Consistent Illumination within Optical See-Through Augmented Environments

Oliver Bimber, Anselm Grundhöfer, Gordon Wetzstein and Sebastian Knödel
Bauhaus University
Bauhausstraße 11, 99423 Weimar, Germany,
{oliver.bimber, anselm.grundhoefer, gordon.wetzstein, sebastian.knoedel}@medien.uni-weimar.de

Abstract

We present techniques which create a consistent illumination between real and virtual objects inside an application specific optical see-through display: the Virtual Showcase. We use projectors and cameras to capture reflectance information from diffuse real objects and to illuminate them under new synthetic lighting conditions. Matching direct and indirect lighting effects, such as shading, shadows, reflections and color bleeding can be approximated at interactive rates in such a controlled mixed environment.

1. Introduction

To achieve a consistent lighting situation between real and virtual environments is important for convincing augmented reality (AR) applications.

A rich pallet of algorithms and techniques have been developed that match illumination for video- or image-based augmented reality. However, very little work has been done in this area for optical see-through AR. For the reasons that are discussed in [1], we believe that the optical see-through concept is currently the most advanced technological approach to provide an acceptable level of realism and interactivity.

The Virtual Showcase [2] is an application specific optical see-through display. Figure 1 illustrates our latest prototype. It consists of up to four tilted CRT screens that are reflected by a pyramid-shaped mirror beam splitter. Wireless infrared tracking determines the observers' perspectives to render high-resolution[1] stereoscopic graphics onto the screens. Video projectors are mounted under its roof and allow a pixel-precise illumination of the real content [3]. Between two and three networked off-the-shelf PCs that are integrated into the Virtual Showcase's frame are used to drive the display. Beside its high resolution, the Virtual Showcase also provides a dark and well controllable environment that makes a consistent illumination of real and virtual components easier to achieve than in real world environments.

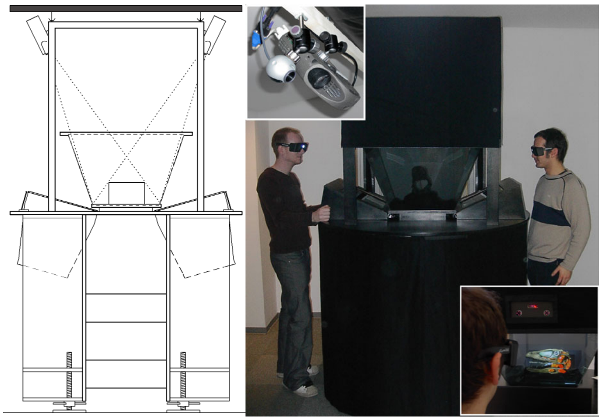

Figure 1: Virtual Showcase prototype with cameras and projectors.

The contribution of this paper is the introduction of methods which create a consistent illumination between real and virtual components within an optical see-through environment – such as the Virtual Showcase. Combinations of video projectors and cameras are applied to capture reflectance information from diffuse real objects and to illuminate them under new synthetic lighting conditions. For diffuse objects, the capturing process can also benefit from hardware acceleration – supporting dynamic update rates. To handle indirect lighting effects (like color bleeding) an off-line radiosity procedure is outlined that consists of multiple rendering passes. For direct lighting effects (such as simple shading, shadows and reflections) hardware accelerated techniques are described which allow to achieve interactive frame rates. The reflectance information is used in addition to solve a main problem of a previously introduced technique which creates consistent occlusion effects for multiple users within such environments [3].

[1] Currently UXGA per user.

2. Related Work

Inspired by the pioneering work of Nakamae et al. [16] and –later– Fournier et al. [9], many researchers have approached to create consistent illumination effects while integrating synthetic objects into a real environment. To our knowledge, all of these approaches represent the real environment in form of images or videos. Consequently, mainly image processing, inverse rendering, inverse global illumination, image-based and photo-realistic rendering techniques are applied to solve this problem. Due to the lack of real-time processing, these approaches are only applicable in combination with desktop screens and an unresponsive[2] user interaction. Devices that require interactive frame-rates, such as head-tracked personal or spatial displays, cannot be supported. Representative for the large body of literature that exists in this area, we want to discuss several more recent achievements:

Boivin et al. [5] present an interactive and hierarchical algorithm for reflectance recovery from a single image. They assume that the geometric model of the scenery and the lighting conditions within the image are known. Making assumptions about the scene's photometric model, a virtual image is generated with global illumination techniques (i.e., ray-tracing and radiosity). This synthetic image is then compared to the photograph and a photometric error is estimated. If this error is too large, their algorithm will use a more complex BRDF model (step by step – using diffuse, specular, isotropic, and finally anisotropic terms) in the following iterations, until the deviation between synthetic image and photograph is satisfactory. Once the reflectance of the real scenery is recovered, virtual objects can be integrated and the scene must be re-rendered. They report that the analysis and re-rendering of the sample images takes between 30 minutes to several hours – depending on the quality required and the scene's complexity.

Yu et al. [22] present a robust iterative approach that uses global illumination and inverse global illumination techniques. They estimate diffuse and specular reflectance, as well as radiance and irradiance from a sparse set of photographs and the given geometry model of the real scenery. Their method is applied to the insertion of virtual objects, the modification of illumination conditions and to the re-rendering of the scenery from novel viewpoints. As for Boivin's approach, BRDF recovery and re-rendering are not supported at interactive frame-rates.

Loscos et al. [13] estimate only the diffuse reflectance from a set of photographs with different but controlled real world illumination conditions. They are able to insert and remove real and virtual objects and shadows, and to modify the lighting conditions. To provide an interactive manipulation of the scenery, they separate the calculation of the direct and indirect illumination. While the direct illumination is computed on a per-pixel basis, indirect illumination is generated with a hierarchical radiosity system that is optimized for dynamic updates [8]. While the reflectance analysis is done during an offline preprocessing step, interactive frame-rates can be achieved during re-rendering. Depending on the performed task and the complexity of the scenery, they report re-rendering times for their examples between 1-3 seconds on a SGI R10000. Although these results are quite remarkable, the update rates are still too low to satisfy the high response requirements of stereoscopic displays that support head-tracking (and possibly multiple users).

Gibson and Murta [10] present another interactive image composition method to merge synthetic objects into a single background photograph of a real environment. A geometric model of the real scenery is also assumed to be known. In contrast to the techniques described above, their approach does not consider global illumination effects to benefit from hardware accelerated multi-pass rendering. Consequently, a reflectance analysis of the real surfaces is not required, indirect illumination effects are ignored, and a modification of the lighting conditions is not possible. The illumination of the real environment is first captured in form of an omnidirectional image. Then a series of high dynamic basis radiance maps are pre-computed. They are used during runtime to simulate a matching direct illumination of the synthetic objects using sphere mapping. Shadow casting between real and virtual objects is approximated with standard shadow mapping. With their method, convincing images can be rendered at frame rates up to 10fps on an SGI Onyx 2. However, it is restricted to a static viewpoint.

3. Diffuse Reflectance Analysis

We want to assume that the geometry of both object types –real and virtual– has been modeled or scanned in advance. While the material properties of virtual objects are also defined during their modeling process, the diffuse reflectance of physical objects is captured inside the Virtual Showcase with a set of video projectors and cameras. This sort of analysis is standard practice for many range scanner setups. But since we consider only diffuse real objects (a projector-based illumination will generally fail for any specular surface), our method can benefit from hardware accelerated rendering techniques. In contrast to conventional scanning approaches, this leads to dynamic update rates. Figure 2 illustrates an example.

[2] Not real-time.

3.1. Calibration

The intrinsic and extrinsic parameters of projectors and cameras within the world coordinate system have to be estimated first. We calibrate each device separately. As described in [3], we interactively mark the two-dimensional projections of known three-dimensional fiducials on a projector's/camera's image plane. Using these mappings, we apply Powell's direction set method [18] to solve a perspective n-point problem for each device. This results in the perspective projection matrices P, C of a projector and a camera. Both matrices incorporate the correct model-view transformations with respect to the origin of our world coordinate system.

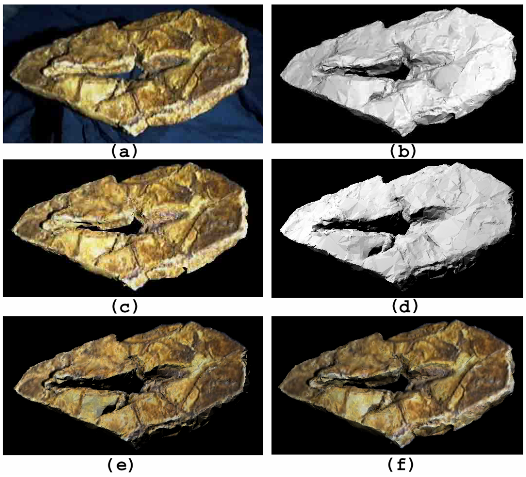

Figure 2: (a) Captured radiance map of a fossilized dinosaur footprint; (b) Intensity image rendered for calibrated projector from (a); (c) Computed reflectance map; (d) Novel illumination situation; (e) Reflectance map under novel illumination from (d); (f) Reflectance map under virtual illumination from (b).

3.2. Capturing Radiance Maps

A video projector is used to send structured light samples to the diffuse physical object and illuminate it with a predefined color C_p and an estimated intensity η. Synchronized to the illumination, a video camera captures an input image. Since this image contains the diffuse reflectance of the object's surface under known lighting conditions it represents a *radiance map*. White-balancing and other dynamic correction functions have been disabled in advance. The parameters of the camera's response function are adjusted manually in such a way that the recorded images approximate the real world situation as close as possible.

Some types of video projectors (such as digital light projectors, DLPs) display a single image within sequential, time-multiplexed light intervals to achieve different intensity levels per color. If such projectors are used, a single snapshot of the illuminated scene would capture only a slice of the entire display period. Consequently, this image would contain incomplete color fragments instead of a full-color image. The width of this slice depends on the exposure time of the camera. To overcome this problem, and to be independent of the camera's exposure capabilities, we capture a sequence of images over a predefined period of time. These images are then combined and averaged to create the final diffuse radiance map I_{rad} (cf. figure 2a).

3.3. Creating Intensity Images

To extract the diffuse material reflectance out of I_{rad} the lighting conditions that have been created by the projector have to be neutralized. OpenGL's diffuse lighting component is given by [17]:

$$I_i = \frac{1}{r_i^2}\cos(\theta_i)(D_l D_m)_i \qquad (3.1)$$

where I_i is the final intensity (color) of a vertex i, D_l is the diffuse color of the light, D_m is the diffuse material property, the angle θ_i is spanned by the vertex's normal and the direction vector to the light source, and the factor $1/r_j^2$ represents a square distance attenuation.

Similar as in [19], an intensity image I_{int} that contains only the diffuse illumination can be created by rendering the object's geometry (with $D_m = 1$) from the perspective of the video camera, illuminated by a point light source (with $D_l = C_p\eta$) that is virtually located at the position of the projector (cf. figure 2b).

In addition, hard shadows are added to the intensity image by applying standard shadow mapping techniques. Consequently, the background pixels of I_{int}, as well as pixels of regions that are occluded from the perspective of the light source are blanked out ($I_{\text{int}}(x, y) = 0$), while all other pixels are shaded under consideration of equation 3.1.

3.4. Extracting and Re-Rendering Diffuse Reflectance

Given the captured radiance map I_{rad} and the rendered intensity image I_{int}, the diffuse reflectance for each surface that is visible to the camera can be computed by:

$$I_{ref}(x,y) = \frac{I_{rad}(x,y)}{I_{\text{int}}(x,y)} \quad \text{for } I_{\text{int}}(x,y) > 0,$$
$$I_{ref}(x,y) = 0 \qquad \text{for } I_{\text{int}}(x,y) = 0 \qquad (3.2)$$

We store the reflectance image I_{ref}, together with the matrix C and the real object's world-transformation O_c that is active during the capturing process within the same data-structure. We call this data structure *reflectance map* (cf. figure 2c).

The captured reflectance map can be re-rendered together with the real object's geometric representation from any perspective with an arbitrary world-transformation O_a. Thereby, I_{ref} is applied as projective texture map with the texture matrix[3] set to $O_a^{-1}O_cC$. Enabling texture modulation, it can then be re-lit virtually under novel illumination conditions (cf. figures 2d, 2e and 2f).

3.5. Shortcomings and Solutions

The basic reflectance analysis method as described above faces the following problems:

(1) due to under-sampling, surfaces which span a large angle ϕ_i between their normal vectors and the direction vectors to the camera can cause texture artifacts if I_{ref} is re-mapped from a different perspective;

(2) a single reflectance map covers only the surface portion that is visible from the perspective of the camera;

(3) the radiance map can contain indirect illumination effects caused by light fractions that are diffused off other surfaces (so-called *secondary scattering*). The intensity image I_{int}, however, does not contain secondary scattering effects since a global illumination solution is not computed. Consequently, the extracted reflectance is incorrect at those areas that are indirectly illuminated by secondary scattering;

(4) the projector intensity η has to be estimated correctly;

To overcome the under-sampling problem, we define that only surfaces with $\phi_i \leq \phi_{\max}$ are analyzed. All other surfaces will be blanked-out in I_{ref} (i.e., $I_{ref}(x,y) = 0$). We found that $\phi_{\max} = 60°$ is appropriate. This corresponds to the findings in [19], describing a maximum angle between projector and projection surface.

Multiple reflectance maps that cover different surface portions can be captured under varying transformations O_c or C. They are merged and alpha blended during re-mapping them via multi-texturing onto the object's geometric representation. This ensures that regions which are blanked out in one reflectance map can be covered by other reflectance maps. To generate seamless transitions between the different texture maps, bi- or tri-linear texture filtering can be enabled.

Illuminating the entire scene can cause an extreme secondary scattering of the light. To minimize the appearance of secondary scattering in I_{rad}, we divide the scene into discrete pieces and capture their reflectance one after the other. For this, we can apply the same algorithm as described above. The difference, however, is that we illuminate and render only one piece at a time which then appears in I_{rad} and I_{int}. By evaluating the blanked out background information provided in I_{int}, we can effectively segment the selected piece in I_{int} and compute its reflectance. This is repeated for each front-facing piece, until I_{ref} is complete.

We estimate the projector's intensity η as follows: First, we generate a reflectance map with an initial guess of η. This reflectance map is then re-mapped onto the object's geometric representation, which is rendered from the perspective of the camera and illuminated by a virtual point light source with η located at the projector. The rendered radiance map I_{radv} is then compared to the captured radiance map I_{rad} by determining the average square distance error Δ among all corresponding pixels. Finally, we find an approximation for η by minimizing the error function f_Δ. For this we apply Brent's inverse parabolic minimization method with bracketing [7]. By estimating η, we can also incorporate the constant black-level of the projector.

4. Augmented Radiosity

In computer graphics, the radiosity method [11] is used to approximate a global illumination solution by solving an energy-flow equation. Indirect illumination effects, such as secondary scattering can be simulated with radiosity. The general radiosity equation for n surface patches is given by:

$$B_i = E_i + \rho_i \sum_{j=1}^{n} B_j F_{ij} \qquad (4.1)$$

where B_i is the radiance of surface i, E_i is the emitted energy per unit area of surface i, ρ_i is the reflectance of

[3] Including the corresponding mapping transformation from normalized device coordinates to normalized texture coordinates.

surface i, and F_{ij} represents the fraction of energy that is exchanged between surface i and surface j (the form-factor).

The simulation of radiosity effects within an optical see-through environment that consists of diffuse physical and virtual objects, is facing the following challenges and problems:

(1) light energy has to flow between all surfaces – real ones and virtual ones;

(2) physical objects are illuminated with physical light sources (i.e., video projectors in our case) which do not share the geometric and radiometric properties of the virtual light sources;

(3) no physical light energy flows from virtual objects to real ones (and vice versa). Consequently, the illuminated physical environment causes (due to the absence of the virtual objects) a different radiometric behavior than the entire environment (i.e., real and virtual objects together).

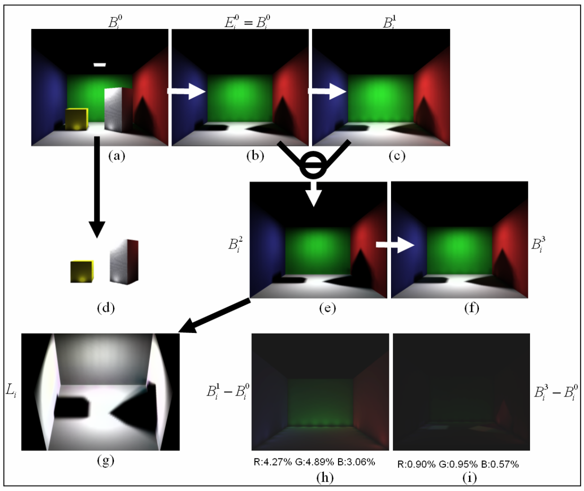

Figure 3: Multi-Pass Radiosity.

An example is illustrated in figure 3a[4]. The entire environment consists of three walls, a floor, two boxes and a surface light source on the ceiling. We want to assume that the walls and the floor are the geometric representations of the physical environment, and the boxes as well as the light source belong to the virtual environment. While the diffuse reflectance ρ_i of the physical environment can be automatically captured (as described in section 3), it has to be defined for the virtual environment. After a radiosity simulation[5] of the entire environment the radiance values B_i^0 for all surfaces have been computed[6]. Color-bleeding and shadow-casting are clearly visible.

4.1 Virtual Objects

For virtual objects, the computed radiance values are already correct (cf. figure 3d). The rendered image represents a radiance map that is generated from one specific perspective. Rendering the virtual objects from multiple perspectives results in multiple radiance maps that can be merged and alpha blended during re-mapping them via multi-texturing onto the virtual geometry (as described for reflectance maps in section 3.5). In this case, our radiance maps are equivalent to light maps that are often being applied during pre-lighting steps to speed up the online rendering process.

The pre-lit virtual objects can simply be rendered together with their light maps and can be optically overlaid over the physical environment.

4.2 Physical Objects

The physical surfaces, however, have to emit the energy that was computed in B_i^0 (cf. figure 3b). To approximate this, we first assume that every physical surface patch directly emits an energy E_i^0 that is equivalent to B_i^0. If this is the case, fractions of this energy will radiate to other surfaces and illuminate them in addition. This can be simulated by a second radiosity-pass (cf. figure 3c), which computes new reflectance values B_i^1 for all the physical surfaces, by assuming that $E_i^0 = B_i^0$, and not considering the direct influence of the virtual light source.

If we subtract the radiance values that have been computed in both passes we receive the scattered light only. That is, the light energy radiated between the physical surfaces $B_i^1 - B_i^0$ (cf. figure 3h).

Consequently,

$$B_i^2 = B_i^0 - \left(B_i^1 - B_i^0\right) \qquad (4.2)$$

approximates the energy that has to be created physically on every real surface patch. To prove this we can apply a third radiosity pass to simulate the energy flow between the patches (cf. figure 3f). We can see that the remaining energy $B_i^1 - B_i^0$ will be nearly added, and we receive:

$$B_i^3 = B_i^2 + \left(B_i^1 - B_i^0\right) \approx B_i^0 \qquad (4.3)$$

By removing the virtual objects from the environment and simulating the second radiosity pass, light energy will also be radiated onto surfaces which were originally blocked or covered by the virtual objects (either

[4] We have chosen a physical mock-up of the Cornell room since it is used in many other publications as a reference to evaluate radiosity techniques.

[5] We applied a hemi-cube-based radiosity implementation with progressive refinement, adaptive subdivision and interpolated rendering for our simulations.

[6] Note, that the upper index represents the radiosity pass.

completely or partially). Examples are the shadow areas that have been cast by the virtual objects. This can be observed in figure 3h and figure 3i. Consequently, negative radiance values are possible for such areas after applying equation 4.2. To avoid this, the resulting values have to be clipped to a valid range.

The average deviations between B_i^0 and B_i^1, as well as between B_i^0 and B_i^3, within the three spectral samples red (R), green (G), and blue (B) are presented below figures 3h and 3i, respectively. Treating a video projector as a point light source B_i^2 can be expressed as a simplified version of equation 4.1:

$$B_i^2 = \rho_i L_i F_i \qquad (4.4)$$

where L_i is the irradiance that has to be projected onto surface i by the projector, and F_i is the form-factor for surface i, which is given by:

$$F_i = \frac{\cos(\theta_i)}{r_i^2} h_i \qquad (4.5)$$

where θ_i is the angle between a surface patch's normal and the direction vector to the projector, r_i is the distance between a surface patch and the projector, and h_i is the visibility term of the surface patch, seen from the projector's perspective.

Extending and solving equation 4.4 for L_i, we receive (cf. figure 3g):

$$L_i = \frac{B_i^2}{\rho_i F_i} \eta \qquad (4.6)$$

To cope with the individual brightness of a video projector, we add the intensity factor η. How to estimate η for a specific projector was described in section 3.5. To be consistent with our previously used terminology, we call L_i the *irradiance map.*

4.3 Limitations

The computed radiance and irradiance values are view-independent. Consequently, irradiance maps for the real objects and radiance maps for the virtual objects can be pre-computed offline.

The real objects are illuminated with projected light during runtime by rendering the generated irradiance map from the viewpoint of the projector (e.g., as illustrated in figure 3g). Virtual objects are rendered with the computed light maps (e.g., as illustrated in figure 3d) and are then optically overlaid over the real environment. Due to the view-independence of the method, the augmented scene can be observed from any perspective (i.e., head-tracking and stereoscopic rendering are possible). However, the scene has to remain static, since any modification would require to re-compute new radiance and irradiance maps throughout multiple radiosity passes. This is not yet possible at interactive rates.

Figure 4 shows a photograph of (a) the physical object under room illumination, (b) a screen-shot of captured reflectance maps that have been re-rendered under novel lighting conditions, (c) a screen-shot of the simulated radiance situation B_i^0, and (d) a photograph of a physical object that has been illuminated with L_i. Note, that small deviations between the images can be contributed to the responds of the digital camera that was used to take the photograph, as well as to the high black-level of the projector that, for instance, makes it impossible to create completely black shadow areas.

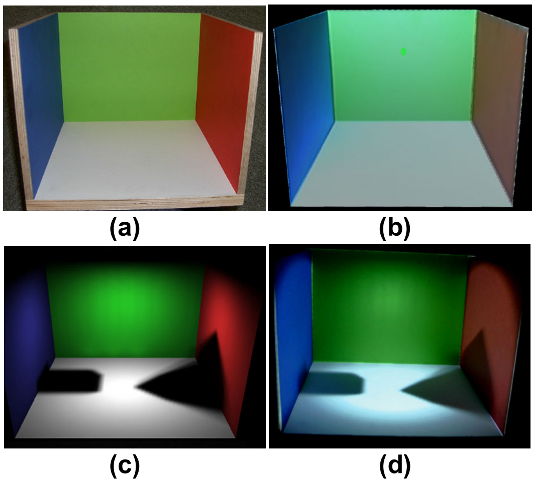

Figure 4: (a) Photograph of original object under room illumination; (b) Screen-shot of captured reflectance re-lit with virtual point light source and Phong shading; (c) Screen-shot of simulated radiosity solution with captured reflectance, virtual surface light source (shown in figure 3), and two virtual objects (show in figure 3); (d) Photograph of original object illuminated with the computed irradiance.

5. Interactive Approximations

In the following section we describe several interactive rendering methods that make use of hardware acceleration. In particular we discuss how to create matching shading, shadow and reflection effects on real and virtual objects. Indirect lighting effects such as color-bleeding, however, cannot be created with these techniques. Yet, they create acceptable results at interactive frame rates for multiple head-tracked users and stereoscopic viewing on conventional PCs.

5.1 Shading

The generation of direct illumination effects on virtual surfaces caused by virtual light sources is a standard task of today's hardware accelerated computer graphics technology. Real-time algorithms, such as Gouraud shading or Phong shading are often implemented on graphics boards.

Consistent and matching shading effects on real surfaces from virtual light sources can be achieved by using video projectors that project appropriate irradiance maps onto the real objects. Raskar et al. [19] show how to compute an irradiance map to lift the radiance properties of neutral diffuse objects with uniform white surfaces into a pre-computed radiance map of a virtual scene illuminated by virtual light sources. An irradiance map that creates virtual illumination effects on diffuse real objects with arbitrary reflectance properties (color and texture) can be computed as follows:

First, the real objects' captured reflectance map (I_{ref}) is rendered from the viewpoint of the projector and is shaded with all virtual light sources in the scene. This results in the radiance map I_{rad_1}. Then I_{ref} is rendered again from the viewpoint of the projector. This time, however, it is illuminated by a single point light source (with $D_l = 1 \cdot \eta$) which is located at the position of the projector. This results in the radiance map I_{rad_2}. Finally, the correct irradiance map is computed by:

$$L = \frac{I_{rad\_1}}{I_{rad\_2}} \tag{5.1}$$

Note that equation 5.1 correlates to equation 4.6. The difference is the applied illumination model. While equation 4.6 is discussed with respect to an indirect global illumination model (radiosity), equation 5.1 applies hardware accelerated direct models (such as Phong or Gouraud shading). It is easy to see that I_{rad_1} is the opponent to B_i^2 and that I_{rad_2} corresponds to $\rho_i F \eta$.

Note also that this method is actually completely independent of the real objects' reflectance. This can be shown by equalizing I_{rad_1} with I_{rad_2} through equation 3.1. In this case the diffuse material property D_m (i.e., the reflectance) is canceled out. Consequently, I_{rad_1} and I_{rad_2} can be rendered with a constant (but equal) reflectance (D_m). If we choose $D_m = 1$ then the irradiance map is simply the quotient between the two intensity images $I_{\text{int\_1}}$ and $I_{\text{int\_2}}$ that result from the two different lighting conditions – the virtual one and the real one.

The irradiance map L should also contain consistent shadow information. How to achieve this is outlined below. Figure 5 illustrates examples with matching shading effects[7].

5.2 Shadows

We can identify six types of shadows within an optical see-through environment:

(1) shadows on real objects created by real objects and real light sources;
(2) shadows on virtual objects created by virtual objects and virtual light sources;
(3) shadows on virtual objects created by real objects and virtual light sources;
(4) shadows on real objects created by real objects and virtual light sources;
(5) shadows on real objects created by virtual objects and virtual light sources;
(6) occlusion shadows;

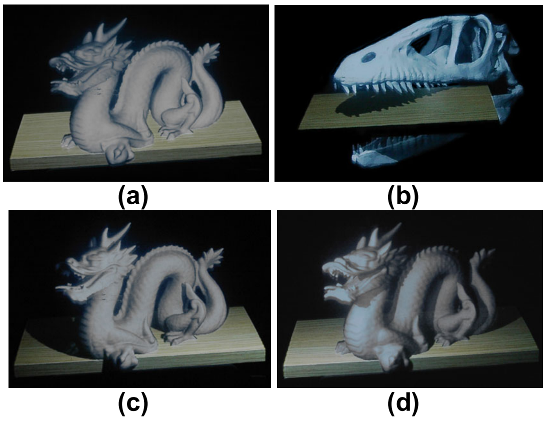
(a) (b) (c) (d)

Figure 5: (a) Unrealistic uniform illumination with shadow type 6 (the wooden plate is real, the dragon and the dinosaur skull are virtual); (b)-(d) Realistic illumination under varying virtual lighting conditions with matching shading and shadows (types 2,3,5, and 6).

The first type of shadow is the result of occlusions and self-occlusions of the physical environment that is illuminated by a physical light source (e.g., a video projector). Since we focus on controlling the illumination conditions within the entire environment via virtual light sources we have to remove these shadows. This can be achieved by using multiple synchronized projectors that are able to illuminate all visible real surfaces. Several techniques have been described which compute a correct color and intensity blending for multi-projector displays [14, 19, 21].

[7] Note that we have chosen a simple wooden plate to demonstrate and to compare the different effects. However, all techniques that are explained in this paper can be applied to arbitrary object shapes.

The second and third shadow types can be created with standard shadow mapping or shadow buffering techniques. To cast shadows from real objects onto virtual ones, the registered geometric representations of the real objects have to be rendered together with the virtual objects when the shadow map is created (i.e., during the first shadow pass). Such geometric real world representations (sometimes called *phantoms* [6]) are often rendered continuously to generate a realistic occlusion of virtual objects by real ones. Note that these hardware accelerated techniques create hard shadows while global illumination methods (such as radiosity) can create soft shadows. Texture blending, however, allows ambient light to be added to the shadow regions. This results in dark shadow regions that are blended with the underlying surface texture, instead of creating unrealistic black shadows.

Shadow types number 4 and 5 can also be created via shadow mapping. However, they are projected on the surface of the real object together with the irradiance map L, as discussed in section 5.1. Therefore, I_{rad_1} has to contain the black (non-blended) shadows of the virtual and the real objects. This is achieved by rendering all virtual objects and all phantoms during the first shadow pass to create a shadow map. During the second pass the shaded reflectance texture and the generated shadow texture are blended and mapped onto the objects' phantoms. A division of the black shadow regions by I_{rad_2} preserves these regions. Note that a blending of the projected shadows with the texture of the real objects occurs physically if the corresponding surface portions are illuminated (e.g., by a relatively small amount of projected ambient light).

Occlusion shadows [3] are special view-dependent shadows created by the projectors on the real objects' surfaces. We have introduced them to achieve a realistic occlusion of real objects by virtual ones. They are normally not visible from the perspective of the observer, since they are displayed exactly underneath the graphical overlays. Occlusion shadow-maps, however, have also to be blended to the irradiance map L before it is projected.

5.3 Reflections

Using hardware accelerated cube mapping techniques, the virtual representation of the real environment (i.e., the objects' geometry together with the correctly illuminated reflectance map) can be reflected by virtual objects (cf. figure 6). Therefore, only the registered virtual representation of the real environment has to be rendered during the generation step of the cube map. Virtual objects are then simply rendered with cube mapping enabled. Note, that for conventional cube mapping, reflection effects on a virtual object are physically correct for only a single point – the center of the cube map frusta. To create convincing approximations this center has to be matched with the virtual object's center of gravity, and the cube map has to be updated every time the scene changes.

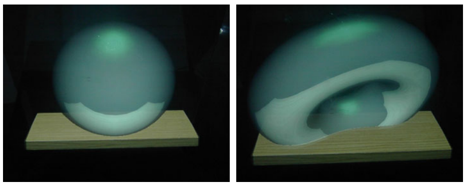
(a) (b)

Figure 6: (a) A virtual sphere and (b) a virtual torus reflecting and occluding the real object (wooden plate).

5.4 Occluding Occlusion Shadows

The occlusion shadow method [3] is currently one of two functioning solutions that can create consistent occlusions effects for optical see-through displays. A main drawback of this approach is its limited support for multiple users: If the same real surfaces are simultaneously visible from multiple points of view (as it is the case for different observers), individual occlusion shadows that project onto these surfaces are also visible from different viewpoints at the same time (cf. figure 7a).

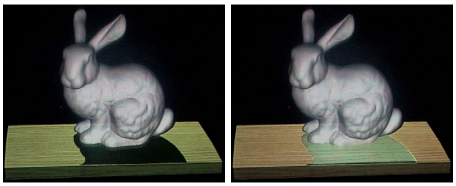
(a) (b)

Figure 7: (a) Occlusion shadow of second observer is clearly visible; (b) Wrongly visible occlusion shadow is covered by optically overlaying the corresponding part of the reflectance map.

Although two different approaches have been presented in [3] that reduce the effects of this problem, it is not completely solved for any type of surface. Knowing the reflectance information of the real surfaces, however, leads to an effective and general solution:

As described in sections 5.1 and 5.2, the real objects are illuminated by projected light (also containing occlusion shadows for all observers) and the virtual objects are shaded, rendered and optically overlaid over the real scene (on top of each observer's occlusion shadows). In addition to the virtual scene, we render the portions of the real scene (i.e., its registered reflectance map) that are covered by the occlusion shadows of all other observers.

Remember that these reflectance-map portions are illuminated and shaded under the same lighting conditions as their real counterparts (outlined in sections 5.1 and 5.2). This creates seamless transitions between the real and the virtual parts.

For each observer the occlusion shadows of all other observers are rendered into the stencil buffer first. This is done by rendering the real scene's geometry from each observer's perspective and adding the corresponding occlusion shadows via projective texture mapping, as described in [3]. The stencil buffer has to be filled in such a way that the area surrounding the occlusion shadows will be blanked out in the final image. Then the real scene's reflectance map is rendered into the frame buffer (also from the perspective of the observer) and is shaded under the virtual lighting situation. After stenciling has been disabled, the virtual objects can be added to the observer's view (cf. figure 7b).

6. Summary and Future Work

The overall goal of this work is to enhance realism for optical see-through AR environments. To achieve this we have develop techniques which allow creating consistent illumination effects between real and virtual objects. We have implemented and demonstrated these techniques based on the Virtual Showcase, since this display provides a well controllable environment.

We use video projectors and cameras as essential components of the Virtual Showcase. They allow to retrieve information out of the Virtual Showcase's inside at dynamic update rates, and to illuminate real objects on a per-pixel basis in real time. Currently only reflectance data is scanned from real objects. In the future, other surface information, such as geometry and external emission can be measured as well. The scanned geometry information will lead to the development of an automatic registration procedure for real objects.

Using the reflectance information, Augmented Radiosity has been described as a global illumination technique for optical see-through devices that is able to create a high level of realism. Only static augmented environments can be created with this method. However, due to its view-independency, head-tracking and stereoscopic rendering is possible. To evaluate interactive global illumination methods in combination with this technique belongs to our list of future tasks.

To reach interactive rendering frame rates, hardware accelerated methods have been used to generate convincing approximations of a consistently and realistically illuminated augmented environment. Throughout several rendering passes shading, shadow mapping and cube mapping techniques have been applied in combination with the captured reflectance information to achieve this. Instead of capturing the reflectance map of a real object and computing a radiance map, its physical radiance can be captured directly after the synthetic illumination information (i.e., shading and shadows, as described in section 5.1) have been created on its surface. This radiance map can then be applied in combination with the reflection and occlusion shadow techniques described in sections 5.3 and 5.4. The advantage of this approach is that the illumination information on the real object has to be computed only once – before it is projected onto its surface. The resulting radiance map can simply be captured with the video camera and contains all important information of the real environment, such as reflectance, shading and shadows.

The occluding occlusion shadow technique allows to solve the multi-user limitation that is linked to the original occlusion shadow idea. In addition, it allows to make seamless transitions on the mixed reality continuum [15], that are important for applications of the Virtual Showcase, such as digital storytelling [4]. The biggest problem of this extension, however, is a slight color inconsistency between the real object and the virtual overlay. This results from photometric deviations between the CRT screens and the video cameras with the observers' visual perception of the real object. Currently we adjust this manually by modifying the physical and the synthetic illumination until the real object and the virtual overlay appear to coincide visually. Automatic calibration techniques need to be developed in the future. To reduce geometric misalignments caused by small registration errors, the edges of the virtual overlay can be blurred in addition.

A next important step towards realism will be to visually enhance the virtual components while retaining interactive frame rates. Advanced rendering techniques, such as light fields [12, 20] and hardware-accelerated procedural shading technology might allow blurring the boundaries between real and virtual even further.

Acknowledgements

We thank Werner Purgathofer for fruitful discussions on the occluding occlusion shadows topic. The Virtual Showcase project is supported by the European Union, IST-2001-28610.

References

[1] Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B. Recent Advances in Augmented Reality. IEEE Computer Graphics & Applications, vol. 21, no.6, pp. 34-47, 2001.

[2] Bimber, O., Fröhlich, B., Schmalstieg, D., and Encarnação, L.M. The Virtual Showcase. IEEE Computer Graphics & Applications, vol. 21, no.6, pp. 48-55, 2001.

[3] Bimber, O, and Fröhlich, B. Occlusion Shadows: Using Projected Light to Generate Realistic Occlusion Effects for View-Dependent Optical See-Through Displays. In proceedings of ACM/IEEE International Symposium on Mixed and Augmented Reality (ISMAR'02), pp. 186-195, 2002.

[4] Bimber, O., Encarnação, L.M, and Schmalstieg, D. The Virtual Showcase as a new Platform for Augmented Reality Digital Storytelling. In proceedings of IPT/EGVE 2003 workshop, pp. 87-95, 2003.

[5] Boivin, S. and Gagalowicz, A. Image-based rendering of diffuse, specular and glossy surfaces from a single image. In proceedings of ACM Siggraph, pp. 107-116, 2001.

[6] Breen, D.E., Whitaker, R. T., Rose, E. and Tuceryan, M. Interactive Occlusion and Automatic Object Placement for Augmented Reality. Computer and Graphics Forum (proceedings of EUROGRAPHICS'96), vol. 15, no. 3, pp. C11-C22, 1996.

[7] Brent, R.P. Algorithms for minimization without derivatives. Prentice-Hall, Engelwood Cliffs, NJ, 1973.

[8] Drettakis, G. and Sillion, F. Interactive update of global illumination using a line-space hierarchy. In proceedings of ACM Siggraph'97, pp- 57-64, 1997.

[9] Fournier, A. Gunawan, A.S., and Romanzin, C. Common illumination between real and computer generated scenes. In proceedings of Graphics Interface'93, pp. 254-262, 1993.

[10] Gibson, S. and Murta, A. Interactive rendering with real-world illumination. In proceedings of 11th Eurographics Workshop on Rendering, pp. 365-376, 2000.

[11] Goral, C.M., Torrance, K.E., Greenberg, D.P., and Battaile, B. Modeling the interaction of light between diffuse surfaces. Computer In proceedings of ACM Siggraph'84, vol. 18, no. 3, pp. 212-222, 1984.

[12] Levoy, M. and Hanraham, P. Light field rendering. In proceedings of ACM Siggraph'96, pp. 31-42, 1996.

[13] Loscos, C., Drettakis, G., and Robert, L. Interactive virtual relighting of real scenes. IEEE Transactions on Visualization and Computer Graphics, vol, 6, no. 3, pp. 289-305, 2000.

[14] Majumder, A. He, Z., Towles, H., and Welch, G. Achieving color uniformity across multi-projector displays. In proceedings of IEEE Visualization, pp. 117-124, 2000.

[15] Milgram, P. and Kishino, F. A Taxonomy of Mixed Reality Visual Displays. IEICE Transactions on Information Systems E77-D, vol.12, pp. 1321-1329, 1994.

[16] Nakamae, E., Harada, K., Ishizaki, T., and Nishita, T.A montage method: The overlaying of computer generated images onto background photographs. In proceedings of ACM Siggraph'86, pp. 207-214, 1986.

[17] Neider, J., Davis, T., and Woo, M. OpenGL Programming Guide. Release 1, Addison Wesley, ISBN 0-201-63274-8, pp.157-194, 1996.

[18] Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. Numerical Recipes in C - The Art of Scientific Computing (2nd edition), Cambridge University Press, ISBN 0-521-43108-5, pp. 412-420,1992.

[19] Raskar, R. Welch, G., Low, K.L., and Bandyopadhyay, D. Shader Lamps: Animating real objects with image-based illumination. In Proc. of Eurographics Rendering Workshop (EGRW'01), 2001.

[20] Wood, D.N., Azuma, D.I., Aldinger, K., Curless, B., Duchamp, T., Salesin, D.H., and Stuetzle, W. Surface Light Fields for 3D Photography, In proceedings of ACM Siggraph'00, pp 287-296, 2000.

[21] Yang, R., Gotz, D., Hensley, J., Towles, H., and Brown, M.S. PixelFlex: A reconfigurable multi-projector display system. In proceedings of IEEE Visualization, pp. 167-174, 2001.

[22] Yu, Y., Debevec, P., Malik, J., and Hawkins, T. Inverse global illumination: Recovering reflectance models of real scenes from photographs. In proceedings of ACM Siggraph' 99, pp. 215-224, 1999.