Inverse Rendering from a Single Image

Samuel Boivin
Dynamic Graphics Project, University of Toronto, Canada
Andre Gagalowicz
Mirages Project, INRIA-Rocquencourt, France

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 or synthetic 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 captured image. We present several synthetic images that are compared to the original ones, and some possible applications in augmented reality such as novel lighting conditions and addition of synthetic objects.

1. Introduction and Motivations

Research in Computer Graphics has been more and more developed over the past few years. This domain has given the opportunity to produce photorealistic images using physical or empirical techniques. Even if the resulting images were often spectacular, full realism is underachieved when comparing the computer-generated images with real images captured with a camera. A new field called Image-Based Rendering enhances the quality of image synthesis, by directly using the real images to create synthetic ones. A subfield known as Inverse Rendering aims to estimate object reflectances (BRDF) inside a real scene. Using this photometric reconstruction, it is possible to create new synthetic images under novel illumination conditions. Moreover, almost all the techniques in inverse rendering use a 3D geometrical model and in some cases the positions and the intensities of the light sources. Consequently many augmented reality applications become appliable. We can add or remove some objects, and then compute the new interactions between the assembled objects of the scenes. Many authors have contributed to the resolution of the inverse rendering problem.[6,10,11,14-19,21-27,32] These works can be divided into several different categories, depending on the complexity of the scene: one isolated object or a full 3D scene, and the complexity of the illumination: local or global. A lot of work has been accomplished in the determination of the BRDF for an isolated object under specific illumination conditions[14,17-19,24-26] or under general unknown illumination conditions.[20] Some of these techniques are able to produce the exact BRDF from a set images and they generally use a tailored approach to achieve this goal. Moreover, the emphasis of these past works are on the elimination of the costly measures incurred by the use of a gonioreflectometer, rather the creation of new synthetic images. Recently, several other methods have been proposed to extend the photometric reconstruction to augmented reality applications such as viewpoint moving and illumination changes for example.[6,32,15,16] These contributions generally use a sparse set of photographs to estimate the full BRDF of materials inside a real scene.[6,32,15,16] This often generates additional work for the user, especially if several images have to be taken under specific viewpoints.[32] Fournier et al.[11] proposed another approach that estimates only diffuse reflectances using a single image. We extend this work by introducing a new hierarchical system to estimate the full BRDF of objects from a single image, following our previous works in the inverse rendering field.[21,3,1,4] This paper is a description of this work and it includes a new experimental validation on a synthetic scene comparing real and recovered parameters for different BRDF.

2. PreviousWork

All the techniques and ideas in this paper have been made possible by works about photorealistic rendering including global illumination and ray tracing, image-based modeling and BRDF modeling. However, this paper falls mainly within the description of inverse rendering, image-based rendering and reflectance recovery. We limit here the overview of the previous methods to the most relevant algorithms to our technique. Therefore, the background described here includes only techniques which take into account a full 3D scene and use global illumination. A complete overview of all the existing algorithms is available in Refs. 4 and 2.

2.1. Reflectance Recovery from Several Images

Debevec[6] used global illumination for augmented reality applications. To insert new objects inside a real image, he needed to take into account interreflections and computed 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 manually calculated 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 perfect diffuse parameters are set by an automatic procedure.

Yu et al.[32] proposed a complete solution for the recovery of surface BRDF from a sparse set of images captured with a camera; 12 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 intensity conversion function (camera transfer function).[7] Using an image-based modeling software such as Façade,[8] a 3D geometrical model of the scene was built from the set of images. All the data were then utilized to recover the BRDF of the modeled surfaces. Their method minimized the error in the parameters of the Ward's anisotropic BRDF model[29] 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.[15] proposed a method based on an original idea from Fournier et al.[11] Their algorithm recovered 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 the insertion/removal of objects and to the modification of the lighting conditions of the original scene. More recently, Loscos et al.[16] 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.

2.2. Reflectance Recovery from a Single Image

A pioneering work in this domain was completed by Fournier et al.[11] 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 were considered to be perfectly diffuse, and they used their reprojection onto the real image to estimate their reflectances. A radiosity-based algorithm then computed an image applying these reflectances to a progressive radiosity technique[5] to obtain a new synthetic image.

An extension of the previous method was developed by Drettakis et al.[10] They proposed an interactive version of the initial paper and added a vision algorithm for the camera calibration and the automatic positioning of the 3D geometrical model. They described a slightly different technique for the estimation of the reflectances of the surfaces and they used a hierarchical radiosity algorithm[13] to compute a new synthetic image similar to the real one.

An approach similar to that of Fournier et al. was chosen by Gagalowicz.[21] 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 image using an iterative method that minimizes the error between the real image and the synthetic one. Note, however, that the 3D geometrical model obtained in the process was built from two stereo images. This technique is limited to a pure lambertian approximation of the surface reflectances. An extension of this work has been realized by Boivin et al.,[4] who introduced a new technique taking into account complex BRDFs of objects inside a real scene. They proposed a hierarchical and iterative method which minimizes the error between the real and the synthetic image to estimate various types of BRDF, such as anisotropic surfaces. They applied their work to augmented reality applications.

3. Data and Work Base

3.1. Two Fundamental Data

The method that we propose here requires two data. First of all, we need a full three-dimensional geometrical model of the scene including the intensities and the positions of the light sources. The construction of the 3D model can be achieved by many different ways including manual ones. We used Maya (Alias Wavefront) to manually position the 3D geometrical models of objects in the original image and to approximately build the light sources. All the camera parameters have been recovered using the Dementhon and Davis[9] technique combined with a downhill simplex minimization method.[12] However, many other techniques can be used to obtain the camera parameters and the 3D geometrical model.[8] Moreover, in our algorithm, all these reconstructed objects must be grouped by the type of reflectance. This means that the user must declare inside a groupall the objects which are supposed to have the same BRDF (for example perfectly diffuse or isotropic). This is a very important heuristic, because the inverse rendering algorithm will now be able to compute or attribute reflectances to objects which are not directly seen in the original image. This structuring of data also allows for some augmented reality applications, such as viewpoint modification and object insertion for example. This grouping operation is a very fast manual operation performed during of after the modeling step. Finally, the second data that we need is one single image of the real scene captured using any camera*, without any constraint on the position of the observer.

3.2. Accuracy of the Geometrical Model

The precision required by the inverse algorithm for the positioning of the geometrical model tolerates several pixels of difference between the projection of the model and the real objects in the image. The acceptable number of misclassified pixels depends on the size of the projected object in the original image. For example, if the projection of all objects belonging to the same group has a total number of ten visible pixels, then the inverse algorithm will compute the wrong BRDF when at least about three or four of the ten pixels do not belong to the currently analyzed objects. We use very classical filtering methods, such as edge detectors, edge removal filters and

* We used a 3xCCD Sony camera, DCR-VX1000E.

a planar approximation, to reduce inconsistencies with the geometrical model by minimizing the number of pixels assigned to a wrong object.

4. Our Inverse Rendering Algorithm

The inverse rendering algorithm can be described using two concepts: an **iterative** one and a **hierarchical** one (see Figure 1). When the algorithm starts, it considers all the objects inside the scene as perfectly diffuse. The BRDFs of all the objects are initialized to the average of the radiances computed from the pixel intensities[†] covered by the projection of the group in the original image.

Following this diffuse assumption, our algorithm compute a new synthetic image using photo-realistic rendering techniques.[‡] Our inverse method attempts to minimize the error between the real and the synthetic image in order to obtain the best possible approximation for the BRDF. The iterative step seeks the best parameters following a given assumption about the BRDF. The hierarchical step changes the hypothesis regarding the BRDF if the iterative step fails to obtain a small error between the real and the synthetic image.

Each time a new image has been generated, an image difference is then computed to determine which object BRDF must be changed. If the perfectly diffuse assumption produces a big error between the two images for a given group, then the inverse rendering algorithm chooses another hypothesis regarding the reflectance of this group. It tries a more complex BRDF model (a perfectly specular one here). Again, *Phoenix* generates a new synthetic image using the new hypothesis, and the inverse algorithm computes a new error image to determine which object BRDF must be modified. As we can see, the inverse algorithm uses more and more complex hypotheses (hierarchical principle) to obtain the correct BRDF and the corresponding parameters. Several hypotheses are successively applied and the algorithm stops when the error between the real and the synthetic image is smaller than a global user-defined threshold. The determination of the thresholds is not critical to our method and it can be found in Refs. 2 and 4.

4.2. Computing the Ward's BRDF parameters

All the BRDF parameters that are estimated here, come from the Ward's BRDF model.[29] We chose the same BRDF model as Yu et al.[32] 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. Furthermore, this model avoids the costly computation of the Fresnel term which has been replaced by a normalization factor. A detailed description of this BRDF model can be found in Ref. 29.

[†] 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.[28] However a more powerful algorithm could be applied if we had more than one photograph of our scene.[7]

[‡] We use our own rendering software called *Phoenix*[2] to compute the new images, but any global illumination software such as Radiance[30] can be used as well

4.1. Overview

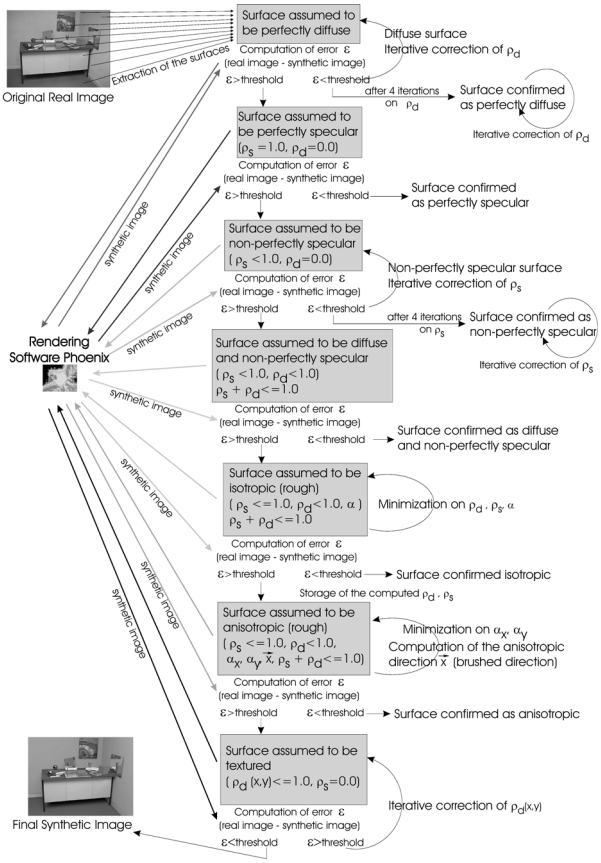

Figure 1. General iterative and hierarchical algorithm for reflectance recovery. Each surface of the scene is analyzed separately, depending on the assumption about its reflectance (perfectly diffuse, perfectly specular, etc.). If the assumption is false (the error between the real and the synthetic 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 modified to be perfectly specular for example).

4.2.1. Perfectly Diffuse Surfaces

The perfectly diffuse case is very simple because only one parameter (ρ_d) has to be computed. During the first iteration, all objects are assumed to be perfectly diffuse. Every reflectance for each group is initialized to the average of the radiances covered by the projection of the group in the original image. *Phoenix* generates a new synthetic image using these reflectance. A new error is computed as the ratio between the average of the radiances 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 Eq. 1). This error balances the original diffuse reflectance, and after several iterations an optimum value of ρ_d is found.[§]

[§] it is shown in Ref. 2 that only 4 iterations are sufficient to converge to an optimum value of ρ_d

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

where:
$\widehat{B_{o_j}}$ 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.

Since the average radiance $\hat{B}_j$ 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:

$$\rho_{di_{k+1}} = \rho_{di_k} \times \hat{\varepsilon}_i \quad (2)$$

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

and

$$f(\hat{\varepsilon}_j) = \begin{cases} 0 & \text{if } \hat{\varepsilon}_j \geq (1+\lambda) \cdot md \\ 1 & \text{else} \end{cases}$$

where:
$\hat{\varepsilon}_i$ and $\hat{\varepsilon}_j$ 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 covers a small amount of pixels). An example of iterative correction of ρ_d (and ρ_s) is provided by Figure 2 on a simple real interior scene, containing both diffuse and specular objects.

After several iterations, a new error image is still computed as the difference between the real and the latest synthetic image. If this error remains bigger than a user-defined threshold for a given group, then the algorithm now decides that all these objects are perfectly specular.

4.2.2. Perfectly and Non-Perfectly Specular Surfaces

In the case of perfectly specular surfaces, it is extremely easy to compute the reflectance parameters, because ρ_d has a null value and ρ_s is constant (ρ_s =1). A new synthetic image can be immediately generated taking into account the new BRDF. But, if the new error for objects assumed as perfectly specular remains large, the algorithm tries to enhance the ρ_s parameter. This new type of BRDF corresponds to the non-perfectly specular case. This specular parameter is modified according to equation 3 applied to ρ_s instead of ρ_d. The images of Figure 2 have been generated using this technique and clearly shows a significant decrease in the error during the inverse rendering iterations.

If the resulting synthetic image still differs from the original one in terms of error (image difference by group of objects), then the diffuse and specular hypothesis is applied.

Figure 2. 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 perfect diffuse and perfect specular assumptions (fifth iteration) and the synthetic image with pure diffuse and non-perfect 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.

4.2.3. Both Diffuse and Specular Surfaces

In the Ward's BRDF model,[29] we now consider the case where ρ_d and ρ_s have a non-null value. All the surfaces are assumed perfectly smoothed which means that there is no roughness factor to compute.

These two parameters can be analytically estimated by minimizing the error between the real image and the synthetic image as a function of ρ_d and ρ_s:

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

where:
nbg, the number of pixels covered by the group projection.
B_{synth},B_{org} the pixel intensities converted to radiances respectively for the synthetic and the original images.

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

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

In practice, such surfaces in real cases are very rare but not impossible. For example, the top face of the desk in Figure 9 presents some photometric properties very close to this approximation.

4.2.4. Isotropic Surfaces

In order to solve the case of isotropic surfaces, we must now find three parameters: the diffuse reflectance ρ_d, the specular reflectance ρ_s and the roughness parameter α.[29] In most cases, a direct minimization algorithm can be used to find these parameters. However, we have shown in Ref. 4 that it is not always easy to minimize such a function. Therefore, it could be useful to separate the case $\rho_s = 1$ from the other cases. We then minimize these two cases separately using a downhill simplex method[12] and we choose the parameters which produce the smallest error.

Figure 3 shows the result of these minimizations: the aluminium surface (in the center of image) has been simulated as isotropic, and an optimum value of $\rho_d = 0.04$ and $\rho_s = 0.96$ has been found. However the error image shows that a better approximation might be possible for this particular surface. The error remains important in the region bordering the specular reflection area of the two books on this surface. Therefore a more complex BRDF is needed and the algorithms now attempts to simulate the surface as an anisotropic one.

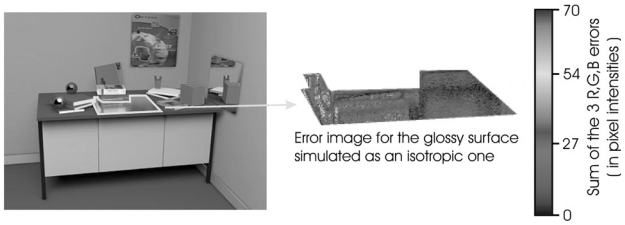

Figure 3. Approximation of the aluminium surface (anisotropic) of the real image (left) by an isotropic surface in the synthetic image (center). The error between these two images for the aluminium surface is visible in the right image. We note that the error is still important in the area of the specular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

4.2.5. Anisotropic Surfaces

In the case of isotropic surfaces, we saw that we had three parameters to compute (ρ_d, ρ_s and α). For the anisotropic case, we must now compute the anisotropy direction ($\vec{x}$) and two other roughness parameters (α_x, α_y) replacing the previous α in the isotropic case. It has been shown in Ref. 4 that a direct minimization algorithm to estimate these parameters produces results of poor quality even if the method converges. Therefore we propose a direct estimation of the anisotropy direction from the original image.

If we could zoom in, we could see that an anisotropic surface has small wave-like features (roughness) on the surfaces characterized by a common direction. This direction called the brushed direction is the anisotropy direction that we are looking for. These waves are clearly visible on the left image of Figure 4 computed for an anisotropic surface. However, they are not directly visible from the original image: the left image of Figure 4 is displayed as a 3D surface and it is produced from several processing steps that are described below.

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).

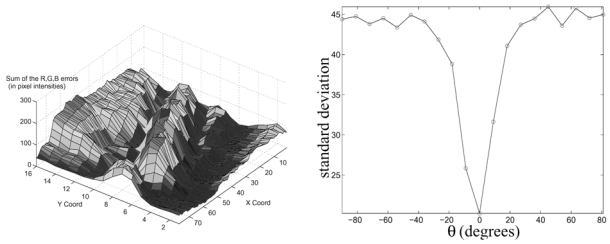

Figure 4. The selected object used here to recover the anisotropy direction is the violet book of the lower left real image of figure 9. The 3D surface (left image) shows the error image for the difference between the perfect specular reflection area of this selected object, and its corresponding area 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 5).

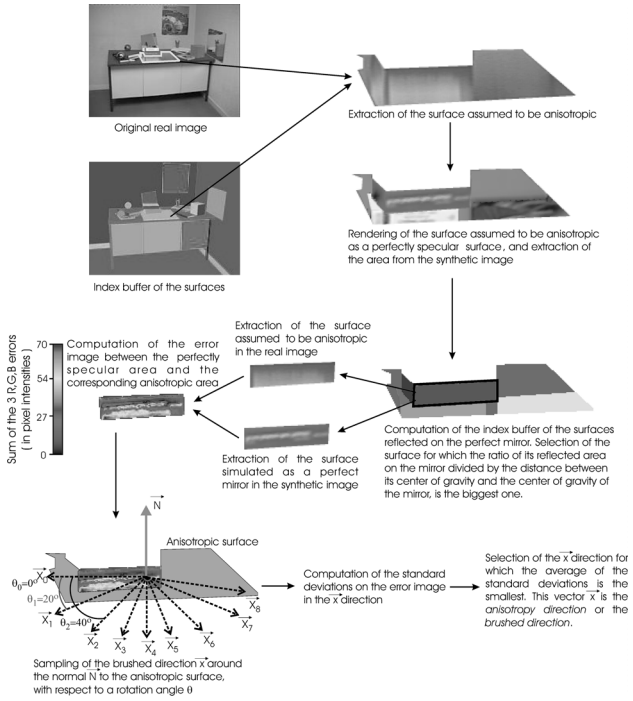

Figure 5. Computation method of the anisotropy direction $\vec{x}$ for a glossy surface.

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 such a 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 4). Figure 5 summarizes the complete procedure.

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.

4.2.6. Textured Surfaces

When the anisotropic simulation of a surface still produces large 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 Ref. 31 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 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.

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 surfaces has been reached. The textures of the poster and the books in the rerendered images of Section 4.3.2 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. As an example, consider 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 then do we extract the marble characteristics and independently of the rest of the scene? This is an extremely hard problem, and Y. Sato et al.[26] have stated that no algorithm has yet been proposed to solve it using a single image.

Figure 6. From left to right: original anisotropic floor, floor simulated as an isotropic object, and the error image between the original and the rerendered images.

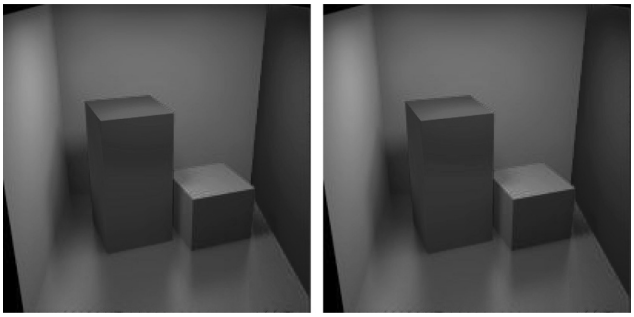

Figure 7. Left: the original computer-generated image. Right: the new synthetic image produced by our inverse rendering technique.

4.3. Results and Computation Times

4.3.1. Comparison of Recovered Parameters

In this section, we propose to give the values obtained for the recovered BRDF of a computer-generated scene (see left image of Figure 7). We compare them to the original known values used to render the original image with *Phoenix*.

The first level of the hierarchy in the inverse rendering process computes all the parameters of the surfaces in a straightforward manner. However, the error remains large for the floor and the next levels are tested for this object. The specular assumptions (perfectly, non-perfectly, both diffuse and specular) produced large errors forcing the algorithm to choose the isotropy hypothesis. During the isotropy case, a global minimum has been found for ρ_d, ρ_s and α, and the synthetic image is visually very close to the original as shown by Figure 6. However, as we only set 1% for the maximum tolerated error to

switch from the isotropy hypothesis to the anisotropy, our method tries to simulate the floor as an anisotropic object.

Using the method described in Section 4.2.5, our algorithm finds all the reflectance parameters for the anisotropic object. All the recovered values are summarized in Figure 8 and the final resulting image is shown in Figure 7.

Surface	Var	Real	Computed
Left wall	ρ_d	(0.66, 0.66, 0.66)	(0.65916, 0.66075, 0.66037)
	ρ_s	(0.0, 0.0, 0.0)	(0.0, 0.0, 0.0)
Right Wall	ρ_d	(0.69, 0, 0.95)	(0.69002, 0.0, 0.95901)
	ρ_s	(0.0, 0.0, 0.0)	(0.0, 0.0, 0.0)
Back Wall	ρ_d	(0.65, 0.65, 0.0)	$(0.64997, 0.65067, 4 \cdot 10^{-7})$
	ρ_s	(0.0, 0.0, 0.0)	(0.0, 0.0, 0.0)
Ceil	ρ_d	(1.0, 1.0, 1.0)	(1.0, 1.0, 1.0)
	ρ_s	(0.0, 0.0, 0.0)	(0.0, 0.0, 0.0)
Big Block	ρ_d	(0.77, 0.0, 0.0)	$(0.77002, 2 \cdot 10^{-4}, 3 \cdot 10^{-6})$
	ρ_s	(0.0, 0.0, 0.0)	(0.0, 0.0, 0.0)
Small Block	ρ_d	(0.0, 0.76, 0.26)	(0.0, 0.75802, 0.25912)
	ρ_s	(0.0, 0.0, 0.0)	(0.0, 0.0, 0.0)
Floor	ρ_d	(0.1, 0.1, 0.1)	(0.10013, 0.10045, 0.09981)
	ρ_s	(0.9, 0.9, 0.9)	(0.89909, 0.90102, 0.89903)
	θ	0.0°	2.8°
	α_x	0.07	0.06999
	α_y	0.11	0.1101

Figure 8. Comparison between the recovered reflectance parameters and their original values. Note that Ceil is not directly visible in the original image. When this happens, the algorithm considered this object as a perfect diffuse white object. In practice, if such a case happens, the user should find an object whose photometric properties are close to Ceil. Ceil will then be declared in the same group as this object.

4.3.2. Rerendered Scenes

All the following synthetic images have been generated using Phoenix as the rendering and inverse rendering software. The first synthetic image at the top right of Figure 9 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 and 14 rerendering iterations were necessary to generate the final image.

The inverse algorithm required 4 hours and 40 minutes to produce the image at the bottom right of Figure 9. Roughly 4 hours of this time were necessary to recover the anisotropic BRDF of the aluminium surface. The final rendering stage required 32 minutes to render the final image (100 bounced rays have been used for the anisotropic surface).

The images of Figure 11 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 without much difficulty. New viewpoints can be generated and new illumination conditions can be created as well.

5. Conclusion and Future Work

In this paper, we have presented a new technique to determine an approximation of the reflectance properties of the surfaces of a 3D scene, and we have proposed an experimental validation of our method. An incremental and hierarchical algorithm iteratively estimates various types of reflectance parameters, including anisotropic and textured surfaces. 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 gives the opportunity to display the original scene from novel viewpoint, with unrestricted illumination conditions and with the addition, removal and modification of objects.

Figure 9. Example of a pure diffuse approximation of a whole 3D scene. From left to right: the original image captured with a camera, the synthetic image and a synthetic image generated under a new viewpoint. The perfect diffuse assumption is realistic enough for many surfaces, except the computer monitor and the door. Moreover, even if the desk is a real anisotropic surface, a pure diffuse approximation produces a realistic enough result for this object. Note that a book on the left bookshelf has not been modeled. Due to the filtering step and the principle of the method, this does not disturb the inverse rendering case. However, this remains true only for small objects that do not interact much with the real environment. A "very" large error in the modeling step would definitely produce wrong results.

Figure 10: Two different examples of synthetic images (right) rerendered from a single real image (left). We note that the perfectly diffuse assumption is realistic enough for many surfaces including the walls, the oor , the desk among others.

Figure 11: Examples of several augmented reality applications. All these new images were rendered using our global illumination software *Phoenix*, which rst recovered the surface re ectances from the bottom left image of Figure 9. The top left image shows the original scene with some objects removed: the feet of the desk and the red cube. Note that the right mirror has taken into account the modi cation. The right top image shows the original scene rendered under a novel viewpoint. The bottom left image shows the scene with modi ed 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.

Currently, our work has some limitations, especially regarding textured surfaces. Until now, we are not 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 remains restricted to a single image.

Moreover, several other extensions are possible because of the hierarchical property of our technique. For instance, we may extend the reflectance recovery algorithm to objects that have more complex photometric properties such as light beams, small fires and caustics as a few examples.

Acknowledgments

The authors would like to thank Jean-Marc V´ezien for providing the 3D geometrical model and the camera calibration for the scenes shown in the results section. We also acknowledge Glenn Tsang for his helpful suggestions regarding the writing of this paper.

References

1. Jacques Blanc-Talon and Dan Popescu. *Advances in Computation: Therory and Practice*, chapter Advanced Computer Graphics and Vision Collaboration Techniques for Image-Based Rendering (Samuel Boivin and Andr´e Gagalowicz). Nova Science Books and Journals, New York, NY., 2000.
2. Samuel Boivin. *Simulation Photor´ealiste de Sc`enes d'Int´erieur `a partir d'Images R´eelles*. PhD thesis, Sp´ecialit´e Informatique, ´ Ecole Polytechnique, Palaiseau, January 2001.
3. Samuel Boivin and Laroussi Doghman. A new rendering technique for the realistic simulation of natural scenes. In *Proceedings of IMAGECOM'96*, pages 302–307, 1996.
4. Samuel Boivin and Andr´e Gagalowicz. Image-based rendering of diffuse, specular and glossy surfaces from a single image. *Proceedings of SIGGRAPH 2001*, pages 107–116, August 2001. ISBN 1-58113-292-1.
5. 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 (SIGGRAPH '88 Proceedings)*, volume 22, pages 75–84, August 1988.
6. 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, *Proceedings of SIGGRAPH 98*, Computer Graphics Proceedings, Annual Conference Series, pages 189–198. Addison Wesley, July 1998.
7. Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In Turner Whitted, editor, *Proceedings of SIGGRAPH 97*, Computer Graphics Proceedings, Annual Conference Series, pages 369–378. AddisonWesley, August 1997.
8. Paul Ernest Debevec. *Modeling and Rendering Architecture from Photographs*. PhD thesis, University of California, Berkeley, 1996.
9. 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.
10. 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. SpringerWien, June 1997.
11. 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.
12. Press W. H., Teukolsky S.A., VetterlingW.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.
13. Pat Hanrahan, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. *Computer Graphics (Proceedings of SIGGRAPH 91)*, 25(4):197–206, July 1991.
14. G. Kay and T. Caelli. Inverting an illumination model from range and intensity maps. *CGVIP: Image Understanding*, 59:183–201, 1994.
15. 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 Rechercherche en Informatique en Automatique (INRIA), Grenoble, France, April 1999.
16. 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.
17. 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.
18. 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.
19. 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.
20. Ravi Ramamoorthi and Pat Hanrahan. A signal-processing framework for inverse rendering. *Proceedings of SIGGRAPH 2001*, pages 117–128, August 2001. ISBN 1-58113-292-1.
21. A. Rosenblum. *Data Visualization*, chapter Modeling Complex indoor scenes using an analysis/synthesis framework (Andr´e Gagalowicz). Academic Press, 1994.
22. Imari Sato, Yoichi Sato, and Katsushi Ikeuchi. Illumination distribution from brightness in shadows: Adaptive estimation of illumination distribution with unknown reflectance properties in shadow regions. In *Proceedings of IEEE ICCC'99*, pages 875–882, September 1999.
23. 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.

24. Yoichi Sato and Katsushi Ikeuchi. Temporal-color space analysis of reflection. *Journal of Optical Society of America*, 11(11):2990–3002, November 1994.
25. Yoichi Sato and Katsushi Ikeuchi. Reflectance analysis for 3d computer graphics model generation. *Graphical Models and Image Processing*, 58(5): 437–451, 1996.
26. 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. AddisonWesley, August 1997.
27. 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.
28. Jack Tumblin and Holly Rushmeier. Tone reproduction for realistic images. *IEEE Computer Graphics and Applications*, 13(6):42–48, November 1993.
29. Gregory J. Ward. Measuring and modeling anisotropic reflection. In Edwin E. Catmull, editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 265–272. ACM Press, July 1992.
30. Gregory J. Ward. The RADIANCE lighting simulation and rendering system. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 459–472. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.
31. George Wolberg. *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos, 1990.
32. 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 (SIGGRAPH 1999 Proceedings)*, volume 19, pages 215–224. Addison-Wesley Longman, August 1999.

Biography

Samuel Boivin is currently a post-doctoral researcher in the Dynamic Graphics Project at the University of Toronto. He received an M.S. in Computer Graphics from Compi`egne University of Technology (U.T.C) in 1995, and a Ph.D. in Computer Science from Ecole Polytechnique in Palaiseau (France) in January 2001. His current research interests are physical simulation and inverse rendering. Among his publications, he recently presented a new image-based rendering technique at the SIGGRAPH 2001 conference, co-authored with Andr´e Gagalowicz.

Andr´e Gagalowicz is a research director at INRIA (France). He is the creator of the first laboratory involved in image analysis/synthesis collaboration techniques. He graduated from Ecole Superieure d'Electricit´e in 1971, obtained a Ph.D in Automatic Control from the University of Paris XI (1973), and a state doctorate in Mathematics from the University of Paris VI (1983). His research interests are in 3D approaches for computer vision, computer graphics, and their cooperation. He has published more than one hundred publications related to these fields in many journals and conferences, and he contributed in the writing of five books.