Shader Lamps
Animating Real Objects with Image-Based Illumination

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 appearance directly on the object. Furthermore the projected imagery can be used to reproduce alternative appearances, including alternate shading, lighting, and even animation. Because the approach is to effectively "lift" the visual properties of the object into the projector, we call the projectors *shader lamps*.

Some limited effects have previously been demonstrated along these lines for specific applications, however the real challenges to realizing this as a new medium for computer graphics lies in addressing the problems related to complete illumination of non-trivial physical objects. Our approach offers a very compelling method of visualization for a variety of applications including dynamic mechanical and architectural models, animated or "living" dioramas, artistic effects, entertainment, and even general visualization for problems that have meaningful physical shape representations. We present and demonstrate methods for using multiple shader lamps to animate physical objects of varying complexity, from a flower vase, to some wooden blocks, to a model of the Taj Mahal.

Keywords:

Engineering Visualization, HCI (Human-Computer Interface), Illumination Effects, User Interfaces, Virtual Reality, Head Mounted Displays

1. INTRODUCTION

Graphics in the World. Reproducing real scenes with a computer has been a focus of the graphics community for almost as long as there has been a graphics community. Whether using geometric models and physically-based rendering, or image-based models and image-based rendering, the idea is 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 in the real world. Primarily this has been done visually for either special effect in movies, or in real time for *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* [Underkoffler97, Underkoffler99a] 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. In general the broad goals of computer graphics are many, including visualization, intuition, communication, imagination, art, and entertainment. During an invited talk at Microsoft Research in 1996 (associated with SIGGRAPH 96) Jim Kajiya noted that

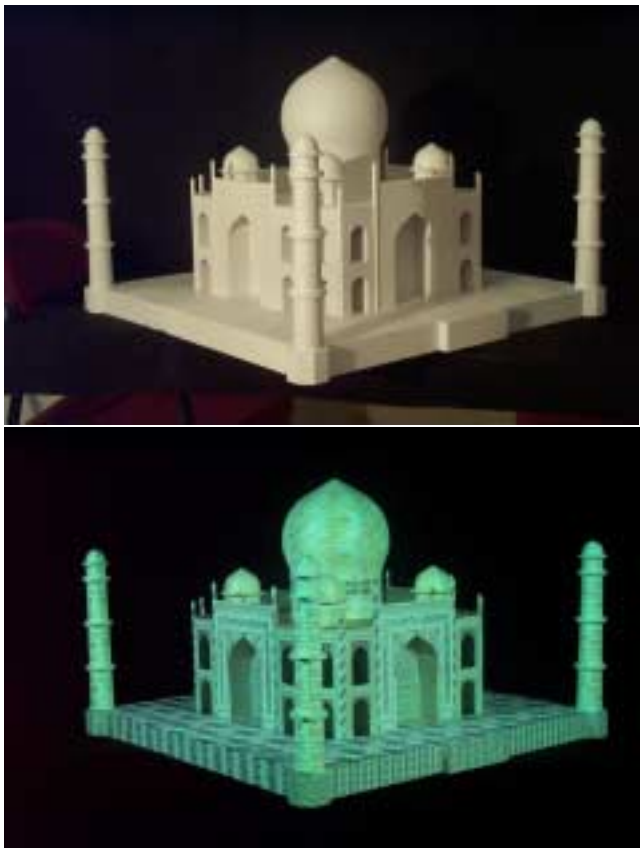


Figure 1: Underlying physical model of Taj Mahal (top) and enhanced with *shader lamps* (bottom).

> "Computer Graphics is useful not only for augmenting one's own imagination but [for] stimulating the imagination of others. We can use it to codify, transmit, store, and communicate experience and ideas. Computer graphics as a medium is only just emerging." [Kajiya96]

With respect to the stimulation and communication of ideas, we are struck that 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. (See for example [Howard99].) The architects that we have spoken with, and many books on the subject, note that while it is true that designers cannot do without CAD tools any more, "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) for example. 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. Our goal is to enjoy the many advantages of the natural physical interface, in particular the auto-stereoscopic nature of viewing physical objects, and the richness of computer graphics.

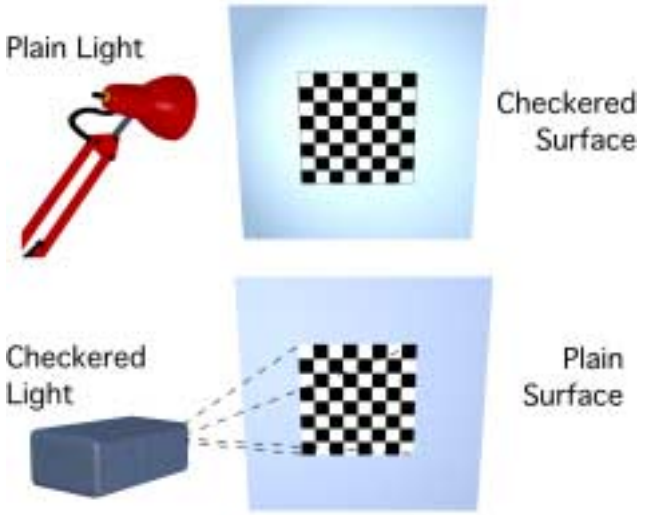


Figure 2: Concept of *shader lamps*. Physical textures (above) and shader lamp textures (below).

Image-Based Illumination. Normally in the physical world, the color, texture, and lighting associated with the surface of a physical object are an integral part of the object. In computer graphics this is typically modeled with a BRDF for the surface. When we illuminate the object with a white light, the surface reflects particular wavelengths of light, and we perceive the respective surface attributes. 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 the same. 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 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, we have achieved results that are quite realistic and compelling for a broad range of surfaces and attributes.

The need for an underlying physical model is arguably unusual for computer graphics, however it is not for architects [Howard99], artists, and computer animators. In addition, various approaches to automatic three-dimensional fabrication or *solid freeform deformation* are steadily becoming a reality, albeit an expensive reality at the present time. (Methods include Laminate Object Manufacturing, Stereolithography, and Fused Deposition.) It is not unreasonable to argue that three-dimensional printing and faxing is coming. In the mean time, if necessary one can use a 3D probe device. We used such a device (Faro) for our Taj Mahal model.

We previously presented preliminary thoughts and results in workshop settings [xxxx_98, xxxx_99]. 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 lamps to animate physical objects of varying complexity—a smooth flower vase and a relatively complicated model of the Taj Mahal. Using our methods one can create compelling dynamic mechanical or architectural models, "living" dioramas, and eventually vibrant hand-held physical user-interface objects.

2. PREVIOUS WORK

Tangible luminous interfaces. At least since 1997 John Underkoffler et al. at the MIT Media Lab have been using projectors as a means to injecting imagery into the real physical surroundings of a room or a designated workspace [Underkoffler97, Underkoffler99a, Underkoffler99b]. Beyond simply projecting information, Underkoffler et al. have articulated and implemented the elegant and useful idea of an "I/O Bulb" (device) that both projects and captures imagery in the environment. The work we present here is distinct from, but complementary to, this work at MIT. A primary distinction is that their main focus is the 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 physical objects (building "phicons") in the scene, track their 3D pose (2D position and orientation), and project light from overhead to reproduce the appropriate sunlight shadows. We are intrigued by the elegant interactive and functional simulation component of this work, as indicated later in "Future Work." However we are primarily interested in the use of physical objects as a *truly* three-dimensional display devices for more general computer graphics and visualization, including the stimulation and communication of ideas, and aesthetic (artistic) applications. We find appeal in the notion of separating physical objects and their visual appearance properties, and we seek to address the many challenges to making image-based illumination practical as a medium for computer graphics.

Modeling and rendering architecture from photographs. The "Facade" project and related work by Paul Debevec et al. at the University of California at Berkeley on modeling and rendering architecture from a sparse set of photographs is relevant and complementary in a particularly exciting way [Debevec96]. While their goals are quite different from ours, and we are not limited to working with models of human-made structures, there are significant similarities. In particular their hybrid approach to using geometry and images to reproduce physical human-made structures is similar in principal, and addresses similar (but different) challenges. Most similar and relevant are the challenges 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, we face them with real (light projected) imagery and physical models. We are excited about the eventual mainstream availability and use of tools for both parts of the problem. In the future one could (for example) use Facade tools to build a hybrid geometry and image model of a university campus, and then use our shader lamp techniques to animate a scaled physical model, effectively creating a "living diorama" of the campus.

"Displacements." A compelling image-based example of something similar to our notion of a shader lamp is work by Michael Naimark (Interval Research) in 1984 [Naimark84]. In a

San Francisco Museum of Modern Art exhibit titled "Displacements," Naimark used projectors to present some in-place image-based modeling and rendering content. The image-based content was captured by using a rotating movie camera to film the contents of a living room, replete with furniture and people. The room and furniture were then painted completely white (neutral), and the captured imagery was re-projected back onto the walls using a rotating projector that was precisely registered with the original camera. The most relevant aspect of this work is the explicit separation and then later merging of the physical and visual properties of a real scene. We are interested in pursuing the full exploitation of this idea, including the manipulation of the visual properties, and in solving the challenges related to multiple overlapping sources of illumination.

Theater and entertainment. Theatrical scene (set) and lighting designers have used colored and even "textured" lighting to stimulated moods and ideas, and to simulate the effects of real lighting in a scene. In fact, the use of 3D objects as graphical display devices is similar to theater in that often seeing the content in person, with the rich physical and optical characteristics, is far more compelling than seeing a 2D version through conventional display devices. Of course the computer graphics special effects demonstrated in modern movies today offer an example of the richness and flexibility generally not possible in the physical theater. In fact, with the realization of our methods, it is reasonable to consider the inclusion of some computer-generated special effects directly on physical objects in live theater, thus realizing some of the richness of each medium. A limited but compelling (and possibly seminal) example of this idea is the use of projectors to animate artificial human heads in the Walt Disney World "Haunted Mansion" attraction. Projected imagery animates four neutral busts of singing men, and a patented projector and fiber-optic setup animates the head of the fictional fortune teller "Madame Leota" inside a real crystal ball [Liljegren90].

Figure 3: From the Walt Disney World "Haunted Mansion," still cells of animated faces projected onto neutral busts (left), and Madame Leota's head (right).

On a more physically grand scale, projectors have recently been used to render a variety of lighting and projected imagery on a very large architectural scale. For example, in 1952 Paul Robert-Houdin used sounds and colored lights on a building for nighttime entertainment. The most well-known modern realization of this idea is the Son et Lumiere (light show) at/on the Blois castle in the Loire Valley (France). In addition the medium is now being used elsewhere around the world at sites such as the Forum (Rome), the Parthenon (Athens); Greenwich Palace; Independence Hall (Philadelphia); the Pyramids of Giza (Cairo); the Red Fort (Delhi); and the ruins of Teotihuacán (near Mexico City); Most recently dynamic imagery was and the Millenium celebration at the three Pyramids at Giza in Cairo (Egypt).

Finally, to realize the general application of this technique one must, among other things, have a method for pre-warping imagery to "fit" the physical object so that it appears correct to local viewers. It is worthy to note that this problem is very similar to that faced by Julie Dorsey et al. [Dorsey91] in trying to model the appearance of theatrical backdrops so that they appear correct from the audience's perspective. We use techniques that build on this (similar to [Raskar98]) to render onto the potentially very non-planar surfaces of physical objects, and new techniques to address the occlusion, sampling, and blending issues.

3. THE RENDERING PROCESS

The appearance of a surface is decided by the radiance at that surface. Hence, it should be possible to reproduce the same appearance on neutral surfaces by rearranging the incident radiance. This result should not be surprising, as that is what we are used to typically with projector screens. Indeed, in the following derivation we will see that reproducing the surface appearance on neutral surfaces is equivalent to rendering the image for a given viewer location and warping it to render from the projector lamp with intensity modifications.

First, let us consider the rendering equation, which is essentially a geometrical optics approximation as explained in [Kajiya86]. The radiance at 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

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

and $g(x, \theta, \phi)$ is the geometry term (visibility and distance), $L_e(x, \theta, \phi)$ is the emitted radiance of the point (non-zero only for light sources), and $F_r(x, \theta, \phi, \theta_i, \phi_i)$ is the bidirectional reflection distribution function (BRDF) for 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 but with diffuse reflectance is given by,

$$L'(x, \theta, \phi) = g(x, \theta, \phi) k_u(x) L_P(x, \theta_P, \phi_P) \cos(\theta_P) \qquad (2)$$

where $L_P(x, \theta_P, \phi_P)$ = radiance of projector in the direction (θ_P, ϕ_P), $k_u(x)$ = surface diffuse reflectance factor at (x), (θ_P, ϕ_P) is the incident direction at the surface point (x). The radiance is converted into a discretized pixel value using filtering and tone representation.

Clearly, we can reproduce equivalent radiance for a given viewer location, by solving Eqn. (2) for L_P:

$$L_P = \frac{L(x, \theta, \phi)}{k_u(x) \cos(\theta_P)} \quad \text{for } k_u > 0. \qquad (3)$$

Using any rendering technique, we can first compute $L(x, \theta, \phi)$ at a surface point for the given viewer location, and then compute the corresponding projector image intensities using the equation above. This is somewhat unusual as normally intensities are computed as an image seen by the viewer and not associated with surface points in object-space. The method of warping this image so that it appears to be captured from the lamp's viewpoint is well known in the image-based rendering literature [Chen93] [Mcmillan95]. Thus, in one way, this is equivalent to rendering and

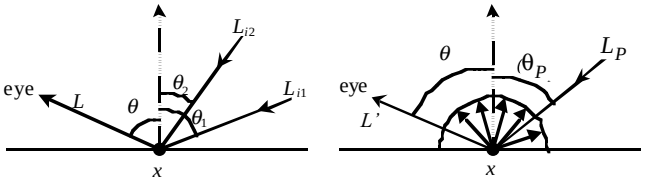


Figure 4: (left) The radiance at a point in the direction (θ, ϕ) (right) The radiance as a result of illumination from a projector lamp. By rearranging the parameters in the optical path, the two can be made equal.

warping the image followed by intensity correction. For a changing viewer location, view-dependent shading under static lighting conditions, can also be implemented [Debevec98][Levoy97][Gortler96]. However, the warping can be avoided by realizing that the display medium is the same as the object. A modification of the general rendering method is required: the eye-point for shading calculations is at a different location than the center of perspective projection. The visibility calculations can be performed without any modification from the projector lamp's viewpoint because the viewer naturally sees only the physically unoccluded parts of the real objects.

Since we want to animate the physical objects, we are forced to use real-time techniques. In current real-time 3D rendering APIs, the solution to the general rendering equation is approximated. The BRDF computation is divided into view-dependent specular, and view-independent diffuse and ambient terms. View-independent shading calculations can be done by assuming the user and the projector lamp are at the same location. Rendering of shadows is also view-independent (although not supported directly), and they are computed using the traditional two-pass shadow-buffer technique. For view-dependent shading, such as specular highlights, however there is no existing support. Appendix I describes a simple modification that allows rendering view-dependent shading without additional cost.

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, secondary scattering is unavoidable and can potentially affect the quality of the results. On the other hand, the secondary scattering can be used to our advantage in cases where underlying virtual object is purely diffuse. The geometric relationships, also known as form factors, among parts of the physical objects are naturally the same as that among parts of the virtual object. Suppose, we consider only the direct illumination during view-independent shading calculations (as usually is the case with real-time APIs). After the appropriate intensity correction Eqn. (3), we will be able to generate the correct radiance at patch i due to m different virtual light sources

$$B_{i\text{-direct}} = k_{d_i} \sum_m B_m F_{i,m}$$

where k_d is the diffuse reflectance, B_m is the radiance of virtual light sources and $F_{i,m}$ is the form factor between the light sources and this patch. However due to secondary scattering, if neutral surfaces have diffuse reflectance k_u, the perceived radiance includes the secondary scattering due to the n patches:

$$B_{i\text{-actual}} = k_{d_i} \sum_m B_m F_{i,m} + k_u \sum_n B_n F_{i,n}.$$

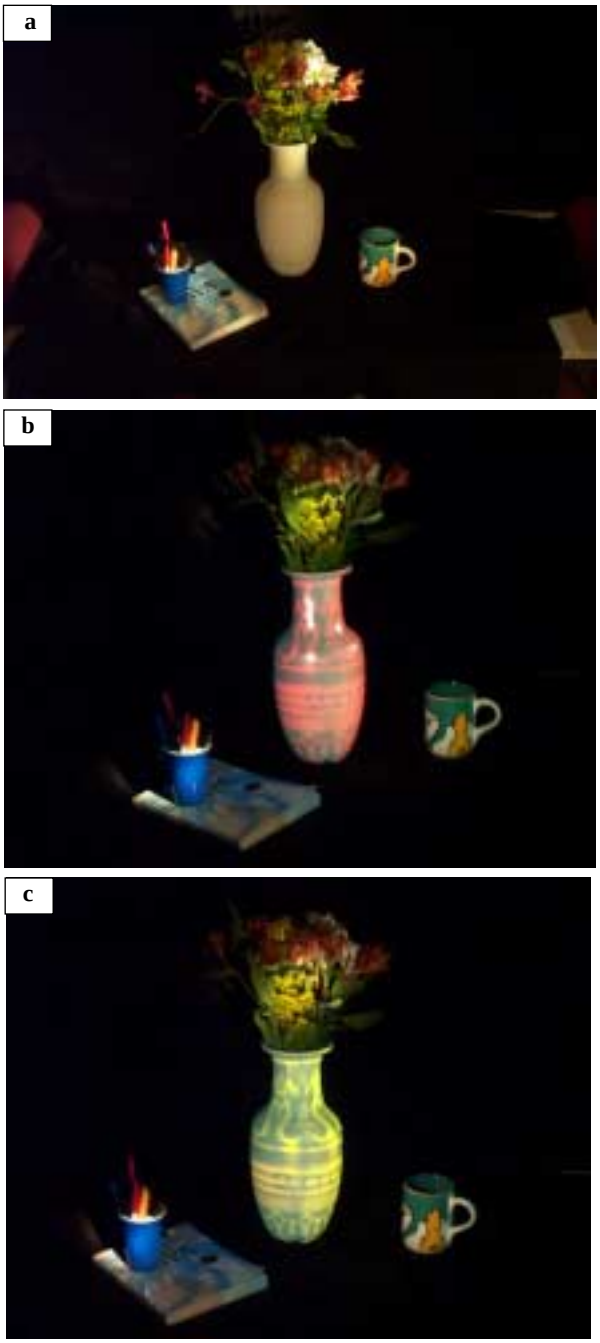


Figure 5: (a) The underlying physical object is a white diffuse vase (b) The vase can be effectively 'painted' by projecting an image with view-independent diffuse shading, textures and intensity correction. Some view-dependent effects such as specular highlights are generated for a given user location by modifying reflectance properties of the graphics model. (c) The same vase with different material properties.

This is very close to the radiosity solution for non-emitters considering all the m light sources and n patches,

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

Thus, the secondary contribution from neutral surfaces is not accurate but still results in the 'spilling' of colors on neighboring parts of the physical objects. Figure 6 shows green paper with spill over, with and without illumination.

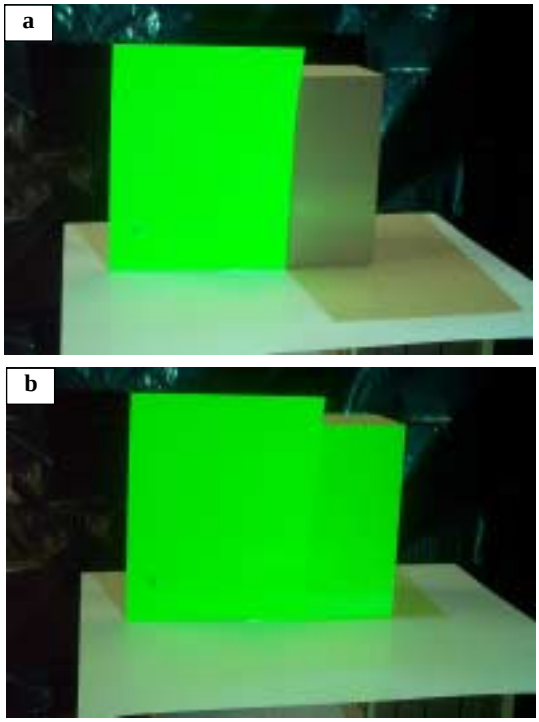


Figure 6: (a) A green paper illuminated with white light (b) The white diffuse surface on the right is illuminated with green light. In this special case, the secondary scattering off the white surface below is similar for both parts.

3.2 Illumination of All Visible Surfaces

A complete illumination of physical object clearly needs more than one lamp. 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 the set of viewpoints for cameras so that every visible part of every surface is imaged at least once, using a hierarchical visibility algorithm. 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. However, the techniques are still very limited and it is treated as a problem of 2D registration between a single image and its projection on an object. The overall process is very well-defined in advance and used in large entertainment centers or theaters. With the advent of digital light projectors and real-time 3D graphics, we believe that some additional tools can make the task of animating neutral physical objects as simple as rendering on a computer screen.

Some major challenges have kept such efforts to only large-scale implementations. Many people consider the task of aligning the images correctly with the physical object to be very cumbersome. This is usually solved by tedious electro-mechanical adjustments and then kept in place by rigid mechanical construction. The other important problem is dealing with shadows due to self-occlusion with respect to the projectors. We treat the problem of alignment as essentially a problem of 3D calibration of a pin-hole projection device. We illuminate the shadowed parts of the object by adding more projectors and then address the issue of merging overlapped images from multiple projectors.

4.1 Authoring and Alignment

One of the important tasks in achieving compelling visualization is to create association between the physical objects and the graphics primitives that will enhance those objects when projected. We need the physical object as well as its geometric 3D representation. For most applications, the physical object is already available but not necessarily its 3D graphics model. As mentioned in Section 1, many hardware and software solutions are now available to scan 3D objects and create highly detailed, textured graphics models. On the other hand, when the 3D definition is available, a single colored physical model can be created using 3D printers. In our case we used a touch probe 3D scanner to record key features and then used a commercial 3D modeling tool to assign textures and materials. The authoring can also be done interactively by 'painting' directly on top of the physical objects. As demonstrated in the video, the result of user interaction is 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. Then a user may be able to make decisions about, for example, which texture image should be used for the face of a building model, or what color distribution will look better for a physical object.

Calibrating a projector with respect to the physical objects involves finding its internal parameters and the rigid transformation between the coordinate system of the objects and the projector. This is a classical computer vision problem [Faugeras93]. For our demonstrations, we take a set of fiducials with known 3D locations on the physical object and then find corresponding projector pixels that illuminate them. This allows us to compute a 3x4 perspective projection matrix up to scale, which is decomposed to find the internal and the external 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 to take into consideration the reflectance of the neutral surface and the local orientation of the surface with respect to the projector. To compute the correction using the cosine term at each projector pixel, we need the direction of the normal at the surface illuminated by the pixel. For polygonal graphics models, the surface normal is available only at the vertices. We instead use a simple approximation inside our rendering program by illuminating a white diffuse version of the graphics model with a virtual white light placed at the location of projector lamp. The resultant intensities are smooth across curved surfaces due to shading

interpolation and directly proportional to the cosine of the angle between view vector and surface normal. For angles α greater than sixty degrees, the correction $(1/(\cos(\alpha)))$ is greater than a factor of two. To use the limited dynamic range of the projectors more efficiently, we do not illuminate surfaces facing at angles greater than sixty degrees. This avoids the low sampling rate of projected pixels on oblique surfaces and also minimizes the misregistration effects due to the errors in geometric calibration. During the calculations to find overlap regions, described below, the highly oblique surfaces are considered not illuminated.

4.3 Occlusions and Overlaps

A major problem with using projectors is the presence of shadows due to self-occlusion on the physical object. A single projector can only partially illuminate a closed object. Using additional projectors is an obvious choice. However this leads to the more difficult problem of seamlessly merging images from multiple projectors. This situation is analogous to image based rendering techniques, where warping a single depth-enhanced image creates dis-occlusion artifacts. When multiple source images are warped to the target image, the color assigned to a pixel is derived from either a single image (where they overwrite each other) or as a weighted combination of pixels from multiple images. With projectors the resulting intensity is always the sum of all intensities and there is no 'winning' pixel.

The luminance in the overlap region may be much greater than that in regions illuminated by only one projector. Thus in addition to geometric alignment between projected images, it is also necessary to achieve intensity normalization. The problem of generating seamless images using multiple projectors has been explored for large wide-field-of-view displays [Panoram] [Trimensions] [Raskar99], as well as two-dimensional arrays of flat projections [Humphreys99] [Czernuszenko97]. In such cases, the overlap region is typically a (well-defined) contiguous region on display surface as well as in each projectors frame buffer. The intensity of projector pixels is weighted using feathering (also known as intensity roll-off or soft-edge) techniques so that the overlapping images blend to create a single seamless image.

However, in our case, the physical model is usually made up of non-convex objects or a collection of disjoint objects resulting in overlap regions that are fragmented in each projector's frame buffer. Traditional feathering techniques weight the pixel intensities proportional to the distance to the nearest boundary (or invisible) pixel in the source image. The weights multiply the intensities in the final rendered image and range between [0, 1.0]. The pixels near the boundary of a source image contribute very little, so that there is a smooth transition to the next source image. This works well only when the target image is a single continuous surface at and around the overlap. Some examples are the final images in photo-mosaics [Szeliski97] and tiled displays on planar or curved surfaces [Panoram] [Trimensions][Raskar99].

We describe a new modified feathering algorithm for assigning intensity weights when the surfaces at which the source images are merged may have depth discontinuities. This blending technique can also be used in image based rendering to determine the contribution of pixels in each image to the novel view.

Our 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;

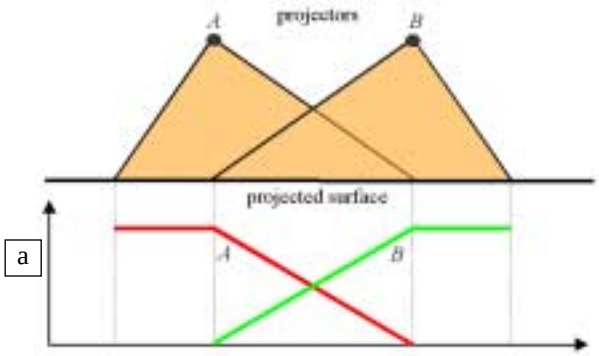
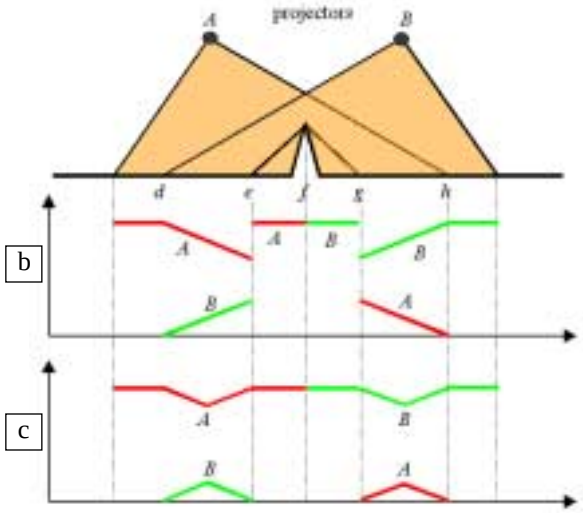


Figure 7: Intensity weights using traditional feathering method (a) Simple intensity ramps on planar overlap create smooth transitions.



(b) Concave object in overlap region, the weight at point *g* due to projector A is not zero creating a large change in A's contribution (c) By considering depth layers in the modified feathering technique, intensities due to each projector change smoothly on real surfaces.

2. The intensity contribution of a projector along a surface on the physical object changes smoothly so that projectors which differ in color properties do not create visible discontinuity in 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.

When the illuminated surface is continuous, the conditions (2) and (3) are essentially the same. Note that it is not always possible to satisfy the conditions (2) or (3) even for smooth continuous surfaces. So they are used only as guidelines. For surfaces with depth discontinuity with respect to any projector, it is more important to ensure (2) than (3). This is because, in practice, it is relatively easy to achieve (or maintain) precise geometric calibration but difficult to ensure color equality among a set of

projectors. Broadly, the three guidelines essentially suggest solving the feathering problem at each 'depth layer' separately. A depth layer corresponds to contiguous regions in the depth buffer with no discontinuity between any neighboring pixels of the region. Such a solution will avoid feathering intensities across depth discontinuities. Traditional feathering methods use the distance to the nearest boundary (i.e. zero contribution) pixel to find the weight. Instead, we first find pixels corresponding to regions illuminated by a single projector and assign an intensity weight of 1.0. Then, for the remaining pixels, the basic idea behind our technique is to find the shortest distance to a pixel region with weight 1.0 and which is also in the same depth layer. The assigned weight for this pixel is inversely proportional to this distance.

For a practical implementation we use two buffers, an overlap buffer and a depth buffer. For the sake of simplicity, we will use the word 'pixel' to mean the pixel in the framebuffer as well as the 3D point illuminated by the pixel. In the overlap buffers, integer values are assigned to a pixel depending on how many pixels from other projectors overlap with this pixel. If the pixel does not illuminate any useful region, the value is zero. If no other projectors overlap with this pixel, the value is one. The overlap regions (i.e. overlap count of two or more) are computed using the traditional shadow-buffer algorithm that finds pixels in the current view lit by a point light source. The light sources in this case are the other projector lamps. The depth buffer simply stores the resultant depth values when the graphics model of the physical object is rendered. First we find overlap boundaries in overlap buffer between pixels with value 1 and pixels with value more than 1. In the depth buffer, we find boundaries between pixels with a large depth difference. The shortest distance (in pixel units) for each pixel in the overlap region is then found by finding the euclidean distance to the nearest pixel in the region of count 1, ignoring paths that cross over a depth discontinuity. For some pixels in the overlap region, no pixel of count 1 is found in the same depth layer and the distance is set to a large value. Finally, using the traditional 3D warping technique, we find all corresponding pixels that illuminate the same 3D point and assign a weight inversely proportional to the computed shorted distance. For example, if corresponding pixels P_a and P_b in two overlapping projectors A and B have assigned distance of d_a and d_b, then P_a is assigned the weight

$$\frac{1/d_a}{1/d_a + 1/d_b} = \frac{d_b}{d_a + d_b}$$

and similarly P_b is assigned the weight

$$\frac{d_a}{d_a + d_b}.$$

5. ISSUES

Today's projectors have a limited depth of field and hence cannot create images that remain in focus over the a large physical object. The problem due to secondary scattering cannot be completely avoided which makes reproducing the behavior of surfaces with very low reflectance is difficult. The same problem is made even worse by the 'black level' of the projectors i.e. the non-zero illumination when the rendered pixel color is black. A major problem during implementation is the non-linearity of projector illumination with respect to the values in the framebuffer. We compensate for the non-linearities with a gamma factor in the weighting functions, but that does not solve the problem completely.

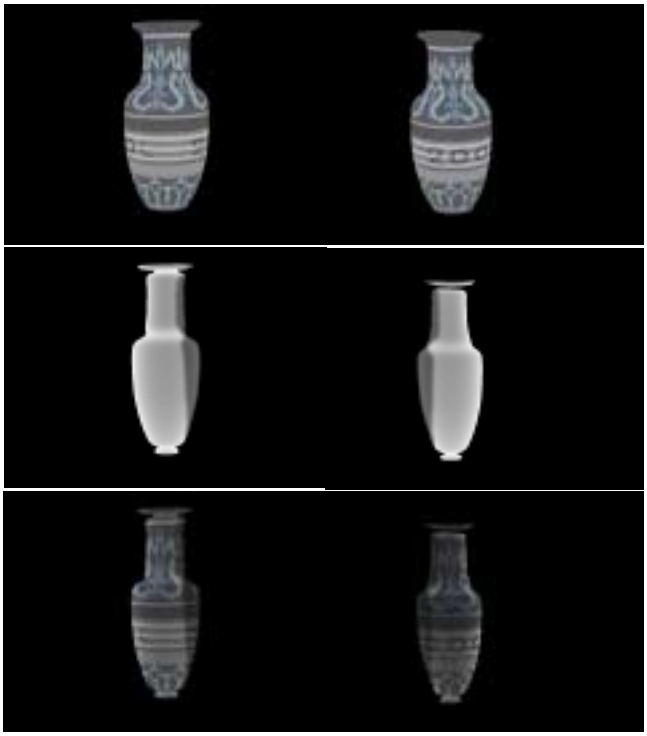


Figure 8: The vase is illuminated by two projectors. (a-b) Images rendered by first and second projectors. (c-d) The intensity weight images, including elimination of oblique parts, and correction for surface orientation and overlap (e-f) Final projected images after intensity normalization.

In terms of user interaction, shadows of the user on the projected surface can be disturbing. However, the method has the great advantages of not requiring the user to wear stereo-glasses or head-mounted displays.

6. IMPLEMENTATION

For the setup, we used two Sony VPL6000U projectors displaying at 1024x768 resolution. The OpenGL rendering programs run on a Windows NT PC with Wildcard graphics card. The vase is made up of clay and is approximately 12 cm x 12 cm x 35 cm. The Taj Mahal model is wooden and spray painted white. The dimensions are approximately 70 cm x 70 cm x 35 cm. Both objects were scanned with a 3D touch probe sensor which gives readings with an accuracy of 0.5 mm. Since the vase is a surface of revolution, we recorded points on the curve and created a surface model using Rhino3D modeling package. The Taj Mahal was scanned in by recording key features. (We collected approximately 100 points for the Taj Mahal, and 25 for the vase.) The vase model is made up of 7000 triangles. 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 find the viewer location.

The projectors are calibrated by finding pixels that illuminate known 3D fiducials on the model. Selected approximately 20 points on the model. Then we align a projected cross-hair by moving it in the projector image-space. The 3x4 perspective projection matrix and its decomposition into internal and external 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 maximum re-projection error was less than two pixels and the images from the two projectors appear geometrically aligned on the physical model. The intensity weights for projector pixels are computed during preprocessing and it takes approximately 10 second for each projector. The intensities during rendering are modified using alpha blending available in the graphics hardware. More details and high resolution colored images are available at the anonymous website http://members.xoom.com/shaderlamps.

Figure 9: (a) We use a 3D touch probe scanner to create a 3D model of the real object. (b) The projectors are calibrated with respect to the model finding which pixels (the center of the cross) illuminate known 3D features.

7. POSSIBILITIES

We believe the potential of shader lamps extends much beyond what we have described or even imagined. While we have focussed on techniques for creating image based illumination with traditional computer graphics, additional technologies such as tracking, vision-based recognition and smart building blocks can take this medium into new territories.

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 demonstrated by [Underkoffler99b], city planners can move around physical scaled blocks and visualize global effects such as shadows and wind patterns in 3D. For stage shows, we can change not just the backdrops, but also simulate seasons or aging of the objects in the scene. In the video, we show how motion can be simulated out of stationery objects by changing the texture mapped on the objects. Interesting non-photo-realistic effects can also be generated.

With simple head tracking, we have demonstrated how a vase made of clay can appear to be made of metal or plastic. It is easy to render other view dependent effects such as reflections. The concept can be extended to much larger setups. Sculptures often make clay models of large statues before they create a mold. It may be useful for them to visualize how the geometric form they have created will look with different material or under different conditions in context of other objects. We believe image based illumination can be very effectively used in movie studios where miniature models are painstakingly built and then updated with fine detail. With tracked motion camera, it is even possible to project the silhouettes of moving virtual characters, so that the post-processing task of inserting computer graphics characters can be simplified.

When multiple people want to simultaneously want to look the enhanced object, we can track and illuminate moving physical objects with registered colors and textures. 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.

Our video shows a demonstration of interactive spray painting on top of real objects. A useful tracked input device could be a “paint brush” that allows natural haptic feedback. The result of the interaction is then stored to make it a truly 3D paint system. We are also excited about a 2-handed 3D modeling and 3D painting setup where user's viewpoint, input device and a course shaped object (such as a sphere) are tracked. The user can literally create and add surface properties to a virtual object that's always registered with the sphere.

8. CONCLUSION

We have described a new paradigm for 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. 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. As we have seen, the concept of shader lamps attempts 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 architectural design, art and entertainment.

9. REFERENCES

[Brooks99] Fred Brooks, personal communication.

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

[Czernuszenko97] M. Czernuszenko, D. Pape, D. Sandin, T. DeFanti, L. Dawe, and M. Brown. The ImmersaDesk and InfinityWall Projection-Based Virtual Reality Displays. In Computer Graphics, May 1997.

[Debevec96] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. “Modeling and Rendering Architecture from Photographs,” in SIGGRAPH '96, August 1996.

[Debevec98] Paul Debevec, Yizhou Yu and George Borshukov, Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping, Proc. of 9th Eurographics Workshop on Rendering, Vienna, Austria, June, 1998

[Dorsey91] Dorsey, Julie O’B., Fransco X. Sillion, Donald Greenberg. 1991. “Design and Simulation of Opera Lighting and Projection Effects,” SIGGRAPH 91 Conference Proceedings, Annual Conference Series, ACM SIGGRAPH, Addison-Wesley, pp. 41-50, 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", in SIGGRAPH96, August 1996

[Howard99] *HowardModels.com*, 7944 Central Avenue, Unit 3, Toledo, Ohio 43617. Available at http://www.howardweb.com/model/index.html [cited January 9, 2000].

[Humphreys99] Humphreys, Greg and Pat Hanrahan. “A Distributed Graphics System for Large Tiled Displays” in Proceedings of IEEE Visualization 99, San Fransisco, CA, October 24-29, 1999.

[Kajiya86] KAJIYA, J. T. The rendering equation. Computer Graphics 20, 4 (1986), 143--151.

[Kajiya96] Jim Kajiya. 1996. “The Future of Computer Graphics,” invited talk, Microsoft Research Campus, Building 12, San Juan Room, May 28, 1996. Available from http://www.research.microsoft.com/siggraph/talks/1996.05/ [cited January 8, 1999].

[Knoll92] Wolfgang Knoll and Martin Hechinger, 1992. *Architectural Models: Construction Techniques*, McGraw-Hill Publishing Company, ISBN 0-07-071543-2.

[Levoy96] M. Levoy and P. Hanrahan, Light Field Rendering, in SIGGRAPH96, August 1996

[McMillan96] McMillan, Leonard, and Gary Bishop. Plenoptic Modeling, Proceedings of SIGGRAPH 95, (Los Angeles, CA), August 6-11, 1995. pp 39-46.

[Liljegren90] Gordon E. Liljegren and Eugene L. Foster. 1990. “Figure with Back Projected Image Using Fiber Optics,” US Patent # 4,978.216, Walt Disney Company, Burbank California, USA, December 18, 1990.

[Naimark84] Michael Naimark, “Displacements,” an exhibit at the San Francisco Museum of Modern Art, San Francisco, CA (USA), 1984.

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

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

[Raskar98] Raskar, Ramesh, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, and Henry Fuchs. 1998. “The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays,” in SIGGRAPH 98 July 1998

[Raskar99] Raskar, Ramesh, Michael S. Brown, Ruigang Yang, Wei-Chao Chen, Greg Welch, Herman Towles, Brent Seales, Henry Fuchs. 1999. "Multi-Projector Displays Using Camera-Based Registration," in Proceedings of IEEE Visualization 99, San Fransisco, CA, October 24-29, 1999.

[Stuerzlinger 99] W. Stuerzlinger, Imaging all Visible Surfaces, in Proceedings Graphics Interface '99 (Kingston, Ontario), pp. 115-122, June 1999.

[Szeliski97] Richard Szeliski and Heung-Yeung Shum, "Creating Full View Panoramic Mosaics and Environment Maps", in SIGGRAPH 97, August 1997.

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

[Underkoffler97] John Underkoffler. 1997. “A View From the Luminous Room,” Personal Technologies, Vol. 1, No. 2, pp. 49-59, June 1997.

[Underkoffler99a] John Underkoffler, Brygg Ullmer and Hiroshi Ishii. 1999. “Emancipated pixels: real-world graphics in the luminous room,” in Proceedings of the SIGGRAPH 1999 annual conference on Computer graphics, August 8 - 13, 1999, Los Angeles, CA (USA), Pages 385 – 392

[Underkoffler99b] John Underkoffler and Hiroshi Ishii. Urp: A Luminous-Tangible Workbench for Urban Planning and Design, in Proceedings of Conference on Human Factors in Computing Systems (CHI '99), (Pittsburgh, Pennsylvania USA, May 15-20, 1999), ACM Press, pp. 386-393.

[xxxx_98] workshop presentation (details sanitized for blind review process).

[xxxx_99] workshop presentation (details sanitized for blind review process).

APPENDIX

As described in Section 3, for diffuse shading, the viewer location could be assumed to be the location of the projector lamp. The location itself is defined by the perspective projection matrix used for rendering. However, for view-dependent lighting calculations for effects such as specular highlights, the eye-point should be at the specified head-tracked viewer-location. Although unusual, this is a minor modification. Real-time rendering APIs, however, do not support this feature. For the convenience of anyone who wants to implement this, we give here a brief outline of an OpenGL program that achieves the same effect without any additional cost.

```
glMatrixMode( GL_PROJECTION );
// internal/external params of proj matrix
glMultMatrix(inverse(xform for eye-point))
glMatrixMode( GL_MODELVIEW );
glLoadIdentity();
glMultMatrix(xform for eye-point)
// set light position
// draw scene
```