

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

Bruno Mercier, Daniel Meneveaux, Alain Fournier

▶ **To cite this version:**

Bruno Mercier, Daniel Meneveaux, Alain Fournier. A Framework for Automatically Recovering Object Shape, Reflectance and Light Sources from Calibrated Images. International Journal of Computer Vision, 2007, 73 (1), pp.77-93. 10.1007/s11263-006-9273-y . hal-00331473

HAL Id: hal-00331473
https://hal.science/hal-00331473

Submitted on 17 Oct 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

International Journal of Computer Vision manuscript No.
(will be inserted by the editor)

Bruno Mercier · Daniel Meneveaux · Alain Fournier

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

Abstract In this paper, we present a complete framework for recovering an object shape, estimating its reflectance properties and light sources from a set of images. The whole process is performed automatically. We use the shape from silhouette approach proposed by R. Szeliski in [40] combined with image pixels for reconstructing a triangular mesh according to the marching cubes algorithm. A classification process identifies regions of the object having the same appearance. For each region, a single point or directional light source is detected. Therefore, we use specular lobes, lambertian regions of the surface or specular highlights seen on images. An identification method jointly (i) decides what light sources are actually significant and (ii) estimates diffuse and specular coefficients for a surface represented by the modified Phong model [25]. In order to validate our algorithm efficiency, we present a case study with various objects, light sources and surface properties. As shown in the results, our system proves accurate even for real objects images obtained with an inexpensive acquisition system.

Keywords shape from silhouette · marching cubes · multiple light sources detection · reflectance properties recovery · reshading

1 Introduction

Since the early years of computer vision, much effort has been dedicated for automatically digitizing shape and reflectance of real objects. For instance, Stanford Digital Michelangelo [24] project aims at digitizing large statues, Callet et al. [3] reconstruct plaster statuettes covered with bronze. Hasenfratz et al. [13] detect and place a real actor in a virtual environment so that shadows and lighting be properly computed. In [36], Sato et al. propose to estimate shape as well as reflectance parameters for real textured objects, using multiple range images. Relighting objects have also been proposed using reflectance fields [7,14].

In most cases, the acquisition hardware plays a major role for reconstructing objects from images and research efforts have increased a lot during the last decade. Our concern is about objects described only with a set of calibrated photographs.

Light sources estimation is also a key issue for many applications related to computer vision, image processing or computer graphics. For example, segmentation algorithms can be improved if the incoming light direction is known [33]. Moreover, shape from shading approaches rely on some information concerning light sources [46,22,2], as well as augmented reality where virtual objects have to be lit with real light sources. Today applications in computer graphics are more and more concerned with this topic, especially with image-based rendering approaches.

Our method can be applied to several calibrated images of a single object and does not need any additional test pattern or specific object (such as a metallic sphere for instance). It rather relies on a preprocessing step aiming at recovering a geometric description of the object seen. The basis of our work is the voxel-based shape from silhouette technique presented by Szeliski in [40]. The marching cubes algorithm is combined with image pixels for providing a triangular mesh defining the object shape as well as the surface normal [29].

Based on this geometry representation, our method automatically and robustly recovers multiple light sources together with the surface reflectance parameters. The object surface is assumed lambertian or glossy, without anisotropy, off-specular reflections or hair for instance. First, our algorithm classifies the object surface into regions according to the appearance. Second, for each region, reflectance properties are estimated together with one light source (point or directional). Finally, reflectance parameters and light sources are refined using an error estimate based on original image pixels. When specular highlights are visible on the images,

B. Mercier / D. Meneveaux
SIC Laboratory, University of Poitiers,
Bât. IFMI, Bd M. et P. Curie, 86962 Futuroscope Chasseneuil, France
Fax: +33(0)549-496-570
E-mail: {mercier/daniel}@sic.univ-poitiers.fr

A. Fournier
Imager Lab, University of British Columbia
Department of Computer Science, X635-2366 Main Mall
Vancouver, BC V6T 1Z4, Canada

we also propose a method for improving the light sources estimation accuracy. As shown in the results, our method has been successfully applied to various synthetic and real objects, with lambertian and glossy surfaces.

In this paper, our main contribution concerns the description of a complete framework for recovering geometry, normals, brdfs and light sources from a set of calibrated images. This work relies on the geometric reconstruction method described in [29] and extends/improves the brdf and light sources estimation presented in [28]. Here, we provide: (i) the physical justifications of our approach, (ii) the use of specular highlights seen on the acquired images for improving detection accuracy and (iii) new views of the objects with varying lighting conditions. Our method has proven robust even with images of real objects obtained with an inexpensive acquisition system.

The remaining of this paper is organized as follows. Section 2 describes work most related to shape, reflectance and light sources estimation from images. Sections 3 and 4 present our object acquisition and reconstruction systems. Sections 5 and 6 detail our light sources estimation and reflectance properties identification. Section 7 presents the results provided by our method.

2 Previous Work

2.1 Geometry Reconstruction

The literature concerning geometry reconstruction is vast and this section only presents a quick run through the area for the most closely related work.

Stereo vision [11,4] methods use two cameras located close one to another so that images of the object be slightly different. Internal and external camera parameters are used to deduce depth with the help of textures, laser grids or structured light.

Shape from shading methods aim at recovering objects shape with the assumption that surfaces are lambertian (or almost lambertian) [38,17].

Shape from silhouette approaches [27,5] are more adapted to our problem since we do not want to make any assumption: objects can have lambertian or glossy surfaces, with or without textures; we do not want to use active laser grids or test patterns to reconstruct the object geometry. Most shape from silhouette methods rely on a voxel-based datastructure. The approach described by R. Szeliski [40] is often used as a basis. However, with such methods, cavities cannot always be reconstructed properly. For instance, a coffee cup handle will be recovered since the hole can be deduced from the visual hull on some images, but the inside will be filled-up with material (unless there is no bottom). Improvements have been proposed for solving this problem with voxel coloring [37], space carving [20] or gradient voxel flux [10] with the assumption that objects surface is mostly lambertian.

From voxels, the marching cubes algorithm can easily generate a triangular mesh corresponding to the object surface [39]. For example, Hasenfratz et al. use such a reconstruction method for interactively integrating a real person into a virtual environment [13]. As explained in [29], the object shape accuracy can be improved with the help of image pixels. Our global framework makes use of this latter method, summed up in the following section.

2.2 BRDF Estimation

Various models have been proposed for surface reflectance properties [19,15,42,31,21], providing reasonable approximations of many real materials, such as metals, plastics, anisotropic surfaces, etc.

Sato et al. [36,35] recover reflectance properties from multiple photographs of a textured object with known illumination conditions. The object shape is reconstructed, using 12 range images. Then, three terms of a simplified Torrance-Sparrow reflection model are estimated with the help of the reconstructed object shape and 120 color images. Kay et al. [18] describe a method to recover the surface reflectance using the Torrance-Sparrow reflection model. For this method, a depth map is required as well as 8 images obtained with different point light sources.

Some approaches have also been proposed for estimating reflectance properties in real scenes (instead of single objects). Boivin et al. [1] approximate the reflectance properties from a single photograph of a real scene containing untextured surfaces and known light sources. Yu et al. [44] use a geometric model of the scene and a set of calibrated high dynamic range photographs (150) taken for recovering diffuse and specular reflectance parameters (12 photographs specifically to get specular highlights on surfaces) and constructing high-resolution diffuse albedo maps for each surface. Specular reflectance for a surface is estimated only if at least one highlight is visible on the images.

To our knowledge, all the existing methods make the assumption that illumination conditions or some geometry information or both are known. In our case, geometry has to be reconstructed from the initial set of images; with our method, in addition to reflectance properties, light sources are automatically estimated.

2.3 Light Sources Detection

Starting with the pioneering work of Pentland [32], many techniques have been developed to estimate light sources properties from one single image. One single directional light source can be estimated with the assumption that the viewed scene represents a convex object with sharp contours [43, 41]. A directional light source can also be estimated with the additional help of shadows [34,30]. Guillou presents a method for estimating several point or directional light sources, with the help of depth and reflectance maps [12].

Some authors have favored the use of several images for estimating multiple directional light sources. For example, Hougen et al. [16] solve a system of constraints while Zhou et al. [47] use a pair of stereo images.

In [8], Debevec proposes a method for acquiring a radiance map with photographs of a spherical surface mirror, such as a polished steel ball. Powell et al. [33] have also used specular spheres to estimate the position of several point light sources from a set of images while lambertian spheres are used in [45].

In our work, the object is only represented by a series of calibrated images. We do not want to make any assumption about the surface reflectance properties. Moreover, neither the position/direction nor the number of light sources is known in advance.

3 Acquisition System

Our method applies to a set of images representing an object without cavity. For each image, we make the assumption that the camera position and orientation are known. In practice, we use as well synthesized images and photographs of real objects. Virtual objects are convenient for validating the method and providing good quality results since camera parameters and object geometry are precisely known. For real objects, we devised the acquisition system described in Figure 3.1.

Object and light sources are fixed on a turntable: a camera is located on a tripod with fixed focal length, aperture and shutter speed. The acquisition process consists in turning the object and taking two photographs every 5 degrees. The first one is overexposed with an additional light source for separating the object from the background (a lambertian black cloth) while the second one is used for acquiring the actual object radiance (see images on Figure 3.2).

During one turn, the camera position does not change. After one turn, the camera is raised of several centimeters. In practice, only 1 turn (72 viewpoints) is necessary for recovering the object shape. This system has also been used for acquiring complete image-based objects: several turns are also used for estimating light sources position.

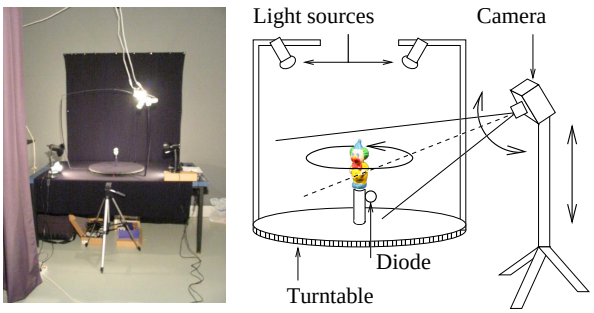


Fig. 3.1 Acquisition system: the camera is placed on a tripod and the object/light sources are fixed on a turntable.

3.1 Image Preprocessing



a. *b.*

Fig. 3.2 a. Overexposed image using an additional light source for segmentation; b. Image used for reflectance and light sources estimation.

We did not use any test pattern for estimating camera parameters. Instead, the camera position is manually measured. It is placed on a tripod at about 1.5 meters of the object.

The camera orientation is set to the rotation axis of the turntable. Therefore, the object is placed on a cylindric base at the center of the turntable; when the table is turning, the base should remain at the same position on the camera screen (up to a pixel). Since the camera is placed on a tripod, the distance to the object remains the same and the focus does not change. The red diode is used only in the first image for estimating the camera elevation angle which remains fixed for one turn.

A global coordinate system is defined: the turntable center corresponds to the origin; the z axis represents the up (vertical) vector; the camera is placed on the $(0yz)$ plane for the first photograph, thus defining the y axis. The coordinate system is associated with the turntable so that the camera is in effect turning around the object.

Finally, the camera position is estimated in this coordinate system: (i) the horizontal and vertical distances between the turntable center and the camera are measured once and for all (y and z values) and (ii) a rotation matrix is computed every time the table is turned (every 5 degrees, using a mark on the table).

Even though the center of projection is not perfectly measured, the camera does not move. Thus the error is only relative and only tends to modify the absolute object size of less than 1 millimeter.

4 Geometry Reconstruction

4.1 Overview

Before estimating light sources and reflectance properties, a preprocessing step recovers a voxel-based geometry of the object, using a *shape from silhouette* approach. With the help of image pixels, we adapt the *marching cubes* algorithm for providing a polygonal surface. Then, a normal is estimated for each voxel either with the marching cubes triangles or with a method using only voxels. This section recalls the

reconstruction approach proposed in [29] in order to set the notations used in the following sections.

4.2 Shape From Silhouette

4.2.1 Octree Generation

As a first approximation of the object geometry, we used the shape from silhouette approach proposed in [40]. With this approach, all the images are used iteratively to hierarchically sculpt the object. The shape initially corresponds to one voxel. It is recursively refined: for each viewpoint, the voxels obtained so far are projected onto the image plane and compared to image pixels. They can be seen *inside/outside* the object or *ambiguous* i.e. containing the object surface (Figure 4.1). Only ambiguous voxels are subdivided into 8 sub-voxels. The process is repeated until no ambiguous voxels exist or a minimum size criterion has been reached.

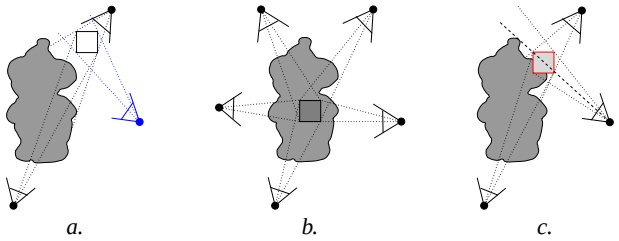


Fig. 4.1 Voxel seen ; a. outside the object; b. inside the object; c. ambiguous.

4.2.2 Practical Aspects

In practice, we use ray tracing for projecting voxels onto the image plane. Each pixel corresponds to a ray originating from the image center-of-projection and going through the pixel (Figure 4.2). These rays are used for (i) improving the marching cubes reconstruction algorithm and (ii) estimating light sources and reflectance properties. They are called *pixel-rays* from now on. Pixel-rays corresponding to the object silhouette are called *in-rays* since they hit the object and rays corresponding to background are called *out-rays*.

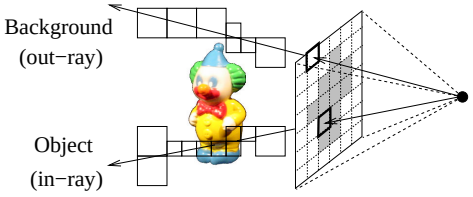


Fig. 4.2 Pixel-rays.

4.3 Pixel-rays for Marching Cubes

The set of resulting ambiguous voxels is called the *discrete surface*. For marching cubes, this surface needs to be 6-connected which is not ensured by the previous algorithm. To achieve this goal, we propose to modify the discrete surface with an additional process applied every time ambiguous voxels are subdivided. A final operation reduces the surface thickness while keeping a 6-connection. More details can be found in [29].

4.3.1 Original Marching Cubes

For reconstructing a triangular mesh from a set of ambiguous voxels, marching cubes [26] are well-suited. For each voxel, vertices are classified as inside or outside the object surface (Figure 4.3.a). When an edge has a vertex inside *and* a vertex outside the object, it is intersected by the object surface. In some cases, the intersection point can be estimated and triangles are placed inside voxels using one of the 14 possible configurations. When the intersection points cannot be estimated, the edge centers are used for generating triangles. However, as explained in the following, we propose to use out-rays with marching cubes so that triangles fit the model shape more precisely. Consequently, the surface normal estimation is also more precise for recovering light sources and reflectance properties.

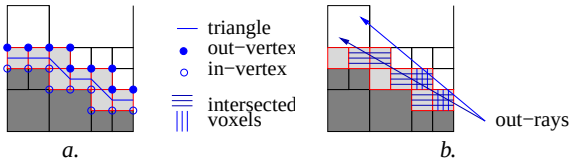


Fig. 4.3 a. Mesh reconstruction using edge centers; b. out-rays can be used for refining the mesh.

4.3.2 Refining the Method

When a out-ray intersects several ambiguous voxels, it is used in all intersected voxels for helping the triangles construction (see Figure 4.3.b).

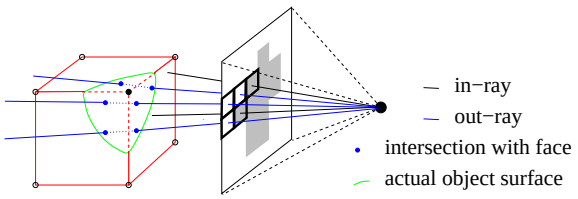


Fig. 4.4 Out-rays should not touch the surface inside an ambiguous voxel; in-rays are not used.

Our algorithm firstly computes the intersection points between out-rays and voxel faces (Figure 4.4). Then, a line

corresponding to the object surface is estimated on each face. Finally, the intersection between this lines and the voxel edges defines the points used for marching cubes.

For two adjacent faces (resp. voxels), the corresponding surface lines have to be connected as shown in Figure 4.5.a (resp. 4.5.b).

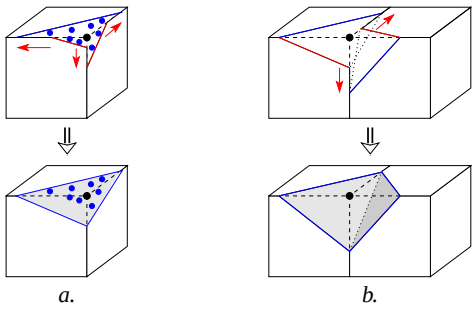


Fig. 4.5 Surface continuity.

4.4 Surface Normal

For estimating light sources properties and the object BRDF, we need to define a normal for each surface voxel. This normal has also to be used for rendering new images of the object.

Inside each ambiguous voxel, several triangles define the object surface. Using triangles located in only one voxel leads to bumpy normals, introducing artifacts for light sources estimation or during the rendering process. This is why we propose to smooth normals according to neighbor voxels triangles. In our application, a parameter called *smoothing distance* is fixed by the user.

A normal estimated from the generated triangular mesh is quite representative of the object surface. However, it is also possible to define a normal in each voxel directly with the discrete surface [29]. Particularly, if a surface mesh is not needed, acceptable results can be obtained.

5 Light Sources Detection

5.1 Overview

We use the reconstructed geometry information for developing our light sources detection methods.

Each pixel of an image corresponds to the radiance emitted from the object to the camera (from one voxel to the camera). Each voxel is seen from several viewpoints; thus, for each voxel it is possible to store the set of radiances associated with pixels (we use at least 60 per voxel).

We distinguish two types of light source (point and directional) and two types of surface (diffuse and glossy). The broad lines of our light sources estimation algorithm are the following:

- classify voxels into regions according to hue and orientation;
- for each region, estimate the type of surface (diffuse or glossy);
- for each region, search for a point light source;
- for each region, search for a directional light source;
- for each region, identify parameters for light sources types and positions jointly with surface properties;
- use regions altogether to validate light sources positions / directions and surface properties with an error estimate.

With a glossy surface, when the error produced is greater than a user-defined threshold, light directions can be found using specular highlights seen on images.

For a single object, interreflections due to the environment are negligible (dark environment). However, diffuse or specular self-interreflections sometimes disturb the BRDF estimation. This paper does not solve the problem.

5.2 Notations and Assumptions

For real-world light sources, light bulbs can be very different according to geometry, wavelength range, power, etc. In practice only a few types of light sources are commonly estimated from images or used for virtual environments: spots, surface, directional or point light sources. In this paper we are interested in recovering point light sources and directional light sources. A point light source is described by a 3D point and its power while a directional light source is described by a 3D unit vector and its associated power.

For a surface element dA around a point x, the radiance corresponds the power per unit projected area per unit solid angle in a given direction $\overrightarrow{\omega}$:

$$L(x,\overrightarrow{\omega}) = \frac{d^2\Phi}{\cos\theta_r dA d\omega}$$

where Φ corresponds to light flux, θ_r is the angle between surface normal and outgoing direction, $d\omega$ is the corresponding solid angle.

Radiance also varies according to incoming light, surface orientation and object intrinsic properties: its BRDF or Bidirectional Reflectance Distribution Function. The BRDF represents the ratio between the incoming and the outgoing flux:

$$f(x,\overrightarrow{\omega_i},\overrightarrow{\omega_r}) = \frac{dL_r(x,\overrightarrow{\omega_r})}{L_i(x,\overrightarrow{\omega_i})cos\theta_i d\omega_i}$$

where L_i corresponds to radiance impinging on the surface from the direction $\overrightarrow{\omega_i}$ and L_r is the radiance reflected in the direction $\overrightarrow{\omega_r}$. There is a wealth of literature concerning BRDF models. Each model can be physically correct or not with respect to energy conservation law, Helmholtz reciprocity or Fresnel equation [25,6,42]. Various types of surfaces have been represented: with specular aspect (Figure 5.1), anisotropic effects, microfacets, etc.

For a surface lit by a light source S from a given direction $\overrightarrow{\omega_i}$, the reflected radiance $L_r(\overrightarrow{\omega_r})$ corresponds to the product

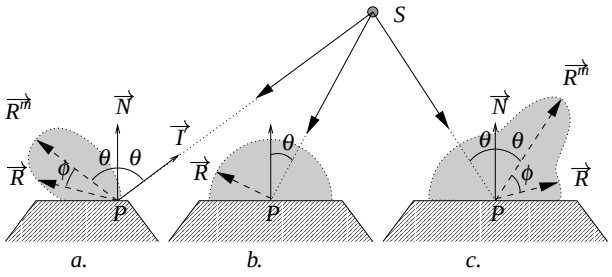


Fig. 5.1 Light flux reflected by a surface; *S* corresponds to a point light source; a. with a specular surface; b. with a perfectly diffuse surface; c. with a glossy surface.

$f(\overrightarrow{\omega_i}, \overrightarrow{\omega_r}).L_i(\overrightarrow{\omega_i})cos\theta_i d\omega_i$. In our case, neither f nor L_i are known, the problem is ill-posed. However, if we restrict incident radiance to directional and point light sources, it is possible to estimate f and light source properties up to a factor (actually corresponding to light source emittance, denoted by L_s in the following).

In this work, for f we choose the *modified Phong* model proposed in [25] since it (i) is physically plausible, (ii) represents diffuse and glossy surfaces, (iii) requires only 3 coefficients. Using this model, a specular lobe is located in the mirror direction of the incoming light. Thus, we do not deal with anisotropic surfaces or off-specular peaks. With more complex surface properties, it could be possible to fit the parameters of another model such as [42,21]. However, the number of samples has to be much denser and the fitting is very sensitive in most cases. According to the modified Phong model, reflected radiance (Figure 5.1.c) is expressed as:

$$L_r = \frac{L_s K_d}{\pi r^2}\cos\theta + \frac{(n+2)L_s K_s}{2\pi r^2}\cos\theta\cos^n\phi$$

where L_r is the radiance reflected from P to $\overrightarrow{R}$; L_s is the radiance emitted by a light source S, arriving at P; r is the distance between S and P; K_d and K_s are respectively diffuse and specular coefficients for reflection (between 0 and 1); n defines the specular lobe size; θ is the angle between the normal $\overrightarrow{N}$ of the surface and the direction $\overrightarrow{I}$ of the light source (incident direction); ϕ is the angle between $\overrightarrow{R^m}$ (mirror reflection direction) and $\overrightarrow{R}$.

5.3 Voxels Classification

The object description (geometry and reflectance properties) as well as the high number of images make the problem difficult to solve in one go. Let us recall that our algorithm is based on a geometry reconstruction producing a set of more than a hundred thousand voxels. Ideally, when all the voxels have the same reflectance properties and are lit by the same, unique light source, the problem is easier to solve. In this paper we do not make this assumption; the surface can be made up with different types of materials and lit by several light sources. Note that for one object, finding a high number of light sources (more than 5 or 6) does not make sense since the object is almost uniformly lit.

To simplify the light sources detection, it is necessary to group voxels according to reflected radiance and surface orientation. This classification process is composed of two parts. The first one identifies regions of voxels having the same hue (chrominance) while during the second part, voxels of each region are classified according to their normal orientation.

5.3.1 Reflectance-Based Classification

Let us recall that for a given voxel V, the set of radiance samples (denoted by L) corresponds to the set of pixels viewing V. We use the radiosity value for discarding voxels in the shade. For each voxel, we estimate the radiosity value (light flux emitted per unit surface):

$$R = \frac{d\Phi}{dA} = \int_{\Omega=2\pi} L_r(x, \overrightarrow{\omega}).d\omega = \frac{2\pi}{NS}\sum_{j=0}^{N} L_j$$

where $(\sum_{j=0}^{N} L_j)/N$ is the average radiance emitted by the surface inside a voxel, S is the surface area contained in a voxel.

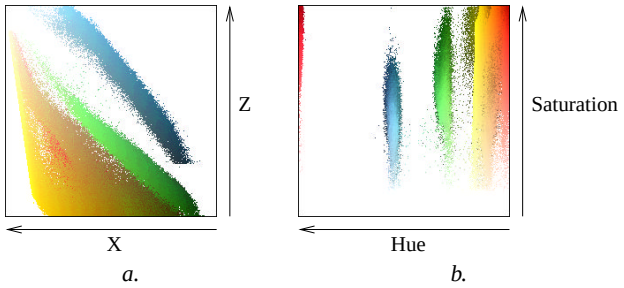
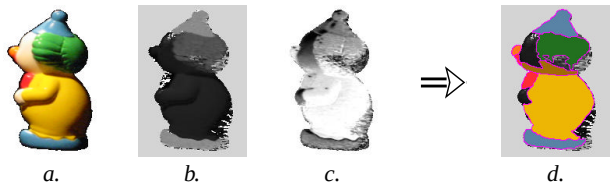


Fig. 5.2 Chrominance-based classification on for a real object photographs after having removed low radiance samples: a. with X-Z values (XYZ color-space), it is difficult separate colors; b. with Hue-Saturation values (HSV color-space).

In practice, we suppose that S is constant for all the voxels. A low radiosity value indicates the corresponding voxel is in the shade. However, acquired images are composed of RGB values, unadapted to precisely estimate radiance, radiosity and chrominance. This is why we choose to translate the color of each pixel into two different color-spaces: HSV (Hue, Saturation, Value) and xyz. Experimentally, compared to other color-spaces, HSV provide better results for our test objects: see Figure 5.2.

Hue and *saturation* express chrominance while *y* is used for radiance. In the general case, hue can be assumed constant whichever incoming flux power; the saturation value indicates hue reliability. Voxels with low saturation and radiosity values (lower than a given threshold) are then dismissed and the classification process is performed according to the *hue* value in HSV space (see Figure 5.3).

This classification process allows:



a. *b.* *c.* *d.*

Fig. 5.3 Images of the object: a. with RGB values; b. with Hue values; c. with saturation values; d. first classification results. Voxels representing a white or dark surface are not classified (thus not used for detecting light sources).

– to match voxels having the same hue
– to dismiss, for each voxel, some in-rays that does not correspond to the voxel average hue.
– to identify bright in-rays that are used for detecting light sources and estimating the specular component of the surface.

5.3.2 Normal-Based Classification

For each class previously defined, voxels having the same orientation are grouped so that they be likely lit by the same light source. Before this processing, voxels with several specular lobes are ignored: we denote by D^{max} the direction corresponding to the highest radiance sample L^{max}; we account for all the radiances $L > 0.9 \times L^{max}$ which direction is denoted by D; if the angle between D and D^{max} is greater than 30 degrees we consider that there is more than one specular lobe. Some voxels are ignored when (i) they contain less than 10 radiance samples or (ii) the angle between D^{max} and the voxel normal is greater than 80 degrees. The classification process provides voxels subclasses (or voxels subregions) with similar normals (about 20 degrees).

5.4 Type of Surface Estimation

We propose two different methods for finding a light source from a class of voxels. The first one applies to diffuse surfaces while the second one applies to glossy surfaces. The type of surface, diffuse or glossy is estimated with the help of a variation coefficient V^{class} computed from radiance samples:

$$V^{class} = \sqrt{\frac{\sum_{i=1}^{NbV} \sum_{j=1}^{NbL_i} \left(\frac{L_{i,j} - L_i^{moy}}{L_i^{moy}} \right)^2}{\sum_{i=1}^{NbV} NbL_i}}$$

where NbV represents the number of voxels in the class, V_i is the i^{th} voxel, $L_{i,j}$ is the j^{th} radiance sample of V_i, and L_i^{moy} is the average radiance of V_i; NbL_i corresponds to the number of radiance samples in the voxel V_i.

This expression derives from the evaluation of a variation coefficient V for each voxel ($V = \sigma / L^{moy}$ with a variance $\sigma^2 = \sum_{j=1}^{NbL} \left(L_j - L^{moy}\right)^2 / NbL$).

For a perfectly diffuse surface, a given voxel emits the same radiance in all directions (see Figure 5.1.b) so that variance equals zero as well as V^{class}. V^{class} increases according to the specular aspect of the surface. Since we make the assumption that a surface is defined by K_d, K_s and n, V^{class} is much sensitive to ratio K_s/K_d. Conversely, n has low incidence on V^{class}. Thus, V^{class} is a good estimator for the surface type. Experimentally:

– if $V^{class} < 0.15$, the surface is considered as *perfectly diffuse*,
– if $V^{class} > 0.30$, the surface is estimated *glossy*,
– if $0.15 \leq V^{class} \leq 0.30$, we reach no decision about the surface type. For such a class we apply algorithms for both diffuse and glossy surfaces.

Later on, as described in Section 5.7, an identification process estimates appropriate values for K_d, K_s and n.

For each subclass defined above, both a point light source and a directional light source are estimated. The choice of the best light source is determined, according to the error between the initial radiance samples (corresponding to pixel values) and recomputed radiances. Let us first describe our method for estimating a point light source.

5.5 Point Source Detection

For each voxel V_i, the direction of incoming light is firstly estimated; the final position of the point light source is deduced from all those directions if the point source is close enough to the surface (Figure 5.4), else the algorithm concludes it is a directional light source.

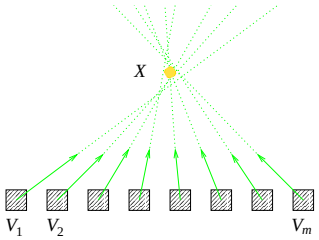


Fig. 5.4 The intersection of all the incident directions defines a position x of the light source.

5.5.1 Diffuse Surfaces

The radiance emitted by a diffuse surface element is constant whichever reflection direction. Since in practice this radiance is not exactly constant, we use the average radiance for all the voxel samples:

$$L_V = \frac{\sum_{j=0}^{NbL} L_j}{NbL}$$

N being the number of radiance samples for a voxel V. This radiance corresponds to the product $L_s K_d \cos\theta_V$. For all the voxels of a given class, L_s and K_d are assumed constant. The emitted radiance only depends on θ, the angle between the normal $\overrightarrow{N}$ of this voxel and the incident direction $\overrightarrow{I}$.

L_V is maximal when $\theta_V = 0$. For each class, the voxel V^{ref} having the highest radiance L_V^{ref} is chosen as a reference for initializing our iterative process: its normal is used as the incident direction $\overrightarrow{I_V^{ref}}$, with $\theta_V^{ref} = 0$ and $L_V^{ref} = L_s K_d$. Consequently, for each voxel V in the class, θ_V can be estimated:

$$\theta_V = \arccos(L_V / L_V^{ref})$$

The estimation of θ_V does not directly provide the incident direction but a cone of directions (Figure 5.5.a). For V, the incident direction belongs to the plane defined by the centers of V, V^{ref} and the incidence direction $\overrightarrow{I_V^{ref}}$ (Figure 5.5.b). The intersection between the cone of directions and this plane gives 0, 1 or 2 incident directions.

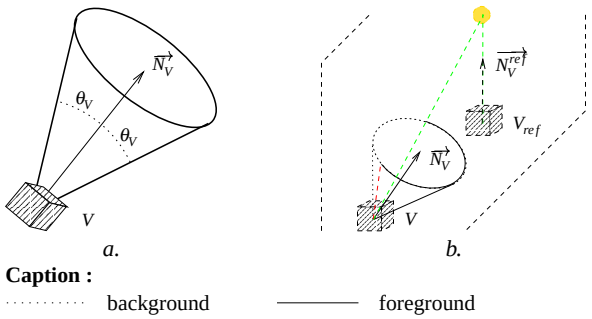


Fig. 5.5 a. Cone corresponding to incident directions (θ_V known); b. intersection between the cone and the plane containing the point light source and voxels.

These solutions are obtained by solving a system of three equations with three unknowns:

$$||\overrightarrow{I_V}|| = 1; \qquad \overrightarrow{I_V}.\overrightarrow{N_V} = L_V / L_V^{ref}; \qquad \overrightarrow{I_V}.\overrightarrow{N_{plane}} = 0$$

Momentarily the algorithm ignores the voxels having two possible incident directions. When a single incident direction is estimated for a voxel, the corresponding line is represented by the intersection of two orthogonal planes (given by $ax + by + cz + d = 0$). The set of these equations is represented in the following matrix form:

$$\begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ \vdots & \vdots & \vdots \\ a_{2n} & b_{2n} & c_{2n} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -d_1 \\ -d_2 \\ \vdots \\ -d_{2n} \end{pmatrix}$$

In this matrix system $MX = D$, X ideally corresponds to the point light source coordinates: $X = M^{-1}D$. Unfortunately, in most cases M is not invertible; instead of M^{-1} we rather estimate a pseudo-inverse matrix M^{pi}.

For each previously dismissed voxels (having two correct incident directions), one of the two directions (the most consistent with the estimation of X) is added to the matrix system. A new estimation of M^{pi} and X is evaluated.

However, the quality of this estimation highly depends on the choice of V^{ref}: the normal of V^{ref} is rarely perfectly aligned with the light source direction. Our iterative algorithm refines the solution with the help of the following error criterion:

$$E_d = \sum_{i=1}^{2m} (M_i X - D_i)^2,$$

where M_i is the i^{th} row of the matrix M, and X is the estimated position of the light source; m corresponds to the number of incident directions (m lines are defined by $2m$ equations of planes). E_d provides the final result significance. Selecting a different voxel for estimating the product $L_s K_d$ can reduce E_d (experimentally, from the voxel class barycenter, the incoming light direction D_e is correct but the light source position is too close). We choose V_e the voxel located on D_e (see Figure 5.6); the value of its radiance is $L_e = L_s K_d cos(\theta_e)$, where θ_e is known.

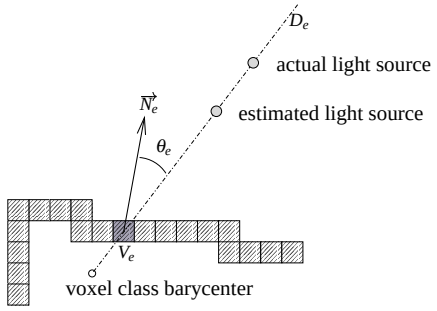


Fig. 5.6 Choice of a new voxel V_e for estimating point light source.

To improve the estimation of $L_s K_d$, this process is repeated as long as the estimated error E_d decreases. This last process significantly reduces E_d. Finally, with this method, light source estimation is precise at about one centimeter when the distance between the object and the light source is lower than or equal to 9 meters with a 1-meter diameter object.

5.5.2 Glossy Surfaces

For non-lambertian surfaces, the specular lobe can be used to estimate more precisely the light source position; the incident flux L_s is reflected by the surface in the mirror direction $\overrightarrow{R^m}$. We propose to estimate $\overrightarrow{R^m}$ for each voxel and deduce the light source direction.

Unfortunately, the assumption that $\overrightarrow{R^m}$ is defined by the highest radiance sample $\overrightarrow{R^{max}}$ cannot be made (see Figure 5.7.a). The specular lobe (around the mirror direction $\overrightarrow{R^m}$) is defined by several radiance samples. We propose to represent the lobe as a curved surface. The corresponding coefficients are identified according to radiance samples. The most obvious solution is to use the cosine function appearing in the *modified Phong* model. However, depending on the coefficients, cosine oscillations could introduce bias in the

estimation. With equivalent results for $\overrightarrow{R^m}$, a parabolic surface facilitates the estimation. The equation of such a surface is:

$$L_{\alpha,\beta} = a(\alpha - \delta_\alpha)^2 + b(\beta - \delta_\beta)^2 + c$$

¿From this equation, the mirror direction $\overrightarrow{R^m}$ is defined by $(\delta_\alpha, \delta_\beta)$ (see Figure 5.7.b).

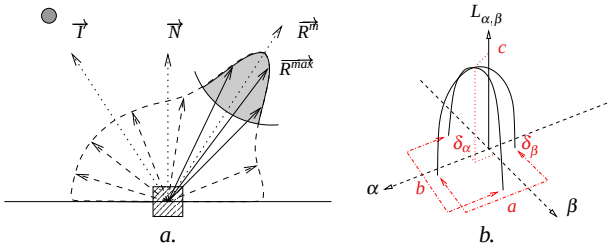


Fig. 5.7 a. Radiance samples for a voxel, the grey tint contains radiance samples of the specular lobe; b. radiance emitted by a surface point according to the angles α and β (polar coordinates of the reflected direction).

Coefficients a, b, c, δ_α and δ_β are identified using a set of radiance samples. Therefore, the algorithm picks the samples around $\overrightarrow{R^{max}}$ with a maximum of 20^o (Figure 5.7.a). All the coefficients are then estimated with a *gradient descent* method on an error corresponding to the square difference between (i) each radiance sample and (ii) the recomputed radiance for the same direction.

This process is applied to each voxel, providing the light source direction. As in Section 5.5.1, these directions are stored in a matrix system and the light source position is estimated with a pseudo-inverse matrix.

5.6 Directional Source Detection

In this analysis, we also distinguish diffuse surfaces from glossy surfaces, with the assumption that the incident direction is the same for all the voxels in a subclass.

5.6.1 Diffuse Surfaces

The radiance emitted by a voxel V is the same for all emitting directions, it is expressed as:

$$L_V = L_s K_d \cos(\theta_V) = L_s K_d (\overrightarrow{I}.\overrightarrow{N_V})$$

where $\overrightarrow{N_V}$ is the surface normal inside V and $\overrightarrow{I}$ is the searched incident direction. In this equation, L_s, K_d, and $\overrightarrow{I}$ are the same for all the voxels of a class. Again, this system can be described in a matrix form $MX = D$, where X represents the product $L_s K_d \overrightarrow{I}$, D is a vector containing L_V values and M corresponds to the set of vectors $\overrightarrow{N_V}$. As above, it is solved with a pseudo-inverse matrix.

5.6.2 Glossy Surfaces

As for point sources detection, specular lobes can be used. Once a direction has been estimated for each voxel (for a given class), the light source direction is defined as the average of all estimated directions.

5.7 Joint Identification

Each light sources estimation algorithm is independently applied for each voxel subclass, providing one point light source and one directional light source (let us recall that when a surface could not be surely classified as diffuse or glossy, four analysis are performed). For each analysis, we estimate an error E_a:

$$\sum_{i=1}^{NbV} \sum_{j=1}^{NbL_i} [(\frac{L_s K_d}{\pi r^2} \cos\theta_i + \frac{(n+2) L_s K_s}{2\pi r^2} \cos\theta_i \cos^n \phi_{i,j}) - L_{i,j}]^2$$

where $L_{i,j}$ corresponds to the radiance sample j of the voxel V_i; the parameters $L_s K_d$, $L_s K_s$ and n are unknown. We apply an identification algorithm with the help of a gradient descent method in order to both ameliorate the parameters estimation and reduce E_a. The parameters actually chosen for the light source and surface properties are those leading to the lowest value of E_a.

For each class, this analysis finally defines (i) the type of light source (point or directional), (ii) its position or direction, (iii) the type of surface (diffuse or glossy), and (iv) the surface reflectance parameters (diffuse and specular reflection parameters). Note that this parameter fitting does not modify the surface normal.

A final step groups the detected light sources according to their types and position/orientation, so as to keep the error E_a as low as possible.

5.8 Glossy Surfaces and Highlights

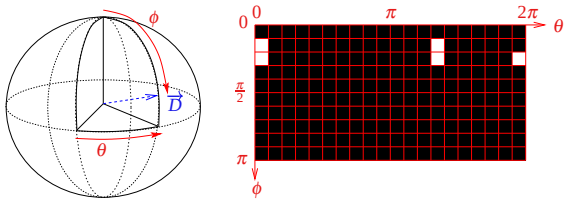
Our light sources estimation method is more accurate with diffuse surfaces than with glossy objects. For instance the light sources used for the glossy clown shown in Figure 7.4 were recovered with an error of about 16 degrees. Conversely, for the (close to diffuse) wood flower, the error is less than 11 degrees (Table 7.6, rows *general method*). This problem is essentially due to the lack of radiance samples for each voxel. When only 4 or 5 radiance samples are used for estimating the specular lobe direction, the incoming direction is not reliable. Cumulated with the normal estimation error due to the reconstruction process, the light source direction estimated for each voxel is not always very accurate.

Instead of using only the *object-space* for estimating light sources direction, we propose to use additionally the *image-space* information. With images, specular highlights correspond to gloss reflected by several voxels. Thus, a more reliable information can be used additionnally. This global approach allows: (i) to use voxels lit by several light sources

and (ii) to estimate a light source direction only for relevant groups of voxels.

We propose to estimate a set of directional light sources with the help of highlights seen on all the images. Highlights are defined by overexposed pixels, with very high saturation.

For each detected specular highlight, the corresponding 4 or 5 pixels are projected onto the voxel geometry. For each intersected surface voxel, the light source (mirror) direction is expressed in spherical coordinates, see Figure 5.8.a. All the images provide a set of directions, corresponding to one or several directional light sources. These directions can be grouped in classes for determining a list of final light sources with their parameters. Therefore we apply a K-means algorithm on the set of directions with the angle difference as a distance criterion. Unfortunately the number of light sources is not known in advance.



a. *b.*

Fig. 5.8 a. Spherical coordinates: a direction is represented by 2 angles $\theta(0..2\pi)$ and $\phi(0..\pi)$; b. set of directions represented on a cylindrical image with θ and ϕ axes: 2 directions appear. In order to have a homogeneous repartition of directions in the image, a weight is associated with each pixel according to solid angle.

For automatically estimating this number, we store the set of directions in an image where axes correspond to θ and ϕ angles (see Figure 5.8.b). A segmentation algorithm provides the light sources number and a first estimation of their directions used as a seed for the K-means algorithm.

6 Final Reflectance Properties Estimation

From the set of light sources obtained so far, the object reflectance properties are further refined. Voxels lit by the same unique light source are grouped according to reflectance properties. A final identification algorithm provides the BRDF coefficients, using the error E_a. For reducing noise and grazing angles imprecisions, we use only radiance samples corresponding to a small incidence angle ($\theta \leq 45^o$, see Figure 6.1).

7 Results

Some images and videos can be found on the web pages dedicated to our project:
www.sic.sp2mi.univ-poitiers.fr/ibr-integration/index_en.html
www.sic.sp2mi.univ-poitiers.fr/ibr-integration/ijcv.html

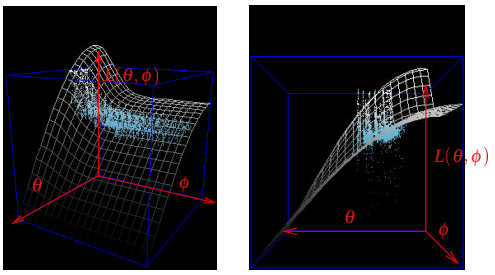
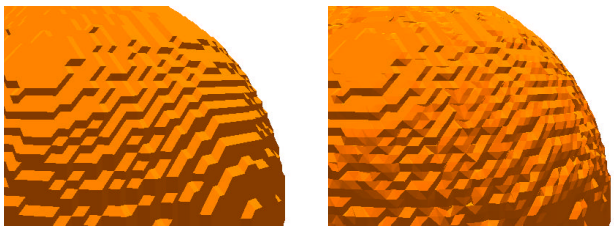


Fig. 6.1 Estimated BRDF together with actual radiance samples, using radiance samples with $\theta \leq 45^o$.

7.1 Geometry Reconstruction

7.1.1 Object Shape

Before actually using our method with real objects, validation has been done with known objects (a sphere and a cube). As shown in Table 7.1 and Figure 7.1, using image pixels with marching cubes (MC in the Table) improves substantially the shape accuracy.



a. *b.*

Fig. 7.1 Rendering using triangles; a. with *edge centers* marching cubes; b. with pixel-rays marching cubes.

When the hierarchy depth increases, the two methods tend to provide the same results because the number of pixel-rays becomes lower in each voxel. Our method will obviously be more accurate with a low-depth hierarchy.

Table 7.1 Average distance between the reconstructed triangles and the object surface (used images resolution: 256x256).

For a 1-meter diameter Sphere			
Octree depth	5	6	7
Edge center MC	*10.8 mm*	*9.3 mm*	*8.9 mm*
Pixel-rays MC	*6.4 mm*	*6.7 mm*	*6.7 mm*
For a 1-meter width Cube			
Octree depth	5	6	7
Edge center MC	*26.4 mm*	*19.4 mm*	*16.7 mm*
Pixel-rays MC	*21.2 mm*	*19.1 mm*	*18.6 mm*

Note that the cube is the worst case for the type of reconstruction. Faces are always difficult to reconstruct with a shape from silhouette approach (as any flat surface) since the camera viewpoint is never perfectly located on the polygon

plane. For the last row and last column in table 7.1, the triangles generated with the edge centers method are aligned "by chance" with the actual cube. Consequently, the generated triangles have a more suitable normal.

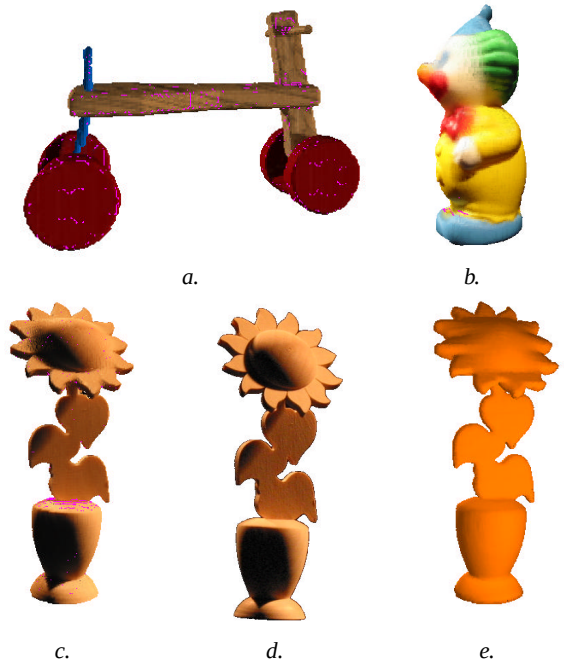


Fig. 7.2 Images of reconstructed objects: triangle color corresponds to the average radiance for each voxel; a. for a virtual quad; b. for a real clown (toy); c. for a real wood-flower; d. actual photograph; e. relighting with lambertian model and modified lighting.

7.1.2 Normal Estimation

For estimating normal quality, we have compared the estimated normal with the actual (known) surface normal (Table 7.2)

Table 7.2 Average angles difference (degrees) between the estimated normal and the object normal.

For a 1-meter diameter Sphere			
Octree depth	5	7	
Smoothing distance	1	1	3
Discrete normal	*11.1°*	*12.1°*	*2.3°*
Edge center MC	*5.7°*	*6.3°*	*2.1°*
Pixel-rays MC	*3.3°*	*4.9°*	*1.8°*

A surface normal obtained with the help of pixel-rays is sensibly more precise (20-25%) than with edge centers marching cubes or a discrete surface. Our method generates an error of less than 5 degrees even without any smoothing. When smoothing, normal precision is about one degree (with a smoothing distance of 5 voxels). Note that with a discrete surface, the smoothing distance should not be less than 3 voxels. New views of virtual and real objects are presented in Figure 7.2.

7.2 Light Sources Detection

7.2.1 From Randomly-Generated Samples

For validating our method, a set of voxels lit by only one light source have been randomly placed. Radiance samples directions and a normal have also been randomly generated for each voxel with the known light source and surface properties (L_s, K_d, K_s and n). Then we have applied our light sources detection algorithms with the assumption that light source and surface properties are unknown, only geometry and radiance samples are used. As shown in Table 7.3, our point light source detection method is more precise for diffuse surfaces. Nevertheless it fits any surface type and automatically estimates a directional light source when the point light source is too far away. Moreover, the estimation of E_d (see Section 5.5.1) has actually been used to reject inaccurate point source detection (for a glossy surface). E_d increases with the distance between light source and object. Note that in the case of a directional light source, direction and surface coefficients (L_sK_d, L_sK_s and n) are always very precise.

Table 7.3 Point light source detection with a 1-meter diameter object.

Surface type	*Object-source distance*	*Estimated final source*	*Error on*		
			source pos/dir	L_sK_d L_sK_s	n
diffuse	$0-9m$	point	$< 1cm$	1%	×
	$> 9m$	directional	$< 1^o$		
glossy	$0-2m$	point	$< 15cm$	1%	5%
	$6m$	none	$1m$		
	$> 6m$	directional	$< 1^o$		

7.2.2 From a Series of Images

Before applying light sources detection methods, we have to estimate the object geometry from images. For virtual objects, the results presented below have been obtained with a series of more than 300 images stored as lightfields (see [23] for more details). Each object is lit by two directional light sources and one point light source placed at 35cm from the object.

Virtual Objects with Known Reflectance Properties
We applied our detection method on a sphere for different surface reflectance properties. Images have been generated with our ray tracing algorithm, so that we can compare the recovered light sources with the actual ones (Table 7.4).

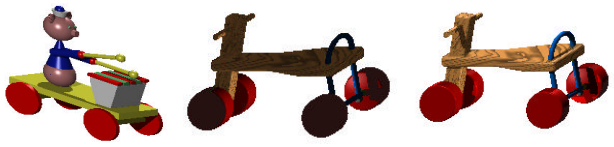
Light sources detection is very sensitive to geometry reconstruction. Nevertheless, our method has proven robust: multiple directional light sources have been recovered with

Table 7.4 Multiple light sources detection with a 1-meter diameter sphere (point light source - object distance: 35 cm; 7-depth octree; smoothing distance: 3 voxels).

Surface type	*Error on*			
	light position	*light direction*	L_sK_d L_sK_s	*n*
diffuse / glossy	$< 4mm$	$< 1^o$	$< 10\%$	$< 20\%$

an 1-degree inaccuracy whereas the surface normal is estimated with an 1.8-degrees inaccuracy (see Table 7.2).

Virtual Objects with more Complex Properties
We haved also used objects with a more complex geometry and composed of several surface types with textures (see Figure 7.3). Images have been generated with the PovRay software in order to use various reflectance properties different from Phong's illumination model.



a. *b.* *c.*

Fig. 7.3 Images of some virtual objects we used (resolution: 256x256); b. quad lit by only one point light source (and an ambient term); c. quad lit by one point and two directional light sources.

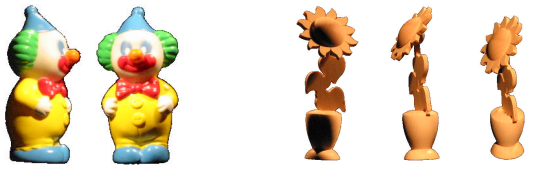
The results written in Table 7.5 have been obtained with objects lit by three light sources (as in Figure 7.3.c).

Table 7.5 Multiple light sources detection with 1-meter width virtual objects (point light source - object distance: 35 cm).

Surface type	*Error on*		*Unexpected light sources*
	light position	*light direction*	
diffuse / glossy	$< 5cm$	$< 5^o$	0 - 1 directional

The results are correct even though surfaces are textured (glossy wood) and some objects edges are curved and chamfered. Directional light sources are accurate to within 5 degrees but reflectance properties are less precisely estimated: values can be up to twice higher/lower than actual values. For removing useless light sources, the radiance of voxels is re-estimated for every light source. When the estimated radiance is not coherent with the radiance samples, the associated light source is automatically discarded.

Real Objects with Small Cavities
We used our acquisition system (see Section 3) for generating calibrated images of real objects: Figure 7.4. For the clown, 72 photographs have been acquired with one complete turn of the acquisition table. Our detection method has been applied to these 72 images. We have chosen to restrict the geometry reconstruction to a 7-depth octree in order to have enough radiance samples in the voxels. For the wood flower, we used 504 photographs acquired with seven complete turns of the acquisition table. We used an octree depth equal to 8 for the shape from silhouette reconstruction.



a. *b.*

Fig. 7.4 Photographs of real objects lit by two spots (object-sources distance: 1 m); a. clown (height: 8 cm; images resolution: 256x256); b. flower (height: 12 cm; images resolution: 512x512).

For these two real objects, light sources detection results are presented in Table 7.6.

Table 7.6 Two light sources detection with real objects (spot light sources - object distance: 1 m).

Object	*Detection algorithm*	*Error on sources direction*	*Unexpected light sources*
Clown (glossy plastic)	general method	12^o&16^o	2 directional
	specular highlights	7^o&11^o	0
Flower (diffuse wood)	general method	6^o&11^o	1 directional
	specular highlights	11^o&17^o	0

According to the objects size and the light sources position, each light source can be seen as a spot emitting in a 7 degrees cone. Thus, light directions can only vary of 7 degrees on the object surface. This is why light sources are considered as directional by our algorithm.

For both the clown and the flower, we obtained an estimation of each directional light source with an error of less than 11 degrees. These results have been obtained despite small unreconstructed cavities. The general method provides better results for the flower since the surface is nearly diffuse. For the clown, indirect lighting also corrupts radiance estimation. However, the method using specular highlights has provided correct results.

7.3 First Relighting Results

Geometry, surface reflectance, light sources and photographs are combined for each voxel. Such information can be used for relighting objects as realistically as possible.

Figure 7.5.b shows that an image of the clown (re)shaded by the set of detected light sources is close to the photograph (figure 7.5.a), except for the silhouette. Some specularity information has also been lost, essentially due to saturation

of radiance values on the original photographs. Figure 7.5.c represents the same reshaded image with an octree depth equal to 8. Note that the specular highlights are sharper but some noise appears in the classification process due to the low number of radiance samples in voxels. High dynamic range images with higher resolution, could also provide better highlights at the expense of a higher acquisition time and disk space. If we reduce one light source radiance (see figure 7.5.d) or add a new light source (see figure 7.5.e and 7.5.f), the new synthesized images remain realistic.

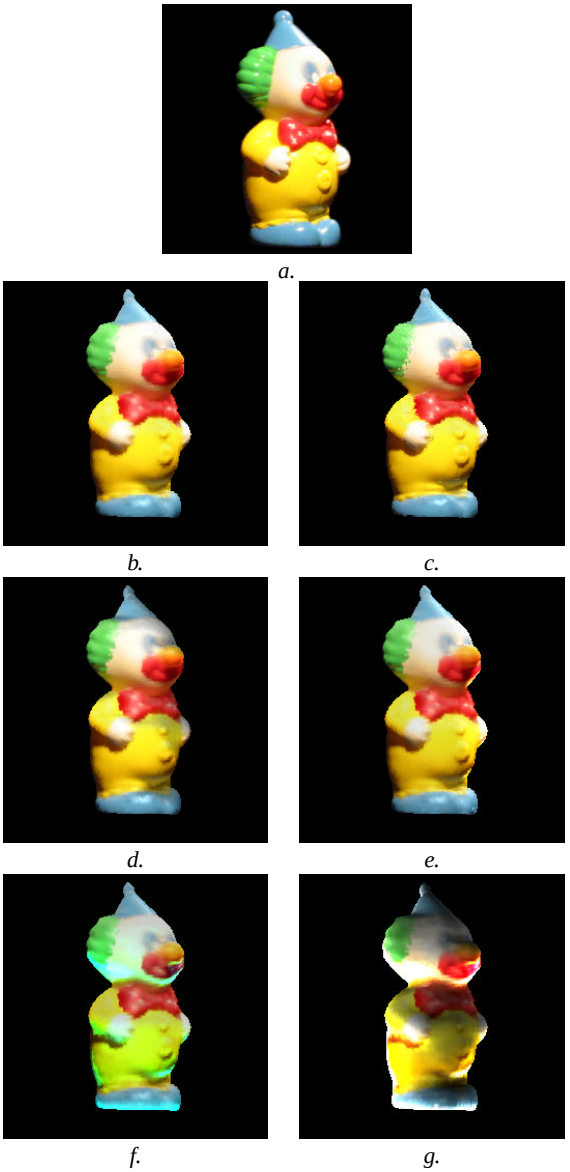


Fig. 7.5 Relighting of a real object, using an octree depth equal to 7; a. photograph; b. reshading using the two detected light sources; c. same image with a 8-depth octree, note that the specular highlights are sharper but some noise appears around the hair; d. reshading with only one light source; e. reshading with an additional white light source; f. reshading with an additional cyan light source; g. reshading without any powerful light source facing the object, note that the specular highlights on the right feet and bow tie have disappeared.

Visual artefacts can appear in some cases: for example, some light sources can be estimated for regions actually in the shade. Then, when relighting the object, incoherencies between regions can appear on the image for some viewpoints.

7.4 Computing time

We used a Dual Intel Xeon 2.4GHz processors with 1GB of RAM. Table 7.7 shows computing time and memory cost for the use of 72 images representing the clown. Acquiring the 72 images required about 30 minutes.

Table 7.7 Computing time and memory cost for all the process with a series of 72 photographs representing the clown.

Clown reconstruction			
	Octree depth	7	8
Precomputation time	*Geometry reconstruction (octree, mesh, normals)*	*14'30"*	*27'00"*
	Light sources and BRDF estimation	*32'30"*	*6'30"*
Relighting time	*with one light source (im. resolution: 256x256)*	*3"*	*6"*
	for each additional light (im. resolution: 256x256)	*+1.5"*	*+3"*
Number of voxels		*13 500*	*54 900*
Number of triangles		*27 000*	*110 000*
Number of randiance samples per voxel		*90*	*23*

According to our experiments, an additional level in the octree multiplies the computing time by two even though the number of voxels is four times higher. This is due to a regular grid used for tracing rays through the octree. Note also that the time required for estimating light sources and surface properties decreases according to the octree depth since the number of voxels is higher and the number of radiance samples in each voxel is much lower.

8 Limitations

For distinguishing two light sources, our detection method requires an angle difference of at least 30 degrees. We also made the assumption that all the light sources are white. However, since the problem is ill-posed, no automatic method could properly estimate both reflectance and light sources emittance properties.

Objects composed of a high variety of textures can be difficult to process. When the number of voxels in each class is too low, reflectance and light sources properties cannot be precisely recovered.

For recovering reflectance properties, diffuse inter-reflections are ignored. Consequently, some artifacts can affect the object (re)shading.

9 Conclusion

This paper presents a method for estimating geometry, reflectance properties and light sources position from images of an object. Our method has been validated with virtual and real objects using an inexpensive acquisition system. Despite unreconstructed small cavities, multiple light sources have been recovered with an inaccuracy of about ten degrees. Specular highlights seen on images provide an additional information used for estimating directional light sources. They could also be matched and used for refining point light sources detection.

In the future, we aim at combining the reconstructed information for relighting image-based objects with diffuse and specular inter-reflections. High dynamic range images [9] would also provide a more precise estimation of radiance samples reflected by the object. The acquisition process would thus require several photographs with various shutting speeds.

Acknowledgements We would like to pay a tribute to Alain Fournier (Imager Lab, University of British Columbia, Vancouver) with whom the overall framework around this paper has been initiated in 1999. The quad has been designed by Stewart Cowley who kindly accepted to let us use this model for our work. Also many thanks to Philippe Dubois (SIC lab) who built the acquisition system.

References

1. Boivin, S., Gagalowicz, A.: Image-based rendering of diffuse, specular and glossy surfaces from a single image. In: SIGGRAPH'2001, Annual Conference Series, pp. 107–116 (2001)
2. Brooks, M.J., Horn, B.K.P.: Shape and source from shading. Tech. rep., MIT (1985)
3. Callet, P., Zymla, A.: Rendering of binary alloys. In: ICCVG 2004, pp. 469–476 (2004)
4. Chen, Q., Medioni, G.: A volumetric stereo matching method: Application to image-based modeling. In: CVPR, pp. 29–34. IEEE Computer Society (1999)
5. Chien, C.H., Aggarwal, J.K.: Volume/surface octrees for the representation of three-dimensional objects. Comput. Vision Graph. Image Process. **36**(1), 100–113 (1986)
6. Cook, R.L., Torrance, K.E.: A reflectance model for computer graphics. ACM Transactions on Graphics **1**(1), 7–24 (1982)
7. Debevec, P., Hawkins, T., Tchou, C., Duiker, H.P., Sarokin, W., Sagar, M.: Acquiring the reflectance field of a human face. In: "Siggraph 2000, Computer Graphics Proceedings", pp. 145–156 (2000)
8. Debevec, P.E.: Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. ACM Computer Graphics **32**(Annual Conference Series), 189–198 (1998)
9. Debevec, P.E., Malik, J.: Recovering high dynamic range radiance maps from photographs. ACM Computer Graphics **31**(Annual Conference Series), 369–378 (1997)
10. Esteban, C.H., Schmitt, F.: Silhouette and stereo fusion for 3d object modeling. In: 3DIM 2003, pp. 46–53 (2003)
11. Faugeras, O., Keriven, R.: Complete dense stereovision using level set methods. Lecture Notes in Computer Science **1406**, 379+ (1998)
12. Guillou, E.: Simulation d'environnements complexes non lambertiens à partir d'images : application à la réalité augmentée. Ph.D. thesis, Université de Rennes 1 (2000)
13. Hasenfratz, J.M., Lapierre, M., Gascuel, J.D., Boyer, E.: Real-time capture, reconstruction and insertion into virtual world of human actors. In: Vision, Video and Graphics, pp. 49–56. Eurographics (2003)
14. Hawkins, T., Wenger, A., Tchou, C., Gardner, A., Göransson, F., Debevec, P.E.: Animatable facial reflectance fields. In: EGSR, Rendering Techniques, pp. 309–321 (2004)
15. He, X.D., Torrance, K.E., Sillion, F.X., Greenberg, D.P.: A comprehensive physical model for light reflection. ACM Computer Graphics **25**(Annual Conference Series), 175–186 (1991)
16. Hougen, D., Ahuja, N.: Estimation of the light source distribution and its use in integrated shape recovery from stereo and shading. In: Proc 4th IEEE ICCV, pp. 148–155 (1993)
17. Hougen, D.R., Ahuja, N.: Adaptive polynomial modelling of the reflectance map for shape estimation from stereo and shading. In: CVPR, pp. 991–994 (1994)
18. Kay, G., Caelli, T.: Inverting an illumination model from range and intensity maps. CVGIP **59**(2), 183–201 (1994)
19. K.E.Torrance, E.M.Sparrow: Theory for off-specular reflection from roughened surface. Journal of Optical Society of America **57**, 1105–1114 (1967)
20. Kutulakos, K.N., Seitz, S.M.: A theory of shape by space carving. Tech. Rep. TR692, (1998)
21. Lafortune, E., Foo, S.C., Torrance, K.E., Greenberg, D.P.: Non-linear approximation of reflectance functions. Computer Graphics **31**(Annual Conference Series), 117–126 (1997)
22. Lee, C.H., Rosenfeld, A.: Improved methods of estimating shape from shading using the light source coordinate system. Artificial Intelligence **26**, 125–143 (1985)
23. Levoy, M., Hanrahan, P.: Lightfield rendering. Computer Graphics **30**(Annual Conference Series), 31–42 (1996)
24. Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, S., Koller, D., Pereira, L., Ginzton, M., Anderson, S., Davis, J., Ginsberg, J., Shade, J., Fulk, D.: The digital michelangelo project: 3D scanning of large statues. In: Siggraph 2000, Computer Graphics Proceedings, pp. 131–144 (2000)
25. Lewis, R.: Making shaders more physically plausible. Computer Graphics Forum **13**(2) (1994)
26. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. ACM Computer Graphics **21**(Annual Conference Series), 163–169 (1987)
27. Martin, W.N., Aggarwal, J.K.: Volumetric descriptions of objects from multiple views. IEEE Transactions on Pattern Analysis and Machine Intelligence **5**(2), 150–158 (1983)
28. Mercier, B., Meneveaux, D.: Joint estimation of multiple light sources and reflectance from images. In: ICCVG 2004, pp. 66–71 (2004)
29. Mercier, B., Meneveaux, D.: Shape from silhouette: Image pixels for marching cubes. Journal of WSCG 2005 **13**, 112–118 (2005)
30. Nillius, P., Eklundh, J.O.: Automatic estimation of the projected light source direction. In: CVPR, pp. I:1076–1083 (2001)
31. Oren, M., Nayar, S.K.: Generalization of Lambert's reflectance model. Computer Graphics **28**(Annual Conference Series), 239–246 (1994)
32. Pentland, A.: Finding the illuminant direction. JOSA **72**, 448–455 (1982)
33. Powell, M., Sarkar, S., Goldgof, D.: A simple strategy for calibrating the geometry of light sources. PAMI **23**(9), 1022–1027 (2001)
34. Sato, I., Sato, Y., Ikeuchi, K.: Illumination distribution from shadows. In: CVPR99, pp. I: 306–312 (1999)
35. Sato, Y., Ikeuchi, K.: Reflectance analysis for 3D computer graphics model generation. Graphical models and image processing: GMIP **58**(5), 437–451 (1996)
36. Sato, Y., Wheeler, M.D., Ikeuchi, K.: Object shape and reflectance modeling from observation. ACM Computer Graphics **31**(Annual Conference Series), 379–388 (1997)
37. Seitz, S., Dyer, C.: Photorealistic scene reconstruction by voxel coloring. In: CVPR, pp. 1067–1073 (1997)
38. Singh, H., Chellappa, R.: An improved shape from shading algorithm. Tech. Rep. CS-TR-3218, Department of Computer Science, University of Maryland Center for Automation Research, College Park, MD (1994)

39. Slabaugh, G., Culbertson, B., Malzbender, T., Schafer, R.: A survey of methods for volumetric scene reconstruction from photographs. In: VG01, pp. 81–100 (2001)
40. Szeliski, R.: Rapid octree construction from image sequences. In: CVGIP: Image Understanding, vol. 1, pp. 23–32 (1993)
41. Vega, O., Yang, Y.: Default shape theory: With application to the computation of the direction of the light source. CVGIP **60**(3), 285–299 (1994)
42. Ward, G.: Measuring and modeling anisotropic reflection. In: ACM (ed.) SIGGRAPH'92, pp. 265–272. ACM (1992)
43. Yang, Y., Yuille, A.: Source from shading. In: IEEE CVPR, pp. 534–539 (1991)
44. Yu, Y., Debevec, P., Malik, J., Hawkins, T.: Inverse global illumination: Recovering reflectance models of real scenes from photographs. In: ACM (ed.) SIGGRAPH'99, pp. 215–224 (1999)
45. Zhang, Y., Yang, Y.: Illuminant direction determination for multiple light sources. In: CVPR00, pp. I: 269–276 (2000)
46. Zheng, Q., Chellappa, R.: Estimation of illumination, albedo, and shape from shading. PAMI **13**(7), 680–702 (1991)
47. Zhou, W., Kambhamettu, C.: Estimation of illuminant direction and intensity of multiple light sources. In: ECCV02, p. IV: 206 ff. (2002)