

# Acquiring a Radiance Distribution to Superimpose Virtual Objects onto a Real Scene

Imari Sato, Yoichi Sato, Katsushi Ikeuchi

*Abstract*—This paper describes a new method for superimposing virtual objects with correct shadings onto an image of a real scene. Unlike the previously proposed methods, our method can measure a radiance distribution of a real scene automatically and use it for superimposing virtual objects appropriately onto a real scene. First, a geometric model of the scene is constructed from a pair of omni-directional images by using an omni-directional stereo algorithm. Then radiance of the scene is computed from a sequence of omni-directional images taken with different shutter speeds and mapped onto the constructed geometric model. The radiance distribution mapped onto the geometric model is used for rendering virtual objects superimposed onto the scene image. As a result, even for a complex radiance distribution, our method can superimpose virtual objects with convincing shadings and shadows cast onto the real scene. We successfully tested the proposed method by using real images to show its effectiveness.

*Keywords*— computer graphics, computer vision, augmented reality, illumination distribution measurement, omni-directional stereo algorithm

## I. INTRODUCTION

The seamless integration of virtual objects with an image of a real scene is an important step toward the long term goal of achieving photographic realism of synthesized images, and techniques for merging virtual objects with a real scene attract a great deal of attention in the fields of both computer graphics and computer vision research. The synthesized world called augmented reality allows us to see the real scene with virtual objects superimposed onto the scene and to handle phenomena not only in the real world but also in a virtual world, while virtual reality technologies immerse a user in a fully computer-generated scene.

To achieve a high quality synthesized image in augmented reality systems, three aspects have to be taken into account: *geometry*, *illumination*, and *time*. More specifically, the virtual object has to be positioned at a desired location in the real scene, and the object must appear at the correct location in the image (*consistency of geometry*). Also, shading of the virtual object has to match that of other objects in the scene, and the virtual object must cast a correct shadow, i.e., a shadow whose characteristics are consistent with those of shadows in the real scene (*consistency of illumination*). Lastly, motions of the virtual object and the real objects have to be coordinated (*consistency of time*).

In the past, consistency of geometry and consistency of time have been intensively investigated in the field of augmented reality.<sup>1</sup> Namely, 3D position sensors of various

modalities are used for measuring the position and orientation of a user's viewpoint in real time, so that a virtual object can be superimposed onto the image that the user is looking at (for instance, [2], [16], [3]).

On the other hand, few methods have been proposed for using correct illumination to superimpose virtual objects onto an image of a real scene. This is because real scenes usually include both direct and indirect illumination distributed in a complex way and it is not easy to obtain correct illumination models to be used for augmented reality systems.

Pioneering work in this field was proposed by Fournier et al. [10]. Fournier et al.'s method takes into account not only direct illumination but also indirect illumination by using the radiosity algorithm, which is commonly used for rendering diffuse interreflection [5]. This method is effective for modeling subtle indirect illumination from nearby objects. However, this method requires a user to specify the 3D shapes of all objects in the scene. This object selection process could be tedious and difficult if a scene were full of objects. Also, since this method computes global illumination using pixel values of an input image, it is required that the image have a reasonably wide field of view. Even so, this method cannot model direct illumination from outside of the input image unless a user specifies the positions of all lights.

Later, Drettakis et al. [8] extended Fourier et al.'s work. Drettakis et al.'s method made the creation of the 3D model much easier using computer vision techniques. They also introduced the use of a panoramic image built by image mosaicing to enlarge the field-of-view of the input image, and the use of hierarchical radiosity for efficient computation of global illumination. However, this method still requires a user to define the vertices and topology of all objects in the scene, and it is often the case that the achieved field-of-view is not wide enough to cover all surfaces in the scene. This causes the same limitation on direct illumination outside the input image as in Fournier et al.'s method.

Recently, Debevec [6] introduced a framework of superimposing virtual objects onto an image of a real scene with correct illuminations. This method first constructs a light-based model: a representation of a scene which consists of radiance information of the surfaces of the scene. The model is constructed by mapping reflections on a spherical mirror onto a geometric model of the scene.<sup>2</sup> To accurately record the brightness of the reflections, a use of a high dy-

I. Sato, Y. Sato, and K. Ikeuchi are with Institute of Industrial Science, The University of Tokyo, Tokyo, Japan. Email: {imarik, ysato, ki}@iis.u-tokyo.ac.jp.

<sup>1</sup>For a good survey of augmented reality technologies, see [1].

<sup>2</sup>State et al. previously introduced the use of a steel ball to capture the reflections at a single point. The method then placed a virtual object at the same location using an environment map texture extracted from the steel ball image [16].

dynamic range image of the scene generated from a series of images taken with different sensor integration time was addressed. Then, global illumination was computed among three scene components: distant scene (light-base model), local scene (nearby objects), and synthetic objects, while light reflected back to the distant scene was ignored. It was reported that the method was able to superimpose virtual objects onto the image with convincing shadings. However, this method cannot take into account the influence of the local scene on the light-base model. Since the light-base model is constructed separately from the local scene and light from the local scene to the light-base model is ignored during the global illumination computation, this method cannot handle the cases where objects in the local scene cause a serious radiance change of the light-base model, e.g., there is a light source in the local scene. Accordingly, geometries and reflectance properties of all objects in the local scene have to be given a priori, while shape and reflectance modeling of real objects itself is a difficult research topic. Moreover, as in the previous methods, this method requires a user's efforts to construct a light-base model: specifying a geometric model of the distant scene, selecting viewing points for observing the mirror so that the reflections on the mirror can cover the entire geometric model of the scene. A sampling frequency of the reflected light differs depending on the 3D geometry of the surface points on the mirror and the camera's viewing directions. This could cause poor sampling of the reflected light in some portions of the constructed light-base model. Hence, the accuracy of the constructed light-base model depends on the user's efforts, and this would not be desirable in some augmented reality applications.

To summarize, the previously proposed methods suffered from two difficulties: how to construct a geometric model of the scene, and how to capture a wide field of view of the scene. With regard to the first difficulty, no simple solution has yet been proposed, and its construction still requires user's efforts. With regard to the second difficulty, special equipment such as a spherical mirror or a panoramic image built by mosaicing was proposed. However, multiple images taken from different viewing angles are still necessary to capture the radiance of the entire scene, and therefore the image registration process is also required.

The purpose of this study is to present a new, efficient method for automatically measuring a radiance distribution of a real scene and for using it to superimpose virtual objects appropriately onto a real scene. For consistency of illumination, the proposed method is able to automatically measure a radiance distribution of a real scene by using a set of omni-directional images taken by a CCD camera with a fisheye lens.

There are three reasons why we use omni-directional images rather than images taken by a camera with an ordinary lens. First, because of fisheye lens' wide field of view, e.g., 180 degrees, we can easily capture illumination from all directions from far less number of omni-directional images. Second, since a fisheye lens is designed so that an incoming ray from a particular direction is projected onto

a particular point on an imaging plane, we do not have to concern ourselves with computing directions of incoming rays and considering a sampling frequency of the incoming rays. Third, we are also able to use the directions of the incoming rays for automatically constructing a geometric model of the scene with fisheye lens' wide field of view. By using omni-directional images, we are able to overcome the two difficulties that the previously proposed method encountered

Using an omni-directional stereo algorithm, the proposed method first constructs a geometric model of the scene from a pair of omni-directional images taken from different locations. Then radiance of the scene is computed from a sequence of omni-directional images taken with different shutter speeds and mapped onto the constructed geometric model. We refer to this geometric model with the radiance as a radiance map. The construction of a radiance map is necessary in order to compute a radiance distribution seen from any point in the scene. In other words, without constructing a radiance map, we can determine only the radiance distribution seen from the particular point where the omni-directional image was captured. To overcome this limitation, our method measures the radiance distribution of the scene as a triangular mesh. Once a radiance map is constructed as a triangular mesh, an appropriate radiance distribution can be used for rendering a virtual object and for generating shadows cast by the virtual object onto the real scene wherever the virtual object is placed in the scene.

For consistency of geometry, the camera calibration algorithm proposed by Tsai [19] is used for estimating the transformation between the 3D coordinate system of the real scene (*the world coordinate system*) and the 2D coordinate system of an image onto which the virtual objects are superimposed (*the image coordinate system*). This camera calibration algorithm takes into account various kinds of effects, including lens distortion and alignment of a CCD chip and the optical axis of a lens. As a result, the transformation can be estimated reliably.

In this work, we are not concerned with the consistency of time, and only static scenes are considered. We assume that there is no moving object in an image of the real scene, and therefore we do not have to coordinate virtual object motions and real object motions.

Our method superimposes virtual objects onto a real scene by the following three steps:

1. We define the world coordinate system in the real scene. Then we estimate the transformation between the world coordinate system and the image coordinate system.
2. We construct a radiance map of the real scene by using an omni-directional stereo algorithm.
3. We use the world-to-image transformation and the real radiance distribution to superimpose virtual objects with correct geometry and illumination onto an image of the real scene.

The rest of the paper is organized as follows. Section II explains how to determine the transformation between the

world coordinate system and the image coordinate system. Section III describes how to measure a radiance distribution of the real scene by using a pair of omni-directional images. Section IV explains how to superimpose virtual objects onto the real scene by using the world-to-image transformation and the measured radiance distribution. Section V shows experimental results of the proposed method applied to real images of both indoor and outdoor environments. Section VI presents concluding remarks.

## II. CONSISTENCY OF GEOMETRY

In this section, we describe how to define the world coordinate system in the real scene, and how to determine the transformation between the 3D coordinate system of the real scene and the 2D coordinate system of an image onto which virtual objects are superimposed.

### A. Definition of World Coordinate System

We take an image of the real scene by using a color CCD camera. Virtual objects are later superimposed onto the input image. Then we place a calibration board with regularly spaced dots, e.g., 81 dots in our experiments, in the scene. Without changing the camera setting, we take another image of the scene so that all dots on the calibration board appear in the calibration image. (For instance, see Fig. 5 (b).)

Using the calibration board, the world coordinate system is defined in the real scene such that a) the calibration board becomes a plane of  $z = 0$ , b) the center dot on the calibration board becomes the origin of the world coordinate system, and c) two edges of the calibration board are parallel to the  $x$ -axis and the  $y$ -axis of the world coordinate system. Once the world coordinate system is defined, we can place virtual objects at arbitrary locations.

### B. Transformation between the World Coordinate System and the Image Coordinate System

After defining the world coordinate system in the real scene, we estimate the transformation between the world coordinate system and the image coordinate system. For this estimation, we use the camera calibration algorithm proposed by Tsai [19], which is known to be able to estimate camera parameters reliably by taking into account various effects causing image distortion, e.g., radial distortion, displacement of the image center, and mismatching between camera and frame-grabber scanning rate.

Tsai's camera model gives a transformation between a 3D world coordinate system and a 2D image coordinate system, e.g., the projection of a 3D point in the scene onto the input image, generating a 3D ray extending from the camera projection center through an image pixel.

Using the transformation between the world coordinate system and the image coordinate system, we can compute where a virtual object appears in the input image once the object is placed in the real scene.

## III. CONSISTENCY OF ILLUMINATION

In the following subsections, we explain how to measure a radiance distribution of a real scene; this distribution will then be used for rendering virtual objects superimposed onto the real scene.

In Section III-A, we describe an omni-directional stereo algorithm for determining the 3D location of distinct features in the scene such as a fluorescent lamp on a ceiling. After the 3D locations of distinct features are obtained, the whole shape of the scene is approximated as a 3D triangular mesh whose vertices are the obtained features (Section III-B). In Section III-C, we describe how to estimate a scene radiance from an irradiance measured as the brightness of the omni-directional image. Then the brightness of the omni-directional image is used for determining the radiance distribution of the scene whose shape is obtained as a triangular mesh.

### A. Locations of Distinct Features from Omni-directional Stereo

A CCD camera with a fisheye lens<sup>3</sup> is used to take omni-directional images of the real scene. The camera is placed at two known locations in the scene to capture two omni-directional images from different locations.

The imaging system used in our method is illustrated in Fig. 1.  $C1$  and  $C2$  are the camera projection centers at each of the two locations. These two locations,  $C1$  and  $C2$ , are known a priori as a user places the camera at these locations. In this paper, we denote the omni-directional image taken at  $C1$  as  $FEV1$ , and the other one taken at  $C2$  as  $FEV2$ .

A real scene contains a very wide range of radiance. Therefore, due to the limited dynamic range of a CCD camera, pixel values of an image taken with one shutter speed cannot measure radiance in the scene accurately. To avoid this problem, multiple images taken with different shutter speeds are combined to produce each omni-directional image with a virtually extended dynamic range [7].

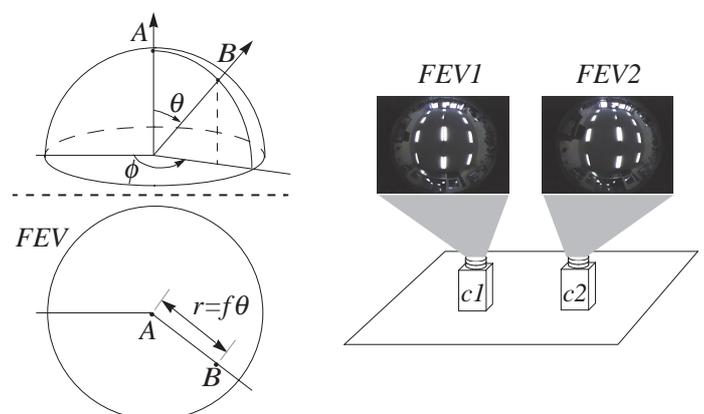


Fig. 1. omni-directional image acquisition system

<sup>3</sup>CCD color camera (Victor KYF-57) and fisheye lens (Fit Corporation FI-19 with field of view of 180 degrees)

The fisheye lens used in the image acquisition system is designed so that an incoming ray to the lens is projected onto an imaging plane as

$$r = f\theta \quad (1)$$

where  $f$  is the focal length of the lens,  $r$  is the distance between the image center and the projection of the ray, and  $\theta$  is an incident angle of the ray (Fig. 1).

Using this projection model, the incident angle of the ray is given as  $\theta = r/f$ , where  $r$  is determined from the image coordinate of the point corresponding to the ray. For instance, if a direct light source appears as a point in the omni-directional image, the direction from the camera projection center to the light source is determined from the image coordinate of the point in the image.

Most of the incoming light energy in a real scene comes from direct light sources such as a fluorescent lamp or a window to the outside, while the rest of the incoming light energy comes from indirect illumination such as reflection from a wall. For this reason, it is important to know the accurate locations of direct light sources to represent an illumination distribution of a real scene.

Fortunately, direct light sources usually appear as significantly bright points in an omni-directional image. Therefore, it should be relatively easy to identify direct light sources in the image. In our method, we extract feature points with high contrast in the two omni-directional images  $FEV1$  and  $FEV2$  by using the feature extraction algorithm proposed by Tomasi and Kanade [17]. In the algorithm, an image pixel with high gradient values in two orthogonal directions, e.g., a corner point, is extracted as a feature point.

After feature points are extracted in  $FEV1$  and  $FEV2$ , 3D coordinates of points in the real scene corresponding to the extracted feature points are determined by using a stereo algorithm [9].

Consider a 3D line that extends from the camera projection center through the extracted feature point on the image plane. The 3D line is given as

$$L1(t) = C_1 + t\mathbf{v}_{1i} \quad (i = 1, 2, \dots, N) \quad (2)$$

$$L2(s) = C_2 + s\mathbf{v}_{2j} \quad (j = 1, 2, \dots, N) \quad (3)$$

where  $N$  is the number of the extracted feature points,  $C1$  and  $C2$  are the camera projection centers,  $\mathbf{v}_{1i}$  and  $\mathbf{v}_{2j}$  are the directional vectors of the 3D line, and  $t$  and  $s$  are scalar values of the line for  $FEV1$  and  $FEV2$ , respectively.

As described in the beginning of this section,  $C1$  and  $C2$  are known a priori, and  $\mathbf{v}_{1i}$  and  $\mathbf{v}_{2j}$  are given as

$$\mathbf{v} = [\sin\theta \cos\phi, \sin\theta \sin\phi, \cos\theta]. \quad (4)$$

Hence, the two lines  $L1(t)$  and  $L2(s)$  are defined uniquely for the extracted feature points.

Once we have obtained 3D lines corresponding to the extracted feature points in  $FEV1$  and  $FEV2$ , we establish

correspondence of the 3D lines between  $FEV1$  and  $FEV2$ . If a feature point  $i$  in  $FEV1$  and a feature point  $j$  in  $FEV2$  correspond to the same 3D point in the scene, the 3D lines obtained from the two feature points  $L1(t)$  and  $L2(s)$  must intersect. Then a coordinate of the 3D point is determined from the intersection of the two lines.

However, due to various kinds of errors such as an error in  $C1$  and  $C2$ , and an error in  $\mathbf{v}_{1i}$  and  $\mathbf{v}_{2j}$ , the two lines may not intersect. Therefore, we consider that the two lines intersect if the distance between the two lines is sufficiently short. The distance between the two 3D lines is given by substituting  $t$  and  $s$  from the following equations into Equation (2) and Equation (3) [11].

$$t = \frac{\text{Det}\{(C_2 - C_1), \mathbf{v}_{2i}, \mathbf{v}_{1i} \times \mathbf{v}_{2j}\}}{|\mathbf{v}_{1i} \times \mathbf{v}_{2j}|^2} \quad (5)$$

$$s = \frac{\text{Det}\{(C_2 - C_1), \mathbf{v}_{1i}, \mathbf{v}_{1i} \times \mathbf{v}_{2j}\}}{|\mathbf{v}_{1i} \times \mathbf{v}_{2j}|^2} \quad (6)$$

If a 3D line from one omni-directional image, e.g.,  $FEV1$ , has multiple candidates for a matching line from the other omni-directional image, e.g.,  $FEV2$ , we select a matching line in  $FEV2$ , so that the feature point in  $FEV2$  has a pixel coordinate closest to that of the feature point in  $FEV1$ .

### B. The Entire Shape of the Scene

In the previous section, we described how to determine 3D coordinates of distinct points in the real scene such as a fluorescent lamp on a ceiling and a window to the outside. However, 3D coordinates of the remaining part of the real scene cannot be determined in the same manner. This part includes a wall, a ceiling, and other object surfaces that act as indirect light sources and therefore do not appear as distinct points in omni-directional images.

In our method, 3D coordinates for that part are approximated by using the 3D coordinates of distinct features in the scene. In particular, we generate a 3D triangular mesh by using the distinct feature points. First, we construct a 2D triangular mesh by applying 2D Delaunay triangulation to the extracted feature points in  $FEV1$ . That determines the connectivity of a 3D triangular mesh whose vertices are the 3D points corresponding to the feature points in  $FEV1$ . Then, using the connectivity, a 3D triangular mesh is created from the 3D feature points.

The obtained triangular mesh approximates an entire shape of the real scene, e.g., the ceiling and walls of a room, which act as direct or indirect light sources.

Note that the fisheye lens we used in our experiments has a field of view of 180 degrees, and therefore we can obtain only the upper half of the scene model from a pair of the omni-directional images. Thus, a horizontal ground plane is assumed and added to the model as a lower half of the model in our experiments. However, if it is necessary, we are able to obtain the lower half of the model by using another omni-directional image pair which captures the left part of the scene.

### C. Radiance Distribution of the Scene

After the shape of the real scene is obtained as a triangular mesh, the radiance of the scene is estimated by using the brightness of the omni-directional images.

First, we explain the relationship between the radiance at a point on an object and the irradiance at the corresponding point in an omni-directional image. The radiance  $L$  of a small patch in the real scene ( $\delta O$  in Fig. 2) can be computed from the irradiance  $E_c$  at a small patch on the imaging plane  $\delta I$ . This irradiance on the imaging plane is measured as a pixel brightness in *FEV1* or *FEV2*.

As shown in Fig. 2, consider a small patch on a real object surface  $\delta O$  that acts as a light source, and a small patch  $\delta I$  on an imaging plane. In Fig. 2,  $d$  is the diameter of the fisheye lens,  $f$  is the distance from  $\delta I$  to the lens,  $z$  is the distance from the lens to  $\delta O$ ,  $\alpha$  is the angle between the surface normal at  $\delta O$  and the ray from  $\delta O$  to the lens center,  $\theta$  is the angle between the lens optical axis and the ray from  $\delta O$  to the lens center, and  $\beta$  is the angle between the lens optical axis and the ray from  $\delta I$  to the lens center.

The apparent area of the patch  $\delta I$  as seen from the center of the lens is  $\delta I \cos \beta$ , while the distance of this patch from the center of the lens is  $f/\cos \beta$ . Thus, the solid angle subtended by the patch  $\delta I$  is  $\delta I \cos \beta / (f/\cos \beta)^2$ .

Similarly, the solid angle of the patch  $\delta O$  as seen from the center of the lens is  $\delta O \cos \alpha / (z/\cos \theta)^2$ .

Since the ratio of the two solid angles is  $(\frac{d\beta}{d\theta})^2$ , we must have

$$\frac{\delta I \cos \beta}{(f/\cos \beta)^2} = \left(\frac{d\beta}{d\theta}\right)^2 \frac{\delta O \cos \alpha}{(z/\cos \theta)^2} \quad (7)$$

From the projection model of the fisheye lens, we have  $\beta = \tan^{-1} \theta$  and  $\frac{d\beta}{d\theta} = \frac{d \tan^{-1} \theta}{d\theta} = \frac{1}{1+\theta^2}$ . Equation (7) becomes

$$\frac{\delta O}{\delta I} = \frac{\cos^3(\tan^{-1} \theta)}{\cos \alpha \cos^2 \theta} \left(\frac{z}{f}\right)^2 (1 + \theta^2)^2. \quad (8)$$

Next, we need to determine how much of the light emitted by the surface makes its way through the lens. The solid angle subtended by the lens, as seen from the patch  $\delta O$ , is

$$\Omega = \frac{\pi}{4} \frac{d^2 \cos \theta}{(z/\cos \theta)^2} = \frac{\pi}{4} \left(\frac{d}{z}\right)^2 \cos^3 \theta. \quad (9)$$

Thus, the power of the light originating on the patch and passing through the lens is

$$\delta P = L \delta O \cos \alpha \Omega = L \delta O \cos \alpha \frac{\pi}{4} \left(\frac{d}{z}\right)^2 \cos^3 \theta \quad (10)$$

where  $L$  is the radiance of the surface in the direction toward the lens.

Since no light from other areas in the scene reaches the patch  $\delta I$ , we have

$$E_c = \frac{\delta P}{\delta I} = L \frac{\delta O}{\delta I} \frac{\pi}{4} \left(\frac{d}{z}\right)^2 \cos^3 \theta \cos \alpha \quad (11)$$

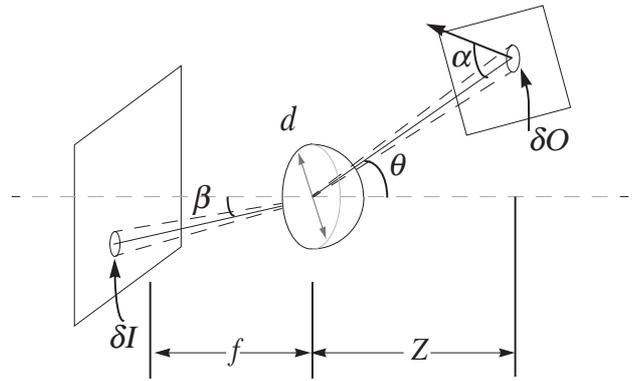


Fig. 2. scene radiance and image irradiance

Substituting Equation (8) into Equation (11), we finally obtain the equation that represents the relationship between the irradiance on the imaging plane and the radiance on the object surface in the scene:

$$E_c = L \frac{\pi}{4} \left(\frac{d}{f}\right)^2 (1 + \theta^2)^2 \cos^3(\tan^{-1} \theta) \cos \theta \quad (12)$$

However, this equation cannot be used to measure irradiance values in the unit of  $Wm^{-2}$  because the equation does not take into account other factors such as D/A and A/D conversions in a CCD camera and a frame grabber. By compensating for  $(1 + \theta^2)^2 \cos^3(\tan^{-1} \theta) \cos \theta$ , we can measure values proportional to real irradiance. The scaling factor can be determined by measuring brightness of the same light source, using both a CCD camera and a photometer, a task which was not performed in this work.

Next, we project the omni-directional image *FEV1* onto the 3D triangular mesh to assign color texture to the mesh. The assigned color texture at a point on the triangular mesh represents the irradiance of the corresponding point in *FEV1*. Therefore, by converting values of the assigned color texture into the radiance  $L$  using Equation (12), we finally obtain the radiance distribution of the scene.

## IV. SUPERIMPOSING VIRTUAL OBJECTS ONTO A REAL SCENE

In the previous sections, we described how to determine the transformation between the world coordinate system and the image coordinate system (consistency of geometry in Section II), and how to measure a radiance distribution of the real scene (consistency of illumination in Section III). In this section, we explain how to superimpose virtual objects onto the real scene by using the transformation and the measured radiance distribution.

First, we describe how to use the radiance map to compute the total irradiance at a point on a virtual object surface or a point on a real object surface. Then, we explain how to compute the color of a virtual object surface and how to generate the shadow cast by a virtual object using the computed irradiance.

It should be noted that superimposing virtual objects without knowing the shapes and reflectance properties of

all nearby objects results in the fact that we cannot model the interreflections between the real objects and the virtual objects. To take the interreflections into account, we need to compute the global illumination in the same manner as in Fournier et al.'s method [10]. However, it is required to define or estimate shapes and reflectance properties of all nearby objects in order to do so. Rather, in this paper, we make the rendering process simple, and we consider only the emitted light from the radiance map.

### A. Total Irradiance from Real Illumination

For rendering a surface of a virtual object and for generating the shadow cast by a virtual object, a total irradiance at the surface from the radiance map has to be obtained.

Consider an infinitesimal patch of the extended light source, of size  $\delta\theta_i$  in polar angle and  $\delta\phi_i$  in azimuth (Fig. 3). Seen from the center point  $A$ , this patch subtends a solid angle  $\delta\omega = \sin\theta_i\delta\theta_i\delta\phi_i$ . If we let  $L(\theta_i, \phi_i)$  be the radiance per unit solid angle coming from the direction  $(\theta_i, \phi_i)$ , then the radiance from the patch under consideration is  $L(\theta_i, \phi_i) \sin\theta_i\delta\theta_i\delta\phi_i$ , and the total irradiance of the surface is

$$E = \int_{-\pi}^{\pi} \int_0^{\frac{\pi}{2}} L(\theta_i, \phi_i) \cos\theta_i \sin\theta_i d\theta_i d\phi_i \quad (13)$$

where the radiance  $L$  is measured as shown in Equation (12) [12].

To compute the irradiance  $E$ , the double integral in Equation (13) needs to be approximated by discrete sampling over the entire hemisphere. In our method, nodes of a geodesic dome [4] are used for the discrete sampling. For each of the nodes, the radiance of a corresponding point in the real scene is used as the radiance of the node. Consider a ray from a point on a virtual object surface to the node. A color texture at an intersection of the ray and the 3D triangular mesh described in Section III-B is obtained as the radiance.

Nodes of a geodesic dome are uniformly distributed over the surface of a sphere. Therefore, by using  $N$  nodes of a geodesic dome in a northern hemisphere as a sampling direction, the double integral in Equation (13) can be approximated as a sampling at equal solid angle  $\delta\omega = 2\pi/N$ . The number of the nodes  $N$  can be adjusted by changing the sampling frequency of a geodesic dome.

Using the discrete sampling, Equation (13) can be approximated as

$$E = \sum_{i=0}^N \frac{2\pi}{N} L(\theta_i, \phi_i) \cos\theta_i. \quad (14)$$

Note that a radiance does not depend on the distance between a viewpoint and a light source. Therefore, the distance from the point on the virtual object to the real scene does not affect the radiance. Also, we assume that the real scene reflects or emits light as a perfect Lambertian plane. That is to say, at each surface point in the scene, light energy is emitted equally in all directions. Due to this assumption, we do not consider directional light sources

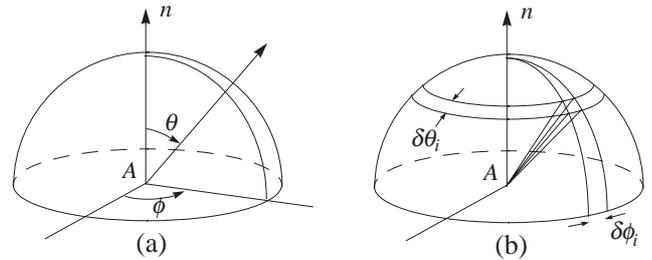


Fig. 3. (a) the direction of incident and emitted light rays (b) infinitesimal patch of an extended light source

such as spotlights in our method. However, by combining multiple omni-directional images taken from different locations, our method could be extended to model changes of a radiance, depending on viewing directions.

### B. Ray Casting for Superimposing Virtual Objects

For superimposing virtual objects onto an input image of a real scene, the ray casting algorithm is used as follows.

1. For each pixel in the input image of the real scene, a ray extending from the camera projection center through the pixel is generated by using the transformation between the world coordinate system and the image coordinate system. Then it is determined whether a ray intersects a virtual object or a real object in the scene, e.g., a tabletop.
2. If the ray intersects a virtual object, we consider that the pixel corresponds to a point on the virtual object surface. Then we compute a color to be observed at the surface point under the measured radiance distribution of the real scene using a reflection model. The computed color is stored in the pixel as the surface color of the virtual object at the pixel.
3. Otherwise, we consider that the pixel corresponds to a point on a real object surface. Then we modify an observed color at the point on the real object surface, so that a shadow cast by the virtual object onto the real object is generated.

We are not concerned here with the problem of occlusion because it is beyond the scope of this paper to measure accurate 3D shapes of real objects.<sup>4</sup> We assume that a virtual object always exists between the camera projection center and real objects in the scene. In other words, seen from the camera projection center, a virtual object exists only in front of real objects, and a real object never occludes the view of the virtual object. If a more dense 3D depth map is obtainable from the process of constructing a radiance map, we can make full use of the depth map for the problem of occlusion.

<sup>4</sup>Techniques for determining correct occlusion between virtual and real objects using the shapes of the real objects are called Z-key. For instance, see [13].

### C. Rendering Virtual Object Surface

As described in the previous section, if a ray through an image pixel intersects a virtual object, the color at the intersection under the measured radiance distribution is computed, and the color is stored at the image pixel as the color of the virtual object surface.

For computing the color on the virtual object surface, a simplified Torrance-Sparrow reflection model [14], [18] was used in our method. The reflection model for the radiance distribution shown in Equation (14) is represented as

$$I_m = K_{d,m} \sum_{i=0}^N L_m(\theta_i, \phi_i) S(\theta_i, \phi_i) \cos\theta_i \quad (15)$$

$$+ K_{s,m} \sum_{i=0}^N L_m(\theta_i, \phi_i) S(\theta_i, \phi_i) \frac{1}{\cos\theta_r} e^{-\frac{\gamma(\theta_i, \phi_i)^2}{2\sigma^2}}$$

$m = R, G, B$

where  $\theta_i$  is the angle between the surface normal and the direction to each point light source,  $\theta_r$  is the angle between the surface normal and the viewing direction,  $\gamma(\theta_i, \phi_i)$  is the angle between the surface normal and the bisector of the light source direction and the viewing direction,  $S(\theta_i, \phi_i)$  are shadow coefficients where  $S(\theta_i, \phi_i) = 0$  if other surface point on the virtual object occludes  $L(\theta_i, \phi_i)$ , and  $S(\theta_i, \phi_i) = 1$  otherwise.  $K_{d,m}$  and  $K_{s,m}$  are constants for the diffuse and specular reflection components, and  $\sigma$  is the standard deviation of a facet slope of the Torrance-Sparrow reflection model.  $K_{d,m}$  and  $K_{s,m}$  are simply given for virtual objects, or they can be determined for real objects by using Sato et al.'s method [15]. Also, the constant  $2\pi/N$  in Equation (14) is included in the constants  $K_{d,m}$  and  $K_{s,m}$  in Equation (15).

### D. Soft Shadow Cast by a Virtual Object

If a ray through an image pixel does not intersect with a virtual object, the image pixel corresponds to a point on a real object surface. The virtual object may occlude some of incoming light to the point on the object surface. Thus, the color of the image pixel needs to be modified, so that a shadow cast by the virtual object is created on the real object surface.

A shadow cast by a virtual object is created as follows:

1. Obtain a 3D coordinate of a point on a real object surface where a ray through an image pixel intersects the real object. We know the shape of the plane onto which a virtual object is placed, because a plane of  $z = 0$  is defined on the plane as explained in Section II-A. Hence, we compute an intersection between the ray and the plane of  $z = 0$ , and generate shadows cast on the plane. In this work, we do not deal with shadows cast by virtual objects onto real objects other than the plane. However, our method can be easily extended to handle other real objects if their shapes are obtained from the process of constructing a radiance map or by other means.

2. Compute a total irradiance  $E_1$  at the surface point from the radiance distribution of the real scene. In this

case, a virtual object does not occlude incoming light at the surface point (Fig. 4.a). As described in Section IV-A, the total irradiance  $E_1$  can be computed using Equation (14).

3. With a virtual object placed in the real scene, compute a total irradiance  $E_2$  at the surface point that is not occluded by the virtual object. In this case, the virtual object occludes some of the irradiance from the real scene (Fig. 4.b). As a result, the total irradiance  $E_2$  becomes smaller than the total irradiance  $E_1$ . The total irradiance  $E_2$  can be obtained as

$$E_{2,m} = \sum_{i=0}^N \frac{2\pi}{N} S(\theta_i, \phi_i) L_m(\theta_i, \phi_i) \cos\theta_i \quad m = R, G, B \quad (16)$$

where  $S(\theta_i, \phi_i) = 0$  if the virtual object occludes  $L(\theta_i, \phi_i)$ , and  $S(\theta_i, \phi_i) = 1$  otherwise.<sup>5</sup>

4. Compute the ratio of the total radiance  $E_2$  to the total radiance  $E_1$ . Then multiply the ratio  $E_2/E_1$  to the color at the intersection between the ray and the plane of  $z = 0$ . The ratio represents how much of the irradiance at the intersection would still be preserved if the virtual object were placed in the scene.

By multiplying the ratio  $E_2/E_1$  to the observed color of the image pixel  $I_m$ , we obtain the color  $I'_m$  that would be the color of the image pixel if there were a virtual object. A similar discussion can be found also in [10].

$$I'_m = I_m \frac{E_{2,m}}{E_{1,m}} \quad m = R, G, B \quad (17)$$

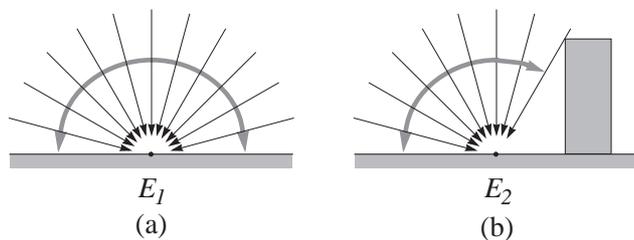


Fig. 4. total irradiance (a) without virtual objects (b) with virtual objects

## V. EXPERIMENTAL RESULTS

We have tested the proposed method by using real images taken in both indoor and outdoor environments. First, we described experimental results for an indoor environment in Section V-A, and then in Section V-B we presented experimental results for an outdoor environment.

### A. Experimental Results for an Indoor Scene

An image of a tabletop and miscellaneous objects on the tabletop in our laboratory was taken. From the same cam-

<sup>5</sup> Alternatively, instead of just blocking the radiance  $L(\theta_i, \phi_i)$ , the scene radiance of the blocking surface point could be used as a secondary light source with its own radiance.

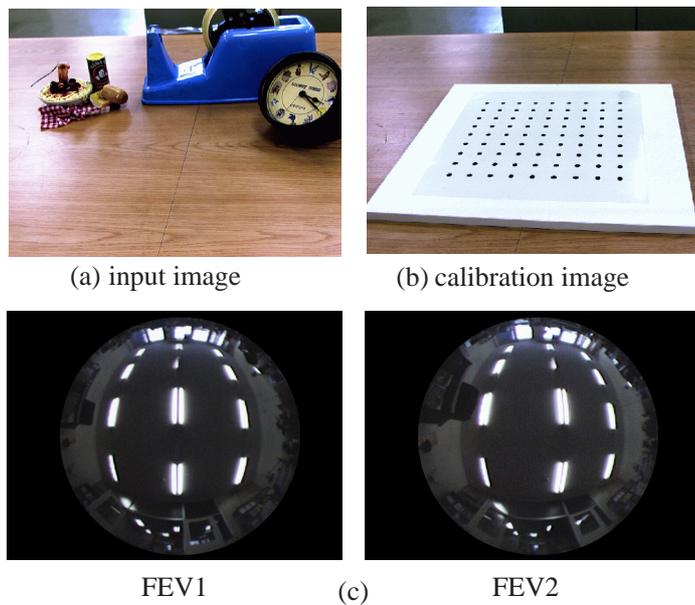


Fig. 5. (a) input image (b) calibration image (c) omnidirectional images



Fig. 6. measured radiance distribution

era position, another image of the tabletop with a calibration board was taken. The input image and the calibration image are shown in Fig. 5 (a) and (b).

First, regularly spaced dots on the calibration board were extracted in the calibration image to determine their 2D image coordinates. From pairs of the 2D image coordinates and the 3D world coordinates that were given a priori, the transformation between the world coordinate system and the image coordinate system was estimated by using the camera calibration algorithm as described in Section II.

By using the imaging system illustrated in Fig. 1, two omnidirectional images of the scene, e.g., the ceiling of the laboratory in this experiment, were taken. Fig. 5 (c) shows the two omnidirectional images.

First, feature points were extracted from each of the omnidirectional images as described in Section III-A. Then

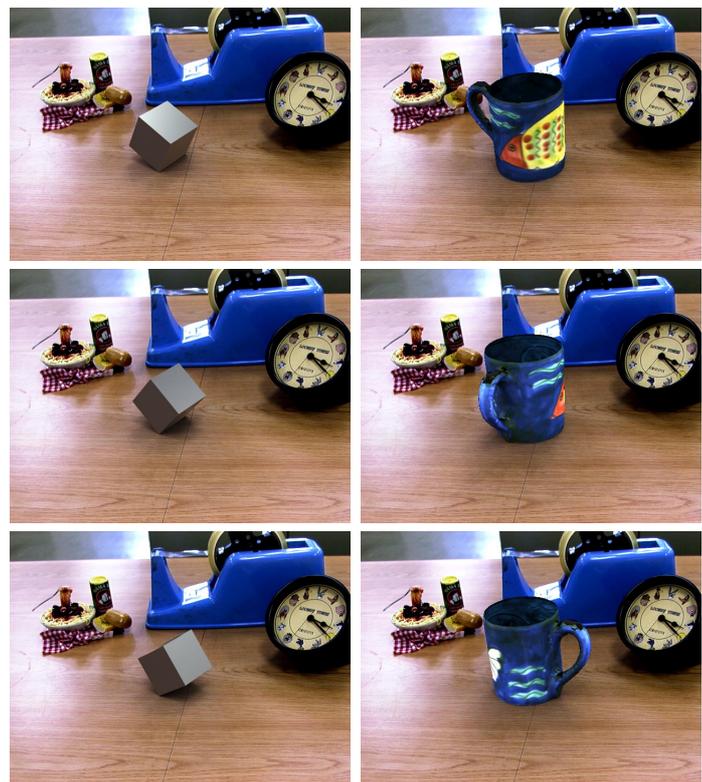


Fig. 7. images synthesized with our method



Fig. 8. images synthesized with our method: appearance changes observed on a metallic hemisphere

pairs of corresponding feature points in the two omnidirectional images were found to determine 3D coordinates of the feature points. Finally, a triangular mesh was constructed from the feature points to represent the radiance distribution of the scene as explained in Section III-B. However, due to the lack of distinct feature points along the floor in the scene, the obtained triangular mesh did not include vertices along the floor. Therefore, vertices representing the floor were automatically added afterward based on the assumption that the size of the floor was approximately the same as the size of the reconstructed ceiling.

Fig. 6 shows the obtained triangular mesh that represents the radiance distribution as its color texture, called a radiance map. The reconstructed scene model shows the size of the room within an error of 8 percent.

Using the world-to-image transformation and the radiance distribution of the scene, a virtual object was superimposed onto the input image of the scene as explained in Section IV. To generate the images in Fig. 7, 1048 nodes of a geodesic dome were used for sampling the radiance

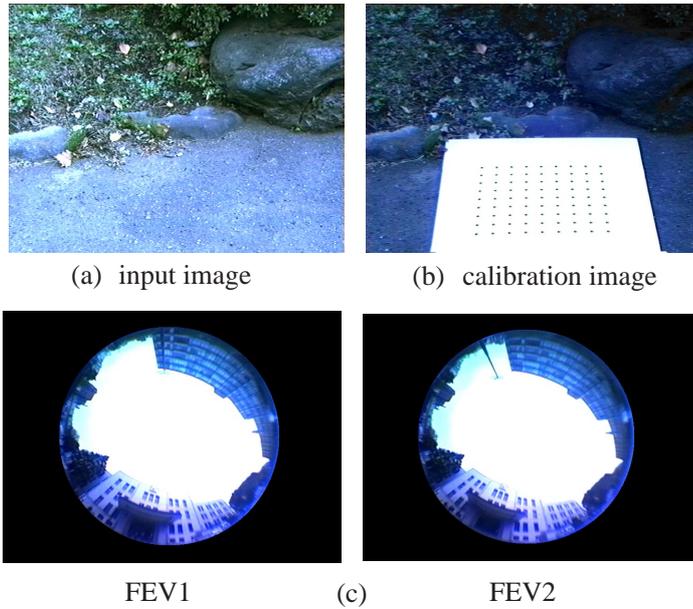


Fig. 9. (a) input image (b) calibration image (c) omni-directional images

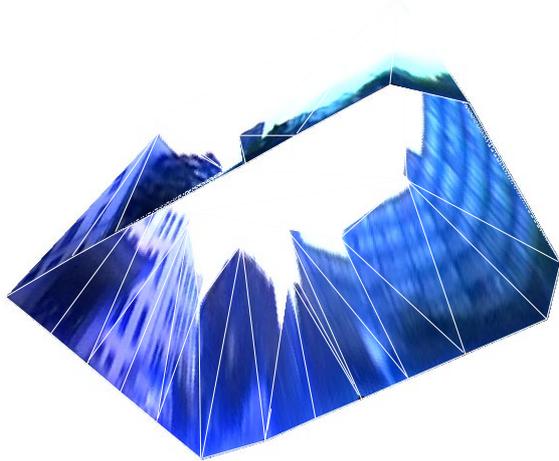


Fig. 10. measured radiance distribution

distribution from the scene.<sup>6</sup>

In the images synthesized by our method, shading of the virtual object blends well into the scene. Also, the virtual object casts a shadow with a soft edge on the tabletop in the same way as do the other objects in the scene.

To demonstrate the effectiveness of constructing a radiance map as a geometric model of the scene with its radiance value, we superimposed a metallic hemisphere at three different locations on the tabletop. The synthesized results are shown in Figure 8. Using the geometric information of the radiance map, an appropriate radiance distribution is

<sup>6</sup>The model of the mug used in the images was created by using the method proposed by Sato et al.[15].

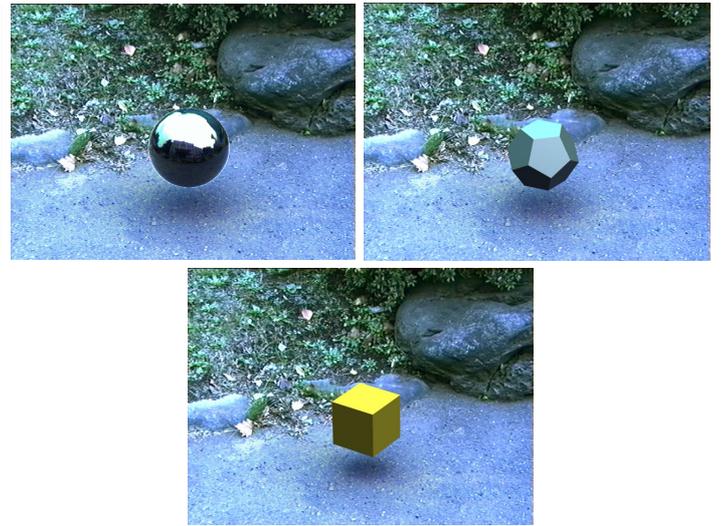


Fig. 11. images synthesized with our method

computed at each location of the object and is used for superimposing the object, along with correct shadings, onto the scene. As a consequence, appearance changes are observed on the object surface as it moves on the tabletop.

### B. Experimental Results for an Outdoor Scene

We also applied our method to real images taken in an outdoor environment. The input images used in this experiment are shown in Fig. 9.

In the same way as with the indoor images, the transformation between the world coordinate system and the image coordinate system was estimated by using the calibration image. Then, the two omni-directional images of the outdoor scene were used to compute the radiance distribution of the scene. The obtained radiance distribution is shown as a color-textured triangular mesh in Fig. 10.

From this figure, we can see that the shape of the outdoor scene was captured reasonably well. In the radiance distribution as a color-textured mesh, the sky is represented as a planar region extending between the tops of the surrounding buildings because the sky did not contain any distinct feature points to be used as vertices of the triangular mesh.

Using the radiance distribution of the outdoor scene, several virtual objects were superimposed onto the input image of the scene. Fig. 11 shows the superimposed images. The virtual objects that we used in this experiment are a shiny metallic sphere, a rough metallic dodecahedron, and a yellow matte cube, all of which are floating at the same location in the outdoor scene. We can see that shadings and shadows are represented appropriately in the synthesized images by using our method. In particular, the reflection of the surrounding buildings appears appropriately on the metallic sphere. Unless the real radiance distribution of the scene is used as in our method, a convincing reflection on a virtual object as shown in this figure cannot be created.

## VI. CONCLUSIONS

In this paper, we proposed a new method for superimposing virtual objects onto an image of a real scene by taking into account the radiance distribution of the scene.

In our method, a camera calibration algorithm is used for matching geometry between virtual objects and the real scene. For matching illumination, the radiance distribution of the real scene is measured by using two omni-directional images of the scene.

Unlike the previously proposed methods, our method can automatically measure an entire radiance distribution of the scene by using multiple omni-directional images. As a result, our method can superimpose virtual objects with convincing shadings and shadows onto the real scene. In addition, we obtain the radiance distribution as a triangular mesh representing the approximate shape of the scene. Therefore, a correct radiance distribution can be used for rendering virtual objects and generating shadows cast by the virtual objects wherever the objects are placed in the real scene.

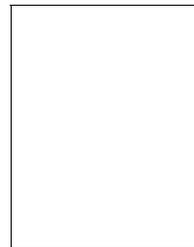
To demonstrate the effectiveness of the proposed method, we have successfully tested our method by using real images taken in both indoor and outdoor environments.

## ACKNOWLEDGMENTS

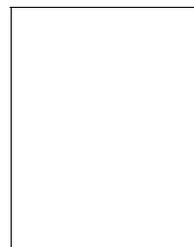
This research was supported in part by the Ministry of Education, Science, Sports and Culture grant-in-Aid for Creative Basis Research 09NP1401 and grant-in-Aid for Scientific Research (B) 09450164. We are also grateful to the anonymous reviewers whose suggestions helped to improve the paper.

## REFERENCES

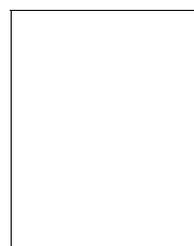
- [1] R. T. Azuma, "A survey of augmented reality," *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 4, pp. 355-385, August 1997.
- [2] R. T. Azuma and G. Bishop, "Improving static and dynamic registration in an optical see-through HMD," *Proc. SIGGRAPH 94*, pp. 197-204, July, 1994.
- [3] M. Bajura, H. Fuchs, and R. Ohbuchi, "Merging virtual objects with the real world: seeing ultrasound imagery within the patient," *Proc. SIGGRAPH 92*, pp. 203-210, 1992.
- [4] D. H. Ballard & C. M. Brown, *Computer Vision*, Prentice-Hall, Englewood Cliffs, NJ., 1982.
- [5] M.F. Cohen, S.E. Chen, J.R. Wallace, D.P. Greenberg, "A Progressive Refinement Approach to Fast Radiosity Image Generation," *Proc. SIGGRAPH 88*, pp. 75-84, 1988.
- [6] P. E. Debevec, "Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography," *Proc. SIGGRAPH 98*, pp. 189-198, July, 1998.
- [7] P. E. Debevec and J. Malik, "Recovering high dynamic range radiance maps from photographs," *Proc. SIGGRAPH 97*, pp. 369-378, August, 1997.
- [8] G. Drettakis, L. Robert, S. Bounoux, "Interactive Common Illumination for Computer Augmented Reality" *Proc. 8th Eurographics Workshop on Rendering*, pp. 45-57, 1997.
- [9] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, Cambridge, MA., 1993.
- [10] A. Fournier, A. Gunawan and C. Romanzin, "Common Illumination between Real and Computer Generated Scenes," *Proc. Graphics Interface '93*, pp.254-262, 1993.
- [11] A. S. Glassner, *Graphics Gems*, AP Professional, Cambridge, MA., 1990.
- [12] B. K. P. Horn, *Robot Vision*, The MIT Press, Cambridge, MA., 1986.
- [13] T. Kanade, A. Yoshida, K. Oda, H. Kano, and M. Tanaka, "A Video-Rate Stereo Machine and Its New Applications," *Proc. IEEE Conf. Computer Vision and Pattern Recognition 96*, pp.196-202, 1996.
- [14] S. K. Nayar, K. Ikeuchi, and T. Kanade, "Surface reflection: physical and geometrical perspectives," *IEEE Trans. PAMI*, vol. 13, no. 7, pp. 611-634, 1991.
- [15] Y. Sato, M. D. Wheeler, and K. Ikeuchi, "Object shape and reflectance modeling from observation," *Proc. of SIGGRAPH 97*, pp. 379-387, August, 1997.
- [16] A. State, G. Hirota, D. T. Chen, W. F. Garrett, and M. A. Livingston, "Superior augmented-reality registration by integrating landmark tracking and magnetic tracking," *Proc. SIGGRAPH 96*, pp. 429-438, August, 1996.
- [17] C. Tomasi and T. Kanade, "Shape and Motion from Image Streams under Orthography: a Factorization Method," *Int'l J. Computer Vision*, vol. 9, no. 2, pp. 137-154, 1992.
- [18] K. E. Torrance and E. M. Sparrow, "Theory for off-specular reflection from roughened surface," *J. Optical Society of America*, vol.57, pp.1105-1114, 1967.
- [19] R. Tsai, "A Versatile Camera Calibration Technique for High Accuracy Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," *IEEE J. Robotics and Automation*, vol. 3, no. 4, pp. 323-344, 1987.



**Imari Sato** is an assistant researcher at the Institute of Industrial Science, the University of Tokyo, Tokyo, Japan. She received a BS in Policy Management from Keio University, Kanagawa, Japan, in 1994. During her junior year she participated in the Tipster text extraction project at the Center for Machine Translation at Carnegie Mellon University as a research assistant. After working at the Robotics Institute at Carnegie Mellon University, she joined the University of Tokyo, in 1997.



**Yoichi Sato** is an assistant professor at the Institute of Industrial Science, the University of Tokyo, Tokyo, Japan. He received a BS in Mechanical Engineering from the University of Tokyo in 1990. He received a MS in Robotics in 1993 and a PhD in Robotics from the School of Computer Science, Carnegie Mellon University in 1997. His primary research interests are computer vision (physics-based computer vision, reflectance analysis for 3D object model generation), computer graphics (virtual reality and augmented reality) and human-computer interaction.



**Katsushi Ikeuchi** is a professor at the Institute of Industrial Science, the University of Tokyo, Tokyo, Japan. He received a B. Eng. in Mechanical Engineering from Kyoto University, Kyoto, Japan, in 1973, and a Ph.D in Information Engineering from the University of Tokyo, Tokyo, Japan, in 1978. After working at the Artificial Intelligence Laboratory at Massachusetts Institute of Technology, the Electrotechnical Laboratory of Ministry of International Trade and Industries, and

the School of Computer Science, Carnegie Mellon University, he joined the University of Tokyo, in 1996. He has received various research awards, including the David Marr Prize in computational vision in 1990, and IEEE R&A K-S Fu Memorial Best Transaction Paper award in 1998. In addition, in 1992, his paper, "Numerical Shape from Shading and Occluding Boundaries," was selected as one of the most influential papers to have appeared in the Artificial Intelligence Journal within the past ten years.