very similar for all θ values. This confirms that a standard minimization algorithm will probably not find a global minimum.

To prove this assumption, we have computed the four images corresponding to the four smallest minima found by a *downhill simplex* minimization algorithm (figure 10). It is interesting to note that the rerendered images remain far from the original one and that the error is bigger than for the isotropic case. This brings us to the conclusion that a minimization procedure is not the correct way to solve the anisotropic case. Therefore, we propose to determine the anisotropy vector $\vec{x}$ directly from the real image.

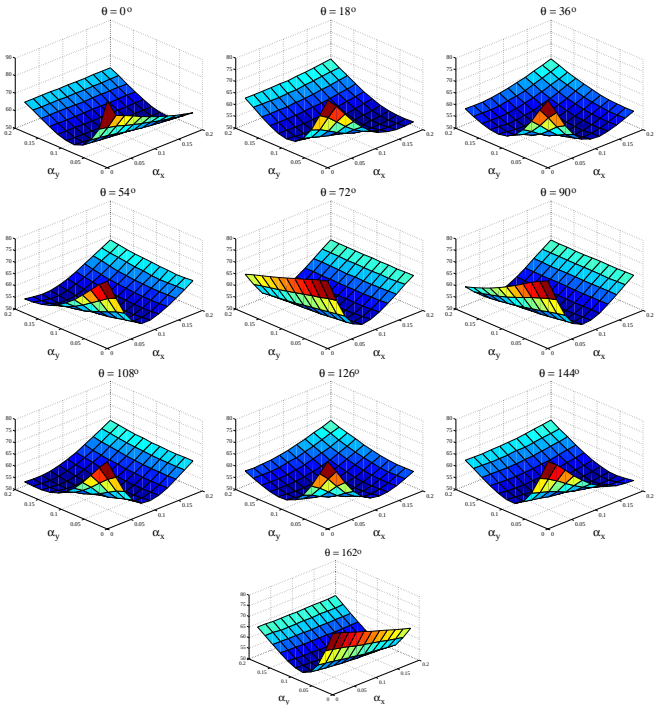

Figure 9: Error function (synthetic image - real image), for different anisotropy directions $\vec{x}$ (the vector is turned around the normal to the surface using a step of 18 degrees) with respect to variations of the roughness parameters α_x, α_y (with a step of 0.018). The diffuse reflectance and the specular reflectance terms have been estimated during the isotropy analysis.

In a first step, we consider the anisotropic surface as a perfect mirror and compute a synthetic image. Next, we estimate the difference between the real image and the synthetic one to visualize the part of the anisotropic mirror where the specular reflection is "extended". This area corresponds to an attenuation of the specular reflection, and this effect is always very important in the direction perpendicular to the brushed direction (or anisotropy direction). In a second step, we compute an index buffer for this mirror of all the surfaces visible through it. We then look for a reference surface that has the biggest reflection area on the anisotropic surface, while being as close as possible to it. This surface is then selected in a such manner that the ratio $\frac{\text{Area(reflected surface)}}{d(S,P)}$ is maximized (with d(S,P), the euclidean distance between the center of gravity of the selected surface and the center of gravity of the anisotropic mirror). The motivation of this choice resides in the fact that surfaces very far from the anisotropic object exhibit a reflection pattern that is too small or too noisy to be usable for the recovery of the brushed direction. In a third step, the anisotropy direction is sampled creating $\vec{x}$ vectors around the normal to the anisotropic surface. Each of these sampled directions determine a direction to traverse the error image and compute the average of the standard error deviations computed in the error image. Finally, the algorithm selects the direction for which this average value is the smallest one (see figure 11). Figure 12 summarizes the complete procedure.

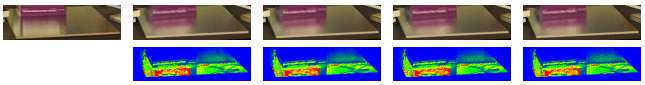

Figure 10: The first image (top left) is the original one (reduced here to the interest area). The next four images have been produced using the four smallest minima found by the minimization algorithm. We can see that all of these images are far from the original one (the vertical black line on the white book (see figure 8) has disappeared from the specular reflection) and that a lot of details have been smoothed. The error colormap remained the same as on figure 8.

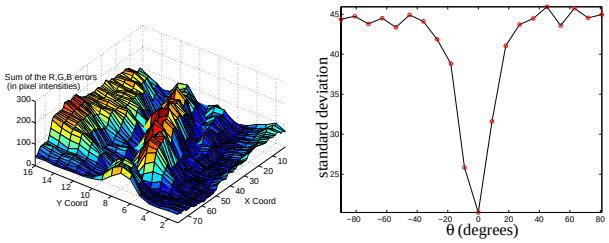

Figure 11: The selected object used here to recover the anisotropy direction is the violet book of the lower left real image of figure 14. The 3D surface (left image) shows the error image for the difference between the perfectly specular reflection area of this selected object, and its corresponding area in the real image. The 2D curve (right) shows the average of the standard error deviations computed from the error image along the sampled anisotropy directions (see also figure 12).

Once the anisotropy direction $\vec{x}$ has been recovered, a *downhill simplex* minimization algorithm is used to estimate the roughness parameters α_x and α_y. Typically, for the synthetic image in the lower right corner of the figure 14, it took 50 iterations and $2h30$ to recover the full BRDF of the anisotropic surface. The algorithm found an optimum anisotropy vector for a rotation angle of 0 degrees and then minimized the error function of the upper left corner of the figure 9. The estimated values of α_x and α_y were 0.01 and 0.062 respectively.

4.7 The case of textured surfaces

When the simulation of a surface as anisotropic still produces big errors in the difference image, we proceed to texture extraction. Extracting the texture from the real image is an easy task that can be realized using the technique proposed by [40] for example. However, we have to extract this texture while taking into account the fact that it already has received the energy from the light sources, and that the pixels covered by its projection in the real image contain this information. Otherwise, if we send the energy of the light sources to these textures again, they will be over-illuminated. Therefore, we introduce here a notion called *radiosity texture* that balances the extracted texture with an intermediate texture in order to minimize the error between the real and the synthetic image. As for the perfectly diffuse reflectance case, this intermediate texture is computed by an iterative method.

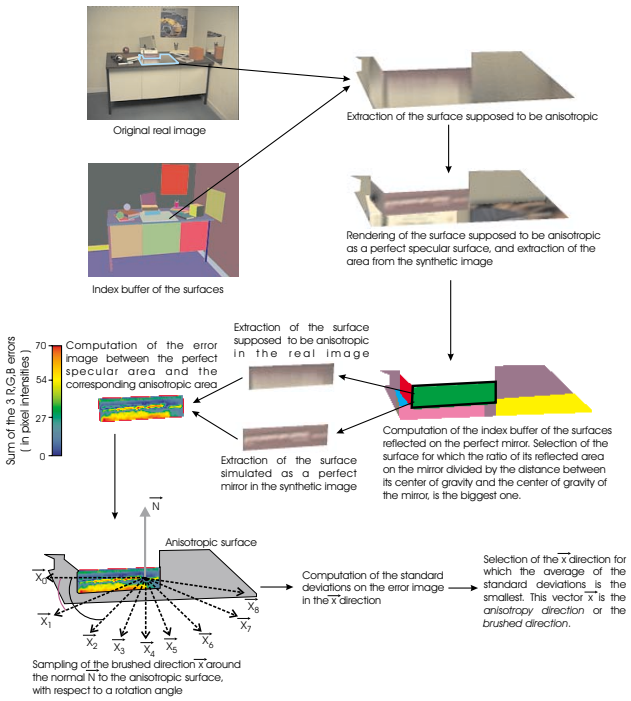

Figure 12: Computation method of the anisotropy direction $\vec{x}$ for a glossy surface.

At the first iteration, the texture used to rerender the image is the texture directly extracted from the real image. At the second iteration, the texture used to obtain the resulting synthetic image is multiplied by the ratio between the newly extracted texture of this synthetic image and the texture of the real image. This iterative process stops when the user-defined threshold for textured surface has been reached. The textures of the poster and the books in the rerendered images of section 7 have been obtained using this technique. The problem of this method is that it computes a texture including the shadows, the specular reflections and the highlights. Typically, suppose that we have a marbled floor on which a sphere is reflected. The texture of this floor in the real image then includes the marble characteristics, its reflectance properties and the sphere reflection including its own reflectance properties. How to extract the marble characteristics only and independently of the rest of the scene ? This is an extremely hard problem, and according to Y. Sato et al. [33] no algorithm has been proposed yet to solve it using a single image.

5 Advanced Analysis of Reflectances

Our inverse rendering procedure provides the opportunity to analyze the reflectances of some surfaces that are not directly seen in the original image. Indeed, if a surface is detected and confirmed as a perfectly or non-perfectly specular one, we can extend our reflectance recovery algorithm to the surfaces that are seen through this mirror in the real image.

First of all, the index buffer of the groups visible through the mirror are computed using a ray tracing algorithm. If there exists a surface in this buffer that was not directly visible before in the real image, then its reflectance is computed taking into account the current assumption made for its group reflectance (the surface has the same photometric properties as its group). In the next iteration, this reflectance is balanced by the mirror reflectance (if it is a non-perfect one), and it is then considered for the correction of the group reflectance (see figure 13).

To our knowledge, this is the first time that an image-based rendering technique deliberatley exploits mirror surfaces to enhance the BRDF recovery process in a scene.

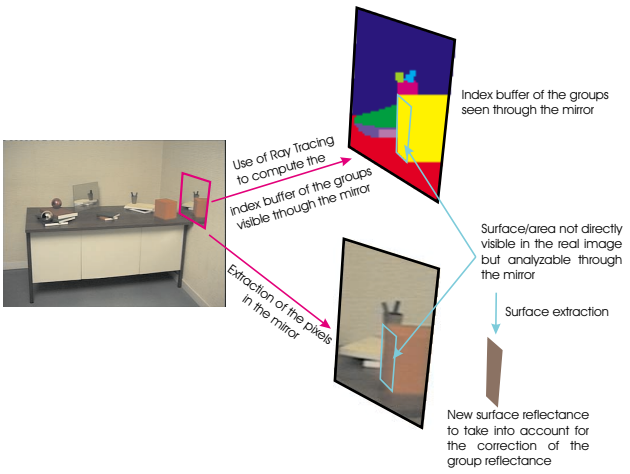

Figure 13: Example of surface not directly seen in the original image. The reflectance of this surface becomes computable through the mirror put on the right wall. This surface belongs to the 'cube' group and now contributes to the estimation of its reflectance. If this face had a specific reflectance (new group), it would be computable only at this point.

6 Optimizations and Determination of Thresholds

Since the reflectance recovery algorithm takes around two hours to simulate an isotropic surface, and two more hours in an anisotropic case, this means that all textured surfaces (which is the final hypothesis after the isotropy and the anisotropy tests) will need four hours to be correctly estimated. This is not acceptable when a lot of surfaces are textured in a real image, but the computing time could be greatly reduced if we can find that the surface is textured before treating the isotropic case. Therefore we introduced a heuristic to solve this problem. It is related to the choice of the thresholds that determine if a surface is correctly simulated. Indeed, after each synthetic image has been produced, it is compared to the real one using a user-defined error threshold to know if the assumptions made about the surface reflectances are correct. For the presented images, the following thresholds were used to produce the results of the section 7. Such thresholds are not critical to the behavior of the technique but will affect speed because it will always find a solution regarding the fixed thresholds.

From the case of the perfectly diffuse assumption up to the isotropic one, the sum of the three R,G,B errors coming from the difference between the real and the synthetic image must have a value smaller than 5%. However, during the non-perfect specular assumption, if the error is greater than 50%, we can directly avoid the isotropic and the anisotropic cases and so greatly increase the performance of the algorithm. We do not have a formal characterization of this optimization, but in practice it seems to work well (see section 7). The isotropic-to-anisotropic threshold has been chosen equal to 1%, to ensure that the algorithm tries the anisotropic case. On the other hand, the threshold used to come to a texture assumption is equal to 5%. Finally, the last threshold is a global threshold that forces all the groups in the synthetic image to have an error smaller than 5%.

7 Results

All the following synthetic images have been generated using *Phoenix* as rendering and inverse rendering software. The first synthetic image at the top right of figure 14 has been generated in 37 minutes using the hierarchical algorithm, from the left real photograph. Two specular surfaces have been recovered and simulated as non-perfect mirrors. Neither the isotropic nor anisotropic hypothe-

ses have been tried thanks to the optimization technique described in section 6, and 14 rerendering iterations were necessary to generate the final image.

The inverse algorithm tooks 4 hours and 40 minutes to produce the image at the bottom right of figure 14. Roughly 4 hours of this time were necessary to recover the anisotropic BRDF of the aluminium surface. The final rendering stage took 32 minutes to render the final image (100 bounced rays have been used for the anisotropic surface).

The images of figure 15 show examples of applications in augmented reality. Some synthetic objects have been added such as a small robot and a luxo-like desk lamp. It is also possible to modify the reflectances easily too. New viewpoints can be generated and new illumination conditions can be created as well.

8 Conclusion and Future Work

In this paper, we have presented a new technique that approximates the reflectance properties of the surfaces of a 3D scene. An incremental and hierarchical algorithm iteratively estimates the diffuse, specular, isotropic and anisotropic reflectance parameters. In a final step, the textured surfaces are considered as a special case of reflectances to be simulated. The method takes as input a single photograph of the scene taken under known illumination conditions as well as a 3D geometric model of the scene. The result is a complete description of the photometric properties of the scene which may be used to produce a photorealistic synthetic image very similar to the real one. We showed that the method is robust and provides the opportunity to visualize the original scene from new angle, with any illumination conditions and with the addition, removal and modification of objects.

Our work has currently some limitations, especially regarding textured surfaces. Until now, we have not been able to discriminate the shadows or highlights from an assumed textured surface. In this regard, it will be interesting to extend our method to these cases, although we think that this is a very difficult problem, if one sticks to the single image assumption.

While many challenges remain, we believe that algorithms for recovering an approximation of the reflectances inside a real scene are an important direction of research for both Computer Vision and Computer Graphics communities. In Computer Vision, it could be possible for example to use our method to enhance the positioning of mirrors using a minimization algorithm between the real and the synthetic image. Regarding Computer Graphics, we may extend the reflectance recovery algorithm to objects that have more complex photometric properties such as light beams, small fires, caustics, etc. The hierarchical property of our technique offers many possible extensions.

Acknowledgments

The authors would like to thanks Jean-Marc Vézien for providing the 3D geometrical model and the camera calibration for the scenes shown in the results section. We also acknowledge him for his helpful scientific commentaries regarding the techniques described in this paper.

References

[1] Arthur Appel. Some techniques for shading machine renderings of solids. *AFIPS 1968 Spring Joint Computer Conf.*, 32:37–45, 1968.

[2] R. Baribeau, M. Rioux, and G. Godin. Color reflectance modeling using a polychromatic laser range sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):263–269, February 1992.

[3] Loren Carpenter. The a-buffer, an antialiased hidden surface method. *Computer Graphics (Proceedings of SIGGRAPH 84)*, 18(3):103–108, July 1984. Held in Minneapolis, Minnesota.

[4] Michael F. Cohen, Shenchang Eric Chen, John R. Wallace, and Donald P. Greenberg. A progressive refinement approach to fast radiosity image generation. In John Dill, editor, *Computer Graphics (Proceedings of SIGGRAPH 88)*, volume 22, pages 75–84, August 1988.

[5] Michael F. Cohen and Donald P. Greenberg. The Hemi-Cube: A radiosity solution for complex environments. In B. A. Barsky, editor, *Computer Graphics (Proceedings of SIGGRAPH 85)*, volume 19, pages 31–40, August 1985.

[6] Kristin J. Dana, Bram van Ginneken, Shree K. Nayar, and Jan J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics*, 18(1):1–34, January 1999. ISSN 0730-0301.

[7] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In Michael Cohen, editor, *Computer Graphics (Proceedings of SIGGRAPH 98)*, Annual Conference Series, pages 189–198. Addison Wesley, July 1998.

[8] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In Turner Whitted, editor, *Computer Graphics (Proceedings of SIGGRAPH 97)*, Annual Conference Series, pages 369–378. Addison Wesley, August 1997.

[9] Paul Ernest Debevec. *Modeling and Rendering Architecture from Photographs.* PhD thesis, University of California, Berkeley, 1996.

[10] D.F DeMenthon and L. Davis. Model-based object pose in 25 lines of code. In *Second European Conference on Computer Vision(ECCV)*, pages 335–343. Springer-Verlag, May 1992.

[11] George Drettakis, Luc Robert, and Sylvain Bougnoux. Interactive common illumination for computer augmented reality. In Julie Dorsey and Philipp Slusallek, editors, *Eurographics Rendering Workshop 1997*, pages 45–56. Springer Wien, June 1997.

[12] Frédo Durand. *Visibilité tridimensionnelle: étude analytique et applications.* Ph.D. thesis, Université Joseph Fourier, Grenoble, France, July 1999.

[13] E. Fiume, A. Fournier, and L. Rudolph. A parallel scan conversion algorithm with anti-aliasing for a general purpose ultracomputer. *Computer Graphics (Proceedings of SIGGRAPH 83)*, 17(3):141–150, July 1983. Held in Detroit, Michigan.

[14] Alain Fournier, Atjeng S. Gunawan, and Chris Romanzin. Common illumination between real and computer generated scenes. In *Graphics Interface '93*, pages 254–262. Canadian Information Processing Society, May 1993. Held in Toronto, Ontario, Canada.

[15] Andrew Glassner. *An Introduction to Ray Tracing.* Academic Press, New York, NY, 1989.

[16] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modeling the interaction of light between diffuse surfaces. In Hank Christiansen, editor, *Computer Graphics (Proceedings of SIGGRAPH 84)*, volume 18, pages 213–222, July 1984.

[17] Press W. H., Teukolsky S.A., Vetterling W.T., and Flannery B.P. *Numerical Recipes in C, The Art of Scientific Computing*, chapter 10.4 Downhill Simplex Method in Multidimensions, pages 305–309. Cambridge University Press, Cambridge, 1992.

[18] Pat Hanrahan, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. *Computer Graphics (Proceedings of SIGGRAPH 91)*, 25(4):197–206, July 1991.

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

[20] Konrad F. Karner, Heinz Mayer, and Michael Gervautz. An image based measurement system for anisotropic reflection. *Computer Graphics Forum*, 15(3):119–128, August 1996.

[21] G. Kay and T. Caelli. Inverting an illumination model from range and intensity maps. *CGVIP: Image Understanding*, 59:183–201, 1994.

[22] Eric P. Lafortune, Sing-Choong Foo, Kenneth E. Torrance, and Donald P. Greenberg. Non-linear approximation of reflectance functions. In *Computer Graphics (Proceedings of SIGGRAPH 97)*, volume 31, pages 117–126, 1997.

[23] C. Loscos, M. C. Frasson, G. Drettakis, B. Walter, X. Grainer, and P. Poulin. Interactive virtual relighting and remodeling of real scenes. Available from www.imagis.imag.fr/Publications RT-0230, Institut National de Recherche en Informatique en Automatique (INRIA), Grenoble, France, April 1999.

[24] Celine Loscos, George Drettakis, and Luc Robert. Interactive virtual relighting of real scenes. *IEEE Transactions on Visualization and Computer Graphics*, 6(3):289–305, 2000.

[25] J. Lu and J. Little. Reflectance function estimation and shape recovery from image sequence of rotating object. In *International Conference on Computer Vision*, pages 80–86, June 1995.

[26] Stephen R. Marschner and Donald P. Greenberg. Inverse lighting for photography. In *Proceedings of the Fifth Color Imaging Conference*. Society for Imaging Science and Technology, November 1997.

[27] Stephen R. Marschner, Stephen H. Westin, Eric P. F. Lafortune, Kenneth E. Torrance, and Donald P.Greenberg. Image-based brdf measurement including human skin. In Dani Lischinski and Greg Ward Larson, editors, *Eurographics Rendering Workshop 1999*. Eurographics, June 1999.

[28] A. Rosenblum. *Data Visualization*, chapter Modeling Complex indoor scenes using an analysis/synthesis framework (André Gagalowicz). Academic Press, 1994.

[29] Imari Sato, Yoichi Sato, and Katsushi Ikeuchi. Illumination distribution from brightness in shadows: Adaptive extimation of illumination distribution with unknown reflectance properties in shadow regions. In *Proceedings of IEEE ICCC'99*, pages 875–882, September 1999.

[30] Kosuke Sato and Katsushi Ikeuchi. Determining reflectance properties of an object using range and brightness images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11):1139–1153, 1991.

[31] Yoichi Sato and Katsushi Ikeuchi. Temporal-color space analysis of reflection. *Journal of Optical Society of America*, 11(11):2990–3002, November 1994.

[32] Yoichi Sato and Katsushi Ikeuchi. Reflectance analysis for 3d computer graphics model generation. *Graphical Models and Image Processing*, 58(5):437–451, 1996.

[33] Yoichi Sato, Mark D. Wheeler, and Katsushi Ikeuchi. Object shape and reflectance modeling from observation. In Turner Whitted, editor, *Computer Graphics (Proceedings of SIGGRAPH 97)*, pages 379–388. Addison Wesley, August 1997.

[34] Siu-Hang Or Tien-Tsin Wong, Pheng-Ann Heng and Wai-Yin Ng. Image-based rendering with controllable illumination. In Julie Dorsey and Phillip Slusallek, editors, *Rendering Techniques '97 (Proceedings of the Eighth Eurographics Workshop on Rendering)*, pages 13–22, New York, NY, 1997. Springer Wien. ISBN 3-211-83001-4.

[35] K.E. Torrance, E.M. Sparrow, and R.C. Birkebak. Polarization, directional distribution, and off-specular peak phenomena in light reflected from roughened surfaces. *Journal of Optical Society of America*, 56(7):916–925, 1966.

[36] Jack Tumblin and Holly Rushmeier. Tone reproduction for realistic images. *IEEE Computer Graphics and Applications*, 13(6):42–48, November 1993.

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

[38] Gregory J. Ward. The radiance lighting simulation and rendering system. In Andrew Glassner, editor, *Computer Graphics (Proceedings of SIGGRAPH 94)*, Annual Conference Series, pages 459–472. ACM Press, July 1994. ISBN 0-89791-667-0. Held in Orlando, Florida.

[39] Hank Weghorst, Gary Hooper, and Donald P. Greenberg. Improved computational methods for ray tracing. *ACM Transactions on Graphics*, 3(1):52–69, January 1984.

[40] George Wolberg. *Digital Image Warping.* IEEE Computer Society Press, Los Alamitos, 1990.

[41] Y. Yu, P. Debevec, J. Malik, and T. Hawkins. Inverse global illumination : Recovering reflectance models of real scenes from photographs. In A. Rockwood, editor, *Computer Graphics (Proceedings of SIGGRAPH 99)*, volume 19, pages 215–224. Addison Wesley Longman, August 1999.

Figure 14: Two different examples of synthetic images (right) rerendered from a single real image (left). We remark that the perfectly diffuse assumption is realistic enough for many surfaces (including the walls, the floor, the desk, etc.).

Figure 15: Examples of several augmented reality applications. All these new images were rendered using our global illumination software *Phoenix*, which first recovered the surface reflectances from the bottom left image of figure 14. The top left image shows the original scene removing some objects (the feet of the desk and the red cube). Note that the right mirror has taken into account the modification. The right top image shows the original scene rendered under a novel viewpoint. The bottom left image shows the scene with modified photometric properties, and the addition of an object (a small robot). The bottom right image presents the scene under novel illumination conditions with the addition and deletion of objects.