Shadow Fields for Deformable Objects

Jiaping Wang
v-jpwang@msrchina.research.microsoft.com

Kun Zhou
kunzhou@microsoft.com

Stephen Lin
stevelin@microsoft.com

Baining Guo
bainguo@microsoft.com

Heung-Yeung Shum
hshum@microsoft.com

July 2005
Technical Report
MSR-TR-2005-96

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
http://www.research.microsoft.com

1 Introduction

The realism of a computer-generated scene can be considerably heightened by including soft shadows of area light sources and environment maps. These shadows provide the viewer with a sense of 3D and facilitate inference of the spatial arrangement of the scene. Although much research in this area has produced numerous techniques for computing soft shadows, generating them efficiently for realistic scenes remains a significant challenge.

Real-world scenes often contain complex characteristics that preclude rapid shadow rendering by previous techniques. For a scene that contains many objects and light sources with complicated geometry, a substantial number of shadow rays need to be traced to obtain acceptable accuracy. Many soft shadow techniques reduce this computational burden by assuming small or point-like light sources, but such illumination conditions are not typical of real scenes which often are illuminated by more general local sources as well as distant global lighting that is typically modelled by an environment map. Furthermore, scenes often contain significant motion, of not only objects but light sources as well, which compounds the processing load due to frame by frame updates.

To address the problem of rapid shadow rendering of dynamic scenes under the combined illumination of moving local light sources and environment maps, the shadow fields technique [1] precomputes the shadowing effects of light sources and occluding objects *independently* at sampled points in a surrounding empty space, and then efficiently combines this information at run-time according to the configuration of the scene. With this approach, real-time shadow generation can be achieved for both rigid transformations of scene entities and articulated motions of objects that consist of rigid segments.

There however exists a broad range of objects whose movements are characterized by significantly more complicated deformations than articulations. Examples include living organisms such as people and animals whose natural shape changes cannot be convincingly modelled as articulated motions. For such objects, we propose in this work a technique for rapid run-time determination of their shadow fields as they progress through general deformations.

A shadow field is a 5D function of the angular direction (2D), radial distance (1D), and visibility (2D) of sample points in the surrounding space of a scene entity. This 5D function is too costly to sample at run-time, but can be precomputed for rigid objects as in [1]. For deformable objects whose shape variations cannot practically be enumerated for precomputation, our proposed method takes advantage of similarities among visibility functions sampled at points along the radial direction. At increasing radial distances, visibility functions expressed in terms of cube maps maintain an approximately similar form while decreasing in size. Based on this general property, our technique samples at run-time a 4D shadow field at a single radius defined by the entity's bounding sphere, and then extrapolates this 4D function to obtain 5D shadow field data using precomputed matrices that transform radiance distributions represented in terms of spherical harmonics to different radial distances.

This solution to soft shadowing of deformable objects generalizes shadow fields to arbitrary motions of light sources and occluders. With this added property, low-frequency shadows from general dynamic objects become possible to generate in real-time. In the remainder of this paper, we begin with an overview of related work and a brief review of shadow fields. Our proposed technique and implementation are then presented, followed by results and conclusions.

2 Related Work

Soft shadowing techniques have been presented at various levels of scene generality. Traditional approaches primarily address shadowing of light sources described by points or small areas. These typically are based on combining multiple shadow maps (e.g., [2,3]) or extending shadow volumes (e.g., [4]). To gain speed at the expense of some accuracy, some related methods compute “fake”, geometrically incorrect soft shadows (e.g., [5,6]). These techniques have become increasingly efficient in recent years, but have not reached the levels of performance needed for real-time generation of shadows in complex, dynamic scenes that include environment map illumination. Since shadow maps and volumes are determined with respect to a specific arrangement of lights and occluders, precomputation is infeasible for reducing run-time processing of dynamic scenes due to their numerous possible configurations.

Self-shadows that arise from global illumination have been generated in previous methods on precomputed radiance transfer (PRT), which precomputes how incoming illumination is shadowed and inter-reflected by an object to produce the incident lighting distribution over its surface. This approach was introduced in [7] for low-frequency lighting, and extensions have been presented in [8] for spatially varying reflectance models and in [9] for greater speed and anisotropic BRDFs. PRT methods for all-frequency shadowing were proposed in [10,11], which utilizes wavelets instead of spherical harmonics for approximating the environment map and lighting effects. These PRT techniques address the transfer of distant directional illumination from environment maps to an object or a fixed surrounding scene, but do not effectively handle dynamic local light sources from which illumination can arrive at an object from multiple directions. In this case, dense and expensive illumination sampling over the object must be done frame-by-frame at run time to utilize PRT information. Annen et al. [12] approximate the incident lighting over an object by interpolating far fewer samples using gradients of spherical harmonic coefficients, but the frequent, time-varying fluctuations of these gradients in dynamic scenes may necessitate dense sampling nonetheless.

More recently, attention has been focused on how to employ PRT in dynamic scenes. Sloan et al. [7] present a neighborhood-transfer technique that records the soft shadows and reflections cast from an object onto surrounding points in a possibly non-static environment. However, the problem of how to combine neighborhood-transfers from multiple objects was not addressed. For scenes with moving objects, Mei et al. [13] rendered shadows using precomputed visibility information for each object [14] with respect to uniformly sampled directions of distant illumination. By assuming parallel lighting from distant illuminants, they show that some degree of shadow map precomputation becomes manageable for environment maps. Precomputation of shadow maps nevertheless remains infeasible for dynamic local light sources.

To address non-rigid motion, PRT methods have been presented for deformable objects. One approach to deformable objects is to compute and interpolate PRT transfer vectors for several key frames of a given animation, and render the pre-animated model under environment maps, as done by James and Fatahaltian [15] in real time. Precomputation of a constrained sequence, however, does not generalize well to dynamic scenes with numerous degrees of freedom. Another approach is to compute self-visibility on the fly, as presented by Kautz et al. [16], but this approach would be expensive for a complex scene containing many objects. Sloan et al. [17] present the use of zonal harmonics for representing spatially variant lighting effects over an object, which allows for efficient rotation of local coordinate frames over a deforming object. While this approach is effective for generating self-shadows, it does not address the shadowing effects of multiple objects upon one another. As with PRT methods for dynamic scenes, PRT techniques for deformable models are intended for use with distant illumination from environment maps, and are not straightforward to modify for efficient handling of local light sources.

3 Shadow Fields

Traditionally, shadows are computed based on the relative positions of a light source and an occluder. Because of this dependence on scene configuration, precomputation becomes difficult due to the large number of possible object arrangements in a dynamic scene. To enable some amount of precomputation in soft shadow generation for dynamic scenes, the shadow fields technique [1] decouples lighting and visibility by modeling the shadowing effects of illuminants and occluders individually. This decoupling allows precomputation that is independent of arrangement. At run time, these precomputed shadowing effects are then efficiently combined according the scene arrangement to give fast performance.

In this section, we briefly review the shadow fields technique, first describing the precomputation of shadow fields and then giving an overview of the run-time rendering process for combining the precomputed shadow fields.

3.1 Shadow Field Precomputation

For a light source, its shadow field is called a source radiance field (SRF), and is precomputed by sampling the incoming radiance distribution at points in an empty space surrounding the illuminant. These points are sampled at uniform intervals angularly and radially. Points at a given radial distance from the center of the light source form a sphere, which is sampled angularly in a uniform distribution. At each of these sample points, the incoming radiance distribution is recorded as a cube map of incoming radiance values from different directions. For shadow fields of occluding objects, called object occlusion fields (OOFs), cube maps are similarly sampled but instead record visibility information as alpha values. For intermediate locations within the sampled points of a shadow field, cube maps are trilinearly interpolated from the cube maps of the eight nearest sample points. For efficient processing, these cube maps are represented in terms of fourth-order spherical harmonics.

3.2 Shadow Field Rendering

To determine the soft shadow at an arbitrary scene point, the shadowing effects of each light and object in the scene as represented by their respective shadow fields are combined according to their relative positions and orientations in the scene. At a given receiver point in the scene, the incoming illumination distribution or visibility function from a given light source or object is given by its cube map corresponding to the distance and direction of the point from the scene entity.

Combining the cube maps of shadow fields involves four types of operations. The first is alignment of shadow fields to the global coordinate frame. Since objects and light sources may have various orientations in a dynamic scene, their shadow fields need to be rotated by a simple coordinate transformation that requires little processing for shadow fields represented in terms of spherical harmonics.

How shadow fields are combined depends upon their relative distances from the receiver point. So the second operation is to sort the centers of the shadow fields from near to far.

For objects that lie closer to the receiver point than a light source, their OOFs are multiplied to determine their aggregate occlusion of the source. Multiplying this product by the SRF of the light source yields the source radiance distribution that that it contributes to the receiver point for the given scene configuration. To determine the soft shadow or reflected radiance at the receiver point, the BRDF of the receiver point is multiplied to this product as well.

After calculating the reflected radiance for each light source in the preceding multiplication step, the final operation adds these components together to obtain the overall soft shadow that is rendered in the image.

4 Object Deformations

For arbitrarily deforming objects, shadow fields cannot practically be precomputed so they must be determined at run time. Examination of many actual shadow field functions $f(\theta, \phi, r) \rightarrow V(\theta_v, \phi_v)$ reveals significant correlation among visibility functions V that differ in radial distance r. Intuitively, as one views a shadow field entity from increasing distances, the form of the visibility function or radiance distribution appears approximately the same, but at decreasing scale, as illustrated in the top row of Fig. 1. Because of this relationship,

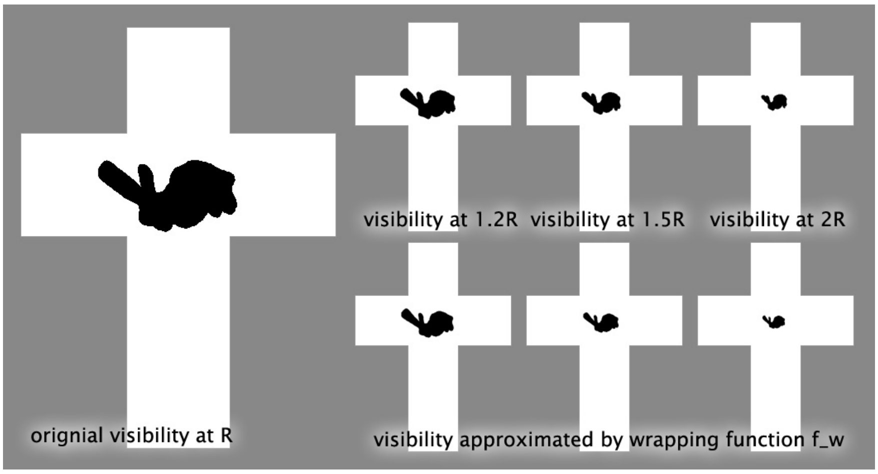


Figure 1: Approximate visibility by shrinking of visibility functions.

it is sufficient for low-frequency shadow generation to compute shadow field cube maps at only a single radius at run time, and then extrapolate cube maps at different radii based on this property, as exemplified in the bottom row of Fig. 1. For cube maps represented in terms of spherical harmonics, this extrapolation can be rapidly performed at run time using precomputed extrapolation matrices that transform cube maps at one radius to other radial distances. With this technique for efficient determination of shadow fields at run time, the shadow field rendering technique of [1] can be employed for arbitrary deformations of objects.

4.1 Visibility Shrinking Approximation

In our method, visibility functions V_r are sampled at points v_r on the radius r of the entity's bounding sphere. For visibility functions with a coordinate frame whose z-axis points towards the center of the bounding sphere, greater radial distances generally result in shrinking in the θ direction while the ϕ parameter is unchanged, as shown in Fig. 1. We express this angular rescaling of the visibility function V_d with respect to θ as

$$V_d(\theta, \phi) = V_r(f_{r/d}(\theta), \phi)$$

where $f_{r/d}$ denotes the rescaling function for angle θ from distance r to distance d.

In principle, the geometry of the object must be known in order to obtain a precise rescaling function. Since calculations based on detailed geometry of the deforming object would prevent real-time processing, we employ an approximation that yields visually similar results for low-frequency lighting effects. As illustrated in Fig. 2, our approach is to assume that object points on the silhouette as seen from the bounding sphere (in blue) are also the silhouette points as

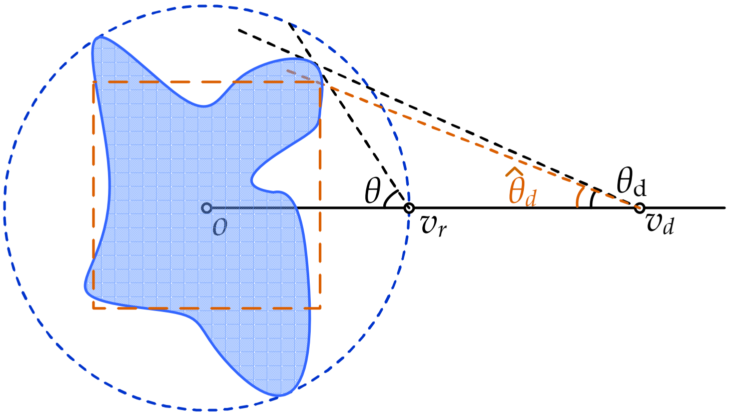


Figure 2: Approximation of visibility shrinking. v_r is a sampled point on the bounding sphere, whose visibility is extrapolated to a point v_d in the same (θ, ϕ) direction but a different radial distance. The silhouette angle $\widehat{\theta}_d$ is our approximation to the actual silhouette angle θ_d.

viewed from other radial distances. To determine the 3D location of a silhouette point seen at a certain angle, we determine the square that, as in Fig. 2, has the following properties: shares its center with the bounding sphere, has a side that is perpendicular to sample direction (θ, ϕ), and is the smallest square that intersects the vector defined by the viewed silhouette direction. The intersection point is taken as the 3D position of the silhouette point. In cases where these conditions cannot be satisfied together, the silhouette point is taken as the intersection of the silhouette vector and the bounding sphere.

From this approximation, the rescaling function from radius r to another radius d is given by

$$\widehat{\theta}_d = f_{r/d}(\theta) = \left\{ \begin{array}{cc} \theta' & \theta' \geq 0 \\ \theta' + \pi & \theta' < 0 \end{array} \right.$$

where

$$\theta' = Tan^{-1}\left(\tfrac{kTan\theta}{1+Tan\theta-kTan\theta}\right), \quad k = r/d.$$

We note that this angular rescaling is dependent on the ratio $k = r/d$, not on the absolute values of d and r, so we denote the rescaling function by f_k henceforth.

The error that results from this approximation is also exemplified in Fig. 2. The silhouette point at direction (θ, ϕ) as seen from a sample point v_r on the bounding sphere gives a silhouette direction $(\widehat{\theta}_d, \phi)$ from point v_d of radial distance d. This differs from the actual silhouette direction (θ_d, ϕ) from v_d. For many object shapes, a small difference in silhouette direction is not visually discernible for low-frequency shadowing effects. There however exist instances where this approximation error may become more evident. One is when the

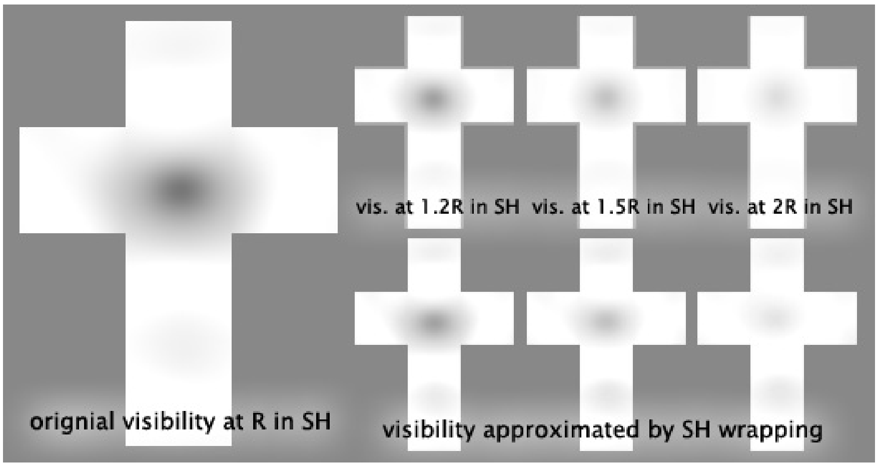

vis. at 1.2R in SH
vis. at 1.5R in SH
vis. at 2R in SH
orignial visibility at R in SH
visibility approximated by SH wrapping

Figure 3: Visibility shrinking approximation in terms of spherical harmonics.

sampled point on the bounding sphere lies on the surface of the entity. This gives a visibility function that is half occluded regardless of the object shape. This problem though is partly mitigated by the visiblity functions of nearby sample points that do not lie on the object surface.

An illustration of the visibility shrinking approximation is shown for the Stanford bunny in Fig. 1, where the approximated visibility functions in bottom row closely resemble the actual visibility functions given in the top row. While some slight discrepancies exist in the approximated visibility function, the approximation is nevertheless adequate for low-frequency shadowing as displayed in Fig. 3.

For SRFs of light sources, visibility functions also express radiant energy in terms of unit solid angle instead of binary visibility. To account for radiant intensity with respect to the solid angle factor, we multiply the radiance by the first derivative of the angular rescaling function f_k, as derived in Appendix A:

$$V_d(\theta,\phi) = V_r(f_k(\theta),\phi)\cdot f_k'(\theta)\cdot\frac{\sin(f_k(\theta))}{\sin\theta}$$

where

$$\begin{aligned} & f_k'(\theta)\cdot\frac{\sin(f_k(\theta))}{\sin\theta} \\ = & \frac{k(Tan^2\theta+1)\cdot\sin(f_k(\theta))/\sin\theta}{(2k^2-2k+1)Tan^2\theta-2(k-1)Tan\theta+1}. \end{aligned}$$

4.2 Precomputed Extrapolation Matrices

Shadow fields are compressed by spherical harmonics for low-frequency lighting effects. To enhance run-time performance, we present a method for extrapo-

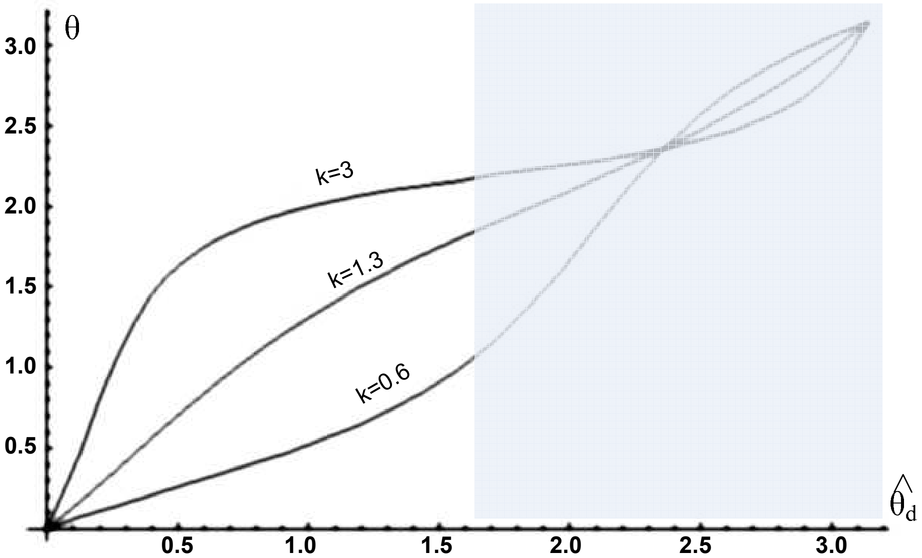


Figure 4: Angular rescaling functions for different radial distance ratios k. Values for $\theta > \pi/2$ are not meaningful since visibility is sampled over a hemisphere, so they consequently rescale zero values in the visibility function.

lating sampled visibility functions to other radial distances using precomputed transformation matrices for spherical harmonics coefficients.

For radial distance d, the spherical harmonic coefficients S_d with basis functions Ψ can be expressed as a linear transformation of the coefficients S_r at radial distance r on in the same direction (θ, ϕ):

$$\begin{aligned} S_d(i) &= \iint_\Omega V_d(\theta,\phi)\Psi_i(\theta,\phi)d\theta d\phi \\ &= \iint_\Omega V_r(f_k(\theta),\phi)\Psi_i(\theta,\phi)d\theta d\phi \\ &= \iint_\Omega \left(\sum_{j\in I} \Psi_j(f_k(\theta),\phi)\cdot S_r(j) \right) \Psi_i(\theta,\phi)d\theta d\phi \\ &= \sum_j \left(S_r(j)\cdot \iint_\Omega \Psi_j(f_k(\theta),\phi)\cdot\Psi_i(\theta,\phi)d\theta d\phi \right). \end{aligned}$$

Our method precomputes transformation matrices for a set of radial distance ratios k as given by the sampling intervals in [1]. With these precomputed matrices, extrapolation of sampled cube maps to other radial distances is computed as:

$$S_d = \mathcal{M} \cdot S_r^T$$

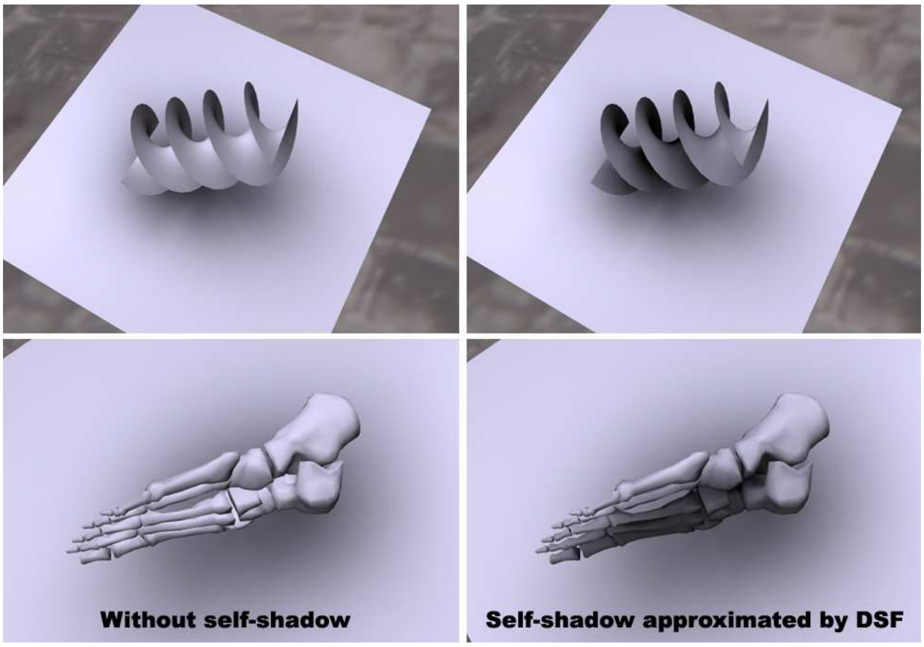


Figure 5: Self-shadow approximation.

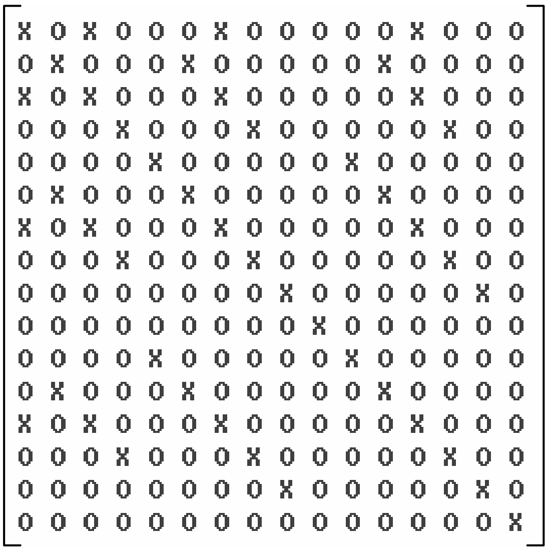
where

$$\mathcal{M}_{i,j} = \iint_{\Omega} \Psi_j(f_k(\theta), \phi) \cdot \Psi_i(\theta, \phi) d\theta d\phi.$$

For non-sampled radial distance ratios k, $\mathcal{M}$ is linearly interpolated from neighboring values.

This extrapolation is used not only for radial distances beyond the bounding sphere, but also for radii within the bounding sphere. Such ratios $k < 1$ expand the original visibility function instead of shrinking it, as shown in the plot of angular rescaling functions in Fig. 4. Moreover, it is used for computing self-shadows on object surfaces. This is in contrast to [1] which precomputes visibility functions directly on sampled points on the surface. This approach circumvents precomputation of self-shadow OOFs, which would not be feasible for arbitrarily deformable objects. Fig. 5 demonstrates self-shadows generated in this manner. More precise self-shadowing that includes interreflection effects may alternatively be computed by PRT techniques for deformable objects as in [17].

For a q-order spherical harmonics representation, the number of elements in $\mathcal{M}$ is q^4. An interesting property of $\mathcal{M}$ is that only $(q+1)(2q^2+4q+3)/3$ of the elements are non-zero and the positions of the non-zero elements are independent of k, as proven in Appendix B. Fig. 6 shows the form of transformation matrices for fourth-order spherical harmonics, in which only 44 elements of the 256 total entries are non-zero. Therefore, calculation and storage of only these non-zero



$$\begin{bmatrix}
X & O & X & O & O & O & X & O & O & O & O & O & X & O & O & O \\
O & X & O & O & O & X & O & O & O & O & O & X & O & O & O & O \\
X & O & X & O & O & O & X & O & O & O & O & O & X & O & O & O \\
O & O & O & X & O & O & O & X & O & O & O & O & O & X & O & O \\
O & O & O & O & X & O & O & O & O & O & X & O & O & O & O & O \\
O & X & O & O & O & X & O & O & O & O & O & X & O & O & O & O \\
X & O & X & O & O & O & X & O & O & O & O & O & X & O & O & O \\
O & O & O & X & O & O & O & X & O & O & O & O & O & X & O & O \\
O & O & O & O & O & O & O & O & X & O & O & O & O & O & X & O \\
O & O & O & O & O & O & O & O & O & X & O & O & O & O & O & O \\
O & O & O & O & X & O & O & O & O & O & X & O & O & O & O & O \\
O & X & O & O & O & X & O & O & O & O & O & X & O & O & O & O \\
X & O & X & O & O & O & X & O & O & O & O & O & X & O & O & O \\
O & O & O & X & O & O & O & X & O & O & O & O & O & X & O & O \\
O & O & O & O & O & O & O & O & X & O & O & O & O & O & X & O \\
O & O & O & O & O & O & O & O & O & O & O & O & O & O & O & X
\end{bmatrix}$$

Figure 6: Pattern of zero elements in fourth-order spherical harmonics transformation matrices. X represents the positions of non-zero elements for all k.

elements is needed.

4.3 Run-Time Processing

Shadow field rendering is performed at run time in three steps. On the GPU, visibility functions are computed at points on the bounding sphere of scene entities that have deformed in the current frame. These visibility functions are then projected into the spherical harmonics domain. We refer to this sampled data as *base sphere visibility* (BSV). In parallel on the CPU, soft shadows are computed at visible surface points in the scene using the BSV and the precomputed transformation matrices $\mathcal{M}$. With the computed vertex colors, object triangles are sent to the graphics hardware for interactive rendering.

BSV Updates In the GPU, the BSV of a scene entity is updated if the shape of the entity has changed. The hardware implementation consists of two hardware passes.

In the first pass, visibility functions are sampled at 12x20 ($\theta \times \phi$) points on the bounding sphere, according to the sampling rate used in [1]. At each of the sample points, visibility sampling with respect to a rasterized object or light source is performed at 32x32 uniformly distributed directions on a hemisphere oriented towards the center of the entity. Sampling is not needed on the hemisphere facing away from entity, since the visibility is known to be always 1 for OOFs and) for SRFs. The sampled data is then organized and stored in

Table 1: Rendering Performance

Model	**No. of Vertices**	**BSV Sampling Time**	**Sph. Har. Conversion Time**	**Shadowing Time** (all vertices)	**FPS**
Dragon	1151	11.0 ms	5.0 ms	37.7 ms	21.3 Hz
Egg	482	7.4 ms	5.0 ms	36.7 ms	22.5 Hz
Plane	8382	72.4 ms	5.0 ms	47.9 ms	13.5 Hz

a (32x12)x(32x20) texture composed of 32x32 sub-textures that each represent visibility at a point.

In the second pass, the visibility functions are converted into fourth-order spherical harmonic coefficients. This conversion involves downsampling the 32x32 sub-textures into 16x16 clips and then taking their inner products with the 16 fourth-order spherical harmonics, which are also encoded in 16x16 clips. This step is processed in the pixel shader, and yields 16 12x20 floating-point textures. This BSV is stored as a 16x240 lookup table for use in soft shadow evaluation.

Soft Shadow Evaluation In the CPU, the BSVs from the set of objects and light sources in the scene are used to determine shadow field information from which soft shadows are computed at screen vertices.

For each screen vertex, the visibility function with respect to each scene entity is computed from the corresponding BSVs. Using the precomputed transformation matrices $\mathcal{M}$, visibility functions are extrapolated from each BSV. For a point at location (d, θ, ϕ) from the center of the bounding sphere, the visibility function is evaluated as:

$$S = \mathcal{M}_k \cdot S_r(\theta, \phi).$$

Since $\mathcal{M}_k$ and $S_r(\theta, \phi)$ are both discretely sampled, trilinear interpolation is employed to obtain $\mathcal{M}$ for arbitrary k from the eight closest sample points. For points that lie beyond the maximum sampled radial distance, the shadowing effect of the entity is disregarded, i.e., $\mathcal{M} = \{0, 0, 0...0\}$. For points inside the minimum radius, $\mathcal{M} = \{1, 0, 0...0\}$ is used instead.

To determine soft shadow values, the computed cube maps for the scene entities are combined along with the BRDF of the receiver point according to the algorithm given in [1].

Rendering Object triangles with vertex colors from the soft shadow evaluation are finally sent to the graphics hardware for interactive rendering in a traditional rendering pipeline. We utilize the vertex object buffer to maximize the throughput of the graphics hardware.

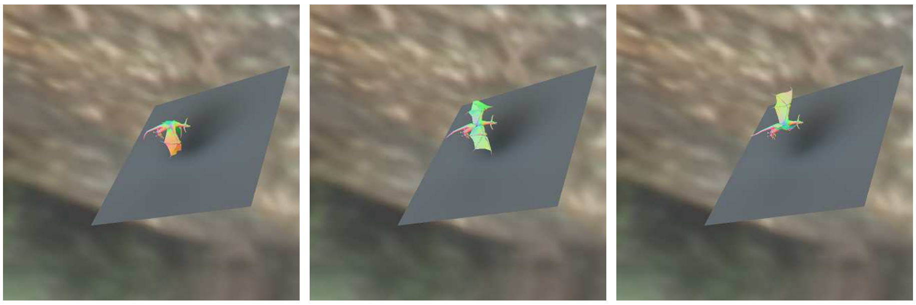


Figure 7: Frames of a flying dragon model.

5 Results

With the proposed technique for shadow field rendering of deformable objects, low-frequency soft shadows from objects with arbitrary motion are generated in real time. Experiments were performed on a 2.8-GHz Pentium IV PC with 1GB RAM and an nVidia FX6800GT graphics card. For the examples shown in this paper, we used a BSV sampling resolution of 12x20 in θ, ϕ, and a visibility mask resolution of 32x32 over a hemisphere. Shadows are computed on a planar area with 160x160 vertices beneath the objects.

We display rendered frames for sequences of a flying dragon in Fig. 7, a transforming egg in Fig. 8, and a bending plane in Fig. 9. For complete sequences of these scenes, we refer the reader to the accompanying video file. The shadow field evaluation speed is approximately 1.41 μs per vertex, and the overall performance data for these deforming models is listed in Table 1. To isolate the performance of our deformation algorithm, these examples are of scenes that contain only the model and environment map illumination. Increasing scene complexity to match the examples given in [1] would increase processing time by approximately the amount reported for those scenes in [1].

In the example figures, the soft shadows can be seen to change according to the deformations of the objects. By sampling points only on bounding spheres and extrapolating efficiently to other radial distances, our method produces shadow field data approximately 20 times faster than full shadow field computation. Processing speed is particularly elevated by the parallel use of the GPU and CPU in our implementation. For a small number of simultaneously deforming objects in a scene, real-time performance can be maintained because the computational bottleneck lies in soft shadow evaluation, not the visibility sampling and spherical harmonic conversion stage.

6 Conclusion

With rapid run-time computation of shadow field information, it becomes feasible to efficiently generate low-frequency shadows for deformable objects in

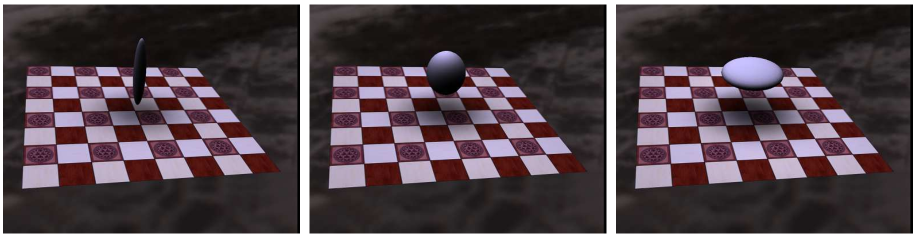


Figure 8: Frames of a transforming egg model.

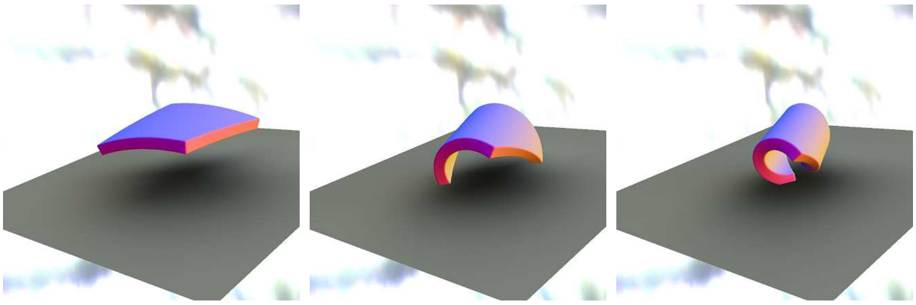


Figure 9: Frames of a bending plane model.

dynamic scenes. This generalization of shadow fields to arbitrary object motions enables soft shadow rendering for animate scenes containing objects such as people and animals, which are central elements in many applications.

A possible direction for future work is to examine different sampling schemes for the purpose of reducing error in extrapolated visibility functions. As mentioned previously, sampling points on the bounding sphere can lead to greater errors when the point lies on the object surface. This error may potentially be decreased by sampling at different radial distances for different (θ, ϕ) directions according to certain geometric criteria. We also plan to investigate how this error may be reduced by partitioning a deformable object into a small number of sub-volumes that are handled separately from one another.

A Angular Rescaling Function

The radiant intensity factor in the angular rescaling function for SRFs can be derived from the following energy preservation relationship, where E_d is the energy at radial distance d, and E_r is the energy at distance r.

$$
\begin{aligned}
E_d \ &= \iint\limits_{4\pi} V_d(\theta,\phi)\sin\theta d\theta d\phi \\
&= \iint\limits_{4\pi} \left[V_r(f_k(\theta),\phi)\cdot f_k'(\theta)\cdot \frac{\sin(f_k(\theta))}{\sin\theta} \right] \sin\theta d\theta d\phi \\
&= \iint\limits_{4\pi} V_r(f_k(\theta),\phi)\cdot f_k'(\theta)\cdot \sin(f_k(\theta)) d\theta d\phi \\
&= \iint\limits_{4\pi} V_r(f_k(\theta),\phi)\cdot d(f_k(\theta))\cdot \sin(f_k(\theta)) d\theta d\phi \\
&= \iint\limits_{4\pi} V_r(\theta',\phi)\cdot d(\theta')\cdot \sin(\theta') d\theta' d\phi \\
&= E_r
\end{aligned}
$$

B Transformation Matrix Structure

Real-valued spherical harmonics are defined as

$$
Y_l^m(\theta,\phi) = \begin{cases} \sqrt{2}K_l^m \cos(m\phi)P_l^m(cos\theta), & m > 0 \\ \sqrt{2}K_l^m \sin(-m\phi)P_l^{-m}(cos\theta), & m < 0 \\ K_l^0 P_l^0(cos\theta), & m = 0 \end{cases}
$$

where P denotes the associated Legendre polynomials and K is a normalization factor:

$$
K_l^m = \sqrt{\frac{(2l+1)}{4\pi}\frac{(l-|m|)!}{(l+|m|)!}}.
$$

Let

$$
V_r(\theta,\phi) = \sum a_l^m Y_l^m(\theta,\phi),
$$

$$
V_d(\theta,\phi) = \sum b_l^m Y_l^m(\theta,\phi).
$$

For OOFs where

$$
V_r(\theta,\phi) = V_d(f_k(\theta),\phi),
$$

f_k is a one-to-one continuous mapping, so there exists an inverse f_k^{-1} such that

$$
V_d(\theta,\phi) = V_r(f_k^{-1}(\theta),\phi),
$$

i.e.,

$$
\sum b_l^m Y_l^m(\theta,\phi) = \sum a_l^m Y_l^m(f_k^{-1}(\theta),\phi).
$$

Based on the equality

$$\begin{aligned}&\iint Y_l^m(f_k^{-1}(\theta),\phi)Y_{l'}^{m'}(\theta,\phi)\sin\theta d\theta d\phi\\ &=\delta_{mm'}\cdot\iint Y_l^m(f_k^{-1}(\theta),\phi)Y_{l'}^{m'}(\theta,\phi)\sin\theta d\theta d\phi,\end{aligned}$$

the relation between a_l^m and b_l^m can be expressed as

$$\begin{aligned}b_l^m &= \iint V_d(\theta,\phi)Y_l^m(\theta,\phi)\sin\theta d\theta d\phi\\ &= \iint\left[\sum_{l',m'} a_{l'}^{m'}Y_{l'}^{m'}(f_k^{-1}(\theta),\phi)\right]Y_l^m(\theta,\phi)\sin\theta d\theta d\phi\\ &= \sum_{l',m'}\left[a_{l'}^{m'}\iint Y_{l'}^{m'}(f_k^{-1}(\theta),\phi)Y_l^m(\theta,\phi)\sin\theta d\theta d\phi\right]\\ &= \sum_{l'\geq m} a_{l'}^m\iint Y_{l'}^m(f_k^{-1}(\theta),\phi)Y_l^m(\theta,\phi)\sin\theta d\theta d\phi\\ &= \sum_{l'\geq m} a_{l'}^m K_l^m K_{l'}^m\int_0^\pi P_{l'}^m[\cos(f_k^{-1}(\theta))]P_l^m(\cos\theta)\sin\theta d\theta.\end{aligned}$$

From this relationship between the spherical harmonics coefficients of V_d and V_r, it can be seen that the non-zero entries in transformation matrices $\mathcal{M}$ lie in fixed positions that are independent of k, and are relatively sparse. For SRFs with a radiant intensity factor, a similar relationship can be proven in similar manner.

References

[1] K. Zhou, Y. Hu, S. Lin, B. Guo, and H.-Y. Shum, "Precomputed shadow fields for dynamic scenes," in *SIGGRAPH '05*, 2005.

[2] P. Heckbert and M. Herf, "Simulating soft shadows with graphics hardware," Carnegie Mellon University, Tech. Rep. CMU-CS-97-104, 1997.

[3] M. Agrawala, R. Ramamoorthi, A. Heirich, and L. Moll, "Efficient image-based methods for rendering soft shadows," in *SIGGRAPH '00*, 2000, pp. 375–384.

[4] U. Assarsson and T. Akenine-Moller, "A geometry-based soft shadow volume algorithm using graphics hardware," in *SIGGRAPH '03*, 2003, pp. 511–520.

[5] E. Chan and F. Durand, "Rendering fake soft shadows with smoothies," in *Euro. Symp. on Rendering*, 2003, pp. 208–218.

[6] C. Wyman and C. Hansen, “Penumbra maps: Approximate soft shadows in real-time,” in *Euro. Symp. on Rendering*, 2003, pp. 202–207.

[7] P. Sloan, J. Kautz, and J. Snyder, “Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments,” in *SIGGRAPH ’02*, 2002, pp. 527–536.

[8] J. Kautz, P. Sloan, and J. Snyder, “Fast, arbitrary brdf shading for low-frequency lighting using spherical harmonics,” in *Euro. Workshop on Rendering*, 2002, pp. 291–296.

[9] J. Lehtinen and J. Kautz, “Matrix radiance tranfer,” in *Symp. on Interactive 3D Graphics*, 2003, pp. 59–64.

[10] R. Ng, R. Ramamoorthi, and P. Hanrahan, “All-frequency shadows using non-linear wavelet lighting approximation,” in *SIGGRAPH ’03*, 2003, pp. 376–381.

[11] ——, “Triple product wavelet integrals for all-frequency relighting,” in *SIGGRAPH ’04*, 2004, pp. 477–487.

[12] T. Annen, J. Kautz, F. Durand, and H.-P. Seidel, “Spherical harmonic gradients for mid-range illumination,” in *Euro. Symp. on Rendering*, 2004, pp. 331–336.

[13] C. Mei, J. Shi, and F. Wu, “Rendering with spherical radiance transport maps,” in *Eurographics ’04*, 2004, pp. 281–290.

[14] M. Ouhyoung, Y.-Y. Chuang, and R.-H. Liang, “Reusable Radiosity Objects,” *Computer Graphics Forum*, vol. 15, no. 3, pp. 347–356, 1996.

[15] D. James and K. Fatahaltian, “Precomputing interactive dynamic deformable scenes,” in *SIGGRAPH ’03*, 2003, pp. 879–887.

[16] J. Kautz, J. Lehtinen, and T. Aila, “Hemispherical rasterization for self-shadowing of dynamic objects,” in *Euro. Symp. on Rendering*, 2004, pp. 179–184.

[17] P.-P. Sloan, B. Luna, and J. Snyder, “Local, deformable precomputed radiance transfer,” in *SIGGRAPH ’05*, 2005.