Combining Dynamic Simulation, High Dynamic Range Photography and Global Illumination

Christine Waggoner
Undergraduate Researcher

Son Chang
Undergraduate Researcher

Dr. Paul E. Debevec
Graduate Mentor

1998 Summer Undergraduate Program in Engineering Research at U.C. Berkeley

Abstract

This paper describes an approach to seamlessly integrate synthetic moving objects into a real scene. We create photorealistic physical simulations by combining work in dynamic simulation, high dynamic range photography and global illumination. We capture realistic physical interactions between synthetic objects by using a rigid body dynamic simulator. To visually integrate our synthetic moving objects into a real scene, we render frames of our physical simulation by using a new lighting technique and a differential rendering technique.

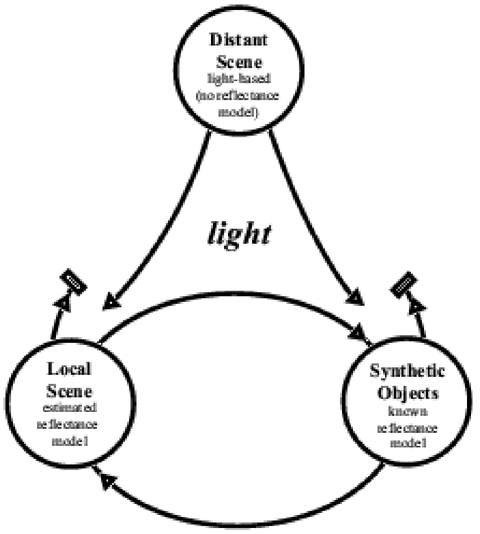

Figure 1: **The Lighting Technique** *This lighting method presented in [1] allows us to calculate correct lighting for synthetic objects to be integrated into a real scene. We divide the scene into three components: the distant scene, the local scene, and the synthetic objects. We render an image using a global illumination calculation of the interplay of light between the three components. However, light from the local scene and synthetic objects is ignored by the distant scene. Hence, there is no reflectance model for the distant scene. An estimated reflectance model is used to calculate the interplay of light between the local scene and the synthetic objects.*

[1] illustrates the lighting technique we use. This technique employs measured scene radiance captured through high dynamic range photography and global illumination. With this technique, we calculate correct lighting for synthetic objects to be introduced into a real scene. To calculate the proper illumination for the new objects, we regard the scene as three components: the distant scene, the local scene and the synthetic objects. The distant scene envelops and illuminates the local scene and the synthetic objects. The local scene is the area where the synthetic objects are placed. In addition, the local scene captures estimated reflectance and shadow data from the newly introduced synthetic objects. Neither the local scene nor the synthetic objects' reflectance information affects the distant scene. When the distant scene, local scene and synthetic objects are in place, we render the entire scene using a global illumination synthetic imaging and lighting system. All three components contribute to a global illumination solution , which may be composited into a photograph of a real scene using a differential rendering technique, also presented in [1]. In our case, the result of applying these techniques is a composite that we store as a single frame in an animation.

We propose extensions to the lighting and rendering techniques described in [1] for the creation of dynamic simulation animations. The lighting and rendering techniques in [1] largely rely on incident light from the distant scene to illuminate the local scene and the synthetic objects. This dependency causes the illumination calculation in the synthetic lighting and imaging system that we use to be computationally intensive. The lighting and imaging system we use divides the global illumination calculation into two parts: the calculation of the contributions of light from "indirect" and "direct" light sources. In the direct portion of the illumination calculation, the light sources are known, whereas the indirect portion of the illumination calculation accounts for light sources that are not known . We recommend identifying concentrated areas of light previously computed in the indirect calculation and adding these areas of light to the direct light source portion of the illumination solution.

We proffer a manual approach to identify concentrated light sources in the scene that are calculated in the indirect calculation and an approach to correctly quantify these light sources' color radiance values. The identification of these concentrated light sources allows us to leverage the powerful lighting and rendering techniques demonstrated in [1], but to also optimize rendering and reducing flickering in our final animation. By identifying concentrated areas of light formerly included in the indirect portion of the illumination calculation a priori, we reduce the number of rays required to be cast to identify these light sources and to accurately render the scene. This optimization is paramount to successfully render an entire animation in a reasonable amount of time. In addition, by identifying these light sources, the illumination calculation is more consistent and reduces flickering which is very important for creating a convincing and aesthetically pleasing animation. We also discuss possible user interfaces for software to further ease the process of creating animations using our methods. We conclude by discussing the significance of combining dynamic simulation, global illumination and high dynamic range photography and by suggesting avenues for future research in this area.

Introduction

Animation is traditionally an extremely long and tedious process. Even today when inverse kinematics, linear and nonlinear interpolation are nearly commonplace, the best animations are created by a patient hand. The success of an animation is determined by how well an animator can capture the nuances of an object's motion. Even for animations that are rooted in the physical world, animations continue to be produced by hand rather than completely simulated by the computer.

In the case of inverse kinematics, an animator specifies handles and constraints on the joints of a geometric model and manually sets the transformations of the model for each keyframe. Linear and nonlinear interpolation reduce the task of creating the frames inbetween the keyframes; however, again, the keyframes are set manually. Fortunately, as human beings, we are extremely sensitive to motion and can distinguish between false and realistic-looking motion. Our keen sense of motion permits animators to iteratively refine an object's movement until the object moves convincingly. However, even our intuition about how to simulate movement is not always physically accurate.

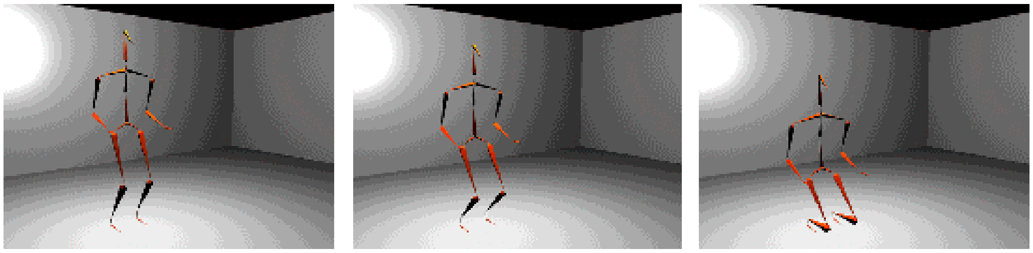

Figure 2: **Inverse Kinematics and Interpolation** *Inverse kinematics involves the creation of a skeleton composed of a hierarchy of joints and the manipulation of handles and constraints on the skeleton. The joint hierarchy propagates information regarding the movements of the skeleton from lower nodes to upper nodes of the hierarchy until the root node of the skeleton is reached. The joint hierarchy permits an animator to easily manipulate the skeleton. Above, we create a character and animate its legs by using inverse kinematics. We manipulate handles and constraints between the hip joints and the ankle joints to bend the character's legs. Using inverse kinematics, we posed the skeleton in two steps: creating the constraints and handles with which to manipulate the skeleton and moving the handles. Otherwise, moving the skeleton would have required more tedious transformations of the limbs of the character. In this animation, the first and third frames are keyframes. The second frame is a resultant frame from the interpolation between the first and third frames.*

While advances such as the widespread use of inverse kinematics and of linear and nonlinear interpolation over fully hand-drawn animations are significant, the dynamic simulation system presented in [3] offers yet another alternative to traditional animation techniques and a new paradigm for dynamic simulation. We use an impulse-based simulator which departs from traditional constraint-based simulation in that explicit constraints are no longer maintained at contact points. This departure is advantageous, adding simplicity and efficiency to the simulation of collisions, rolling, sliding, and resting, thereby easing the production of an animation, and more specifically, of our photorealistically rendered dynamic simulation.

Much like the task of creating an animation, the quest for visual realism in computer graphics has also been an arduous one, but an extremely important area of research. Rendering synthetic objects into real-world scenes contributes to an increasingly important application of computer graphics: creating realistic images of 3D scenes. Commonly available techniques for lighting and rendering a three-dimensional scene demand creative intuition and are labor-intensive. Frequently, scenes are lit with direct (known) light sources only. This tactic does not account for what we term indirect light: light transmitted from an object's own surfaces and from other objects' surfaces. The omission of indirect light when illuminating a scene fully departs from reality where indirect light is a significant factor in the illumination of all objects. The process of placing direct light sources is also flawed in that it is extremely tedious. An adjustment to a single light source necessitates a completely new rendering of the scene. This technique is also problematic in that the person lighting the scene often relies on crude estimations of the placement and coloration of the light. Precise techniques for capturing accurate lighting data exist. Lighting professionals survey environments by using special devices to conduct comprehensive real-world lighting analyses. However, although this technique is precise, it is also laborious and in most cases, requires very costly equipment. Another common technique is to photograph a reference object in the environment where the newly modeled object is to be rendered and then attempt to reproduce the lighting conditions and the surface qualities of the object from the photograph. The reflection mapping technique is useful for reflective surfaces. However, none of these methods is ideal for producing convincing and realistically illuminated images. These methods are insufficient for measuring the contributions of indirect illumination from the environment.

Numerous global illumination algorithms exist which successfully simulate the effects of light from both the direct and indirect portions of the illumination calculation. It is possible to accurately model an entire scene with its full geometric and reflectance (BRDF) characteristics, but often an ambitious task at best. The calculation of a global illumination solution for a large complex environment is extremely time-consuming and challenging. [1] presents a technique that reduces the need to calculate a full global and illumination solution. [1] underscores that it is not always necessary to recalculate the entire global illumination solution with the introduction of a new object to the scene. It is unlikely a new object's contribution to the illumination solution will significantly affect the distant areas of the scene. Fortunately, the integrated approach of using both global illumination and high dynamic range photography presented in [1] is a useful option that handles light from indirect portion of the illumination calculation well. We use this approach and build upon it to create a fully rendered dynamic simulation. This technique may also be applied to all different sorts of animations. It is not limited to being used for dynamic simulations only.

Recent work [2] introduced a technique for obtaining accurate scene radiance measurement from a series of photographs taken at different levels of exposure. This high dynamic range photographic technique measures a range of low levels of indirect radiance from surfaces and high levels of direct radiance from light sources. [1] demonstrates a method that uses this measured radiance to create light-based models. [1] defines the term light-based model to refer to a model of a scene constituted of radiance information. By using measured scene radiance and image-based modeling techniques, it is possible to create a light-based model by mapping scene radiance values to an environment in which synthetic objects can be placed

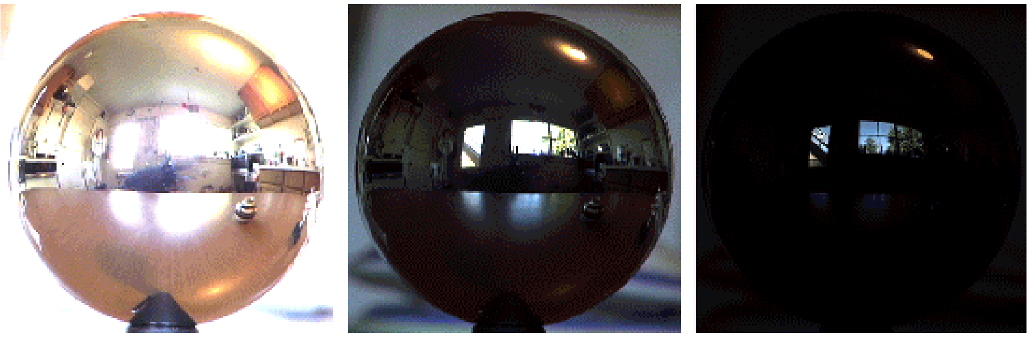

Figure 3: **Capturing Scene Radiance Values From a Probe** *[1] demonstrates a method for capturing the high dynamic range scene radiance values by photographing a mirrored ball on a pen cap positioned on a table top. The three images of the probe are taken at different levels of exposure to capture the full dynamic range of the scene. The scene radiance values obtained from the probe may then be used to create a light-based model of the scene.*

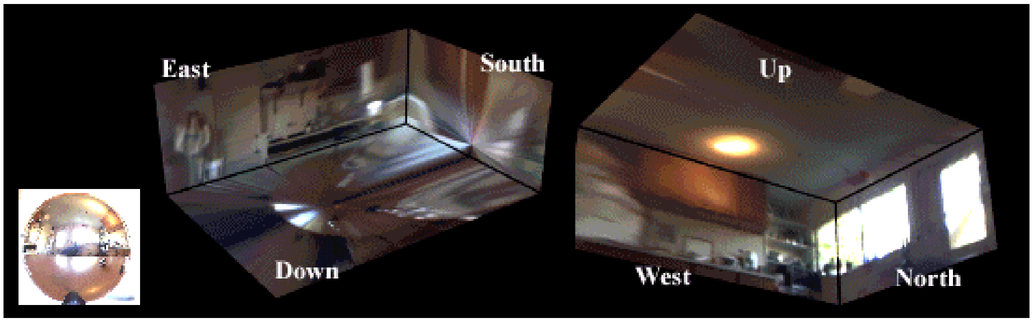

Figure 4: **The Distant Scene, a Light-Based Model** *The distant scene is created by mapping values from a light probe onto a rectangular box. the geometry of the box corresponds to the upper half of a kitchen. the face labeled "Down" coincides with the top of the kitchen table. We may place the local scene and synthetic objects within this model and they will be illuminated with the scene radiance values from the full dynamic range of the original scene.*

In the method presented in [1], the scene is partitioned into three components: the distant scene, the local scene, and the synthetic objects. The distant scene, the first component of the scene, is a light-based model. The distant scene envelops the local scene and synthetic objects. The second component of the scene is the local scene, the portion of the scene that will be affected by the introduction of objects into the scene. The third component of the scene is the new objects introduced into the local scene, the synthetic objects. Approximate geometry and surface properties are used to estimate the interplay of light between the local scene and the synthetic objects. Synthetic objects can be specified with arbitrary material properties. All three components of the scene are rendered together using a synthetic imaging and lighting system, RADIANCE[5] that employs standard global illumination techniques. However, neither the local scene's nor the synthetic objects' contributions to the distant scene's radiance values are considered in the final global illumination calculation. Since there is no exchange of light between the local scene and synthetic objects and the distant scene, we avoid a full global illumination calculation for the entire scene, but receive comparable results to a more complex calculation.

The creation of a dynamic simulation of synthetic objects integrated into a real scene is a potent marriage of dynamics and of new lighting and rendering techniques. Applications of this combination do not remain solely within the realm of entertainment and visual effects, but extend to the fields of engineering, virtual reality, and scientific visualization. Moreover, these two techniques combined yield a method to ameliorate the intensive process of creating a realistically rendered animation. This method reduces the task of keyframing and shortens rendering time.

To further reduce rendering time, we suggest an addition to the method presented in [1]. We suggest replacing concentrated areas of light in the distant scene with impostors1, polygons that mark concentrated areas of light. Impostors are helpful in that they are used to identify large concentrated areas of light in the distant scene a priori. By introducing impostor light sources, we insure that we account for concentrated light sources in the direct calculation of the global illumination solution rather than in the indirect calculation. By using impostors, we use a hybrid deterministic/stochastic calculation technique. We use the term deterministic to describe a technique that repeatedly achieves the same result and we use the term stochastic to describe a technique that employs randomization and generally produces inconsistent results. The basis for our deterministic/stochastic calculation is the equation:

$$L_r(\theta_r, \phi_r) = L_e + \int \int L_i(\theta_i, \phi_i) f_r(\theta_i, \phi_i, \theta_r, \phi_r) |\cos \theta_i| \sin \theta_i d\theta_i d\phi_i$$

In the function above, the value of a ray in a given direction, expressed in terms of luminance or radiance. The function sums together the emitted radiation and the convolution of all incoming radiance with the bidirectional reflectance-transmittance distribution function (BRTDF). In order to simplify the integral, we attempt to remove the two prominent components of the integral:

A. $L_i(\theta_r, \phi_r)$

B. $f_r(\theta_i, \phi_i, \theta_r, \phi_r)$

where A is the incoming luminance distribution and B is the BRTDF. In our hybrid stochastic/deterministic approach, we remove the two prominent components of the integral by computing the direct and indirect components of the calculation. The direct component is a calculation in which rays are traced to random locations on the light sources. During this component of the calculation, the rays will most likely intersect walls or other large obstructions. These rays determine whether light sources illuminating an object are occluded. The specular indirect component is a calculation where rays are distributed about the mirror and transmitted directions using Monte Carlo sampling. During the indirect components of the calculation, hemisphere rays are sent out which will hopefully reveal a bright region of indirect light. Most likely, the direct light sources will be discovered during the sampling process. Finding large indirect sources of illumination requires additional bounces and samples. Consequently, the calculation that correctly identifies large concentrated areas of indirect light is extremely expensive. In RADIANCE, to create an accurate rendering without using impostors, the number of hemisphere samples must be high and the ambient accuracy tolerance setting must be low. This increase in hemisphere samples and reduction in the tolerance setting result in a very prolonged rendering time. Finally, after removing these two components from the integral, the diffusely interreflected component is computed by calculating the simplified integral at dynamically selected locations.

We reduce rendering time and increase the quality of images by removing light with high variance from the indirect illumination calculation. To accurately and to efficiently find the sum of light illuminating objects, we treat concentrated areas of light formerly included in the indirect calculation as impostors. We use the term impostors because large concentrated sources of light formerly included in the indirect calculation masquerade as direct light sources included in the direct calculation. It may at first seem illogical to use light sources originally included indirect calculation of the global illumination solution because these "indirect" light sources are not self-luminous. However, in the real world, there is actually no distinction between "indirect" and "direct" light. "Direct light" and "indirect light" are terms used in computer graphics to categorize portions of a calculation. In relation to our everyday experiences with light, it is illogical that we should insist on making a distinction between concentrated areas of indirect and direct light. For example, we do not make a distinction between a lamp’s fixture and its light bulb. When we sample a light source, its sample should represent the overall light, not a single value corresponding to a value from its luminaire only. For our purposes, we should consider the difference between indirect and direct light as the way that each light source is sampled. Direct light sources and impostors are ear-marked to be sampled whereas other indirect light sources are not. By marking our concentrated areas of indirect light, our lighting and rendering algorithm now allows us to avoid unnecessary sampling and calculation. By using impostors, we sample intelligently and consistently. This reduction in calculation further distinguishes our technique from long global illumination solutions for complex environments. We skirt excessively long rendering times and the identification of light sources formerly included in the indirect calculation allows us to consistently sample high variance light sources. The consistent sampling of these light sources allows us to reduce flickering between frames in an animation that results from stochastic sampling in the indirect calculation.

Creating the Distant Scene

We begin the process of creating our animation by preparing the distant scene. We reiterate that the distant scene is a light-based model and that a light-based model is a representation of the scene that corresponds to the radiance information of the scene. In order to capture the radiance information, we use high dynamic range photographic techniques described in [2] through which we obtain accurate scene radiance measurements from a series of photographs taken at different levels of exposure. We obtain the full dynamic range scene radiance data by photographing a spherical probe in an environment at different levels of exposure. This probe acts as a omnidirectional radiance map. We map the scene radiance values from the probe to a geometric model of the scene. As opposed to texture-mapping, we light-map. Just as with texture-mapping, the geometry of the model onto which we map may vary. However, careful attention should be paid to insure accurate recordings of scene radiance and to capture the desired viewpoint of the scene from the probe. We complete our light-based model of the distant scene by mapping scene radiance values to the geometric model of the distant scene by specifying the glow material property in RADIANCE. The glow material property is emissive only and is not affected by the local scene or the synthetic objects.

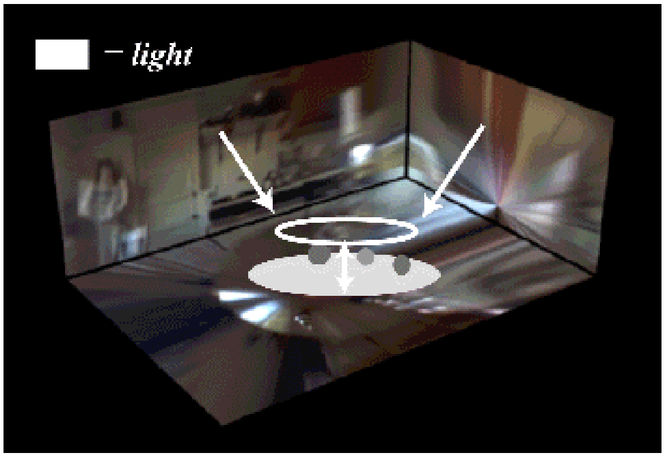

Figure 5: **A Three-Dimensional Representation of the Scene** *The two arrows represent light from the distant scene illuminating the local scene and the synthetic objects. The halo of light represents the interplay of light between the synthetic objects. The vertical arrowed line represents the interplay of light between the synthetic objects and the local scene. Refer to Figure 1 for further elaboration on the composition of the scene.*

Creating the Local Scene

We create the local scene by approximating the surface and material properties of the real scene. The local scene fully participates in the illumination solution and should reflect the real environment as closely as possible. The local scene is enveloped by the distant scene. The local scene is also the component of the scene onto which the synthetic objects will cast shadows and cast light. In some cases, the synthetic objects placed in the local scene can significantly affect it. Consequently, care should be taken when specifying the local scene. For example, if the synthetic objects are very large, the local scene should be comparably large to capture the shadows and reflections. If a shadow from a large synthetic object is not correctly accounted for, the shadow may be cut off abruptly or be cast flatly across components of the distant scene. Circumstances such as unusual shadows being cast or light being emitted by synthetic objects should be taken into careful consideration when deciding which areas will be part of the local scene and the distant scene and how synthetic objects will be placed in the local scene.

Creating a Dynamic Simulation

We use Impulse[4], a rigid body dynamic simulator. In order to create a simulation, polygonal models must be specified according to the Impulse file format2. Concave models and flat planes do not simulate correctly. Each object’s transformation, mass, the coefficient of friction, linear velocity, and angular velocity may be specified. It is also possible to override default values for gravity, a direction vector for gravity, and the coefficient of restitution. We create a simulation by placing objects into the scene, making appropriate settings, running the simulation, and gathering the transformations of each object in the simulation for a specific number of frames.

Integrating Synthetic Objects into the Scene

In order to integrate the synthetic objects into the scene, we use the transformation data we gathered from Impulse. We modified Impulse to output each object’s transformation data for every frame. We place the synthetic objects into the local scene according to the positions specified by Impulse for each frame. Particular care must be taken to place the objects according to the boundaries of the local scene. In addition, it may also be desirable to originally place the objects in an aesthetically pleasing manner according to the viewpoint from which the objects will be seen. The rendering of the scene will eventually be composited with a photograph of a real scene. The viewpoint of each frame must correspond to the viewpoint of the photograph. To animate the viewpoint of the simulation, a corresponding background photograph must be taken as the viewpoint changes for each frame. In addition, the background photograph establishes constraints on the placement of objects in the local scene. For example, the synthetic objects should not be placed on top of the objects pictured in the background plate. Care should also be taken in observing the movement of objects and their effects on the local scene from frame to frame in the animation. The objects should not move beyond the boundaries of the local scene or unnaturally interact with objects pictured in the background plate.

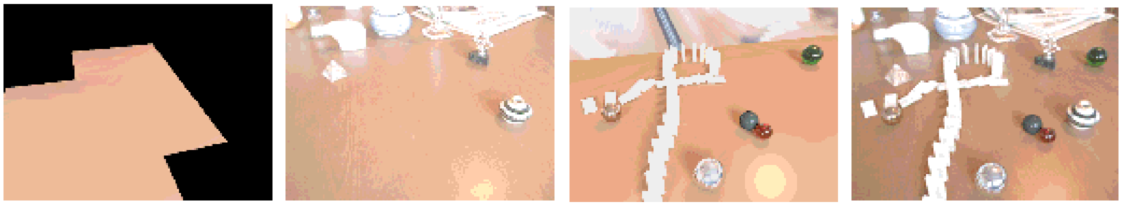

Figure 6: **Creating the Local Scene and Placing Synthetic Objects** *WE create a polygon, shown in the first image above, to set the boundaries in which we may safely place synthetic objects. This polygon helps the person creating the scene to avoid placing synthetic objects on top of objects in the real scene or in areas such as the bright spot on the table. To create this polygon, we project our polygon into the image of the real scene and adjust the polygon’s vertices accordingly. The real scene is the second image in this figure. After placing synthetic objects into the scene, we expand the area of the local scene rather than rendering the oddly shaped polygon as our local scene. By expanding the local scene, we almost completely conceal the edges of the local scene. The expanded local scene is shown in the third image above. The final image is of the synthetic objects. The scene appears naturally and the only trace of the local scene is a barely noticeable line beyond the dominos.*

A Manual Approach to Optimizing Rendering

We optimize the rendering time and reduce flickering of light between frames of our animation by using impostor light sources. We identify concentrated areas of indirect light by sampling areas of the scene radiance map and by intuition. For example, in Figure 4, the overhead light is located on the plane labeled "Up" and the windows are located on the plane labeled "North." We intuit that the windows and the overhead light in the distant scene represent large areas of concentrated light. By using the lighting techniques in [1], the light sources that might normally be considered direct light sources default to being indirect light sources. We convert these indirect light sources to direct light sources to justify the inconsistency of considering areas of light with high variance in the indirect calculation of the global illumination calculation.

Figure 7: **Placing an Impostor Light Source** *To determine the geometry of an impostor light source, we view the Northern wall of the room as a parallel projection at a low exposure. The blackened out area in the second image of this series shows the geometry of the impostor light source. The scene in the last two images is displayed at the exposure with which the scene will be rendered. The polygon placed in front of the concentrated area of light should cover it completely to avoid an incorrect illumination calculation. Note that the geometry of the impostor will not be reflected by other objects in the scene. Instead, the wall will appear in reflections as shown in the third image of the above series. The impostor simply serves to mark and area of light so additional sampling of the area is not needed.*

We create impostors by specifying polygons to cover the high variance areas of light. We also find the three-dimensional points in space that we will use to specify the geometry of each impostor light in the scene. We take a box filter of the area an impostor will cover in the distant scene to determine the impostor's average red, green, and blue color radiance values. When the we render the scene, RADIANCE will treat the impostor as a direct light source with our red, green, and blue radiance values. In addition, in our final renderings, RADIANCE will create the proper reflections of the areas we covered with our impostors. More specifically, we treat the impostor as a light source when computing its contribution to the illumination of some point, but otherwise treat the area of the impostor's polygon according to its material properties. For example, we may specify an impostor for a window on the interior of a room. By using an impostor, in the illumination calculation, we treat the window as a light source, but for the calculation of reflections, the glass of the window and the environment outside the window remain visible.

Shortcomings and Drawbacks

The method we present for creating a high quality rendered dynamic simulation is a somewhat complicated and lengthy process. Although, the method we present is not haphazard, there remains room for improvement. More specifically, we would like to fully automate the process of identifying concentrated areas of light in the distant scene. In addition, the implementation of impostor light sources in RADIANCE does not completely fulfill our needs. When impostors cast light that forms a polygon of light on a surface, the light sources used to account for the area of light of the impostor's geometry are easily distinguishable. For example, when an impostor is used to mark a window, when the light from the window is cast onto a floor plane, the point lights used to account for the window's geometry cast easily distinguishable beams of light. Consequently, the light from the window appears fragmented in an unnatural manner. While impostors reduce flickering between frames in an animation and reduce rendering time, one must use impostors prudently, insuring that the light from the impostor is cast in a natural manner.

In our work, we used a dynamic simulator to create the animation that we rendered. We do not wish to overlook the merits of not using a simulator to create work that is more artistic. The work we present is an effort to display the progress in the quest for visual realism in computer graphics. By using simulation and global illumination techniques, we use the most physically accurate modes for the creating a fully rendered animation that are readily available to us. We hope that the techniques we present will be adopted and built upon to create a range of animations varying in style.

Conclusion

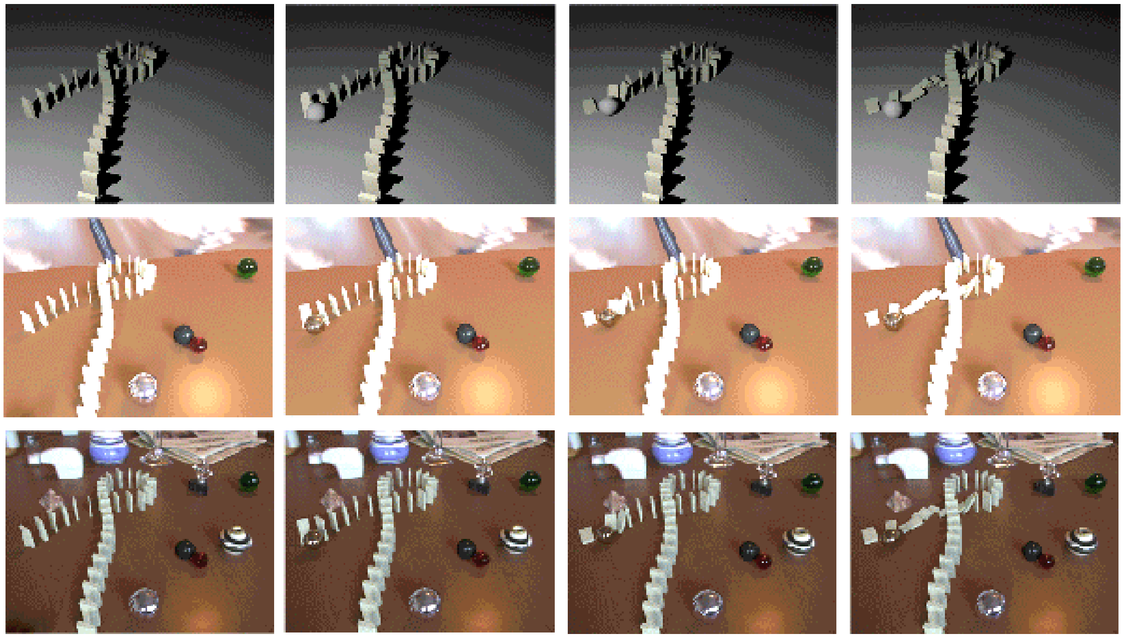

Figure 8: **The Final Product** *After simulating the movement of the synthetic objects, we use the simulator's calculations to render the dynamic simulation. The distant scene provides the scene radiance map. We identify large concentrated sources of indirect light in the distant scene and cover these areas of light with impostors. We render the scene using RADIANCE. We then composite each rendered frame with the background photograph using a differential rendering technique. In the figure above, the first row of images are frames from the simulation. Textures for the final render are not yet assigned and only a point light source illuminates the scene. The second row of images are of renders which have not yet been composited with the background image. Textures and lighting to be used in the final animation are present. The third row of images are of fully rendered frames. These frames have been composited with the background plate using differential rendering. We have also applied a Gaussian blur to the images to further blend the synthetic objects into the real scene.*

We have described an approach to seamlessly integrate synthetic moving objects into a real scene. This work is yet another step in easing the production of realistic-looking animations. We proposed a method to decrease rendering time by identifying concentrated areas of indirect light. In addition, we illuminated the weaknesses of our approach and made suggestions for creating a software aid for the production of a rendered dynamic simulation. In conclusion, we suggest continued research in the area of dynamics and research in an effort to facilitate creating animations of synthetic objects that may be composited into real scenes from multiple viewpoints.

Acknowledgements

The author thanks Son Chang, Soren Spies, Greg Ward Larson, and Brown University. A special thanks goes to Paul Debevec for being a patient mentor and for introducing me to his work in computer graphics. This research was conducted as a project for the Summer Undergraduate Program in Engineering Research at U.C. Berkeley. The author wishes to thank the coordinators of the program and the other participants of the program for their friendship.

References

[1] DEBEVEC, P. E. Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography. In SIGGRAPH ’98 (July 1998).

[2] DEBEVEC, P. E., AND MALIK, J. Recovering High Dynamic Range Radiance Maps From Photographs. In SIGGRAPH ’97 (August 1997), pp. 369-178.

[3] MIRTICH, B. Impulse-based Dynamic Simulation of Rigid Body Systems. Ph.D. thesis. University of California, Berkeley, December 1996.

[4] MIRTICH, B. A rigid body dynamic simulator. Ph.D. thesis software written in C. University of California, Berkeley, December 1996.

[5] WARD, G. J. The radiance lighting and rendering system. In SIGGRAPH ’94 (July 1994), pp. 459-472.

[6] WARD, G. J. Rendering with Radiance, The Art and Science of Lighting Visualization. Morgan Kaufmann Publishers, San Francisco, CA, 1998, ch. 10-13.

[back]

>>HomE<< | m%E | >res&Ume*!<-- | PRoJ-ects