Shader Lamps: Animating Real Objects With Image-Based Illumination

Ramesh Raskar, Greg Welch, Kok-lim Low, Deepak Bandyopadhyay

Abstract

We describe a new paradigm for three-dimensional computer graphics, using projectors to graphically animate physical objects in the real world. The idea is to replace a physical object– with its inherent color, texture, and material properties–with a neutral object and projected imagery, reproducing the original (or alternative) appearance directly on the object. Because the approach is to effectively "lift" the visual properties of the object into the projector, we call the projectors shader lamps. We address the central issue of complete and continuous illumination of non-trivial physical objects using multiple projectors and present a set of new techniques that makes the process of illumination practical. We demonstrate the viability of these tech-niques through a variety of table-top applications, and describe preliminary results to reproduce life-sized virtual spaces.

Shader Lamps:
Animating Real Objects With Image-Based Illumination

Ramesh Raskar[+]   Greg Welch[*]   Kok-Lim Low[*]   Deepak Bandyopadhyay[*]

[+]MERL, Mitsubishi Electric Research Laboratories
[*]University of North Carolina at Chapel Hill

Abstract

We describe a new paradigm for three-dimensional computer graphics, using projectors to graphically animate physical objects in the real world. The idea is to replace a physical object—with its inherent color, texture, and material properties—with a neutral object and projected imagery, reproducing the original (or alternative) appearance directly on the object. Because the approach is to effectively "lift" the visual properties of the object into the projector, we call the projectors *shader lamps*. We address the central issue of complete and continuous illumination of non-trivial physical objects using multiple projectors and present a set of new techniques that makes the process of illumination practical. We demonstrate the viability of these techniques through a variety of table-top applications, and describe preliminary results to reproduce life-sized virtual spaces.

Keywords: Engineering Visualization, Illumination Effects, User Interfaces, Virtual Reality.

1. Introduction

Graphics in the World. Traditionally, computer graphics techniques attempt to "capture" the real world in the computer, and then to reproduce it visually. In later years, work has been done to explore what is in effect the reversal of this relationship—to "insert" computer graphics into the real world. Primarily, this has been done for special effects in movies, and for real-time *augmented reality.* Most recently, there is a new trend to use light projectors to render imagery directly in our real physical surroundings. Examples include the *Luminous Room* [Underkoffler99] and the *Office of the Future* [Raskar98]. What we are pursuing here is a more complete extension of these ideas—the incorporation of three-dimensional computer graphics and animation directly into the real world all around us.

Stimulation and Communication of Ideas. Despite the many advances in computer graphics, architects and city planners (for example) still resort to building physical models when the time comes to seek client or constituent approval [Howard00]. The architects that we have spoken with, and many books on the subject, have noted that while it is true that designers cannot do without CAD tools anymore, "it [the computer] cannot replace the actual material experience, the physical shape and the build-up of spatial relationships." [Knoll92]. Even in this day of computer animation, animators often sculpt a physical model of a character before making computer models. This was the case with Geri in "Geri's Game" (Pixar Animation Studios). One reason for these sentiments and practices is that the human interface to a physical model is the essence of "intuitive". There are no widgets to manipulate, no sliders to move, and no displays to look through (or wear). Instead, we walk around objects, moving in and out to zoom, gazing and focusing on interesting components, all at very high visual, spatial, and temporal fidelity. We all have a lifetime of experience with this paradigm. The ambi-
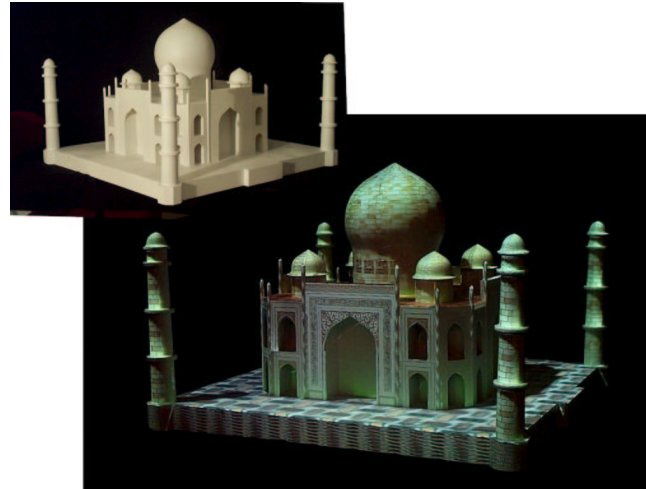

Figure 1: The underlying physical model of the Taj Mahal and the same model enhanced with *shader lamps*.

tious goal of shader lamps is to enjoy some of the advantages of this natural physical interface, in particular, the auto-stereoscopic nature of viewing physical objects, combined with the richness of computer graphics.

Image-Based Illumination. When we illuminate a real object with a white light, its surface reflects particular wavelengths of light. Because our perception of the surface attributes is dependent only on the spectrum of light that eventually reaches our eyes, we can shift or re-arrange items in the optical path, as long as the spectrum of light that eventually reaches our eyes is sufficiently similar. Many physical attributes can be effectively incorporated into the light source to achieve a perceptually equivalent effect on a neutral object. Even non-realistic appearances can be realized. This concept is illustrated in Figure 2. We can use digital light projectors and computer graphics to form *shader lamps* that effectively reproduce or synthesize various surface attributes, either statically, dynamically, or interactively. While the results are theoretically equivalent for only a limited class of surfaces and attributes, our experience is that they are quite realistic and compelling for a broad range of applications.

The existence of an underlying physical model is arguably unusual for computer graphics, however, it is not for architects [Howard00], artists, and computer animators. In addition, various approaches to automatic three-dimensional fabrication are steadily becoming available, e.g. laminate object manufacturing, stereolithography, and fused deposition. It is not unreasonable to argue that three-dimensional printing and faxing are coming.

We previously presented preliminary thoughts and results in workshop settings [Raskar99b]. After further development of our ideas and methods, we are now ready to articulate the idea more completely, and to demonstrate practical methods. We present results using multiple shader

1

lamps to animate physical objects of varying complexity—from a smooth flower vase to a relatively complex model of the Taj Mahal. We also demonstrate some applications such as small "living" dioramas, human-scale indoor models, and hand-held physical user-interface objects.

Contributions

- We introduce shader lamps as a new mode of visualizing 3D computer graphics. Our idea treats illumination basically as a 3D perspective projection from a lamp, and thus, it can be created using traditional 3D computer graphics. We present techniques that can replace not just textures, i.e. diffuse component, but can reproduce virtually any BRDF appearance.

- We present new algorithms to make the process of illumination practical. We first identify a simple radiance adjustment equation for guiding the rendering process and then present methods for the corresponding intensity correction.

- We introduce a new algorithm for determining pixel weights and computing feathering intensities across transitions in projectors' regions of influence in the presence of depth discontinuities.

2. Previous Work

Theater and entertainment. Naimark [Naimark84] used a rotating movie camera to film a living room, replete with furniture and people. The room and furniture were then painted white (neutral), and the captured imagery was projected back onto the walls using a rotating projector that was precisely registered with the original camera. This crucial co-location of the capturing and displaying devices is common to most of the current demonstrations that use pre-recorded images or image-sequences. A limited but compelling example of this idea is the projection of pre-recorded video to animate four neutral busts of singing men in the Walt Disney World "Haunted Mansion". In addition, a patented projector and fiber-optic setup animates the head of the fictional fortune teller "Madame Leota" inside a real crystal ball [Liljegren90].

Slides of modified photographs augmented with fine details are also used with very bright projectors to render imagery on a very large architectural scale. A well-known modern realization of this idea is the Son et Lumiere (light show) on the Blois castle in the Loire Valley (France). In addition, the medium is now being used elsewhere around the world. Influenced by Son et Lumiere, Marc Levoy [Levoy00] has recently experimented with projection of imagery onto small-scale fabricated statues. Instead of photographs, he first renders an image of a stored 3D model

similar to our techniques and then manually positions the projector to geometrically register the projected image. The [Hypermask99], an exception in terms of automatic registration, involves projecting an animated face onto a moving mask for storytelling.

All these systems create compelling visualizations. However, the cumbersome alignment process can take several hours even for a single projector. Our technique avoids this problem by forming a 3D geometric understanding using well-known computer vision techniques described in Section 4 and then moves beyond simple image projection to reproduce reflectance properties.

Tangible luminous interfaces. The Luminous Room project treats a co-located camera-projector pair as an I/O bulb to sense and inject imagery onto flat surfaces in the real physical surroundings of a room or a designated workspace [Underkoffler99]. The work we present here is distinct from, but complementary to, this work. A primary distinction is that their main focus is interaction with the information via luminous (lit) and tangible interfaces. This focus is exemplified in such applications as "Illuminating Light" and "URP" (urban planning). The latter arguably bears closest resemblance to our work, in particular the interactive simulation of building shadows from sunlight. The approach is to recognize the 2D physical objects (building "phicons") lying in a plane, track their 2D positions and orientations in the plane, and project light from overhead to reproduce the appropriate sunlight shadows. However, we are primarily interested in the use of physical objects as *truly* three-dimensional display devices for more general computer graphics, visualization and aesthetic (artistic) applications.

Modeling and rendering architecture from photographs. In the "Facade" project, a sparse set of photographs are used to model and render architectural monuments [Debevec96]. This is a good example of a hybrid approach of using geometry and images to reproduce physical human-made structures. The main challenges are related to the occlusion, sampling, and blending issues that arise when re-projecting images onto geometric models. They face these challenges with computer imagery and analytic models, while in shader lamps, we have to face them with real (light projected) imagery and physical models. It would be useful to use Facade tools to build a hybrid geometry and image model of a university campus, and then use the shader-lamp techniques to animate a scaled physical model, effectively creating a "living diorama" of the campus.

To realize the general application of this technique, one must, among other things, have a method for pre-warping the imagery to "fit" the physical object so that it appears correct to local viewers. Some limited 2D warping effects have been achieved by [Dorsey91] to model the appearance of theatrical backdrops so that they appear correct from the audience's perspective. The Office of the Future project [Raskar98] presents rendering techniques to project onto non-planar surfaces. We use techniques that build on this to illuminate potentially non-convex or a disjoint set of objects, and present new techniques to address the alignment, occlusion, sampling and blending issues.

3. The Illumination Process

We introduce the idea of rearranging the terms in the relationship between illumination and reflectance to reproduce equivalent radiance at a surface. As shown in flatland in Figure 3, the radiance in a certain direction at point (x), which has a given BRDF in the physical world (left), can be mimicked by changing the BRDF and illuminating the point with a appropriately chosen light source, e.g. a projector pixel (right). Below we identify a radiance adjustment equation for determining the necessary intensity of a projector pixel, given the position and orientation of the viewer and the virtual scene. For a more sys-
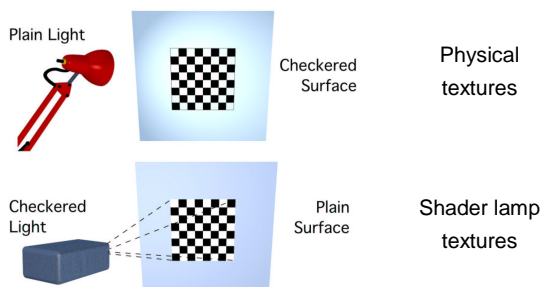

Figure 2: Concept of *shader lamps*.

tematic rendering scheme, we describe the notion of separating the *rendering view*—the traditional virtual camera view, from the *shading view*—the position of the viewer for lighting calculations.

First, let us consider the rendering equation, which is essentially a geometrical optics approximation as explained in [Kajiya86]. The radiance at a visible surface point (x) in the direction (θ, ϕ) that would reach the observer of a physical realization of the scene is

$$L(x,\theta,\phi) = g(x,\theta,\phi)(L_e(x,\theta,\phi) + h(x,\theta,\phi)) \qquad (1)$$

where

$$h(x,\theta,\phi) = \int_i F_r(x,\theta,\phi,\theta_i,\phi_i) L_i(x,\theta_i,\phi_i) \cos(\theta_i) d\omega_i \qquad (2)$$

and $g(x,\theta,\phi)$ is the geometry term (visibility and distance), $L_e(x,\theta,\phi)$ is the emitted radiance at the point (non-zero only for light sources), and $F_r(x,\theta,\phi,\theta_i,\phi_i)$ is the BRDF of the point. The integral in $h(x,\theta,\phi)$ accounts for all reflection of incident radiance $L_i(x,\theta_i,\phi_i)$ from solid angles $d\omega_i$. Radiance has dimensions of energy per unit time, area and solid angle.

Treating the projector lamp as a point emitter, the radiance due to direct projector illumination at the same surface point at distance $d(x)$ but with diffuse reflectance $k_u(x)$ is given by

$$L'(x,\theta,\phi) = g(x,\theta,\phi) k_u(x) I_p(x,\theta_p,\phi_p) \cos(\theta_p) / d(x)^2 \qquad (3)$$

where $I_p(x,\theta_p,\phi_p)$ = radiant intensity of projector in the direction (θ_p,ϕ_p) and is related to a discretized pixel value via filtering and tone representation.

We can reproduce radiance $L'(x,\theta,\phi)$ equivalent to $L(x,\theta,\phi)$ for a given viewer location, by solving Equation 3 for I_p:

$$I_p(x,\theta_p,\phi_p) = \frac{L(x,\theta,\phi) d(x)^2}{k_u(x) \cos(\theta_p)} \quad \text{for } k_u(x) > 0. \qquad (4)$$

Thus, as long as the diffuse reflectance $k_u(x)$ is nonzero for all the wavelengths represented in $L(x,\theta,\phi)$, we can effectively represent the surface attribute with appropriate pixel intensities. In practice, however, the range of values we can display are limited by the brightness, dynamic range and pixel resolution of the projector.

The rendering process here involves two viewpoints: the user's and the projector's. A simple approach would be to first render the image as seen by the user, which is represented by $L(x,\theta,\phi)$, and then use traditional image-based rendering techniques to warp this image to generate the intensity-corrected

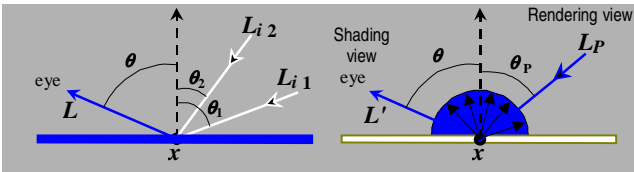projected image, represented by $I_p(x,\theta_p,\phi_p)$ [Chen93, McMillan95]. For a changing viewer location, view-dependent shading under static lighting conditions can also be implemented [Debevec98, Levoy96, Gortler96]. However, the warping can be avoided in the case where the display medium is the same as the virtual object. For a single-pass rendering, we treat the moving user's viewpoint as the shading view. Then, the image synthesis process involves rendering the scene from the projector's view, by using a perspective projection matrix that matches the projector's intrinsic and extrinsic parameters, followed by radiance adjustment. The separation of the two views offers an interesting topic of study. For example, for a static projector, the visibility and view-independent shading calculations can be performed just once even when the user's viewpoint is changing.

To realize a real-time interactive implementation we use conventional 3D rendering APIs, which only approximate the general rendering equation. The BRDF computation is divided into view-dependent specular, and view-independent diffuse and ambient components. View-independent shading calculations can be performed by assuming the rendering and shading view are the same. (The virtual shadows, also view-independent, are computed using the traditional two-pass shadow-buffer technique.) For view-dependent shading, such as specular highlights (Figure 4), however, there is no existing support to separate the two views. A note in the appendix describes the required modification.

3.1 Secondary Scattering

Shader lamps are limited in the type of surface attributes that can be reproduced. In addition, since we are using neutral surfaces with (presumed) diffuse characteristics, secondary scattering is unavoidable and can potentially affect the quality of the results. When the underlying virtual object is purely diffuse, sometimes the secondary scattering can be used to our advantage. The geometric relationships, also known as form factors, among parts of the physical objects, are naturally the same as those among parts of the virtual object. Consider the radiosity solution for a patch i in a virtual scene with m light sources and n patches:

$$B_{i\text{-intended}} = k_{d_i} \sum_j B_j F_{i,j} = k_{d_i} \left(\sum_m B_m F_{i,m} + \sum_n B_n F_{i,n} \right). \qquad (5)$$

Here k_d is the diffuse reflectance, B_j is the radiance of patch j, and $F_{i,j}$ is the form factor between patches. Using shader lamps to reproduce simply the effect of direct illumination (after radiance adjustment), we are able to generate the effect of m light sources:

$$B_{i\text{-direct}} = k_{d_i} \sum_m B_m F_{i,m}. \qquad (6)$$

However, due to secondary scattering, if the neutral surfaces have diffuse reflectance k_u, the perceived radiance also includes the secondary scattering due to the n patches, and that gives us

Figure 3: (Left) The radiance at a point in the direction (θ, ϕ). (Right) The radiance as a result of illumination from a projector lamp. By rearranging the parameters in the optical path, the two can be made equal.
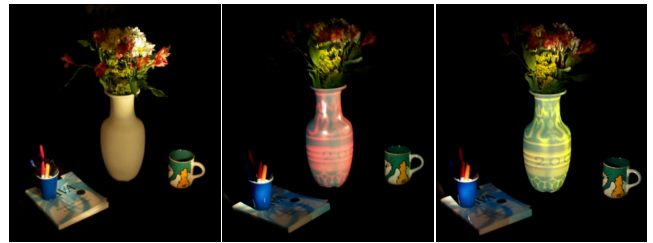

Figure 4: (Left) The underlying physical object is a white diffuse vase. (Middle and right) View-dependent effects, such as specular highlights, can be generated by tracking the user's location and projecting images on the vase.
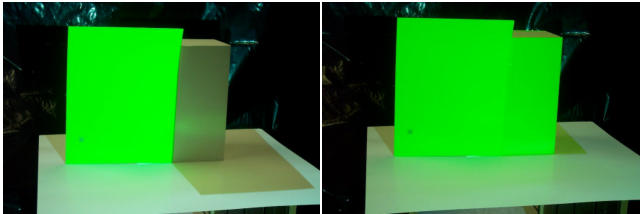
Figure 5: (Left) A green paper illuminated with white light. (Right) The white diffuse surface on the right is illuminated with green light.

$$B_{i\text{-actual}} = B_{i\text{-direct}} + B_{i\text{-secondary}} = k_{d_i} \sum_m B_m F_{i,m} + k_u \sum_n B_n F_{i,n}. \quad (7)$$

The difference between the desired and perceived radiance is

$$\left| (k_{d_i} - k_u) \sum_n B_n F_{i,n} \right|. \quad (8)$$

Thus, in scenarios where k_d and k_u are similar, we get approximate radiosity for "free"—projection of even a simple direct illumination rendering produces believable "spilling" of colors on neighboring parts of the physical objects. From the equation above, the secondary contribution from the neutral surfaces is certainly not accurate, even if we reproduce the first bounce exactly,. The difference is even larger when the virtual object has non-lambertian reflectance properties. We are currently investigating *inverse global illumination* methods so that the projected image can more accurately deliver the desired global illumination effect. Figure 5 shows a green and a white paper with spill over from natural white and projected green illumination. In this special case, the secondary scattering off the horizontal white surface below is similar for both parts.

3.2 Illumination of All Visible Surfaces

One may wonder, given a physical object, what is a good set of viewpoints for the lamps, so that every visible surface is illuminated by at least one lamp. This problem is addressed by [Stuerzlinger99], where he finds, using a hierarchical visibility algorithm, a set of camera viewpoints such that every visible part of every surface is imaged at least once. The problem of determining an optimal set of viewpoints is NP-hard and is related to the art gallery problem [O'Rourke87] known in the field of computational geometry.

4. Methods

The image-based illumination of physical objects has been explored by many. But, we believe, two main challenges have kept the previous efforts to only expensive, large scale, or one-off implementations. (a) First, the geometric registration problem, which is cast as matching the projection of a single 2D image with an object. The projection of a perspective device has up to 11 degrees of freedom (6 external and 5 internal) [Faugeras93], therefore, any effort to manually achieve the registration is likely to be extremely tedious. We propose a new simple technique in subsection 4.1 below. (b) The second problem, which appears to be unexplored, is the complete illumination of nontrivial physical objects in presence of shadows due to self occlusion. With the advent of digitally-fed projectors and real-time 3D graphics rendering, a new approach for image-based illumination is now possible. We approach these problems by creating a 3D geometric understanding of the display setup. In subsection 4.2, we describe an important intensity correction step and in

subsection 4.3, we describe our solution for dealing with shadows.

4.1 Authoring and Alignment

One of the important tasks in achieving compelling visualization is to create the association between the physical objects and the graphics primitives that will enhance those objects when projected. For example, how do we specify which texture image should be used for the face of a building model, or what color distribution will look better for a physical object? We need the physical object as well as its geometric 3D representation, and real or desired surface attributes. As mentioned earlier, many hardware and software solutions are now available to scan/print 3D objects and capture/create highly detailed, textured graphics models. We demonstrate in the video how the authoring can also be done interactively by "painting" directly on top of the physical objects. We also show how the result of the user interaction can be projected on the objects and also stored on the computer. Ideally, a more sophisticated user interface would be used to create and edit graphics primitives of different shape, color and texture.

To align a projector, first we *approximately* position the projector and then *adapt* to its geometric relationship with respect to the physical object. That relationship is computed by finding projector's intrinsic parameters and the rigid transformation between the two coordinate systems. This is a classical computer vision problem [Faugeras93]. As seen in the video, we take a set of fiducials with known 3D locations on the physical object and find the corresponding projector pixels that illuminate them. This allows us to compute a 3×4 perspective projection matrix up to scale, which is decomposed to find the intrinsic and the extrinsic parameters of the projector. The rendering process uses the same internal and external parameters, so that the projected images are registered with the physical objects.

4.2 Intensity Correction

The intensity of the rendered image is modified on a per-pixel basis to take into account the reflectance of the neutral surface, the local orientation and distance with respect to the projector using Equation (4). Since the surface normals used to compute the $1/cos(\theta_P)$ correction are available only at the vertices in polygonal graphics models, we exploit the rendering pipeline for approximate interpolation. We illuminate a white diffuse version of the graphics model (or a model matching appropriate $k_u(x)$ of the physical model) with a virtual white light placed at the location of the projector lamp and render it with black fog for squared distance attenuation. The resultant intensities are smooth across curved surfaces due to shading interpolation and inversely proportional to $(d(x)^2/k_u(x)cos(\theta_P))$ factor. To use the limited dynamic range of the projectors more efficiently, we do not illuminate surfaces with $\theta_P > 60$ (since $1/cos(\theta)$ ranges from 2 to infinity). This avoids the low sampling rate of the projected pixels on oblique surfaces and also minimizes the misregistration artifacts due to any errors in geometric calibration. During the calculations to find the overlap regions (described below), highly oblique surfaces are considered not to be illuminated by that projector. See Figure 7 for an example.

4.3 Occlusions and Overlaps

For complete illumination, using additional projectors is an obvious choice. This leads to the more difficult problem of seamlessly merging images from multiple projectors. A naïve solution may involve letting only a single projector illuminate any given surface patch. But, there are two main issues when dealing with overlapping CRT, LCD or DLP projectors, which compel the use of feathering (or cross-fading) of intensities. The first is the lack of color equivalence between neighboring projectors [Majumder00], due to manufacturing process and temperature color drift during their use. The second is our desire to minimize

the sensitivity to small errors in the estimated geometric calibration parameters or mechanical variations.

Feathering is commonly used to generate seamless panoramic photomosaics by combining several views from a single location [Szeliski97]. Similar techniques are exploited in multi-projector wide-field-of-view displays [Panoram, Trimensions, Raskar99], and two-dimensional arrays of flat projections. In such cases, the overlap region is typically a (well-defined) contiguous region on the display surface as well as in each projector's frame buffer. In the algorithm used in [Szeliski97, Raskar99] the intensity of a pixel is weighted proportional to the Euclidean distance to the nearest boundary (zero contribution) pixel of the (projected) image. The per-pixel weights are in the range [0, 1]. They are multiplied to the pixel intensities in the final rendered image. The pixels weights near the boundary of a source image are near zero and the pixels contribute very little, so that there is a smooth transition to the next source image. This leads to the commonly seen intensity roll-off as shown in Figure 6(a). Under ideal conditions and assuming color equivalence, the weight contribution of both projectors $A+B$ adds up to 1. Even when projector B's color response is different than that of A (say, attenuated—shown as B'), the resultant $A+B'$ (shown in blue) transitions smoothly in the overlap region.

This weight assignment strategy works well only when the target image illuminates a smooth continuous surface at and around the overlap. In our case, the physical model is usually made up of non-convex objects or a collection of disjoint objects resulting in shadows, fragmented overlap regions and, more importantly, overlap regions containing surfaces with depth discontinuities, as shown in Figure 6(c) with a simple occluder. Now, with unequal color response, the resultant weight distribution $A+B'$ has offending sharp changes, e.g. at points f and g. This situation is analogous to image-based rendering (IBR), where warping a single depth-enhanced image creates disocclusion artifacts. When multiple source images are warped to the target image, the color assigned to a pixel needs to be derived (from either a single image where they overwrite each other or) as a weighted combination of corresponding pixels from source images. The feathering, which actually blurs the result, is usually necessary to overcome (minor) color difference
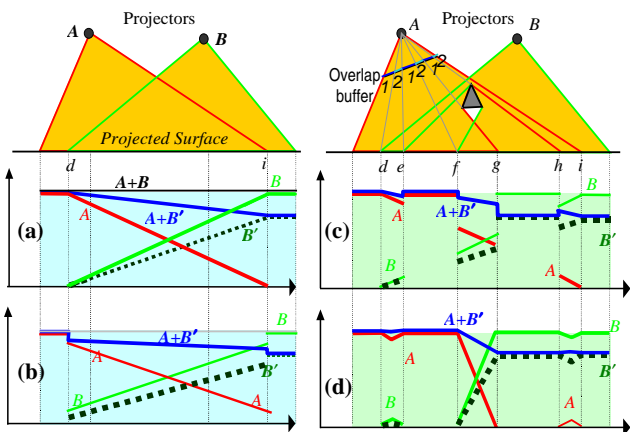

Figure 6: Intensity weights using feathering methods. The plots show the contribution of projectors A, B and B' and the resultant accumulation $A+B$ and $A+B'$ along the lit planar surface. Our technique, shown in (d), creates smooth weight transitions.
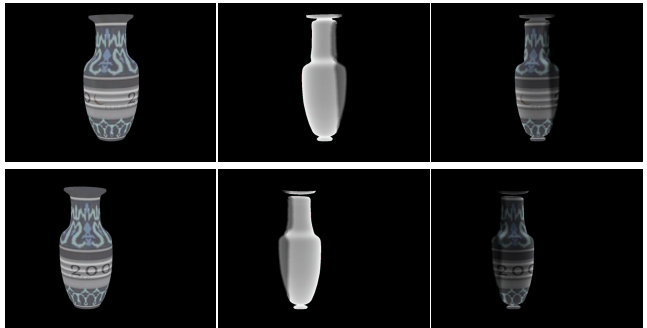

Figure 7: Illuminating a vase (Left) Rendered images (Middle) The intensity weight images, including elimination of oblique parts, and correction for surface orientation and overlap. (Right) Intensity corrected images.

in corresponding pixels in input images and to hide ghosting effects (due to small mis-registration errors). One of the few solutions to this is proposed by [Debevec98], in which they scale the intensities by weights proportional to the angles between the target view and the source views. As mentioned in their paper, "it does not guarantee that the weights will transition smoothly across surfaces of the scene. As a result, seams can appear in the renderings where neighboring polygons are rendered with very different combinations of images." The plots in Figure 6(b) show a sample weighting scheme based on a similar idea and the corresponding problems. Below, we present a global solution using a new feathering algorithm that suits IBR as well as shader lamps. The algorithm is based on the following guidelines:

1. The sum of the intensity weights of the corresponding projector pixels is one so that the intensities are normalized;
2. The weights for pixels of a projector along a physical surface change smoothly in and near overlaps so that the inter-projector color differences do not create visible discontinuity in displayed images; and
3. The distribution of intensity weights for a projector within its framebuffer is smooth so that small errors in calibration or mechanical variations do not result in sharp edges.

In practice, it is easier to achieve (or maintain) precise geometric calibration than to ensure color equality among a set of projectors over a period of time [Majumder00]. This makes condition (2) more important than (3). But, it is not always possible to satisfy condition (2) or (3) (e.g. if the occluder moves closer to the plane so that $f = g$ in Figure 6) and hence they remain as guidelines rather than rules.

The three guidelines suggest solving the feathering problem, without violating the weight constraints at depth discontinuities and shadow boundaries. Traditional feathering methods use the distance to the nearest boundary pixel to find the weight [Szeliski97, Raskar99]. Instead, we first find pixels corresponding to regions illuminated by a single projector and assign them an intensity weight of 1. Then, for each remaining pixel, the basic idea behind our technique is to find the shortest Euclidean distance to a pixel with weight 1, ignoring paths that cross depth discontinuities. The assigned weight is inversely proportional to this distance. Figure 6(d) shows the result of the new feathering algorithm in flatland for two projectors. Even under different color responses, the algorithm generates smooth transitions (see $A+B'$) on the planar surface in presence of shadows and fragmented overlaps. The algorithm can be used for 3 or more projectors without modification.

For a practical implementation, we use two buffers—an overlap buffer and a depth buffer. The depth buffer is updated by rendering the graphics model. The overlap buffer contains integer values to indicate the

number of overlapping projectors for each pixel. The overlap regions (i.e. overlap count of two or more) are computed using the traditional shadow-buffer technique. The algorithm follows:

At each projector,
    Compute boundaries between regions of overlap count 1 and > 1
    Compute depth discontinuities using edge detection in depth buffer
    For each pixel in overlap region
        update shortest distance to overlap count ==1 region ignoring
                    paths crossing depth discontinuity
At each projector,
    For each pixel in overlap region
        Find all corresponding pixels in other projectors
        Assign weights inversely proportional to the shortest distance

For some pixels in the overlap region, such as region [*h,i*] for projector *A*, no nearest pixel with overlap count of 1 can be found, and so the shortest distance is set to a large value. This elegantly reduces the weight in isolated regions and also cuts down unnecessary transition zones. Figure 7 shows the set of images for the illumination of a vase, including weights and intensity corrected images.

5. Limitations

The major limitation of shader lamps is the dependence on the properties of the neutral physical surface and the controlled (dark) ambient lighting. The problem due to secondary scattering cannot be completely avoided, which makes the task of reproducing the behavior of virtual surfaces with very low reflectance very difficult. In addition, traditional projector limitations [Majumder00], such as limited depth of field, reduced dynamic range due to "black level" and non-uniformity, can affect the visual quality.

Although this type of visualization has the advantage that the user is not required to wear stereo-glasses or head-worn displays, the user-induced shadows on the projected surface can be very disturbing. In this paper, we have mainly focussed on the visualization aspect, but a more detailed study of human interaction and issues is necessary.

6. Implementation

For the setup, we used two Sony VPL6000U projectors displaying at 1024×768 resolution. The OpenGL rendering programs run on a Windows NT PC with Wildcard graphics card. The vase is made of clay and is approximately 12 cm × 12 cm × 35 cm. The Taj Mahal model is wooden and spray-painted white. Its dimensions are approximately 70 cm × 70 cm × 35 cm. Both objects were scanned, in about 30 mins each, with a
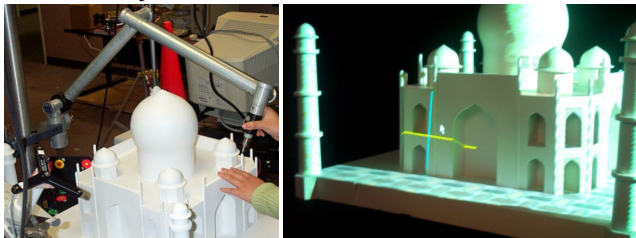

Figure 8: (Left) We use a 3D touch probe scanner to create a 3D model of the real object. (Right) The projectors are calibrated with respect to the model by finding which pixels (center of cross) illuminate the known 3D fiducials.

3D touch probe sensor that gives readings with an accuracy of 0.5 mm (Figure 8 left). The vase model is made up of 7,000 triangles, and the Taj Mahal model is made up of 21,000 triangles and 15 texture maps. For the specular highlight effects, we used the Origin Instruments DynaSight optical tracking system to track the viewer's location.

Each projector is calibrated by finding the pixels that illuminate a set of about 20 known 3D fiducials on the physical model. We accomplish this by moving a projected cross-hair in the projector image-space so that its center coincides with the known fiducials (Figure 8 right). The 3×4 perspective projection matrix and its decomposition into intrinsic and extrinsic parameters of the projector are computed using Matlab. The rendering process uses these parameters so that the projected images are registered with the model. It takes less than five minutes to calibrate each projector. Typically, the re-projection error is less than two pixels and the images from the two projectors appear geometrically aligned on the physical model. The intensity weights for the projector pixels are computed during preprocessing, and it takes approximately 10 seconds for each projector. During rendering, the intensities are modified using alpha-blending available in the graphics hardware. More details and high-resolution colored images are available at the website *http://www.shaderlamps.com.*

7. Applications

In the simplest form, shader lamps can be used to dynamically change the color of day-to-day objects or add temporary markings on them. For example, engineers can mark the areas of interest, like drilling locations, without affecting the physical surface. As seen in Figure 1, we can render virtual shadows on scaled models. City planners can move around such blocks and visualize global effects in 3D on a tabletop rather than on their computer screen. For stage shows, we can change not just the backdrops, but also simulate seasons or aging of the objects in the scene. Instead of randomly beaming laser vector images, we would like to create shapes with laser display on large buildings by calibrating the laser device with respect to a 3D model of the buildings. We can also simulate motion, as shown in the video, by projecting changing texture onto stationary rotationally symmetric objects. Interesting non-photorealistic effects can also be generated.

Tracked viewer. With simple head tracking of the viewer, we have demonstrated how a clay vase can appear to be made of metal or plastic. We could also render other view-dependent effects such as reflections. The concept can be extended to some larger setups. Sculptors often make clay models of large statues before they create the molds. It may be useful for them to visualize how the geometric forms they have created will look with different materials or under different conditions in the context of other objects. By projecting guiding points or lines (e.g. wire-frame), from the computer models, the sculptors can verify the geometric correctness of the clay models. Image-based illumination can be very effectively used in movie studios where miniature models are painstakingly built and then updated with fine details. For inserting synthetic characters into a fly-thru of a miniature set, we can project silhouette of the moving virtual character that looks perspectively correct to the tracked motion camera. This will guide the placement during post-processing because intrinsic camera parameters are not required.

Tracked Objects. We can illuminate objects so that the surface textures appear glued to the objects even as they move. In this case, we can display updated specular highlights even for a static viewer. For example, in showroom windows or on exhibition floors, one can show a rotating model of the product in changing colors or with different features enhanced. In an experimental system, a tracked "paintbrush" was used to paint on a tracked moving cuboid held by the user (Figure 9). The presence of the physical model allows natural haptic feedback. The need to attach a tracker and dynamic mis-registration due to tracker latency are the two main problems [Bandyopadhyay01].

Scaling it up. We have begun to explore extensions aimed at *walk-thru* virtual models of human-sized environments [Low01]. Instead of building an exact detailed physical replica for projection, we are using simplified versions. For example, primary structures of building interiors and mid-sized architectural objects (walls, columns, cupboards, tables, etc.), can usually be approximated with simple components (boxes, cylinders, etc.). As seen in the video, we are using construction Styrofoam blocks. The main architectural features that match the simplified physical model retain 3D auto-stereo, but the other details must be presented by projecting view-dependent images. Nevertheless, our experiment to simulate a building interior has convinced us that this setup can provide a stronger sense of immersion when compared to CAVE[TM] [Cruz-Neira93], as the user is allowed to really *walk around* in the virtual environment. However, because of large concave surfaces (e.g. corners of room), inter-reflection problem becomes more serious. Moreover, since almost all of the surfaces around the user need to be illuminated, it is now easier for the user to occlude some projectors. Strategic placement of projectors is thus more critical, and that (among other things) remains as one of the outstanding challenges.

Ideas. A shader-lamp-guided clay modeling system would be useful as a 3D version of "connect-the-dots" to provide feedback to a modeler. For example, two synchronized projectors could successively beam images of the different parts of the intended 3D model in red and green. A correct positioning of clay will be verified by a yellow illumination. After the shape is formed, the same shader lamps can be used to guide painting of the model, or the application of a real material with matching reflectance properties.

An interactive 3D touch-probe scanning system with closed-loop verification of surface reconstruction (tessellation) could be realized by continuously projecting enhanced images of the partial reconstruction on the object being scanned. This will indicate to the person scanning the required density of points, the regions that lack samples and the current deviation of the geometric model from the underlying physical object.

A useful 2-handed 3D modeling and 3D painting setup would involve tracking the user's viewpoint, input devices and a

Figure 9: A tracked "paintbrush" painting on a tracked cuboid.

coarsely-shaped object (such as a sphere). The user can literally create and add surface properties to a virtual object that is registered with the sphere.

8. Conclusion

We have described a new mode for visualization of 3D computer graphics, which involves light projectors and physical objects to generate rich detailed images directly in the user's world. Although the method is limited when compared to traditional graphics rendered on computer screens, it offers a new way of interacting with synthetic imagery. We have presented new techniques that make image-based illumination of non-trivial objects practical. A rendering process essentially involves user's viewpoint, shape of the graphics objects, reflectance properties and illumination. Traditional computer graphics or head-mounted augmented reality generates the result for all these elements at a reduced temporal (frame rate) or spatial (pixel) resolution. With shader lamps, we attempt to keep the viewpoint and shape at the best resolution and only the added color information is at a limited resolution. We believe the visualization method is compelling for a variety of applications including training, architectural design, art and entertainment.

Acknowledgements

References

[Bandyopadhyay01] D Bandyopadhyay, R Raskar, A State, H Fuchs, *Dynamic Spatially Augmented 3D Painting*. UNC Chapel Hill Tech Report TR01-006, 2001.

[Chen93] S. E. Chen, and L. Williams. *View Interpolation from Image Synthesis*. SIGGRAPH '93, pp. 279-288, July 1993.

[Cruz-Neira93] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti. *Surround-Screen Projection-Base Virtual Reality: the Design and Implementation of the CAVE*. SIGGRAPH '93, July 1993.

[Debevec96] P. Debevec, C. J. Taylor, and J. Malik. *Modeling and Rendering Architecture from Photographs*. SIGGRAPH '96, August 1996.

[Debevec98] P. Debevec, Y. Yu, and G. Borshukov. *Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping*. Proc. of 9th Eurographics Workshop on Rendering, June 1998.

[Dorsey91] J. Dorsey, F. X. Sillion, and D. Greenberg. *Design and Simulation of Opera Lighting and Projection Effects*. SIGGRAPH '91, August 1991.

[Faugeras93] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, Massachusetts, 1993.

[Gortler96] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. *The Lumigraph*. SIGGRAPH '96, August 1996.

[Howard00] *HowardModels.co*m, 7944 Central Avenue, Toledo, OH 43617. *http://www.howardweb.com/model/*.

[Hypermask99] The HyperMask project. *http://www.csl.sony.co.jp/person/nielsen/HYPERMASK/*.

[Kajiya86] J. T. Kajiya. *The Rendering Equation*. Computer Graphics 20(4) (1986), pp. 143-151.

[Knoll92] W. Knoll, and M. Hechinger. *Architectural Models: Construction Techniques*. McGraw-Hill Publishing Company, ISBN 0-07-071543-2.

[Levoy96] M. Levoy, and P. Hanrahan. *Light Field Rendering*. SIGGRAPH '96, August 1996.

[Levoy00] M. Levoy. Personal communication.

[Low01] K. Low, G. Welch, A. Lastra, H. Fuchs. Life-Sized Projector-Based Dioramas: Spatially Real and Visually Virtual. To appear in Sketches & Applications, SIGGRAPH 2001.

[McMillan96] L. McMillan, and G. Bishop. *Plenoptic Modeling*. SIGGRAPH '95, August 1995.

[Liljegren90] G. E. Liljegren and E. L. Foster. *Figure with Back Projected Image Using Fiber Optics*. US Patent # 4,978.216, Walt Disney Company, USA, December 1990.

[Majumder00] A. Majumder, Z. He, H. Towles, and G. Welch. *Color Calibration of Projectors for Large Tiled Displays*, IEEE Visualization 2000.

[Naimark84] M. Naimark. *Displacements*. An exhibit at the San Francisco Museum of Modern Art, San Francisco, CA (USA), 1984.

[O'Rourke87] J. O'Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, New York, 1987.

[Panoram] Panoram Technology. *http://www.panoramtech.com*

[Raskar98] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs. *The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays*. SIGGRAPH '98, July 1998

[Raskar99] R. Raskar, M. Brown, R. Yang, W. Chen, G. Welch, H. Towles, B. Seales, H. Fuchs. *Multi-Projector Displays Using Camera-Based Registration*. IEEE Visualization 99, October 1999.

[Raskar99b] R. Raskar, G. Welch, W. Chen. *Tabletop Spatially Augmented Reality: Bringing Physical Models to Life using Projected Imagery*. Second Int Workshop on Augmented Reality (IWAR'99), October 1999, San Francisco, CA

[Stuerzlinger 99] W. Stuerzlinger. *Imaging all Visible Surfaces*. Graphics Interface '99, pp. 115-122, June 1999.

[Szeliski97] R. Szeliski and H. Shum. *Creating Full View Panoramic Mosaics and Environment Maps*. SIGGRAPH '97, August 1997.

[Trimensions] Trimensions. *http://www.trimensions-inc.com/*

[Underkoffler99] J. Underkoffler, B. Ullmer, and H. Ishii. *Emancipated pixels: real-world graphics in the luminous room*. SIGGRAPH '99, August 1999.

Appendix

As described in Section 3, while the rendering view defined by the projector parameters remains fixed, the shading view is specified by the head-tracked moving viewer. We show a minor modification to the traditional view setup to achieve the separation of the two views using as example OpenGL API.

```
glMatrixMode( GL_PROJECTION );
glLoadMatrix( intrinsic matrix of projector );
glMultMatrix( xform for rendering view )
glMultMatrix( inverse(xform for shading view) );
glMatrixMode( GL_MODELVIEW  );
glLoadMatrix( xform for shading view );
// set virtual light position(s)
// render graphics model
```