RC23053 (W0401-009) January 5, 2004
Computer Science

IBM Research Report

View-in-Context

Aravind Kalaiah, Holly Rushmeier, Fausto Bernardini
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

View-in-Context

Aravind Kalaiah Holly Rushmeier Fausto Bernardini

Abstract

We present a simple and cost-effective solution for augmented reality for pictures. Our approach is robust and achieves realistic results with minimal calibration, equipment, and user expertise. In our scheme the user takes a series of pictures of the site where the virtual object has to be visualized. These pictures are all we need for camera calibration, illumination capture, and rendering. We present a novel camera calibration technique that determines the orientation of a site that is too far to calibrate using the traditional calibration algorithms. We capture the illumination at the site by infering from the shadows cast by a paper-made cube placed at the site. We then combine all the information to place the object at the site with a consistent illumination and shadows.

Key words: Augmented Reality, Image-Based Rendering, Inverse Rendering.

1 Introduction

Changing the living space around us is a common day-to-day activity. In particular, the act of placing a new physical entity in a scene assumes special significance in a variety of industries such as architecture, construction, entertainment, museums, electronic commerce, etc. Visualizing the scene prior to this act could be critical for various reasons such as feasibility analysis, time, cost, and planning. Although traditional techniques of modeling and rendering virtual environments could be used to visualize the entire scene virtually, realism can be difficult to achieve. This is mainly because the parameters required for these simulation algorithms tend to be grossly inaccurate. Moreover such attempts to model a real-world environment is bound to be expensive, cumbersome, and time-consuming.

In this work, we address the problem of placing a synthetic object at any given site in a photograph of a real-world environment. This involves handling three main issues : (1) the position and orientation of the object should be consistent with the rest of the scene, (2) the illumination of the object should be consistent with the rest of the scene, and (3) the illumination of the scene should be changed to account for the synthetic object. There has been considerable research into this area and most solutions involve a combination of controlled setup, geometric modelling of the scene, calibration, user input, and inverse computation [15, 3, 17, 4, 5, 2]. However, a simple, quick, easily mastered, and complete solution to this problem is found lacking. Our approach is motivated to robustly generate a realistic image while keeping the cost, effort, and the user-expertise at low levels. We detail a six-step protocol for capturing the pictures that give us all the information needed for our computation in a efficient manner.

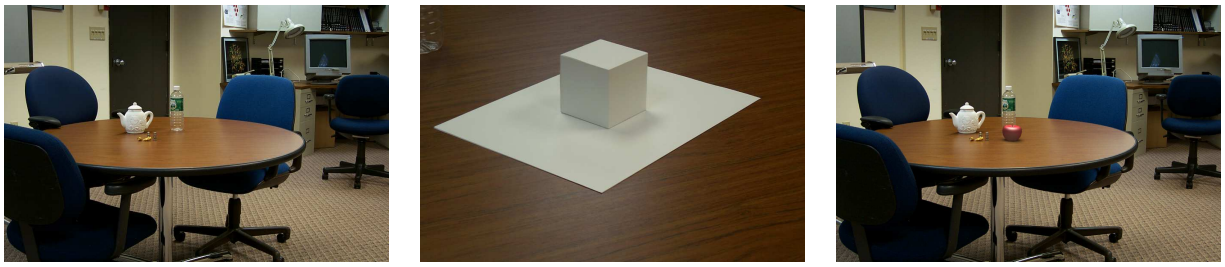
For the purposes of a simple but realistic solution, we make two assumptions: (1) the object is lit by distant lighting, and (2) apart from direct illumination, the only interaction between the synthetic object and the scene involves the intereflection between the object and the base of the site. We show how these assumptions can be used to create a wide range of pictures for augmented reality applications. Our approach is based on determining the distant lighting at a site by infering from the shadows cast by a paper-made cube (see figure 1).

2 Previous Work

Our work primarily draws from two areas of research: illumination recovery and reflectance recovery. While the former deals with determining the position and the intensity of the light sources in the environment, the latter focuses on the estimation of the BRDF (bidirectional reflectance distribution function) of the various surfaces visible in one or more pictures taken by the user.

Methods for recovering scene illumination use either a specialized camera, a light probe, or knowledge of the local environment. Sato et. al. [12] use a omnidirectional stereo camera to capture a view of the illuminating environment. Sato et. al. [13] determine the distant illumination at a site by interpreting the shadows cast by an object of known geometry already in the real scene. Debevec [3] uses a high dynamic range images of a shiny spherical "light probe" to capture the incident illumination. Zhang and Yang [18] use a lambertian spherical light probe to characterize a small number number of directional light sources in a scene.

There are many different techniques [9, 14, 11, 10]that use a combination of multiple images, controlled lighting, and geometric information to locally determine the BRDF of any given surface. Fournier et. al. [7] in their pioneering work describe a solution for augmented re-



(a) (b) (c)

Figure 1: We present a simple start-to-end solution to place a virtual object in the picture of a real-world environment (figure (a)) by determining the incident illumination at the site by interpreting shadows from a paper-made cube (figure (b)). The resulting image is shown in figure (c).

ality assuming all the surfaces in the environment were diffuse. With the knowledge of the geometry, camera parameters, and the lighting attributes they estimate the diffuse albedo of the surfaces by reprojecting them on the real image. The image can then be modified using progressive radiosity techniques. Drettakis et. al. [6] extend this for interactive manipulation with the pre-processing made more robust and convenient using computer vision techniques for camera calibration. Debevec [3] modeled only the local geometry after determining the distant light distribution with a light probe. Debevec proposed an automatic technique to extract the diffuse reflectance properties of the surface while the specular properties were fine-tuned manually. Loscos et. al. [8] use a knowledge of the geometry and the the light sources to recover the diffuse BRDF in the scene even in the presence of highly textured surfaces. Yu et. al. [17] propose an extensive solution to this problem by simulating the light transport and minimizing the error in the parameters of the Ward's anisotropic BRDF model [16]. They use the geometry of the scene and measure the lighting attributes that illuminates the scene. They can then easily introduce virtual objects in the pictures of the real scene by simulating the light transport. More recently, Boivin and Gagalowicz [2] propose an incremental and hierarchical solution that can extract a wide range of reflectance properties from a single photograph given the geometric model of the scene and the lighting attributes.

Of previous methods, our approach follows Debevec [3] most closely. We also divide the problem into estimating distant illumination and local geometry and reflectance. For distant illumination however we use a more commonly available object for light probe than a shiny spherical ball. We use a paper cube that can be readily assembled from common copier/printer paper. Similar to the work of [13] we construct the lighting distribution from examining the shading and shadows of the cubical probe. The averaging effect of diffuse reflectance reduces the dynamic range of luminances to make estimates of illumination. The directionality and frequency of the illumination are indicated by the sharpness of shadows cast by the test object. The advantage is that the illumination is constructed from luminances in the range of the objects in the photograph we are augmenting, rather than the high dynamic range required by the shiny sphere. Unlike Sato et. al. [13], we control the surface onto which shadows are cast, and take into account the effects of intereflection. Using a cube rather than a sphere reduces the demands on accuracy of estimating the camera parameters in the system, and so requires less user expertise. Different views of the cube are readily aligned by indicating vertices on the cube.

3 Data Capture

Our data capture process requires the user to take six pictures (see figure 2). The first set of pictures are taken from a distance that is possibly far from the site. It consists of two pictures taken from approximately the same spot with the first picture being of the scene (including the base) where the synthetic object has to be rendered. The second picture is taken with a plain sheet being placed at the site (for calibrating the camera for the first picture). The remaining pictures are close-up shots of the site. The third picture is taken with a calibration sheet placed at the site and the forth picture is that of a paper-made cube placed on top of a plain sheet. They both are taken with approximately similar camera coordinates and sheet orientation. Then the user goes around and takes the fifth picture capturing the cube from the other side and then takes the sixth picture with the calibration sheet placed at nearly the same orientation as in the seventh. The shadows in the forth and the fifth pictures help us to determine the lighting incident at the site. The third and the sixth pictures are taken to calibrate pictures four and five respectively. If the user is only interested in a closeup shot of the site then we can do without the second picture. We

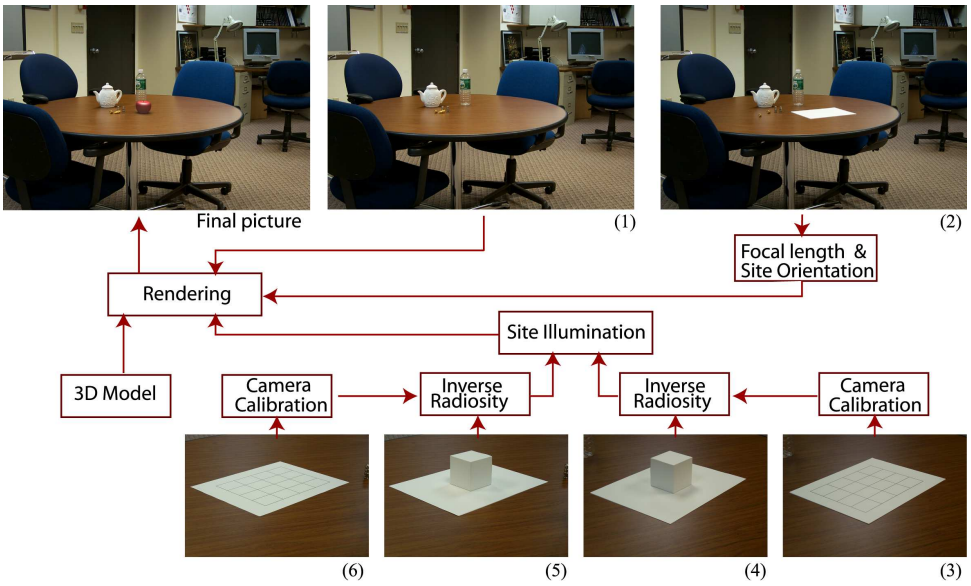
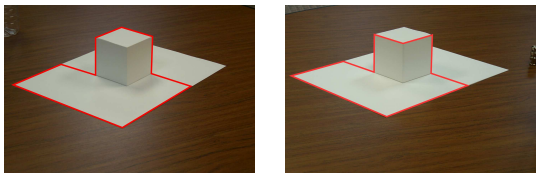


Figure 2: The six pictorial inputs to our algorithm: (1): The original picture in which the virtual object needs to be inserted. (2): The corners of the paper are used to calibrate the camera using our novel calibration technique. (3): A closeup shot of the site with the Tsai [15] calibration sheet. (4): Picture used to determine the incident lighting at the site using the shadows cast by the cube. (5): Same picture as (4) but taken from the other side of the cube. (6): Serves to calibrate picture (5). We use these pictures to generate a new picture with an artificially inserted synthetic object.

denote the pictures by the term $\mathbf{p}_i$ where $i = 1, \ldots, 6$ is the number of the picture in the picture capture sequence as shown in figure 2.

After the pictures are taken, the user loads the pictures into our software and identifies certain features of the pictures. In picture $\mathbf{p}_2$, the user marks the corners of the sheet. The user then identifies the corners of the cube and the sheet in pictures $\mathbf{p}_4$ and $\mathbf{p}_5$ and the calibration points in pictures $\mathbf{p}_3$ and $\mathbf{p}_6$. Note that this stage can also be automated (or atleast computer-assisted) using feature detection algorithms for enhanced efficiency [3].

We start with the processing of the pictures by first calibrating the camera parameters for pictures $\mathbf{p}_1$, $\mathbf{p}_4$ and $\mathbf{p}_5$ (§4). We then correct pictures $\mathbf{p}_4$ and $\mathbf{p}_5$ for the color distortion introduced by the camera (§5.1). We follow this with a radiosity based correction which undoes the inter-reflection between the cube and the sheet in the color-corrected versions of pictures $\mathbf{p}_4$ and $\mathbf{p}_5$ (§5.2). We use the corrected pictures to determine the lighting at the site by inverse computation from the shadows cast by the cube (§5.3). Finally, we render the virtual object into picture $\mathbf{p}_1$ with consistent orientation and illumination (§6). We conclude the paper with results and conclusions in sections §7 and §8 respectively.



(a) (b)

Figure 3: The area enclosed in red in figures (a) is the part of picture $\mathbf{p}_4$ that is used of computing the distant illumination. Figure (b) shows the analogous area for picture $\mathbf{p}_5$.

4 Camera Calibration

The white sheets and the cube seen in the photographs of figure 2 are the common A4 paper that are used for laser printers. To determine the lighting at the site we need to reconstruct a three-dimensional model of the cube sitting on top of the sheet and casting shadows on the sheet. We use the Tsai calibration algorithm [15] on picture $\mathbf{p}_3$ to determine the coordinates of the camera w.r.t. the sheet in picture $\mathbf{p}_4$. This lets us establish a mapping between a three-dimensional cube-on-top-of-sheet model and $\mathbf{p}_4$. However, since all parts of the sheet and the cube cannot be seen in a single picture, we use this mapping for only certain portions of the sheet and the cube as shown in figure 3. The remaining portions are covered in picture $\mathbf{p}_5$ which is in turn calibrated using picture $\mathbf{p}_6$.

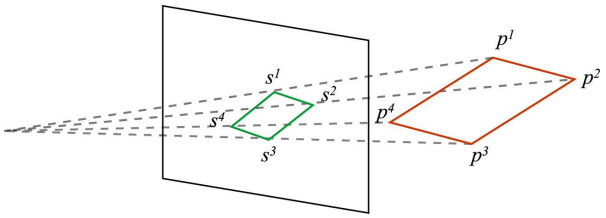
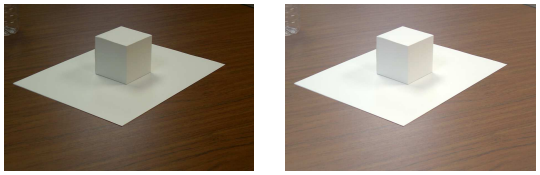


Figure 4: The orientation and the relative distance of the sheet in picture $\mathbf{p}_2$ is determined by using the projection coordinates s_i and the rectangular constraints inherent in the points p_i.



(a) (b)

Figure 5: We correct the original picture $\mathbf{p}_4$ (a) using an approximation of the camera response function to get the corrected picture (b).

While the Tsai calibration gives us the parameters of the camera, this information still cannot give us the depth at any pixel. For this we use our knowledge of the dimensions of the sheet, the dimensions of the cube, and the relative position of the cube w.r.t. the sheet. This allows us to map any pixel (u, v) of pictures $\mathbf{p}_4$ and $\mathbf{p}_5$ to a point on the cube and vice-versa.

The Tsai calibration technique can robustly recover the camera parameters but is impractical for the case where the site is far away. This is because a site that is far away would require a large calibration sheet so that the calibration points are sufficiently spaced in the picture. For this problem, we propose an alternate approach of determining the extrinsic parameters of the camera relative to the focal length of the photo shot. In particular, we determine the orientation of the site and its distance from the center of projection relative to the focal length of the camera. We infer these properties by exploiting the inherent constraints between the corners of the rectangular sheet in picture $\mathbf{p}_2$. Let $p^i = (p^i_x, p^i_y, p^i_z), i = 1 \ldots 4$, be the four corners of the rectangle as shown in figure 4. Let every corner p^i project to a point $s^i = (s^i_x, s^i_y)$ on the image. Then we have the following relationships between them:

$$p^i_x = \frac{s^i_x p^i_z}{f} \quad \forall i = 1, \ldots, 4, \tag{1}$$

$$p^i_y = \frac{s^i_y p^i_z}{f} \quad \forall i = 1, \ldots, 4, \tag{2}$$

$$p^3 = (p^4 - p^1) + (p^2 - p^1) + p^1, \tag{3}$$

$$0 = (p^1 - p^4) \bullet (p^3 - p^4), \tag{4}$$

where f is the focal length of the camera. Equations (1) and (2) impose the constraints of perspective projection, while equations (3) and (4) enforce the rectangular and planar constraints respectively. This gives us 12 linear equations with 13 variables. We solve this linear system in MATLAB by expressing all the terms relative to the focal length f. The camera parameters of $\mathbf{p}_1$ are needed for illumination of the virtual object and to cast shadows at the base. Our relative camera parameters suffice for this task. However, since all the recovered parameters are in the camera coordinates, this does not allow us to specify the dimensions of the virtual object in a real-world metrics such as meters. We leave this factor as a variable and allow the user to scale the virtual object according to his/her needs.

5 Illumination Capture

5.1 Color Correction

When the user is using a simple commercial camera we have to account for the possibility that the exposure time of the different pictures cannot be controlled by the user. That leads to color calibration issues since the site-illumination extracted from pictures $\mathbf{p}_4$ and $\mathbf{p}_5$ may be too bright or dim compared to the actual site-illumination as seen in the original picture $\mathbf{p}_1$. To color-calibrate the three relevant pictures $\mathbf{p}_1$, $\mathbf{p}_4$, and $\mathbf{p}_5$ we use the pixel response curve of the camera. Debevec and Malik [5] recover this curve by using several camera exposures of a scene with varying time intervals. They note that the actual color, Z, observed at any pixel of a picture is a function of the irradiance, E, observed at that pixel by the camera's sensor and the exposure time, $\triangle t$, of the shot. The relationship between them is given by:

$$g(Z) = ln(E) + ln(\triangle t),$$

where $g(\cdot)$ is a smooth monotonic function that they recover. Although we do not recover this function, we use this relationship to color-calibrate the pictures $\mathbf{p}_2$, $\mathbf{p}_4$, and $\mathbf{p}_5$.

Consider any diffuse point x in picture $\mathbf{p}_2$ that is also visible in picture $\mathbf{p}_4$. If we denote its irradiance by E_x, then we have that:

$$g(Z^i_x) = ln(E_x) + ln(\triangle t_i), \quad \text{for } i = 2, 4$$

where Z^i_x is the color of x as seen in picture $\mathbf{p}_i$ and $\triangle t_i$ is the exposure time of the picture $\mathbf{p}_i$. That gives us the relation:

$$ln(\triangle t_4) - ln(\triangle t_2) = g(Z^4_x) - g(Z^2_x). \tag{5}$$

Color calibration then comes down to determining how the pixel value Z_x^4 would change if the camera exposure time was $\triangle t_2$ as opposed to $\triangle t_4$. If we denote $corr(Z_x^4)$ as the corrected pixel value then we have that:

$$\begin{aligned} ln(E_x) &= g(Z_x^4) - ln(\triangle t_4) \\ &= g(corr(Z_x^4)) - ln(\triangle t_2). \end{aligned}$$

So the corrected color can now be written as:

$$g(corr(Z_x^4)) = g(Z_x^4) + ln(\triangle t_2) - ln(\triangle t_4).$$

We note that the term $ln(\triangle t_2) - ln(\triangle t_4)$ is independent of the point x. We estimate this term by the term $g(Z_m^2) - g(Z_m^4)$, where m is any other diffuse point that is visible in both pictures $\mathbf{p}_2$ and $\mathbf{p}_4$. This follows directly from equation (5). We thus have that:

$$g(corr(Z_x^4)) = g(Z_x^4) + g(Z_m^2) - g(Z_m^4).$$

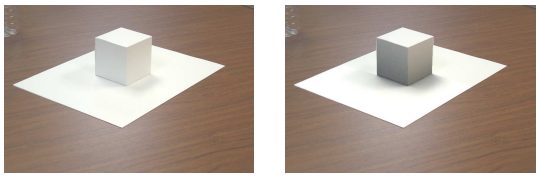
We note that in general, $g(Z)$, exhibits near-linear property for most cameras when the values of Z is in the range $[50, 255]$. If we approximate the function $g(Z)$ to be a linear function, then we have that:

$$corr(Z_x^4) \approx Z_x^4 + Z_m^2 - Z_m^4. \tag{6}$$

The near-linear property of $g(Z)$ is not true for very low values of Z. However, since the illumination computation (that picture $\mathbf{p}_4$ is used for) is dictated by the mid-range color values, we did not see any noticeable effects of this approximation. The point m has to be chosen such that it has a very high irradiance in both figures $\mathbf{p}_2$ and $\mathbf{p}_4$ for approximation (6) to hold true. We chose the point m to be the center of the white sheet in picture $\mathbf{p}_2$. However, since this point is occluded in picture $\mathbf{p}_4$, and we chose the color of the mid-point of the top of the cube in picture $\mathbf{p}_4$ to represent Z_m^4. The three color channels of each pixel of picture $\mathbf{p}_4$ are then corrected using approximation (6). Similarly, we correct picture $\mathbf{p}_5$ separately. Note that we are not calibrating the pictures w.r.t. the target picture $\mathbf{p}_1$. However, since $\mathbf{p}_1$ and $\mathbf{p}_2$ are shot with similar camera locations and lighting, the auto-exposure feature of the camera is bound to use similar exposure times for the two shots.

5.2 Correction by Inverse Radiosity

Using a white paper for the cube and for the paper sheet in pictures $\mathbf{p}_2$, $\mathbf{p}_4$, and $\mathbf{p}_5$ helps us to do the color-calibration and the illumination extraction robustly. It also allows us to account for any color variations due to color-variable lighting (e.g. an orange light from one side and blue from another). However, it also has the undesirable effect that we cannot interpret the pixel values directly since the observed color also includes the illumination due to the inter-reflection between the white-sheet and the cube in pictures $\mathbf{p}_4$ and $\mathbf{p}_5$. We undo this inter-reflection by using an inverse-radiosity solution. This is possible since the paper sheets we normally use are nearly diffuse reflectors, except at grazing angles. Figure 6 shows picture $\mathbf{p}_4$ before and after the correction.



(a) (b)

Figure 6: We eliminate the diffuse inter-reflection between the base paper and the cube seen in picture (a) by using an inverse radiosity solution. Picture (b) shows the resulting shadows without the inter-reflection.

To start with, we partition the sheet and the four sides of the cube into small rectangles that are about 1-2mm wide using the calibration pictures $\mathbf{p}_3$ and $\mathbf{p}_6$. The radiosity B_r of any rectangle r on the sheet is given by:

$$B_r = \rho_d^s I_r^d + \rho_d^s \sum_c B_c F_{r-c} \tag{7}$$

where ρ_d^s is the diffuse albedo of the sheet, I_r^d is incident irradiance at r due to the *visible* part of the distant light, c is any rectangle on the cube, and F_{r-c} is the rectangle-to-rectangle form factor from r to c (see figure 7(a)). We do not include any visibility term inside the integral since the cube is a convex object and hence the visibility term can be handled directly by comparing the normals of r and c during the form factor computation. The direct incident distant light at the sheet can be derived from equation (7) as:

$$I_r^d = \frac{B_r}{\rho_d^s} - \sum_c B_c F_{r-c} \tag{8}$$

Figure 6(b) shows the direct incident irradiance, I_r^d, on the sheet and the cube. We used a ρ_d^s value of 0.9 for our test cases. Changing it by ± 0.05 did not create noticeable changes in our results.

5.3 Incident Lighting

To estimate the incident lighting at the scene we use the implicit information that is contained in the shadow cast by the cube onto the paper. Sato et. al. [13] introduced this method to successfully extract the distant-lighting in both indoor and outdoor scenes. We use their method to estimate the lighting from the visible incident light, I^d, that was determined in §5.2.

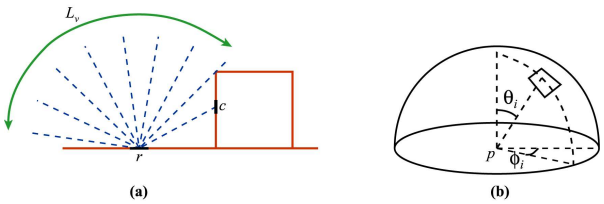


Figure 7: (a) The radiosity at a rectangle r on the sheet is a function of the visible light L_v and the radiosity of the rectangles c on the cube. (b) Any point p on the sheet is lit by the distant lighting coming from all directions of the upper-hemisphere. We partition the hemisphere into patches centered at directions (θ_i, ϕ_i).

Consider any point p on the sheet. Since it rests on a flat surface, it only sees the distant lighting in its upper hemisphere as shown in figure 7(b). We divide the upper hemisphere into N patches with the center of each patch being at an angle of (θ_i, ϕ_i) w.r.t. p. Let ω_i denote the solid angle that the patch subtends at p and let L_i^d be the average radiance through that patch. Furthermore let $S_i(p)$ be the visibility function which takes the value 1 if p is visible in the direction (θ_i, ϕ_i) and 0 otherwise. Then the incident irradiance due to the distant light, I_p^d, can be expressed as:

$$I_p^d = \sum_{i=1}^{N} L_i^d S_i(p) \omega_i \cos\theta_i. \tag{9}$$

Each point p on the sheet gives rise to an equation of the form (9). If we choose M points on the sheet, then we have the following linear system of equations:

$$\begin{aligned} \alpha_{1,1}L_1^d + \alpha_{1,2}L_2^d + \ldots + \alpha_{1,N}L_N^d &= I_1^d \\ \alpha_{2,1}L_1^d + \alpha_{2,2}L_2^d + \ldots + \alpha_{2,N}L_N^d &= I_2^d \\ &\vdots \\ \alpha_{M,1}L_1^d + \alpha_{M,2}L_2^d + \ldots + \alpha_{M,N}L_N^d &= I_M^d \end{aligned}$$

where $\alpha_{j,i}$ is the coefficient of the i^{th} term in the equation corresponding to the j^{th} point (see equation (9)). This is a liner system of M equations with N variables. If the rank of the matrix $\alpha = \{\alpha_{j,i}\}$ is atleast N, then this system can be solved using linear least-squares optimization methods. We use the Levenberg-Marquardt algorithm [1] to solve this system of equations and to determine the incident distant lighting L_i^d.

We use the sheet-partitioning that was used in §5.2 for setting up our linear system. The center of each partition gives us an equation of the form of equation (9). To make the system of equations more robust we add another equation representing the unoccluded illumination

(a) (b)

Figure 8: (a) A closeup of rendering of the virtual apple without inter-reflection between the object and the base. Figure (b) shows the same scene with inter-reflection.

of the sheet. For this we compute the average color of the top of the cube as observed in picture $\mathbf{p}_4$ and add the corresponding equation to the linear system. Further, to make the linear system numerically stable, we have to ensure that the solid angles, $\omega(\theta_i, \phi_i)$, do not differ significantly. We do a near-uniform partitioning of the upper hemisphere as follows:

$$\begin{aligned} \theta &= acos(u_o) \\ \phi &= 2\pi u_1 \end{aligned}$$

where u_0 and u_1 are a uniform partitioning of the closed range (0, 1). We avoid having duplicate light rays with $\theta = 0$ by using a minimum θ threshold of 5 degrees.

6 Rendering

After we have estimated the distant illumination at the site, placing the virtual object in picture $\mathbf{p}_1$ involves three steps: (1) placing the object with the right orientation, (2) casting shadows at the site, and (3) illuminating the virtual object. The first of these issues can be solved using our camera calibration method described in §4 with the size of the object being specified in terms of the focal length f of the camera in picture $\mathbf{p}_1$.

For casting shadows at the site, we first determine a shadow region around the object. We do this by determining a threshold θ amongst the light directions such that all directions with greater θ have negligible intensity. For each pixel in this region we determine the fraction, η, of the distant light intensity incident at that point. Assuming that the base is a perfectly diffuse reflector, we set the new color at the pixel to be a η fraction of its original color. We do not need to know the actual albedo of the base for this because the incoming direct light (under shadowing) is a subset of the original incoming light without shadowing. We illuminate the object using the distant light and the user specified material properties.

While the direct illumination of the object and the base accounts for a significant amount of the overall illumination, the inter-reflection between the two can be significant as well (see figure 8). We do the inter-reflection

by estimating the amount of light bounced back from the base to the object. This is similar to the final "gathering" step in the final calculation of pixel values in a global illumination solution since we have an estimate of the light leaving the base. Light from the object can also be reflected back onto the base. To compute the complete calculation for multiple interreflections between object and base the user needs to estimate the surface BRDF by a semi-manual method such as that described in [3].

7 Results

We show our results on two scenes captured with different lighting and cameras. We needed about 10 minutes for capturing the pictures in each of the two photo sessions. We used 16 values of ϕ and 8 values of θ for a total of 128 lighting directions. We used 1mm-patches for our inverse-radiosity solution. The lighting computation could be done within 5 minutes on our 2.4 GHz Pentium IV PC, although the inverse-radiosity solution was a little longer at 10 minutes. The rendering time depended on the scale of the virtual object. A bigger virtual object leads to larger shadows and thus more computation time. We needed about 10 seconds for the case of the apple while the cow model took a longer time of about 10 minutes. Our results can be seen in figures 9 and 10.

8 Conclusions and Future Work

We present a simple, robust, and cost-effective solution for augmented reality applications. Our streamlined approach can capture the distant illumination at the site and the camera parameters with just six pictures. Our acquisition stage is very robust. Since the color correction accounts for the differences in the exposure time of the pictures, the user does not have to worry about controlling the exposure time. The positioning of the camera for each shot is only specified approximately – hence the user is not constrained to hold the camera perfectly still from shot to shot.

In our approach we assume that the base is a perfectly diffuse reflector. We can improve our results by estimating the full BRDF of the base by extending current such methods used for uniform reflectance [3, 2]. Getting rid of the user inputs should further automate the whole process with potential extension to video sequences. Also, it seems possible to determine the distance of the light sources from the shadow information. Such a method would allow us to extend our approach to a wide variety of scenes.

References

[1] D. M. Bates and D. G. Watts. *Nonlinear Regression and Its Applications*. New York: Wiley, 1988.

[2] S. Boivin and A. Gagalowicz. Image-Based rendering of diffuse, specular and glossy surfaces from a single image. In *Proc. of SIGGRAPH'01*, pages 107–116, 2001.

[3] P. Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proc. of SIGGRAPH'98*, pages 189–198, 1998.

[4] P. Debevec, C. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proc. of SIGGRAPH'96*, pages 11–20, 1996.

[5] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of SIGGRAPH'97*, pages 369–378, August 1997.

[6] G. Drettakis, L. Robert, and S. Bougnoux. Interactive common illumination for computer augmented reality. In *Eurographics Techniques'97*, pages 45–56, June 1997.

[7] A. Fournier, A. Gunawan, and C. Romanzin. Common Illumination Between Real and Computer Generated Scenes. In *Graphics Interface '93*, pages 254–262, 1993.

[8] C. Loscos, G. Drettakis, and L. Robert. Interactive virtual relighting of real scenes. In *IEEE Tran. on Vis. and Comp. Graph.*, volume 6(4), pages 289–305. 2000.

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

[10] S. Marschner, S. Westin, E. Lafortune, K. Torrance, and D. Greenberg. Image-based BRDF measurement including human skin. In *Rendering Techniques'99*, June 1999.

[11] S. R. Marschner and D. P. Greenberg. Inverse lighting for photography. In *Proc. of the Fifth Color Imaging Conference*, November 1997.

[12] I. Sato, Y. Sato, and K. Ikeuchi. Acquiring a radiance distribution to superimpose virtual objects onto a real scene. *IEEE Trans. on Vis. and Computer Graphics*, 5(1):1–12, January/March 1999.

[13] I. Sato, Y. Sato, and K. Ikeuchi. Illumination from shadows. *IEEE Trans. on Patttern Analysis and Machine Intelligence*, 25(3):290–300, March 2003.

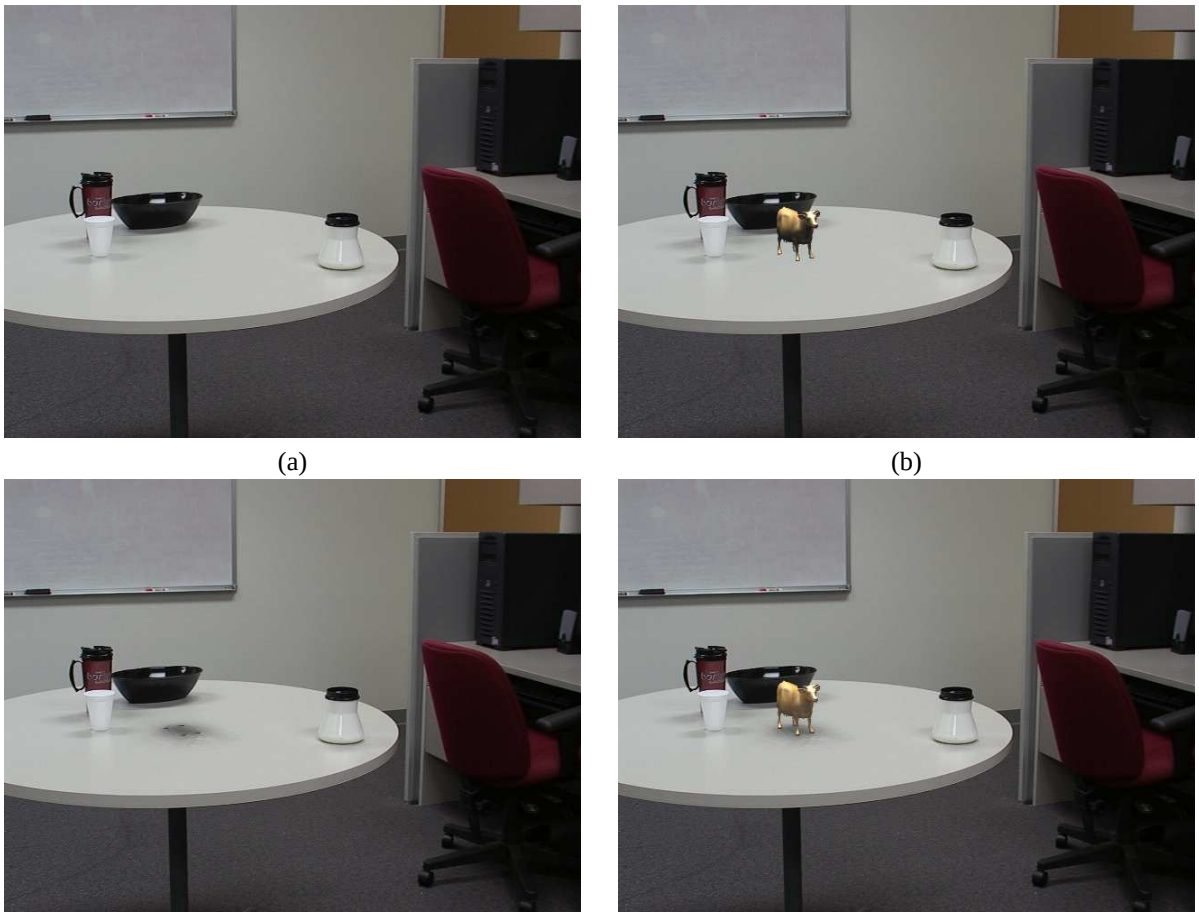
[14] Y. Sato, M. Wheeler, and K. Ikeuchi. Object shape and reflectance modeling from observation. In *Proc. of SIGGRAPH'97*, pages 379–388, August 1997.

[15] Roger Y. Tsai. A versatile camera calibration technique for high accuracy machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Animation*, 3(4):323–344, 1987.

[16] G. Ward. Measuring and modeling anisotropic reflection. In *Proc. of SIGGRAPH'92*, pages 265–272, July 1992.

[17] Y. Yu, P. Debevec, J. Malik, and T. Hawkins. Inverse global illumination: Recovering reflectance models of real scenes from photographs from. In *Proc. of SIGGRAPH'99*, pages 215–224, 1999.

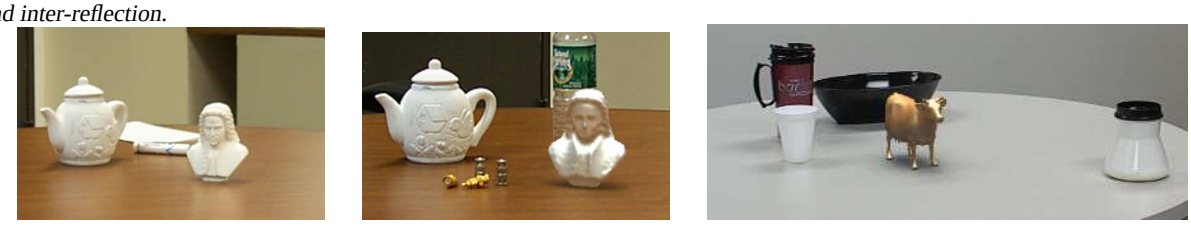
[18] Y. Zhang and Y.-H. Yang. Multiple direction detection with application to image synthesis. *IEEE Trans. on Pat. Anal. and Mach. Int.*, 23(8):913–920, August 2001.



(a) (b)

(c) (d)

Figure 9: Figure (a) shows the original acquired picture $\mathbf{p}_1$, while figure (b) shows the rendering of the cow into the picture but without shadows or inter-reflection. Figure (c) shows shadows cast by the model while figure (d) shows the object with shadows and inter-reflection.



(a) (b) (c)

Figure 10: Figures (a) shows an actual Bach figure, while figure (b) shows a scanned model of the Bach figure placed synthetically using our method. Note that the structure of the shadows are similar in the two cases. Figure (b) can be made to look even closer to figure (a) using knowledge of the BRDF of the base and that of the Bach figure. Figure (c) is a rendering of the cow model using the lighting computed for the scene of figure (1). Notice how its shadows look out of place.