Estimating Positions and Radiances of a Small Number of Light Sources for Real-Time Image-Based Lighting

Claus B. Madsen, Mads K. D. Sørensen, and Michael Vittrup

Laboratory of Computer Vision and Media Technology
Aalborg University, Aalborg, Denmark

Abstract

Image-based lighting (IBL) of virtual objects has become a popular approach to blending virtual and real scenes. In IBL an omni-directional image of a scene is used as the illumination environment for rendering virtual objects. Typically, this rendering is based on global illumination techniques which are far from capable of real-time performance. In this paper we describe how to estimate the positions and radiances of a small number of point light sources, e.g., on the order of 5 to 10, which will produce virtual object appearances which are consistent with those obtained using IBL. The estimated light source parameters can be used directly in OpenGL rendering for real-time performance. We demonstrate the approach on natural scenes.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism, I.4.8 [Image Processing and Computer Vision]: Scene Analysis

1. Introduction

High fidelity compositing of virtual objects into imagery of real scenes is already a highly developed area, and is used extensively for visual effects in films and commercials. One of the main problems in such compositing is to ensure that the lighting used when rendering the virtual objects is consistent with the lighting in the real scene.

Image-based lighting (IBL) is an approach which solves this problem by using omni-directional images to represent the irradiant light at the virtual object's location[4, 5, 9]. Some global illumination approach is then applied to render the virtual objects by placing them inside a large sphere mapped with the omni-directional image. Figure 1 shows an example of such a mapping. While this approach solves the problem of realistic lighting of virtual objects it is not suited for real-time applications such as Augmented Reality (AR).

In AR virtual objects are embedded in real scenes in real-time to allow the user to interact with the scene, and/or to purposively adjust his viewing position and direction in order to explore the augmented scene [2, 3].

The rendering step in IBL using a global illumination technique cannot visualize virtual objects at interactive frame rates. This is mainly due to the fact that computing the radiance at each surface point on the virtual objects involves a summation of radiance contributions from all light sources in the scene. And the large sphere mapped with the omni-directional image in principle represents a very large number (thousands, or hundreds of thousands) of tiny light sources contributing to the irradiance at the surface point.

In this paper we address the problem of trying to achieve IBL in real-time. The only feasible approach seems to be to lower the number of light sources, i.e., approximating the complicated, spatially continuous omni-directional lighting environment with a smaller number of directional light sources. Our approach is to estimate the position and radiance parameters (color and intensity) of some limited number of light sources so that the resulting virtual object lighting as closely as possible approximates what would be obtained with the standard IBL approach. In our experiments we have used up to 10 light sources for this purpose. OpenGL supports at least 8 directional light sources and standard graphics hardware allows real-time rendering, including multiple rendering passes for creating shadows.

A lot of work has been published on different approaches to acquiring the omni-directional real scene illumination in-

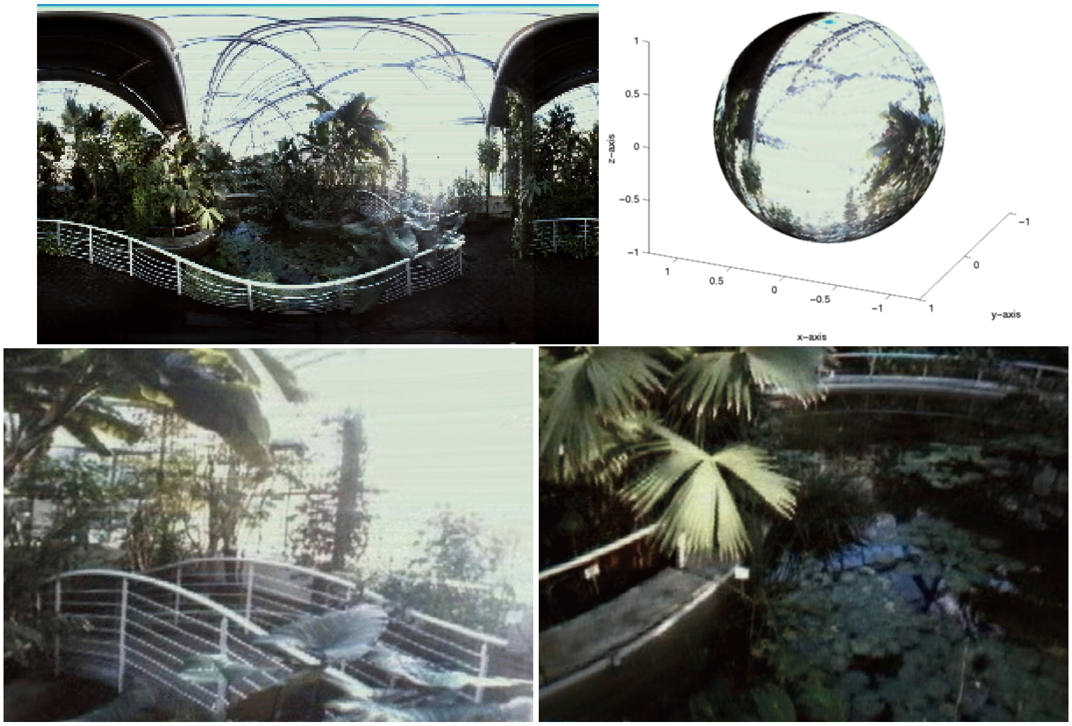

Figure 1: *Top left: full 360 by 180 degree image of a greenhouse in the botanical garden in Prague. Top right: the same image mapped to a sphere (here in a resolution of only 20480 triangles). In both versions it is possible to see the position of the sun, which is the all-dominant light source in the scene, but clearly the plants and the color of the glass roof also influences the lighting conditions. Bottom left and right: two different viewing directions in the botanical garden scenario. Notice the sun light is coming in from the right in the left image, and from the rear in the right image.*

formation. Some researchers use special probes, typically highly reflective metal spheres[4, 5, 8]. Others place cameras with very large field-of-view (up to 180 degrees) in the scene at the approximate location of where the virtual objects will later be placed[9]. A different approach has been to estimate the spherical scene radiance distribution from images of known real objects casting shadows in the real scene[11, 10, 12].

The work presented in this paper is applicable regardless of how the omni-directional radiance distribution is obtained. Our work simply assumes such radiance distribution information is available, (in our experiments omni-directional images are composed from multiple large field-of-view images acquired with a rotating camera).

The paper is organized as follows. In section 2 we present an overview of the approach showing some illustrative results. Section 3 describes the theoretical foundations for the proposed approach to estimating light source parameters. Then section 4 is devoted to presenting how the estimated light source parameters are used in the real-time visualization of virtual objects. Section 5 is devoted to discussing some central issues of the approach and future work. Section 6 offers some conclusions.

2. Context and overview of approach

This research is carried out in the context of a research project where we have omni-directional images of scenarios available. These omni-directional images are composed as mosaics from multiple images acquired with a rotating 180 degree field-of-view camera. Figure 1 shows the full image and a mapping of it to a sphere. In the project such spherical images are displayed to users wearing an orientation tracked Head Mounted Display so the user is able to freely look around in any direction; figure 1 shows two screenshots of what the user may see in this particular example. The user sees the scenario in stereo, but this is not reproduced here.

We wish to insert virtual objects into such scenarios and we naturally want to render the virtual objects under lighting conditions which are consistent with the real scene lighting. As described our objective is to somehow determine the parameters of some virtual lighting condition, which is suitable for real-time rendering, and which as closely as possible mimics the virtual object shading that would have been obtained using normal image-based lighting (IBL). The parameters we are aiming at determining are the directions and radiances (colors and intensities) of a limited number of directional light sources, which can then be used directly in an OpenGL implementation.

The proposed approach to estimating this simplified lighting environment is conceptually very simple, and comprises two steps. First we take a small white virtual sphere and place it in the center of a large sphere mapped with the omni-directional image. Then we render the radiances of the white sphere at many points on the sphere surface taking radiant light contributions into account from every point on the large sphere, i.e., completely as in classic IBL. Subsequently we will refer to the small white sphere as the **virtual probe sphere**, and since the large sphere is a map of the distribution of radiant light from every direction in a real scene we will refer to it as the **scene radiance sphere** (some researchers use the term radiance map[4]). An example of a scene radiance sphere is actually what is displayed in figure 1, and figure 2 shows the computed radiances of the virtual probe sphere for the botanical garden case.

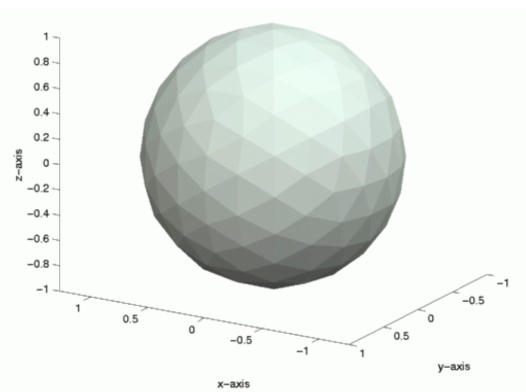

Figure 2: *The virtual probe sphere is a white, perfect Lambertian reflector. This figure shows the radiances of the virtual probe when rendered inside the lighting environment shown in figure 1. Notice the green shine due to reflection from plants, and the 'highlight' due to the incoming sun light.*

After having computed the radiances of the virtual probe sphere under IBL, step 2 is to estimate the parameters of some user specified number of directional light sources, plus the parameters of an optional ambient light term. The estimation is based on an iterative scheme which optimizes over the set of parameters, such that if the virtual probe had been subjected to this simplified lighting its radiances would be similar to those computed in the step 1. That is, the iterative estimation step adjusts the lighting parameters so as to minimize the difference between the radiances of the virtual probe radiances from step 1, and the virtual probe radiances given the currently estimated parameters of the simplified lighting conditions. Figure 3 shows the result of such an estimation.

Figure 3 shows that the complex lighting environment of the botanical garden can actually be simulated quite accurately using only an ambient term plus one directional light source. This allows us to achieve real-time rendering with shading results which are comparable to the complicated true IBL approach.

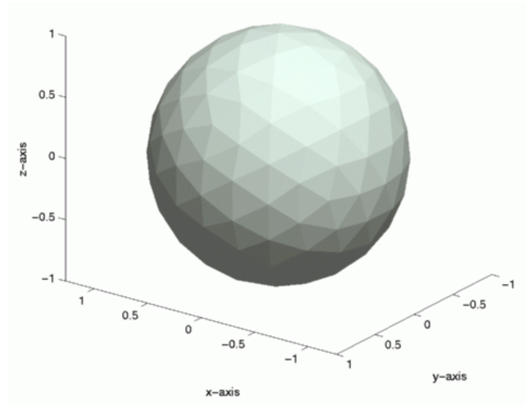

Figure 3: *The virtual probe subjected to virtual lighting consisting of an estimated ambient term, plus an estimated direction and radiance of one directional light source. When comparing to figure 2, which shows the true IBL virtual probe radiance for this example, it is seen that light direction and radiances are very accurately estimated.*

The described steps 1 and 2 represent an off-line phase resulting in lighting parameters which can then be used in an on-line, real-time phase which can render shading and shadows of dynamic virtual objects. This would be impossible to achieve with normal IBL due to the complexity of the global illumination lighting computations.

3. Estimation of light source parameters

In this section we will describe the off-line steps, i.e., the steps involved in estimating the simplified lighting environment prior to the actual real-time rendering of virtual objects. First we describe how to compute the Image-Based Lighting of the virtual probe sphere, i.e., compute the radiances, or the shading as it were, of the virtual white sphere. After that we describe how to use those radiances to estimate the parameters of the chosen number of directional light sources.

3.1. Computing true probe radiances using image-based lighting

In our work we have chosen to use a white sphere with Lambertian reflection properties as the virtual probe. There are multiple reasons for this. First, the probe could be of any color as long as it reflects *some* light in all color bands (we use three color bands, R, G, and B). As long as the same

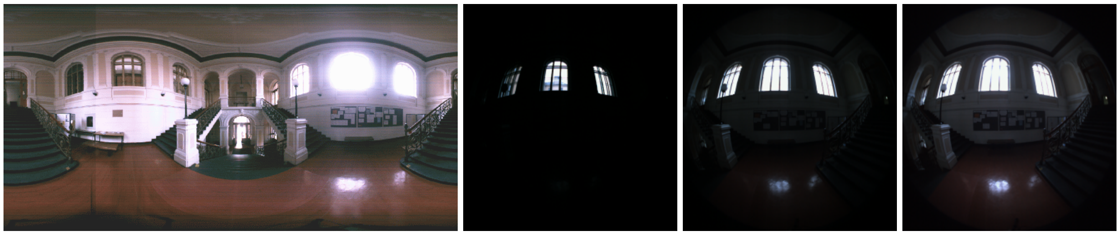

Figure 4: *Leftmost: omni-directional radiance map composited from multiple images acquired with a rotating 180 degrees field-of-view camera in a hall scenario. The three remaining images show the window region acquired with integration times of 23, 63, and 103 milliseconds respectively.*

color is used in the estimation step it will not change the estimated light source parameters. Secondly, a sphere is optimal because it has an even distribution of surface normal directions, enabling us to evenly capture the spatial distribution of light coming from the scene radiance sphere. Thirdly, the virtual probe is chosen as an Lambertian reflector since including a glossy reflection component would involve worrying about viewpoint dependency of the computed radiance. With a Lambertian (purely diffuse) reflection viewpoint dependency is not a problem.

For rendering the virtual probe radiances under the given scene radiance sphere we could choose any reflection model, but we have chosen the Phong shading model[1] due to its simplicity and because it is sufficient for diffuse reflection. Since we only consider diffuse reflection the Phong shading model can be formulated as:

$$R_i^R = \sum_{j=1}^{K} k_d \cdot C^R \cdot L_j^R \cdot (\vec{n}_i \cdot \vec{l}_j) \quad (1)$$

$$R_i^G = \sum_{j=1}^{K} k_d \cdot C^G \cdot L_j^G \cdot (\vec{n}_i \cdot \vec{l}_j) \quad (2)$$

$$R_i^B = \sum_{j=1}^{K} k_d \cdot C^B \cdot L_j^B \cdot (\vec{n}_i \cdot \vec{l}_j) \quad (3)$$

where R_i^R, R_i^G, and R_i^B are the RGB radiances from point number i on the probe surface, k_d is the diffuse reflection coefficient of the probe surface (here set to 1), C^R through C^B are the RGB values representing the color of the probe surface (here all set to 1), L_j^R through L_j^B are the RGB radiances of the jth point on the scene radiance sphere, i.e. the jth light source in the image-based light environment. K is the number of such sources, typically several thousands. $\vec{n}_i$ is the unit normal of the probe surface at point number i, and $\vec{l}_j$ is the unit direction vector from probe point number i to environment light source number j.

In our implementation we have used sub-sampled icosahedra for the virtual probe and the scene radiance spheres. We typically use 320 faces for the virtual probe, and 5120 faces for the scene radiance sphere. For both spheres we use the centroid of each face as the points in conjunction with eqs. 1 through 3. Computing the probe radiances is the most time consuming step in our approach. To compute 320 RGB probe radiances in a 5120 sources image-based lighting environment takes a few minutes in Matlab. In a C++ implementation it would take a few seconds. Figure 2 showed the probe in the botanical garden scenario.

Figure 4 shows an omni-directional scene radiance map from another indoor scenario dominated by lighting coming in from 3 large windows. In this context it is appropriate to introduce how we deal with the problem that there is limited dynamic range in images acquired with a normal camera. The light coming through the windows is so intense that if the windows should not be over-exposed the rest of the scene would be severely under-exposed. We deal with this in the same way as in [4], namely by combining images taken with varying integration times. This way we get much higher dynamic range allowing for non-saturated pixels in the directions of the dominant light sources.

Figure 5 shows the virtual probe radiances corresponding to the scene radiance map in figure 4.

3.2. Approximating true radiances with a few point light sources

Virtual probe radiances as exemplified in figures 2 and 5 form the foundation for the second step in our approach: estimating the parameters of a simplified lighting environment.

Real-time rendering libraries such as OpenGL support at least 8 light sources, plus an ambient light term, and the shading in OpenGL is performed using the Phong shading model[1]. Our objective is to utilize the OpenGL Phong shading to approximate the complex IBL described in section 3.1. I.e., where normal IBL would use several thousands of light sources distributed evenly over all directions in the scene, we are now aiming at figuring out where to place a handful of light sources, and what radiances to give them, in order to get the approximately the same shading result.

When including the ambient term, and allowing for M directional light sources, where M is small, the Phong shading

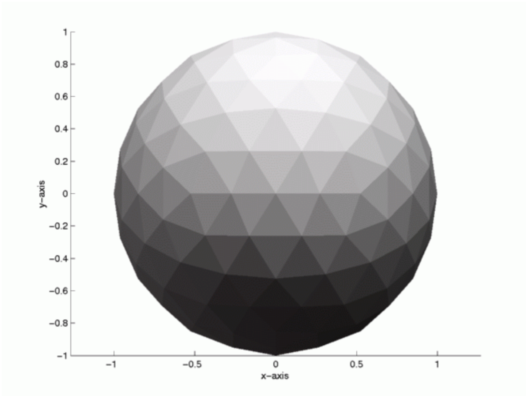

Figure 5: *The radiances of the virtual probe when rendered inside the lighting environment of the hall scene. In this case the virtual probe is shown directly from above and the three windows dominating the scene radiance map are directly in the top relative to the shown probe.*

model can be written as the following (again excluding the specular reflection term):

$$\overline{R}_i^R \;=\; k_a \cdot C^R \cdot L_a^R + \sum_{j=1}^{M} k_d \cdot C^R \cdot L_j^R \cdot (\vec{n}_i \cdot \vec{l}_j) \qquad (4)$$

$$\overline{R}_i^G \;=\; k_a \cdot C^G \cdot L_a^G + \sum_{j=1}^{M} k_d \cdot C^G \cdot L_j^G \cdot (\vec{n}_i \cdot \vec{l}_j) \qquad (5)$$

$$\overline{R}_i^B \;=\; k_a \cdot C^B \cdot L_a^B + \sum_{j=1}^{M} k_d \cdot C^B \cdot L_j^B \cdot (\vec{n}_i \cdot \vec{l}_j) \qquad (6)$$

where $\overline{R}_i^R$, $\overline{R}_i^G$, and $\overline{R}_i^B$ are the RGB radiances of the ith point on the virtual probe in the *simplified* lighting environment (not to confuse them with the radiances computed using IBL, as represented by eqs. 1 through 3). k_a is the coefficient of ambient reflection, L_a^R through L_a^B are the RGB radiances of the ambient light, and all the other parameters have the same meaning as in eqs. 1 through 3.

Now, we are interested in minimizing the difference between the full K sources IBL and the simplified M sources shading. R_i^R, R_i^G, and R_i^B are the probe radiances computed in section 3.1, so they represent three known numbers for each point on the probe surface. Our task is thus to find the parameters which best satisfy the homogeneous equation system:

$$R_i^R - k_a \cdot C^R \cdot L_a^R - \sum_{j=1}^{M} k_d \cdot C^R \cdot L_j^R \cdot (\vec{n}_i \cdot \vec{l}_j) \;=\; 0 \qquad (7)$$

$$R_i^G - k_a \cdot C^G \cdot L_a^G - \sum_{j=1}^{M} k_d \cdot C^G \cdot L_j^G \cdot (\vec{n}_i \cdot \vec{l}_j) \;=\; 0 \qquad (8)$$

$$R_i^B - k_a \cdot C^B \cdot L_a^B - \sum_{j=1}^{M} k_d \cdot C^B \cdot L_j^B \cdot (\vec{n}_i \cdot \vec{l}_j) \;=\; 0 \qquad (9)$$

The unknowns in this equation system are the parameters which we have to estimate. L_a^R, L_a^G, and L_a^B represent the three unknown ambient radiances. The L_j^R, L_j^G, and L_j^B parameters represent the $3M$ unknown radiances of the M light sources. Furthermore, the $\vec{l}_j$ vectors represent $2M$ unknown light source directions if each light direction vector is written as: $\vec{l}_j = \begin{bmatrix} sin(\theta_j)cos(\phi_j) & cos(\theta_j)cos(\phi_j) & cos(\theta_j) \end{bmatrix}$, i.e., the unknowns are the θ_j and ϕ_j parameters.

All in all there are $3+5M$ parameters to estimate. Let N denote the number of points on the virtual probe (typically 320 in our experiments so far). We then have $3N$ equations and $3+5M$ unknowns, and the equations are highly non-linear due to the way the light source direction parameters influence the radiances. We have employed the Newton's iterations method for iteratively solving for the optimal values of the unknowns. A in-depth description of the iterative scheme is given in [6], and it is beyond the scope of this paper to describe it. The method involves computing the Jacobian matrix of the equation system, i.e., the first derivatives of each equation with respect to all the unknown parameters. This is tedious but straight forward in our formulation as given in eqs. 7 through 9.

Figure 6 shows two examples of estimated simple lighting environments corresponding to the hall scenario shown in figure 4. In the latter example, with ambient light and 4 directional sources, the 3 estimated dominant sources correspond exactly to the locations of the 3 windows. The last source corresponds to the diffuse reflection from the almost white wall opposite the windows. Interestingly, the radiance of the estimated ambient term is of a color which is similar to the color of the floor in the scenario. Figure 7 shows the estimated ambient light color.

The issue of which ambient reflection coefficient, k_a, to choose is complex and cannot be dealt with in depth in this paper, but we will give a basic outline of the issue. If k_a is set to zero, no ambient term is estimated and the proposed procedure will attempt to account for all the image-based lighting using the specified number of directional sources. If a non-zero value is chosen, the ambient radiances will be estimated. If k_a is set to 0.1 the estimated ambient radiances will just be 10 times lower than if k_a were set to 1.0. Nevertheless, it is important that the coefficient is set to something which is approximately what will be used for the actual real-time rendering of virtual objects, otherwise the radiance balance between ambient and directional light will not be correct.

As with all iterative schemes the Newton iteration method requires initial values for all unknown parameters. We use 1.0 for all RGB radiances, and initially place the M directional sources so they are fairly evenly distributed over the view sphere. Convergence on the parameter estimation is achieved in about 20 iterations, and it takes a few seconds in our Matlab implementation. The time is primarily spent on rendering the radiances of the virtual probe given the cur-

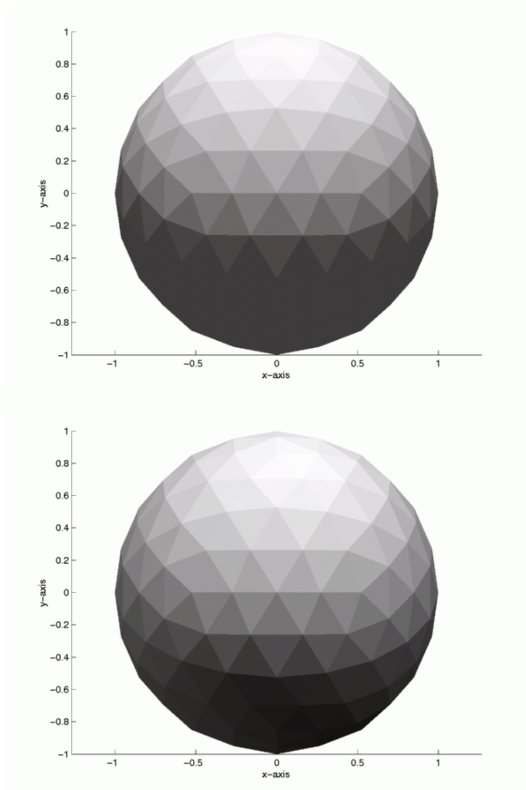

Figure 6: *Two estimated lighting environments visualized by how they would shade the virtual probe. The top example includes an ambient light term and one directional source. The bottom example includes and ambient term and 4 direction sources. Both are visually quite close to the 'ground truth' shown in figure 5, but the bottom figure shows a much better shading depth in the area pointing away from the dominant light sources.*

Figure 7: *The figure shows the color of the estimated ambient light for the hall scenario.*

rent parameter values at each iteration step, (to compute the R_i^R, R_i^G, and R_i^B values for eqs. 7 through 9).

4. Real-time rendering of virtual objects

Having thus described the off-line steps in our approach, the steps leading to the estimated parameters for a simplified lighting environment, we can now proceed to giving a brief overview of the on-line, real-time phase. In the real-time phase we apply the estimated lighting parameters directly in an OpenGL implementation of a renderer, which visualizes two components: 1) the omni-directional scene radiance image mapped to the inside of a large sphere, providing the real image background for the scenario, and 2) any virtual objects one might desire. The first component is not subjected to any lighting, it is just showing the real scene image as a texture. The second component is rendered with shading according to the estimated lighting parameters. Figure 8 shows several views of this for the hall scenario, with arbitrarily distributed white spheres to give an impression of the virtual lighting environment. The implementation runs at +100 frames per second on a GeForce 4 card.

It is easy to see that the distribution of lighting directions is in good correspondence with the real scene, and it can also be seen that the ambient term, which dominates the shading on the dark sides of the balls, gives an appropriate soft reddish shine, consistent with the reddish floor and the creamy colored walls in the scene.

To additionally visualize the results we have implemented a simple shadow algorithm based on a generalization of the perspective projection, to project the virtual objects to planes in the environment. The implemented method is described in e.g. [14, 13], and does not produce soft shadows (penumbra), only umbra. Figure 9 shows a rectangloid object casting shadows on the floor.

The shadows are implemented as alpha overlays, where the alpha values and overlay colors are computed so as to be appropriate given the estimated lighting parameters. Our approach is inspired by [11], and the basic idea is to compute the radiance of a point when it is lit by all sources, and compute the radiance when the point is not lit by the jth source. The ratios between these radiances, a ratio for each of the three RGB components, carry the information needed to change the radiance in a given point in the scene, if the jth source causes a shadow from the virtual object to fall on that point. But where [11] develop the idea for a global illumination approach to compositing virtual objects into real scenes, we have developed a method for establishing the optimal alpha and color values for an overlay which can be used in conjunction with real-time graphics[7]. It is beyond the scope of this paper to recapitulate this method.

Figure 8: *Four different views of the hall scenario with randomly placed white balls to illustrate the shading resulting from the estimated lighting conditions with ambient plus 4 directional sources. The views are rendered with a large field-of-view to make it easier to orient the views relative to the scenario, therefore the balls are sometimes distorted.*

5. Discussion

In this paper we have focused solely on diffuse reflection of the virtual objects. To have more freedom to visualize virtual objects with different reflectance properties it is naturally possible to add a non-zero coefficient of specular reflection, k_s, in the real-time rendering of virtual objects. The estimated locations of directional sources are closely consistent with directions of dominant light in the scene radiance sphere, and therefore adding a specular reflection term in the rendering would give credible results.

Completely specular surfaces would not give good results, though, since the estimated sources are in effect point sized. If one wanted to give the illusion of highly specular surfaces the proposed approach could be combined with a level of environment mapping.

Another subject of future research is the number of estimated light sources used. Currently, the user determines the number of virtual light sources which should be used to approximate the image-based lighting. We will develop methods to automatically find the lowest number of sources which will result in shading errors below some predetermined threshold. An issue in this context is the values used as initial guesses for parameters in the iterative estimation. In this paper we get fine convergence with more or less arbitrarily chosen initial guesses, but preliminary work has shown that automatic determination of initial parameter values is key to seriously minimizing the number of sources. Minimizing the number of sources is important in order to maintain real-time performance for more complicated shadow casting approaches.

6. Conclusion

The paper addressed the problem of trying to achieve real-time performance when rendering virtual objects into real

Figure 9: *A virtual object casting shadows in the hall scenario. The three upward shadows correspond to the three windows casting light from behind the viewing position. A fourth shadow is cast by a weak light source estimated to account for the reflection off the wall in the background.*

scenes while taking real scene lighting into account. With image-based lighting (IBL) as the starting point we presented a method for computing a lighting environment which is of drastically reduced complexity, but still produces shading results that are virtually indistinguishable from what is obtained with non-real-time normal IBL. Normal IBL involves rendering with thousands of light sources, and with the method presented here, good results can be achieved with less than 10.

The presented approach is completely automatic as the user is only required to specify how many light sources to spend on approximating the real scene radiance. The main scope of our future research is to automate the choice of the number of light sources.

Acknowledgments

This research is funded in part by the BENOGO project under the European Commission IST program (IST-2001-39184), and in part by the ARTHUR project (IST-2000-28559). This support is gratefully acknowledged.

References

1. E. Angel. *Interactive Computer Graphics: a Top-Down Approach with OpenGL*. Addison-Wesley, 2nd edition, 2000.
2. R. T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355 – 385, August 1997.
3. R. T. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent advances in augmented reality. *IEEE Transactions on Computer Graphics and Applications*, 21(6):34 – 47, Nov/Dec 2001.
4. P. Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings: SIGGRAPH 1998, Orlando, Florida, USA*, July 2002.
5. P. Debevec. Tutorial: Image-based lighting. *IEEE Computer Graphics and Applications*, pages 26 – 34, March/April 2002.
6. K. Kanatani. *Group-Theoretical Methods in Image Understanding*. Springer Series in Information Sciences. Springer-Verlag, Berlin, Germany, 1990.
7. C. B. Madsen. Detecting, adding, and removing shadows for augmented reality. In *Proceedings: Conference on Image and Vision Computing New Zealand (IVCNZ'02), Auckland, New Zealand*, pages 7 – 13, November 2002.
8. M. W. Powell, S. Sarkar, and D. Goldgof. A simple strategy for calibrating the geometry of light sources. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9):1022 – 1027, September 2001.
9. I. Sato, Y. Sato, and K. Ikeuchi. Acquiring a radiance distribution to superimpose virtual objects onto a real scene. *IEEE Transactions on Visualization and Computer Graphics*, 5(1):1 – 10, March 1999.
10. I. Sato, Y. Sato, and K. Ikeuchi. Illumination distribution from brightness in shadows: Adaptive estimation of illumination distribution with unknown reflectance properties in shadow regions. In *Proceedings: International Conference on Computer Vision, Kerkyra, Greece*, pages 875 – 882, September 1999.
11. I. Sato, Y. Sato, and K. Ikeuchi. Illumination distribution from shadows. In *Proceedings: Conference on Computer Vision and Pattern Recognition, Ft. Collins, Colorado, USA*, pages 306 – 312, June 1999.
12. I. Sato, Y. Sato, and K. Ikeuchi. Stability issues in recovering illumination distribution from brightness in shadows. In *Proceedings: Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA*, pages II 400–407, June 2001.
13. M. Slater, A. Steed, and Y. Chrysanthou. *Computer Graphics and Virtual Environments, from Realism to Real-Time*. Addison-Wesley, 2001.
14. A. Watt and F. Policarpo. *3D Games: Real-Time Rendering and Software Technology*, volume 1. Addison-Wesley, 2001.