

Jiaping Wang · Kun Xu · Kun Zhou · Stephen Lin · Shimin Hu ·  
Baining Guo

# Spherical Harmonics Scaling

**Abstract** In this paper, we present a new SH operation, called spherical harmonics scaling, to shrink or expand a spherical function in frequency domain. We show that this problem can be elegantly formulated as a linear transformation of SH projections, which is efficient to compute and easy to implement on a GPU. Spherical harmonics scaling is particularly useful for extrapolating visibility and radiance functions at a sample point to points closer to or farther from an occluder or light source. With SH scaling, we present applications to low-frequency shadowing for general deformable object, and to efficient approximation of spherical irradiance functions within a mid-range illumination environment.

**Keywords** Signal Processing · Spherical Harmonics · Illumination · Rendering

## 1 Introduction

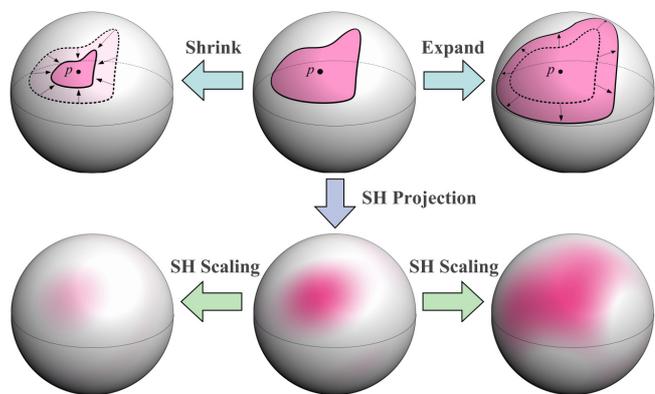
Spherical harmonics (SHs) offer an efficient representation for a band-limited spherical signal. This representation has been widely used in computer graphics for modeling illumination distributions [3], visibility functions [5], and BRDFs [7]. Because of the properties of a spherical harmonics basis, there are several operations defined on spherical harmonics projections that can be efficiently computed and that makes spherical harmonics

Jiaping Wang<sup>†</sup>  
Institute of Computing Technology, Chinese Academy of Science. Graduated School of Chinese Academy of Science  
E-mail: e\_boris2002@hotmail.com

Kun Xu<sup>†</sup> · Shimin Hu  
Tsinghua University  
E-mail: xu-k@mails.tsinghua.edu.cn,shimin@tsinghua.edu.cn

Kun Zhou · Stephen Lin · Baining Guo  
Microsoft Research Asia  
E-mail: {kunzhou,stevelin,bainguo}@microsoft.com

<sup>†</sup>This work was done while Jiaping Wang and Kun Xu were interns at Microsoft Research Asia.



**Fig. 1** Scaling operation. Top row: Scale a spherical functions around  $p$ . Bottom row: The corresponding spherical harmonics scaling.

practical for computer graphics, especially for real-time rendering of global illumination effects.

### 1.1 Related Work

For global illumination, the integral of the rendering equation is computationally expensive and impractical for real-time rendering. An efficient solution to this problem is to represent irradiance and surface reflectance as spherical harmonics projections, and then solve their spherical convolution by a *SH dot product* of their SH coefficient vectors [3]. This approach was utilized in work on precomputed radiance transfer (PRT) [5], which convolves the SH vectors of a BRDF kernel and transferred radiance that accounts for visibility and interreflections.

The basic formulation of PRT was presented for static scenes. To extend the functionality of PRT, operations on spherical harmonics have been proposed to facilitate some change in lighting conditions or object configurations. In [5], spherical harmonic rotations are described for run-time rotation of environment maps without re-projection. *SH rotation* was also employed in the shadow

fields technique [8] for rotation of local light sources and occluding objects. For PRT with a locally deformable object, Sloan et al. [6] presented zonal harmonics (ZH), and demonstrated that a ZH basis is more efficient for rotation and can be implemented in current GPUs.

In [2], a PRT method is introduced for fast evaluation of radiance as a *SH triple product* of lighting, reflectance, and visibility. The triple product presents an operation on spherical harmonic projections that is equivalent to an element-wise multiplication of two spherical functions. Unlike a dot product, the triple product results in a new spherical harmonic projection instead of a scalar.

The triple product is somewhat cumbersome to compute, and would be expensive to employ when a large number of spherical functions are involved. Recently, a new operation on spherical harmonics called *spherical harmonics exponentiation* [4] was proposed for efficient evaluation of visibility in a scene with deformable objects. Instead of computing an expensive SH product per blocker as in previous work, they perform inexpensive vector sums to accumulate the *log* of blocker visibility. SH exponentiation then yields the product visibility vector over all blockers. With this approach, the visibility functions associated with numerous occluders can be efficiently aggregated.

## 1.2 Overview

In this paper, we propose a new SH operation called *spherical harmonics scaling* to shrink or expand a spherical harmonics projected function around a given point on the sphere. The result of this procedure is illustrated in Fig. 1 for the input spherical function shown in the top-center, which is scaled with respect to the point  $p$ . Scaling this spherical function smaller should contract its region boundary towards  $p$ , while a larger scaling should extend its boundaries away from  $p$ . Here we address the problem of how to compute a spherical scaling on a spherical harmonics projection, as illustrated in the bottom row of Fig. 1. We have found that spherical harmonics scaling can be formulated as a linear transformation of SH projections, which is cheap to compute and can be easily implemented on a GPU.

Spherical harmonics scaling brings a new functionality to PRT by extrapolating visibility and radiance functions from a sample point to other points located farther from or closer to a light source or occluding object. We present two real-time applications that take advantage of this property. One is a low-frequency shadowing technique for general object deformation, which constructs approximate shadow fields on the fly by sampling visibility at only a single radial distance from the object and then rapidly scaling these functions to other radii. The second application utilizes spherical harmonics scaling to efficiently approximate spherical irradiance functions for mid-range illumination, such as from an environment map defined on a finite sphere.

## 2 Spherical Harmonics Scaling

In polar coordinates, shrinking and expanding of a spherical function  $F_r(\theta, \phi)$  around the  $\theta = 0$  axis into a new function  $F_d(\theta, \phi)$  can be expressed as

$$F_d(\theta, \phi) = F_r(\tau(\theta), \phi) \quad (1)$$

where  $\tau(\cdot)$  is a monotonically increasing *angular scaling function* that rescales the polar angle  $\theta$  independently of azimuth angle  $\phi$ . For a spherical function  $F_r$  that represents visibility from a given point, moving closer to an occluder results in a scaling of the visibility function in which  $\tau(\theta) < \theta$ . Likewise, moving farther from the occluder corresponds to an angular scaling function with  $\tau(\theta) > \theta$ .

For some spherical functions, such those representing radiance, the change in solid angle due to scaling must be considered for energy preservation. Accounting for solid angles, Eq. (1) can be rewritten as

$$\begin{aligned} F_d(\theta, \phi) \sin \theta d\theta d\phi &= F_r(\tau(\theta), \phi) \sin \tau(\theta) d\tau(\theta) d\phi \\ &= F_r(\tau(\theta), \phi) \sin \tau(\theta) \tau'(\theta) d\theta d\phi. \end{aligned} \quad (2)$$

Combining Eq. (1) and Eq. (2), we define spherical scaling as

$$F_d(\theta, \phi) \sin \theta d\theta d\phi = F_r(\tau(\theta), \phi) \eta(\theta) d\theta d\phi \quad (3)$$

where  $\eta(\theta) = \sin \theta$  without energy preservation, and  $\eta(\theta) = \sin \tau(\theta) \tau'(\theta)$  with energy preservation.

### 2.1 Scaling with SH Representation

When a spherical function is represented by spherical harmonics coefficients, spherical scaling can also be done directly in the spherical harmonics domain. We refer to this operation as *Spherical Harmonics Scaling*. When both  $F_d(\theta, \phi)$  and  $F_r(\theta, \phi)$  are represented by vectors  $S_d$  and  $S_r$  of spherical harmonics coefficients,  $S_d$  can be expressed as a linear transformation of  $S_r$  with respect to SH basis functions  $\Psi$ :

$$\begin{aligned} S_d(i) &= \iint_{\Omega} F_d(\theta, \phi) \Psi_i(\theta, \phi) \eta(\theta) d\theta d\phi \\ &= \iint_{\Omega} F_r(\tau(\theta), \phi) \Psi_i(\theta, \phi) \eta(\theta) d\theta d\phi \\ &= \iint_{\Omega} \left( \sum_{j \in I} \Psi_j(\tau(\theta), \phi) S_r(j) \right) \Psi_i(\theta, \phi) \eta(\theta) d\theta d\phi \\ &= \sum_j \left( S_r(j) \iint_{\Omega} \Psi_j(\tau(\theta), \phi) \Psi_i(\theta, \phi) \eta(\theta) d\theta d\phi \right) \\ &= \sum_j (S_r(j) \cdot \mathcal{M}_{i,j}) \end{aligned} \quad (4)$$

For a given angular scaling function  $\tau(\cdot)$ , spherical harmonics scaling can be reduced to a matrix-vector product:

$$S_d = \mathcal{M} \cdot S_r^T \quad (5)$$

where  $\mathcal{M}_{i,j} = \iint_{\Omega} \Psi_j(\tau(\theta), \phi) \cdot \Psi_i(\theta, \phi) \eta(\theta) d\theta d\phi$  is called the *SH scaling transformation matrix*.

## 2.2 SH Scaling Transformation Matrix

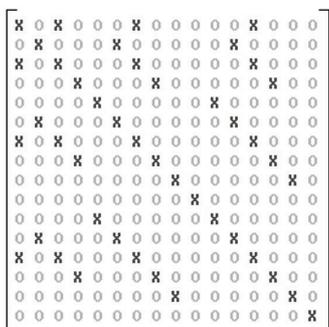
From Eq. (4), the SH scaling transformation matrix  $\mathcal{M}$  is defined as

$$\mathcal{M}_{i,j} = \iint_{\Omega} \Psi_j(\tau(\theta), \phi) \cdot \Psi_i(\theta, \phi) \eta(\theta) d\theta d\phi \quad (6)$$

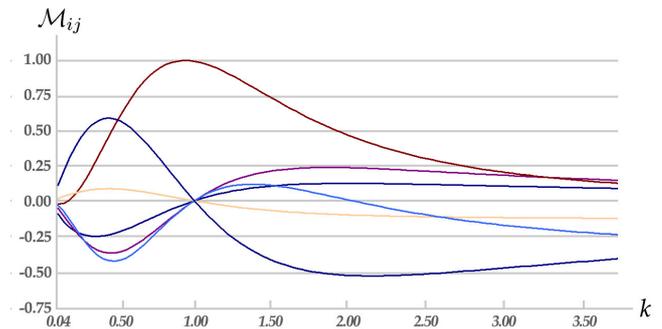
An interesting property of this transformation matrix is that less than one-fifth of its elements are non-zero, and that the matrix locations of these non-zero elements are independent of the angular scaling function  $\tau(\cdot)$ , as proven in the Appendix. Regardless of whether energy is being preserved, exactly  $N(2N^2 + 1)/3$  of the  $N^4$  elements in  $\mathcal{M}$  are non-zero for an  $N^{\text{th}}$ -order spherical harmonics representation. For example, with fourth-order spherical harmonics there are only 44 non-zero elements that are located in the matrix as shown in Fig. 2. Since the locations of non-zero elements in the transformation matrix are independent of angular scaling function  $\tau(\cdot)$ , computational savings can be gained by reducing the matrix-vector product of Eq. (5) to a set of multiplications and sums for only the non-zero elements.

## 2.3 Angular Scaling Function

Analogous to the backward warping function used in image processing, the angular scaling function  $\tau(\cdot)$  represents a backward mapping of polar angles that is defined according to the application. Typically, a series of angular scaling functions  $\tau_k(\cdot)$  parameterized by  $k$  is provided to perform scaling for different instances. Correspondingly, there are different scaling transformation matrices



**Fig. 2** Locations of non-zero elements in fourth-order SH scaling transformation matrices. X represents the non-zero elements.



**Fig. 3** Plot of six randomly selected elements in  $\mathcal{M}$  with respect to  $k$  for scaling visibility functions.

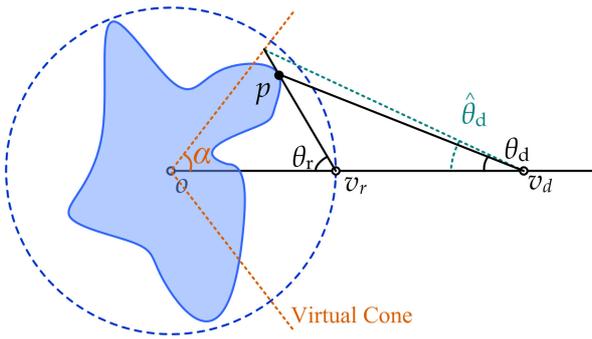
$\mathcal{M}_k$  with respect to different  $k$ . In most cases,  $\mathcal{M}_k$  cannot be expressed analytically, and even if an analytical form exists, such as for  $\tau_k(\theta) = k\theta$ ,  $\mathcal{M}_k$  is often too expensive to calculate. We choose to numerically compute  $\mathcal{M}_k$  for sampled values of  $k$ , and then linearly interpolate for intermediate values. As shown in Fig. 3, elements in different scaling transformation matrices change smoothly with respect to  $k$ , and we have empirically found it sufficient to sample 40 values of  $k$  in an exponential distribution.

## 3 Applications of SH Scaling

### 3.1 Shadow Fields for Deformable Objects

Traditionally, soft shadows are directly computed from the relative positions of a light source and an occluder. Because of this dependence on scene configuration, pre-computation 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 [8] 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 pre-computed shadowing effects are then efficiently combined according to scene arrangement to give fast performance.

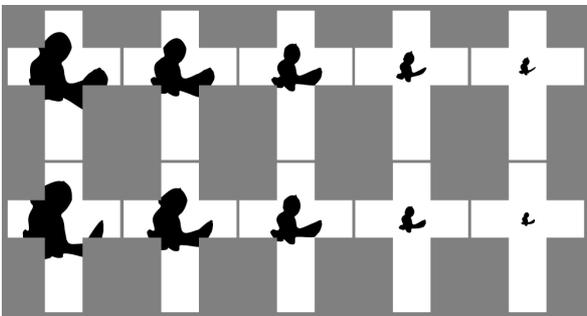
For a light source, its shadow field is called a source radiance field (SRF), and is precomputed by sampling the incoming radiance distribution at points in an empty space surrounding the illuminant. These points are sampled at uniform intervals angularly and radially. Points at a given radial distance 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.



**Fig. 4** Cone geometric approximation.  $v_r$  is a sampled point on the bounding sphere, whose visibility is extrapolated to a point  $v_d$  in the same angular direction but at a different radial distance. The angle  $\hat{\theta}_d$  is our approximation to the actual angle  $\theta_d$  of the object silhouette.

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. These cube maps are represented in terms of fourth-order spherical harmonics to facilitate processing.

For a deformable object, precomputation of shadow fields becomes infeasible due to the large number of possible object configurations. To address this problem, we take advantage of the observation that as one views a shadow field entity from increasing (decreasing) distances, the form of the visibility function or radiance distribution appears approximately the same, but at decreasing (increasing) scale, as illustrated in the top row of Fig. 5. Based on this property, we propose to sample shadow fields at a single radial distance and then rapidly extrapolate samples at other radii by SH scaling. In principle, the geometry of the object must be known in order to obtain a precise angular scaling function  $\tau$ . Since this information is generally unknown, we utilize a *cone geometric approximation* that the object surface forms a cone that faces the sample point as shown in Fig. 4. The angle  $\alpha$  of this virtual cone from the radial direction is



**Fig. 5** Approximate visibility by scaling of visibility functions. Top row: actual visibility at different radii. Bottom row: corresponding visibility computed by cone geometry approximation from visibility acquired from the bounding sphere (radius  $r$ ). From left to right, the radius is  $0.5r$ ,  $0.7r$ ,  $0.9r$ ,  $1.3r$ , and  $2r$ .

set to  $\pi/4$  in our implementation, but may be adjusted by the user to obtain a better approximation for a given object. With this geometric approximation, the angular scaling function  $\tau_k$  from  $\theta_d$  to  $\theta_r$  is given by

$$\theta_r = \tau_k(\theta_d) = \begin{cases} \theta' & \theta' \geq 0 \\ \theta' + \pi & \theta' < 0 \end{cases}$$

where

$$\theta' = \arctan\left(\frac{k \tan \theta}{1 + \tan \theta - k \tan \theta}\right), \quad (k = d/r)$$

The result of this geometric approximation is exhibited for the bird model in Fig. 5, where the approximated visibility functions in the 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.

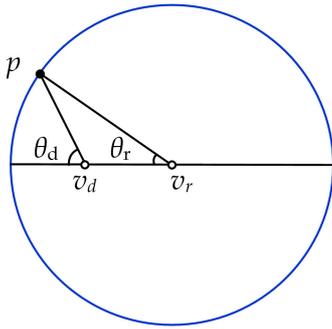
For OOFs of occluding objects, we utilize the SH scaling function of Eq. (1). For SRFs of light sources, the spherical functions also express radiant energy in terms of unit solid angle instead of binary visibility. To account for radiant intensity with respect to solid angle, we use the energy preserving SH scaling function in Eq. (2).

Our rendering algorithm is performed at run time for each frame in two stages. In the first stage, visibility/radiance functions are computed on the fly at  $6^*8^*8$  sample points on the bounding sphere of scene entities. At each sample point, we rasterize a rectangle of  $32^*32$  pixels towards the center of the object or light source, then use the *glReadPixels* function in OpenGL to read back the visibility or radiance functions. Next, the visibility/radiance functions are converted into fourth-order spherical harmonic coefficients  $S_r$ . In the second stage, for each scene vertex, the SH visibility function with respect to each object and the SH radiance function with respect to each light source  $S_d$  are scaled from the corresponding  $S_r$  by computing  $\mathcal{M} \cdot S_r^T$ , where  $\mathcal{M}$  is the corresponding SH scaling transformation matrix. Soft shadow values are then computed according to the algorithm given in [8].

Since visibility functions of deformable objects are sampled and extrapolated on-the-fly by SH scaling, shadow fields of deformable objects/lights are computed at real-time rates. Moreover, self-shadowing on object surfaces can be also approximated within the bounding sphere using ratios of  $k < 1$ , which expand the original visibility function instead of shrinking it.

### 3.2 Mid-Range Environment Lighting

For efficient rendering of global illumination effects with complex distant lighting, several methods based on pre-computed radiance transfer have been proposed (e.g., [5, 2]). These PRT techniques address the transfer of distant directional illumination from environment maps to

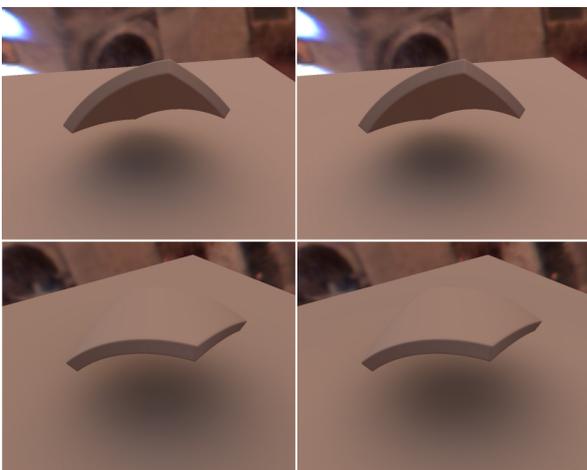


**Fig. 6** SH scaling for mid-range environment lighting.

an object or a fixed surrounding scene, but do not effectively handle lighting from non-distant sources, for which the distribution of incoming source illumination can vary from point to point in a scene. To efficiently address this issue, Annen et al. [1] approximate the incident lighting over an object by sampling the illumination distribution at only a few points (about 8 samples per object) and then interpolating them for other points using gradients of spherical harmonic coefficients.

With spherical harmonic scaling, we present a substantially more efficient technique for the case of environment map illumination that is not distant but originates from a sphere of radius  $R$ . By capturing the single irradiance function  $F_r(\theta, \phi)$  at the center  $v_r$  of the sphere, the irradiance function at any other locations within the sphere can be scaled from  $F_r(\theta, \phi)$ . As shown in Fig. 6,  $\theta_r$  of  $v_r$  corresponds to  $\theta_d$  of  $v_d$ . By denoting the vector from  $v_r$  to  $v_d$  as a signed quantity  $d$ , the angular scaling function  $\tau_k$  from  $\theta_d$  to  $\theta_r$  is given by

$$\begin{aligned} \theta_r &= \tau_k(\theta_d) \\ &= 2 \arctan \left( \frac{\sqrt{\cot^2 \theta_d + 1 - k^2} - \cot \theta_d}{1 + k} \right) \quad (7) \\ k &= d/R \in (-1, 1) \end{aligned}$$



**Fig. 7** Comparison between a fully sampled shadow field (left column) and extrapolation by SH scaling (right column).

**Table 1** Performance data of deformable shadow fields.

Scene	Vert. Model/Plane	FPS
Flying Dragon	1.1k/10k	35.4
Egg + Deform Light	0.5k/10k	14.2
Bending Sponge	8.4k/10k	13.4
Teapot Plane	10k/10k	26.8

Our rendering algorithm involves two steps at each vertex. First, we scale the radiance function  $F_r(\theta, \phi)$  at the sphere's center to the vertex's position to obtain  $F_d(\theta, \phi)$ . Since the vertex can be in arbitrary direction relative to the sphere's center,  $F_r$  is first rotated to that direction and scaled then rotated back. Second, we combine the incoming radiance  $F_d(\theta, \phi)$  represented in a SH basis with the precomputed transport vector/matrix at the vertex to calculate the soft shadow value as described in [5].

## 4 Experimental Results

We have implemented the two applications described in Sec. 3 on a 2.8-GHz Pentium IV PC with 1GB RAM and an nVidia FX6800GT graphics card. A fourth-order spherical harmonics representation is used in all the presented experiments. The overall performance data is listed in Table 1.

*Deformable shadow fields* We display rendered frames for sequences of a flying dragon in Fig. 8, a transforming egg in Fig. 9, and a bending sponge block in Fig. 10. For complete sequences of these scenes, we refer the reader to the supplementary video. The scene in Fig. 8 contains a deformable local light source that consists of a planar red emitter and a green emitter that are connected and have an adjustable angle. In all of these examples, the visibility functions approximated by spherical harmonics scaling generate soft shadows consistent with the deformations of the animated object. Fig. 7 compares the result of our approximation with a fully sampled shadow field as used in [8].

*Mid-Range environment lighting* We show in Fig. 11 some rendered images of a teapot on a plane with mid-range environment lighting. In this example, we only sample the irradiance distribution at the center of the scene; the irradiance distribution at each vertex is then extrapolated using SH scaling. The upper-left corners of each image illustrate the lighting distribution, which consists of a red and a green source, and also shows the position and orientation of the teapot within the environment lighting sphere. Notice that the shadow and shading are changing when the teapot plane is moving in the scene.

## 5 Conclusion

We proposed a new operation on a spherical harmonics basis to efficiently scale spherical functions represented in SH. Spherical harmonics scaling is amenable to GPU implementation, and can be used to extrapolate visibility and radiance functions at a sample point to points closer to or farther from an occluder or light source.

In future work, we plan to examine methods for reducing extrapolation errors in visibility and radiance functions. One source of this error occurs when a sampled point on the bounding sphere lies on the surface of an object. This results in a visibility function that is half occluded regardless of object shape, while direct extrapolation of such a visibility function leads to circular silhouettes. Although this problem is partly mitigated by the visibility functions of nearby sample points that do not lie on the object surface, we plan to examine different sampling schemes to reduce this error. For example, this error may potentially be decreased by sampling at different radial distances for different  $(\theta, \phi)$  directions according to certain geometric criteria.

Extrapolation error may also arise from the geometric approximation described in Sec. 3.1. To address this issue, we plan to investigate the possible solution of partitioning the object or light source into a small number of sub-volumes that are handled separately from one another.

**Acknowledgements** We would like to thank Xu Yang for helpful discussion on spherical harmonics and the anonymous reviewers for their helpful suggestions and comments. Kun Xu and Shimin Hu were supported in part by Natural Science Foundation of China (No.60225016, No.60321002), and the National Basic Research Project of China (No.2002CB312101).

## References

1. Annen, T., Kautz, J., Durand, F., Seidel, H.P.: Spherical harmonic gradients for mid-range illumination. In: Euro. Symp. on Rendering, pp. 331–336 (2004)
2. Ng, R., Ramamoorthi, R., Hanrahan, P.: Triple product wavelet integrals for all-frequency relighting. In: SIGGRAPH '04, pp. 477–487 (2004)
3. Ramamoorthi, R., Hanrahan, P.: An efficient representation for irradiance environment maps. In: SIGGRAPH '01, pp. 497–500 (2001)
4. Ren, Z., Wang, R., Snyder, J., Zhou, K., Liu, X., Sun, B., Sloan, P.P., Bao, H., Peng, Q., Guo, B.: Real-time soft shadows in dynamic scenes using spherical harmonic exponentiation. In: Proc. of the ACM SIGGRAPH 2006 (2006)
5. Sloan, P., Kautz, J., Snyder, J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In: SIGGRAPH '02, pp. 527–536 (2002)
6. Sloan, P.P., Luna, B., Snyder, J.: Local, deformable pre-computed radiance transfer. ACM Trans. Graph. **24**(3), 1216–1224 (2005)
7. Westin, S.H., Arvo, J., Torrance, K.E.: Predicting reflectance functions from complex surfaces. In: SIGGRAPH 1992, pp. 255–264 (1992)

8. Zhou, K., Hu, Y., Lin, S., Guo, B., Shum, H.Y.: Pre-computed shadow fields for dynamic scenes. ACM Trans. Graph. **24**(3), 1148–1155 (2005)

## Appendix: Structure of SH Scaling Transformation Matrix

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:

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

Further, Each spherical harmonics can be decomposed as  $Y_l^m(\theta, \phi) = S_l^m(\phi)T_l^m(\theta)$ , where

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

$$T_l^m(\theta) = \begin{cases} P_l^m(\cos\theta), & m > 0 \\ P_l^{-m}(\cos\theta), & m < 0 \\ P_l^0(\cos\theta), & m = 0 \end{cases}$$

Denote  $\Psi_j(\theta, \phi)$  as  $Y_l^m(\theta, \phi)$  and  $\Psi_i(\theta, \phi)$  as  $Y_{l'}^{m'}(\theta, \phi)$  in Equation 6, Equation 6 can be rewritten as

$$\begin{aligned} \mathcal{M}_{i,j} &= \iint_{\Omega} Y_l^m(\tau(\theta), \phi) \cdot Y_{l'}^{m'}(\theta, \phi) \eta(\theta) d\theta d\phi \\ &= \iint_{\Omega} (S_l^m(\phi)T_l^m(\tau(\theta))) \cdot (S_{l'}^{m'}(\phi)T_{l'}^{m'}(\theta)) \eta(\theta) d\theta d\phi \\ &= \left( \int_0^{2\pi} S_l^m(\phi) S_{l'}^{m'}(\phi) d\phi \right) \cdot \left( \int_0^{\pi} T_l^m(\tau(\theta)) T_{l'}^{m'}(\theta) \eta(\theta) d\theta \right) \\ &= \delta_{mm'} \cdot \int_0^{\pi} T_l^m(\tau(\theta)) T_{l'}^{m'}(\theta) \eta(\theta) d\theta \end{aligned}$$

It can be easily proved that  $\delta_{mm'}$  is non-zero only when  $m = m'$ . This means that  $\mathcal{M}_{i,j}$  is zero for all  $m \neq m'$ , whatever  $\tau(\cdot)$  is chosen. Thus the non-zero entries in the transformation matrices  $\mathcal{M}$  lie at fixed positions that are independent of the angular scaling function  $\tau(\cdot)$ , and that they are relatively sparse.

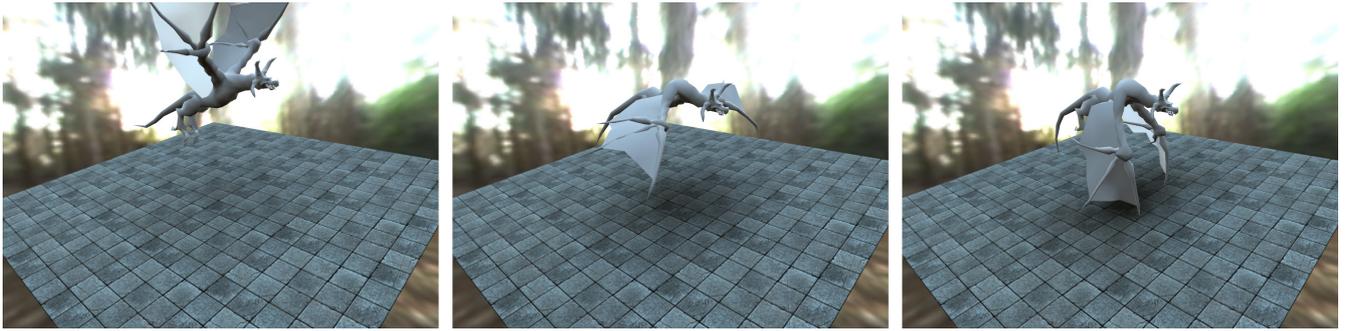


Fig. 8 Shadow fields for deformable objects: Flying dragon.

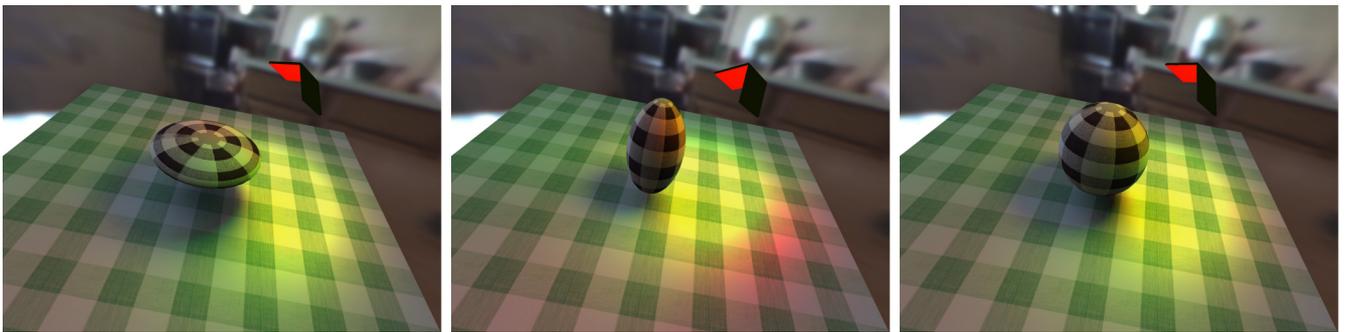


Fig. 9 Shadow fields for deformable objects: Transforming egg with deforming light source.

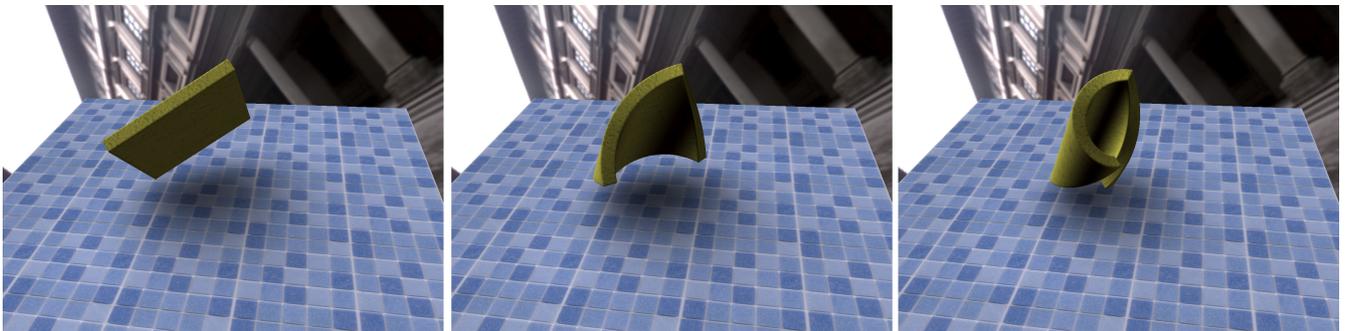


Fig. 10 Shadow fields for deformable objects: Deforming sponge block.

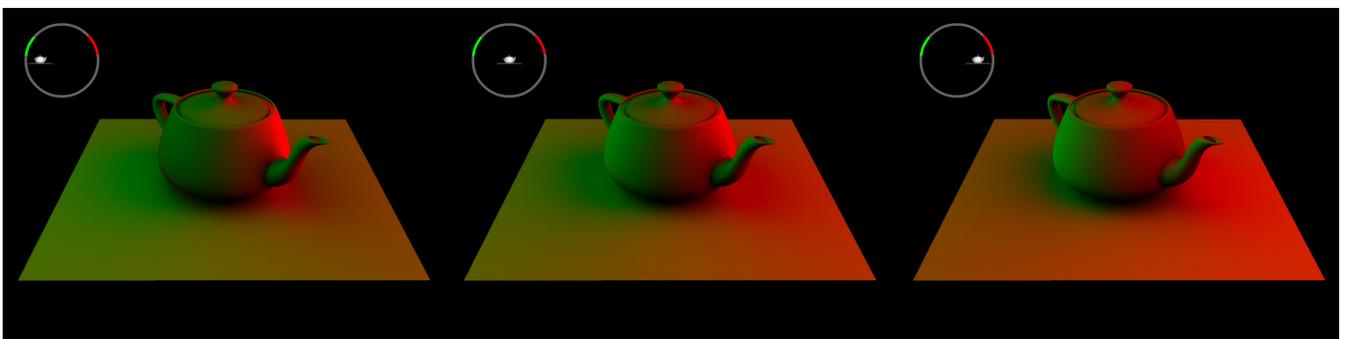


Fig. 11 Mid-range environment lighting: The positions of the teapot relative to the environment light are shown at the upper-left corners.

---

**Jiaping Wang** is a second year Ph.D. student of Institute of Computing Technology, Chinese Academy of Sciences in Graduate School of the Chinese Academy of Sciences. Before that, he received B.S. Degree from Ningbo university in 2002. He has been taking internship in Microsoft Research Asia (MSRA) since 2003. His research interests include appearance modeling and hardware accelerated rendering.

**Kun Xu** is a first year Ph.D. student at Tsinghua University. Before that, he received B.S. Degree from Tsinghua University in 2005. His research interests include real-time rendering and appearance modeling.

**Kun Zhou** is a researcher/project lead of the graphics group at Microsoft Research Asia. He received his B.S. and Ph.D. in Computer Science from Zhejiang University in 1997 and 2002 respectively. His current research focus is geometry processing, texture processing and real time rendering. He holds over 10 granted and pending US patents. Many of these techniques have been integrated in Windows Vista, DirectX and XBOX SDK.

**Stephen Lin** joined Microsoft Research Asia in June 2000, and is currently a researcher of the Internet Graphics group. His research interests include physics-based computer vision and reflectance modelling in computer graphics. He obtained his B.S.E. in electrical engineering from Princeton University and his Ph.D. in computer science and engineering from the University of Michigan.

**Shimin Hu** is currently a Professor in the Computer Science Department at Tsinghua University. His research interests include digital geometry processing, video-based rendering, computer animation, and computer-aided geometric design. He received his Ph.D. from Zhejiang University in 1996. He is on the editorial board of Computer Aided Design.

**Baining Guo** is the research manager of the internet graphics group at Microsoft Research Asia. Before joining Microsoft, Baining was a senior staff researcher in Microcomputer Research Labs at Intel Corporation in Santa Clara, California, where he worked on graphics architecture. Baining received his Ph.D. and M.S. from Cornell University and his B.S. from Beijing University. Baining is an associate editor of IEEE Transactions on Visualization and Computer Graphics. He holds over 30 granted and pending US patents.