A Novel Hemispherical Basis for Accurate and Efficient Rendering

Pascal Gautron[†] Jaroslav Krivanek [‡] Sumanta Pattanaik [§] Kadi Bouatouch [¶]

Abstract

This paper presents a new set of hemispherical basis functions dedicated to hemispherical data representation. These functions are derived from associated Legendre polynomials. We demonstrate the usefulness of this basis for representation of surface reflectance functions, rendering using environment maps and for efficient global illumination computation using radiance caching. We show that our basis is more appropriate for hemispherical functions than spherical harmonics. This basis can be efficiently combined with spherical harmonics in applications involving both hemispherical and spherical data.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism - Rendering, Global Illumination

1. Introduction

High quality rendering and global illumination techniques require an accurate computation of the interaction between light and surfaces. Light/surface interaction models usually rely on hemispherical functions. For example, incident and outgoing radiances at any surface point are defined on a hemisphere, and a BRDF (bi-directional reflectance distribution function) is defined on the cartesian product of two hemispheres. For real time realistic rendering the efficiency and accuracy of the representation of these hemispherical functions is crucial. Researchers have adapted basis functions defined over spheres to represent hemispherical functions such as BRDFs and incident radiance functions (spherical harmonics [RH02, SKS02, WAT92, SAWG91, CMS87], or spherical wavelets [SS95]). However, hemispherical functions introduce discontinuities in the spherical domain at the boundary of the hemisphere and hence their accurate representa-

tion using spherical basis requires a large number of coefficients.

We introduce a *hemispherical* basis that ensures a more accurate representation of hemispherical functions compared to spherical harmonics basis. With minimal effort we combine functions represented by our hemispherical basis with those represented by spherical harmonics for various rendering purposes, such as environment map rendering. This allows our work to be easily integrated into existing algorithms that are designed for using spherical harmonic basis functions.

Our contributions in this paper are:

- The definition of a set of orthogonal basis functions to serve as a hemispherical basis.
- Methods for rotating functions represented using our new basis.
- Application of these basis functions to the representation of BRDFs and hemispherical radiance functions.
- Application of these basis functions to extend the irradiance caching scheme [War94] for non-Lambertian surfaces.

The organization of the paper is as follows. After a description of the state-of-the-art related to functional representations of hemispherical functions (Section 2), we show how the shifting of associated Legendre polynomials can lead to a basis of orthogonal polynomials defined on the

---

[†] University of Central Florida, Orlando, FL, USA - IRISA (Université de Rennes 1), Rennes, France

[‡] University of Central Florida, Orlando, FL, USA - Czech Technical University , Prague, Czech Republic - IRISA (Université de Rennes 1), Rennes, France

[§] University of Central Florida, Orlando, FL, USA

[¶] IRISA (Université de Rennes 1), Rennes, France

hemisphere (Section 3). Section 4 defines methods for efficiently rotating functions represented using our basis functions. In Sections 5 and 6, we present applications of our basis for BRDF representation and radiance caching in the context of global illumination.

2. Previous Work

Three general approaches for functional representation of hemispherical functions have been adopted in graphics. They are spherical harmonics, spherical wavelets and Zernike polynomial representations. A fourth approach, based on Jacobi polynomials, has been defined in the context of radiative transfer.

Spherical harmonics. Spherical harmonics (referred to as SH in the rest of the paper) represent functions defined on the sphere. They are widely used for representation of BRDFs and environment maps [RH02, SKS02, WAT92, SAWG91, CMS87]. Using spherical harmonics for both BRDFs and environment maps, the lighting integral can be computed as a vector dot product. Moreover SH are used to store precomputed shadowing and inter-reflection data [SKS02]. Note that BRDFs, shadowing and inter-reflections at a point are all functions defined on a hemisphere, whereas the SH basis functions are defined on the whole sphere. Hence a larger number of SH coefficients is required for accurate representation.

Sloan et al. [SHHS03] proposed a solution to the domain mismatch by padding a hemispherical function with a mirrored copy of the function itself ("even reflection"). This process significantly improves the accuracy of the representation of a hemispherical function. However, with this approach, the SH coefficients no longer represent the original function. Problems appear when combining with a spherical environment map, since the nonzero values in the lower hemisphere yield erroneous dot product result. To distinguish this process of hemispherical function representation from the standard SH representation we will refer to it as ESH (for Even reflection Spherical Harmonics) representation.

In [SHHS03] Sloan et al. also propose a least squares optimal spherical harmonics (LSOSH) projection for hemispherical functions. Although this method improves the representation accuracy in the upper hemisphere, any hemispherical function projected using this method would contain nonzero values in the lower hemisphere. Therefore this method is only suitable for evaluation purposes, since the added nonzero values yield erroneous results when computing dot product with other LSOSH-projected functions. Consequently this method will not be further detailed.

Spherical wavelets. Wavelets are well known for efficient representation of functions defined on a plane. Schröder and Sweldens [SS95] extended wavelets to the spherical domain for efficient representation of spherical functions. They demonstrated the usefulness of spherical wavelets to accurately represent hemispherical functions, particularly BRDFs. Wavelets are advantageous especially for functions of complex shape. But using a small number of wavelet coefficients to approximate a smooth function might lead to aliasing (see Figure 13 in [SS95]). Aliasing affects the visual quality of the generated images more than an incorrect, although smooth, BRDF shape obtained by spherical harmonics or our new basis. Moreover the hierarchical data structure proposed in [SS95] is not well suited for current graphics hardware.

Zernike polynomials. Zernike polynomials [WC92] are basis functions defined on a disc. Those functions have been adapted by Koenderink et al. [KvDS96] to build a hemispherical basis. The CUReT BRDF database [DvGNK99] uses these basis functions for representing measured BRDFs. However, Zernike polynomials have a high evaluation cost (up to 10 times more than associated Legendre polynomials used for spherical harmonics), and rotation matrices are not available for them.

Makhotkin hemispherical harmonics These basis functions [Mak96, Sme98] are derived from shifted adjoint Jacobi polynomials. The basis functions can be evaluated quickly using a simple recurrence relation, but this representation lacks rotation matrices.

In this paper we propose a new basis defined directly on the hemisphere. Representation of hemispherical functions using this basis is natural and we demonstrate that such a representation is efficient as well.

3. A Novel Hemispherical Basis

In this section, we define a hemispherical basis as an adapted version of SH that relies on the associated Legendre polynomials. In particular, we propose to shift the associated Legendre polynomials to derive our new basis.

Orthogonal Polynomials. A set of polynomials $\{p_l(x)\}$ are said to be orthogonal over an interval $[a,b]$ if $p_l(x)$ is a polynomial of degree l, and for $n,m \geq 0$

$$\int_a^b w(x)p_n(x)p_m(x)\mathrm{d}x = \delta_{nm}c_n \qquad (1)$$

where δ_{nm} is 0 or 1 according as $n \neq m$ or $n = m$, and $w(x)$ is a non negative weighting function. If $c_n = 1$ then the polynomials are orthonormal.

Shifted Polynomials. We define shifting as a linear transformation of x to $k_1 x + k_2$, where $k_1 \neq 0$. If the polynomials $\{p_l(x)\}$ are orthogonal over an interval $[a,b]$ with weight function $w(x)$, then the polynomials $\{p_l(k_1 x + k_2)\}$ are orthogonal over the interval $[\frac{a-k_2}{k_1}, \frac{b-k_2}{k_1}]$ with the weighting

function $w(k_1 x + k_2)$ [Sze75]. If $\{p_l(x)\}$ are orthonormal then $\{(\text{sign } k_1)^l \sqrt{|k_1|} p_l(k_1 x + k_2)\}$ are orthonormal. The polynomials $\{p_l(k_1 x + k_2)\}$ are said to be the shifted versions of $\{p_l(x)\}$.

Associated Legendre Polynomials. The associated Legendre polynomials (or ALP from now on), $\{P_l^m(x)\}$ where $m \in \{0,..,l\}$, are a set of polynomials orthogonal over $[-1,+1]$ with respect to l with the weighting function $w(x) = 1$ [Wei04]:

$$\int_{-1}^{1} P_l^m(x) P_{l'}^m(x) \mathrm{d}x = \frac{2(m+l)!}{(2l+1)(l-m)!} \delta_{ll'}. \qquad (2)$$

Replacing the argument x with $\cos\theta$, we get a set of functions, $\{P_l^m(\cos\theta)\}$ defined over the angular interval $[0,\pi]$. Real valued SH functions, orthogonal over $[0,\pi] \times [0,2\pi)$, are constructed from $P_l^m(\cos\theta)$ as:

$$Y_l^m(\theta,\phi) = \begin{cases} \sqrt{2} K_l^m \cos(m\phi) P_l^m(\cos\theta) & \text{if } m > 0 \\[2mm] \sqrt{2} K_l^m \sin(-m\phi) P_l^{-m}(\cos\theta) & \text{if } m < 0 \\[2mm] K_l^0 P_l^0(\cos\theta) & \text{if } m = 0 \end{cases}$$
$$(3)$$

where K_l^m is the SH normalization value:

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

Shifted ALPs. Using the linear transformation of x to $2x-1$ we get shifted ALPs [Wei04] over the interval $x \in [0,1]$:

$$\widetilde{P}_l^m(x) \;=\; P_l^m(2x-1) \qquad (5)$$

The rationale behind this shifting is that by replacing the argument x with $\cos\theta$, we get functions $\{\widetilde{P}_l^m(\cos\theta)\}$ that are defined in the interval $\theta \in [0, \frac{\pi}{2}]$, the polar angle range of the hemisphere:

$$\widetilde{P}_l^m(\cos\theta) \;=\; P_l^m(2\cos\theta - 1) \text{ with } \theta \in [0, \tfrac{\pi}{2}]. \qquad (6)$$

The shifted ALPs remain orthogonal, but the normalization changes according to the definition given above. The following equation shows the orthogonal relation with respect to l with 1 as the weighting function.

$$\int_0^1 \widetilde{P}_l^m(x) \widetilde{P}_{l'}^m(x) dx \;=\; \int_0^1 P_l^m(2x-1) P_{l'}^m(2x-1) \mathrm{d}x$$
$$= \frac{(m+l)!}{(2l+1)(l-m)!} \delta_{ll'} \qquad (7)$$

From Shifted ALPs to Hemispherical Basis. In the same way that SH functions are constructed from ALPs, we construct our hemispherical basis functions $\{H_l^m(\theta,\phi)\}$ from

shifted ALPs.

$$H_l^m(\theta,\phi) = \begin{cases} \sqrt{2} \widetilde{K}_l^m \cos(m\phi) \widetilde{P}_l^m(\cos\theta) & \text{if } m > 0 \\[2mm] \sqrt{2} \widetilde{K}_l^m \sin(-m\phi) \widetilde{P}_l^{-m}(\cos\theta) & \text{if } m < 0 \\[2mm] \widetilde{K}_l^0 \widetilde{P}_l^0(\cos\theta) & \text{if } m = 0 \end{cases}$$
$$(8)$$

with the following normalization value:

$$\widetilde{K}_l^m = \sqrt{\frac{(2l+1)(l-|m|)!}{2\pi(l+|m|)!}} \qquad (9)$$

By analogy with spherical harmonics, we name the $H_l^m(\theta,\phi)$ basis functions "hemispherical harmonics" (abbreviated by HSH in the remainder of this document). HSH are orthogonal over $[0, \frac{\pi}{2}] \times [0,2\pi)$ with respect to both l and m. 3D plots of the first few HSH functions are given in Appendix A. The following section presents rotation methods for HSH-projected functions.

4. HSH Rotation

In the context of realistic rendering using HSH, one might have to rotate HSH-projected functions efficiently (Section 6.2). This section contains three methods dealing with such rotations: by converting to spherical harmonics, using spherical harmonics rotation matrices, and precomputing rotation matrices.

The first method works as follows : in section 6.1 we define a matrix $\mathbf{C}$ used to convert coefficients from SH to HSH and vice-versa. The HSH rotation can be carried out by converting the HSH coefficients to SH using $\mathbf{C}$, then applying a SH rotation matrix, and finally convert the SH coefficients back to HSH using $\mathbf{C}^{-1}$. Since the SH representation is less accurate than HSH, we propose to use a non-square conversion matrix: the intermediate SH representation uses more coefficients than HSH. It must be noted that this method automatically removes the data that disappears underneath the horizon during rotation (Figure 2(c)).

The two next methods rely on Euler's rotation theorem: any rotation may be described using three angles. We represent our rotations by the ZYZ convention, where Z is the vertical axis of a right handed orthogonal coordinate system.

The rotation of HSH functions around the Z axis is the same as that of SH functions [SAWG91]. The rotation around the Y axis is lossy for all hemispherical functions: after such a rotation, a part of the function must be set to zero (Figure 2). Hence we carry out the Y axis rotation in two steps.

In the first step we delete the hemisphere digon I (Figure 1) that disappears after the rotation around axis Y by angle β (Figure 2(b)). We carry out this deletion as a coefficient

vector transformation with a matrix $\mathbf{M}(\beta)$. The elements of this matrix are defined as:

$$\mathbf{M}_{l,l'}^{m,m'}(\beta) = \int_{H-I} H_l^m(\theta,\phi) H_{l'}^{m'}(\theta,\phi) \sin\theta \mathrm{d}\theta \mathrm{d}\phi \quad (10)$$

where $H-I$ is the hemisphere after the removal of the digon. As the digon deletion adds high frequencies in the signal, this matrix is dense. For practical purposes, we precompute matrices for a set of rotation angles, then linearly interpolate them for any other angle at runtime.
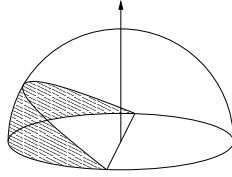

Figure 1: *Hemisphere and a digon [Dutre 2003]. The digon is the shaded part of the hemisphere.*
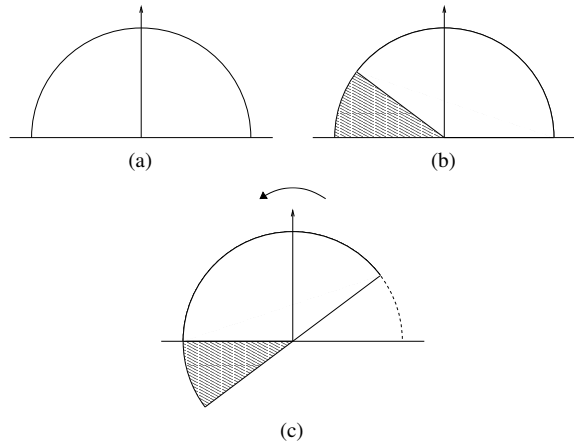

Figure 2: *HSH rotation around Y axis. Hemisphere before rotation (a), shaded digon that is set to zero (b), Hemisphere after rotation (c).*

Once this "deletion" transformation is applied, we have to compute and apply a rotation matrix for rotation around Y. Since HSH definition is close to SH, we chose to use SH rotation matrices [IR96] to derive HSH rotations.

Applying a SH rotation matrix of angle β_{SH} around Y axis to a HSH vector of coefficients $\{f_i\}$ is equivalent to:

- shifting the HSH-projected function f_{HSH} to the whole sphere (Eqs. 11, 12), yielding f_{SH} (Figure 3(b)).

$$f_{HSH}(\theta,\phi) = \sum_i f_i H_i(\theta,\phi) \quad [0,\frac{\pi}{2}] \times [0,2\pi] \rightarrow \mathbf{R} \quad (11)$$

$$f_{SH}(\theta,\phi) = \sum_i f_i Y_i(\theta,\phi) \quad [0,\pi] \times [0,2\pi] \rightarrow \mathbf{R} \quad (12)$$

- applying a SH rotation matrix $R_{\beta_{SH}}^{SH}$ to f_{SH} (Figure 3(c))
- shifting $R_{\beta_{SH}}^{SH}(f_{SH})$ to the upper hemisphere to obtain $R_{\beta_{HSH}}^{HSH}(f_{HSH})$ (Figure 3(d))
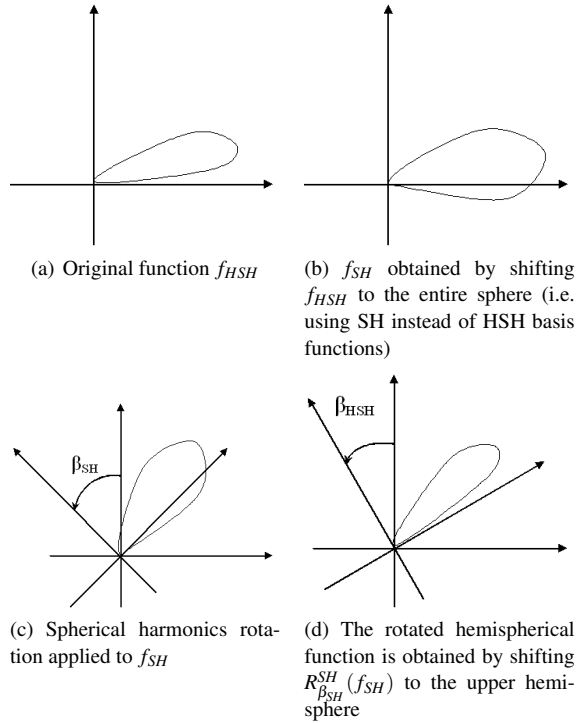

(a) Original function f_{HSH}

(b) f_{SH} obtained by shifting f_{HSH} to the entire sphere (i.e. using SH instead of HSH basis functions)

(c) Spherical harmonics rotation applied to f_{SH}

(d) The rotated hemispherical function is obtained by shifting $R_{\beta_{SH}}^{SH}(f_{SH})$ to the upper hemisphere

Figure 3: *HSH rotation process : the original hemispherical function is first shifted to the whole sphere, then rotated using SH rotation matrices. The last step shifts the rotated function back to the hemisphere.*

Our aim is to determine the relation between β_{SH} and β_{HSH} : since SH and HSH domains are different, $\beta_{SH} \neq \beta_{HSH}$. Using Equation 6, a rotation of angle β_{HSH} on the hemisphere is equivalent to a rotation of angle β_{HSH} (Eq. 13).

$$\beta_{SH} = \arccos(2\cos\beta_{HSH} - 1) \quad (13)$$

This equation shows that the SH rotation matrices can be applied to HSH for a rotation around Y axis provided that no part of the original function moves into the lower hemisphere after rotation (this condition is fulfilled by removing the considered digon (Eq. 10)).

The last rotation method consists of the inclusion of the rotation in the precomputed matrices in Equation 10. The matrix element values are defined by

$$\mathbf{M}_{\mathbf{R}l,l'}^{m,m'}(\beta) = \int_{H-I} H_l^m(\theta,\phi) H_{l'}^{m'}(R_\beta(\theta,\phi)) \sin\theta \mathrm{d}\theta \mathrm{d}\phi$$
$$(14)$$

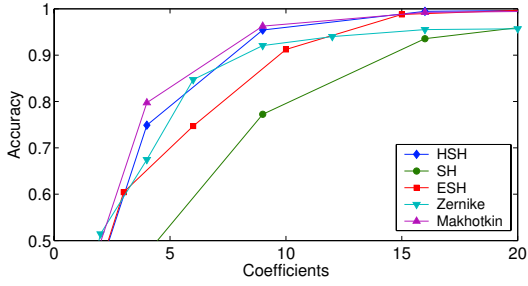where $H-I$ is the hemisphere after removing the digon,

Figure 4: *Accuracy versus number of coefficients for approximation of a Phong lobe, $(\cos\alpha)^5$, using HSH, SH, ESH, Zernike and Makhotkin approaches.*

and $R_\beta(\theta,\phi)$ represents the direction (θ,ϕ) rotated around Y axis by angle β.

The accuracy of this method comes at the expense of memory: an accurate rotation requires a large number of precomputed matrices. The two next sections present applications of HSH in the context of image synthesis.

5. HSH for Hemispherical Function Approximation

The $[0,\frac{\pi}{2}]\times[0,2\pi)$ domain of HSH basis functions makes them ideal for representation of hemispherical functions such as BRDFs and radiance functions around surface points. In this section we show a accuracy comparison between HSH representation and previous methods.

We first show the approximation of a Phong lobe [Pho75] $(\cos\alpha)^5$, where α is the angle between the specular reflection direction and the viewing direction. Figure 4 plots the approximation accuracy as function of the number of coefficients in HSH, SH, ESH, Zernike and Makhotkin's representation. We estimate this accuracy by computing the fraction of the total energy captured for a given number of coefficients [RH02]:

$$\text{Accuracy} = \frac{\sum_{l=0}^{n}\sum_{m=-l}^{l}|f_l^m|^2}{\int_0^{2\pi}\int_0^{\frac{\pi}{2}}f(\theta,\phi)^2\sin\theta d\theta d\phi} \quad (15)$$

The observation in this figure and the results in later sections show that the HSH representation indeed provides a better representation than SH with fewer coefficients. Moreover, our basis performs comparably to Makhotkin's method, and exhibits a higher accuracy than both ESH and Zernike approaches in the Phong case.

HSH Representation of BRDFs. BRDFs are functions defined over the cartesian product of two hemispheres, corresponding to incoming and outgoing directions. One can use either of the following two approaches to represent them using HSH basis. One approach is to use two successive HSH
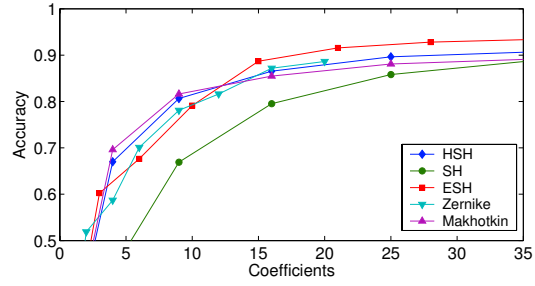
Figure 5: *Accuracy versus number of coefficients for HSH, SH, ESH, Zernike and Makhotkin representations. These values are obtained for an anisotropic Ward BRDF with $k_d = 0$, $k_s = 1$, $\alpha_x = 0.2$, $\alpha_y = 0.5$.*

transformations, one for the incoming hemisphere and the other for the outgoing hemisphere (similar to the approach used by Westin et al. [WAT92] using SH basis). The other approach is to discretize the outgoing hemisphere and use HSH to represent the incoming hemisphere corresponding to each outgoing direction (similar to the approach used by Kautz et al. [KSS02]).

We use the latter approach, in which the n-th order representation of a BRDF $f_{(\theta_o,\phi_o)}$ for any given outgoing sample direction (θ_o,ϕ_o) is

$$f_{(\theta_o,\phi_o)}(\theta_i,\phi_i) \approx \sum_{l=0}^{n}\sum_{m=-l}^{l}c_l^m(\theta_o,\phi_o)H_l^m(\theta_i,\phi_i), \quad (16)$$

where

$$c_l^m(\theta_o,\phi_o) = \int_0^{2\pi}\int_0^{\pi/2}f(\theta_o,\phi_o,\theta_i,\phi_i)H_l^m(\theta_i,\phi_i)\sin\theta_i d\theta_i d\phi_i. \quad (17)$$

We sample the outgoing hemisphere for (θ_o,ϕ_o) using parabolic parametrization proposed by Heidrich and Seidel [HS99].

Figure 5 plots the approximation accuracy of the HSH, SH, ESH, Zernike and Makhotkin's representation of an anisotropic Ward BRDF [War92] as a function of the number of coefficients. The plot shows that although the ESH provides better results in this case, the HSH representation is still more accurate than the SH representation when the same number of coefficients is used. As in the previous example, HSH, Zernike and Makhotkin's method perform comparably.

Figure 6 shows image frames from our GPU based real-time renderer. The teapots shown in the images are lit by a single point light source. Those in the left column are assigned a Phong BRDF, and in the right column an anisotropic Ward BRDF. The renderings in the three rows from top to bottom correspond to analytic BRDF functions, and their HSH and SH representations respectively. 32×32 outgoing sample directions are used for the HSH and SH representations. The coefficients and the basis function val-

ues are stored as textures. For any incoming direction (θ_i, ϕ_i) and outgoing direction (θ_o, ϕ_o), $c_l^m(\theta_o, \phi_o)$ and $H_l^m(\theta_i, \phi_i)$ values are computed by bilinear texture interpolation. Equation 16 for $f_{(\theta_o, \phi_o)}(\theta_i, \phi_i)$ is evaluated using a multi-pass pixel shader program. The images using HSH representation (Figures 6(c) and (d)) are visually closer to the images obtained using analytical BRDFs. Note the ringing in the images obtained using SH representation of the BRDF (Figures 6(e) and (f)).
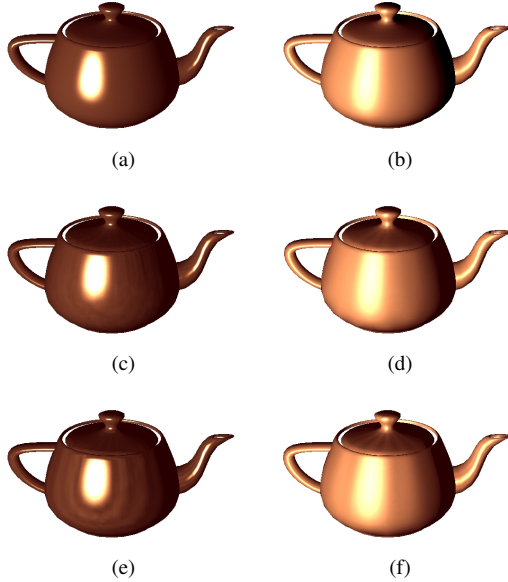

(a)   (b)

(c)   (d)

(e)   (f)

Figure 6: *Images rendered using Phong BRDF (left column) and anisotropic Ward BRDFs (right column). Figures (a) and (b) use analytic BRDFs; Figures (c) and (d) use order 7 HSH representation and Figures (e) and (f) use order 7 SH representation.*

Figures 5 and 6 demonstrate that HSH representation of BRDF using a given number of coefficients outperforms SH representation both in terms of computed and visual accuracy.

6. HSH in Realistic Rendering

6.1. Environment Map Rendering

Hardware rendering of surfaces with arbitrary BRDFs illuminated by environment maps has recently received wide attention [KSS02, RH02, SKS02]. Environment maps are discrete representations of incoming radiance from directions defined on a whole sphere. Consequently they can be approximated using a spherical basis such as SH. In this section, we show that a SH representation of environment maps is straightforward to combine with a HSH representation of BRDFs. We demonstrate interactive hardware rendering using HSH representation for BRDFs and SH representation for environment maps.

Since SH and HSH bases are not mutually orthogonal, we cannot directly compute an integral containing a SH representation of an environment map and a HSH representation of a BRDF. However, incoming light from the environment at any surface point is only defined on a hemisphere. Consequently, we propose to convert the environment map coefficients from SH to HSH after rotating them from the global to the local frame. This conversion is achieved by transforming the rotated SH coefficients with a basis change matrix $\mathbf{C}$. Each element of this matrix is defined by:

$$\mathbf{C}_{l,l'}^{m,m'} = \int_0^{2\pi} \int_0^{\pi/2} H_l^m(\theta, \phi) Y_{l'}^{m'}(\theta, \phi) \sin\theta \, d\theta \, d\phi \quad (18)$$

This matrix is very sparse, nonzero values occur only for $m = m'$, which significantly reduces the cost of the transformation from SH to HSH representation. Note that this transformation is not invertible since the integration in the interval $\theta \in [0, \pi/2]$ only accounts for the information belonging to one half of the sphere, and completely discards the information in the other half. For $l < 3$ and $l' < 3$ the conversion matrix $\mathbf{C}_{3,3}$ is:

$$\begin{pmatrix} 0.71 & 0 & 0.41 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.60 & 0 & 0 & 0 & 0.19 & 0 & 0 & 0 \\ 0.82 & 0 & 0.71 & 0 & 0 & 0 & 0.18 & 0 & 0 \\ 0 & 0 & 0 & 0.60 & 0 & 0 & 0 & 0.19 & 0 \\ 0 & 0 & 0 & 0 & 0.53 & 0 & 0 & 0 & 0 \\ 0 & 0.77 & 0 & 0 & 0 & 0.51 & 0 & 0 & 0 \\ 0.40 & 0 & 0.78 & 0 & 0 & 0 & 0.53 & 0 & 0 \\ 0 & 0 & 0 & 0.77 & 0 & 0 & 0 & 0.51 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.53 \end{pmatrix}$$

Rendering with environment maps proceeds as follows:

- Compute SH representation of the environment map.
- Compute HSH representation of the surface BRDF for every surface in the scene
- For every rendered surface point:

  - Rotate the coefficients of the environment map to fit the local frame associated with the surface point.
  - Transform the coefficients from SH to HSH using the precomputed basis change matrix $\mathbf{C}$.
  - Compute the dot product between the HSH coefficients of the environment map and HSH coefficients of the surface BRDF for the required outgoing direction.

Figure 7 shows renderings of a teapot using three different environment maps. Table 1 compares frame rates achieved by the HSH approximated BRDF against the SH approximated BRDF. Our method achieves similar frame rates to the SH approach for the same number of coefficients since the conversion step does not require much additional computation.

It should be noted that the ESH representation can not be used in this context because of the extra non-zero BRDF

(a) Grace Cathedral   (b) Eucalyptus Grove

(c) St Peter's Basilica

Figure 7: *Environment map rendering using HSH representation of the BRDF (order 10). The glossy teapot is lit by three different environment maps.*

information present in the lower hemisphere: the dot product between the environment map and the BRDF would take into account both upper and lower hemispheres, leading to erroneous results.

Order of representation (n)	2	3	5	8	10
HSH (fps)	220	138	26.2	3.1	1.6
SH (fps)	223	160	29	3.28	1.73

Table 1: *Timing results (in frames per second) for environment maps rendering using HSH and SH representation of BRDFs. The timing was obtained on a 2.4GHz Intel Xeon with ATI Radeon 9800 Pro card for an image size 700x600. The teapot model used has 1873 vertices.*

6.2. Radiance Caching for Efficient Global Illumination Computation

Monte Carlo (MC) ray tracing remains the method of choice for accurately computing global illumination. It is, however, very expensive when it comes to computing illumination on surfaces with low frequency BRDFs. Many sample rays have to be traced to get a reasonable estimate of the following illumination integral at a point.

$$L(\theta_o, \phi_o) = \int_0^{2\pi} \int_0^{\pi/2} L_i(\theta_i, \phi_i) f(\theta_i, \phi_i, \theta_o, \phi_o) \cos\theta_i \sin\theta_i \mathrm{d}\theta_i \mathrm{d}\phi_i$$

(19)

where L is the outgoing radiance, L_i is the incoming radiance and f is the BRDF. Various algorithms have been proposed to reduce the cost of this computation. Ward et al.'s [WRC88] irradiance caching is one of such algorithms. It significantly reduces the amortized cost of the illumination

---

for (every BRDF in the scene) **do**
  - Compute the HSH representation of the BRDF.
end for
for (every surface point $\vec{P}$ visible through the image pixels) **do**
  if (radiance cache samples exist near $\vec{P}$) **then**
    - Compute HSH coefficients of the incoming radiance at $\vec{P}$ by gradient based interpolation.
  else
    - Monte Carlo sample the incoming hemisphere around $\vec{P}$ and get the incoming radiance along a number of incoming directions
    - Compute HSH coefficients of the incoming radiance function at $\vec{P}$ by Monte Carlo quadrature.
    - Compute the gradient of the coefficients at $\vec{P}$.
    - Store the coefficients and the gradients in the radiance cache.
  end if
  - Compute the outgoing radiance at $\vec{P}$ as the dot product of the HSH coefficients of incoming radiance and surface BRDF.
end for

Figure 8: *Outline of our radiance caching algorithm.*

integral estimation at Lambertian surfaces by caching and reusing irradiance.

We extend this algorithm to support non-Lambertian surfaces by caching and reusing the incoming radiance function instead of irradiance. The incoming radiance is represented by hemispherical harmonics.

Figure 8 briefly outlines the steps of our radiance caching algorithm. They are similar to those used by Ward et al. [WRC88] in their irradiance caching. We make use of the same basic observation that the indirect illumination changes slowly over each surface with low-frequency BRDFs. The main difference is that we cache the directional radiance function instead of scalar irradiance. This allows us to compute outgoing radiance from surfaces with non-Lambertian BRDFs.

Though this method can support all types of BRDFs, we would like to point out that the accurate representation of sharp BRDFs requires a large number of HSH coefficients. Hence, for high frequency BRDFs, direct computation of the outgoing radiance by MC importance sampling is likely to be more efficient than interpolation based computation. The various steps of the algorithm shown in Figure 8 are described in the following paragraphs. For a more detailed description of radiance caching please refer to [KGPB04].

BRDF Representation. We precompute the HSH representation of surface BRDFs using the method described in Section 5. Note that we *premultiply* BRDFs by the cosine term,

$\cos\theta_i$, of the illumination integral (Eq. 19) before computing their HSH representation.

Incoming Radiance Computation. We represent the incoming radiance function L_i at a point $\vec{P}$ by a vector of HSH coefficients λ_l^m:

$$L_i(\theta,\phi) \approx \sum_{l=0}^{n} \sum_{m=-l}^{l} \lambda_l^m H_l^m(\theta,\phi), \qquad (20)$$

where n is the order of HSH representation and

$$\lambda_l^m = \int_0^{2\pi} \int_0^{\pi/2} L_i(\theta,\phi) H_l^m(\theta,\phi) \sin\theta \, d\theta \, d\phi. \qquad (21)$$

We compute λ_l^m by MC quadrature with uniform sampling of the hemisphere of incoming directions. The equation for this quadrature is

$$\lambda_l^m = \frac{2\pi}{N} \sum_{k=1}^{N} L_i(\theta_k,\phi_k) H_l^m(\theta_k,\phi_k), \qquad (22)$$

where $L_i(\theta_k,\phi_k)$ is the incoming radiance coming from the sampled direction (θ_k,ϕ_k) and N is the number of sample directions. We use a fixed number N of sample directions.

Radiance Gradient computation. There are two major differences between the gradient based interpolation used here and by Ward and Heckbert [WH92].

- We rotate the radiance functions before interpolation, hence we do not use any rotational gradient.
- We use uniform sampling for computing λ_l^m, whereas Ward and Heckbert [WH92] use cosine sampling for computing irradiance. Hence our translational gradient computation is different.

We compute the translational gradient $\nabla\lambda_l^m = \left[\frac{\partial\lambda_l^m}{\partial x}, \frac{\partial\lambda_l^m}{\partial y}, 0 \right]$ for each λ_l^m. The translational gradient is defined in the local coordinate frame at a sample point $\vec{P}$. The derivative $\partial\lambda_l^m/\partial x$ at $\vec{P}$ is computed by displacing the sample point $\vec{P}$ along the local X axis by Δx to $\vec{P}'$. The coefficient $\lambda_l^{m'}$ at $\vec{P}'$ is computed as

$$\lambda_l^{m'} = \frac{2\pi}{N} \sum_{k=1}^{N} \frac{\|\vec{W}_k - \vec{P}\|^2}{\|\vec{W}_k - \vec{P}'\|^2} \frac{\cos\xi'}{\cos\xi} L_i(\theta_k,\phi_k) H_l^m(\theta_k',\phi_k'),$$

and $\partial\lambda_l^m/\partial x$ is computed as

$$\frac{\partial\lambda_l^m}{\partial x} = \frac{\lambda_l^{m'} - \lambda_l^m}{\Delta x},$$

where the denotation is:

(θ_k,ϕ_k)	Direction of the k-th ray used to sample the hemisphere at $\vec{P}$.
$L_i(\theta_k,\phi_k)$	Incoming radiance from the direction (θ_k,ϕ_k).
$\vec{W}_k$	Ray - surface intersection point for direction (θ_k,ϕ_k).
ξ	Angle between the vector $\vec{W}_k - \vec{P}$ and the surface normal at $\vec{W}_k$.
(θ_k',ϕ_k')	Direction from $\vec{P}'$ to $\vec{W}_k$.
ξ'	Angle between the vector $\vec{W}_k - \vec{P}'$ and the surface normal at $\vec{W}_k$.

The computation of $\partial\lambda_l^m/\partial y$ proceeds in a similar way. The derivation of those formulas is given in [KGPB04].

Radiance Interpolation. We use a weighted interpolation scheme similar to the one proposed in [WH92] for interpolating coefficients λ_l^m at any required surface point $\vec{P}$. The only difference in the schemes is that we replace the use of the rotational gradient by applying a rotation to the vector of coefficients $\Lambda_i = \{\lambda_{l,i}^m\}$. This aligns the coordinate frame at the sample point $\vec{P}_i$ with the frame at $\vec{P}$. The weight $w_i(\vec{P})$ of sample $\vec{P}_i$ is

$$w_i(\vec{P}) = \left(\frac{\|\vec{P} - \vec{P}_i\|}{R_i} + \sqrt{1 - \vec{N}\cdot\vec{N}_i} \right)^{-1}, \qquad (23)$$

where $\vec{N}$ is the surface normal at $\vec{P}$, $\vec{N}_i$ is the surface normal at $\vec{P}_i$, and R_i is the harmonic mean distance to objects visible from $\vec{P}_i$. The coefficient vector $\Lambda = \{\lambda_l^m\}$ of the interpolated radiance is:

$$\Lambda(\vec{P}) = \frac{\sum_S \mathbf{R}_i \left(\Lambda_i + d_x \frac{\partial\Lambda_i}{\partial x} + d_y \frac{\partial\Lambda_i}{\partial y} \right) w_i(\vec{P})}{\sum_S w_i(\vec{P})}, \qquad (24)$$

where $S = \{i | w_i(\vec{P}) > 1/a\}$ and a is a user defined desired accuracy. The definition of the set S is the criterion used to decide which radiance cache samples can be used for interpolation. Derivatives $\partial\Lambda_i/\partial x$ and $\partial\Lambda_i/\partial y$ are the translational gradient components and (d_x, d_y) are the displacements of $\vec{P} - \vec{P}_i$ along the X and Y axes of the sample i's local coordinate frame. $\mathbf{R}_i$ is the HSH rotation (Section 4) that aligns the coordinate frame at $\vec{P}_i$ with the frame at $\vec{P}$. It transforms the whole coefficient vector $\Lambda_i = \{\lambda_{l,i}^m\}$ after the translational gradient is applied.

Outgoing Radiance Computation. As both the incoming radiance L_i and the BRDF f in the illumination integral (Eq. 19) are now represented as HSH coefficient vectors, the integral computation reduces to the dot product

$$L(\theta_o,\phi_o) = \sum_{l=0}^{n-1} \sum_{m=-l}^{l} \lambda_l^m f_l^m(\theta_o,\phi_o),$$
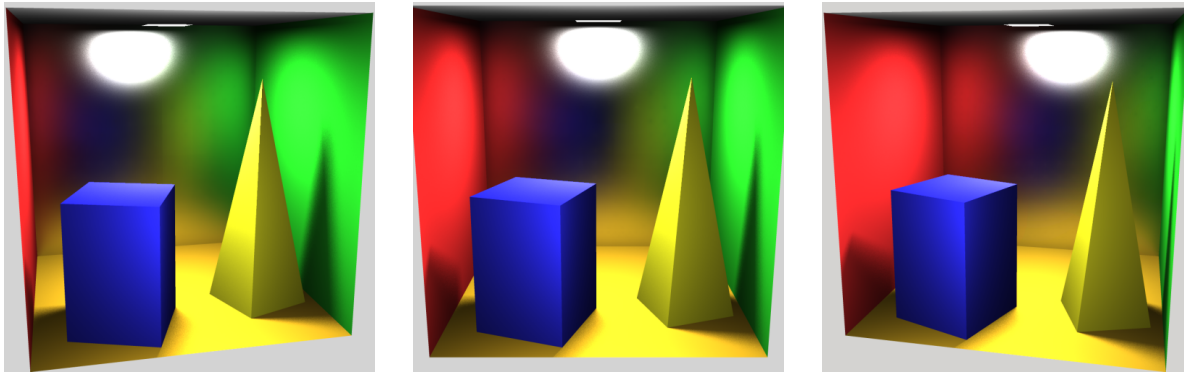
Figure 9: *Cornell box with glossy back wall rendered using radiance caching. Radiance function at sample points are computed as its HSH coefficients. 1121 radiance samples are used to render the center view (b). The number of additional samples needed when the viewpoint changes to left (a) and right (c) is 250 and 198. The size of each of the images is 850×850.*

where λ_l^m is an interpolated incoming radiance coefficient and $f_l^m(\theta_o, \phi_o)$ is a BRDF coefficient at $\vec{P}$ for the outgoing direction (θ_o, ϕ_o).

Results. Figure 9 shows three renderings of a Cornell box with glossy back wall (Phong, exponent 22), taken from three different viewpoints at resolution 850×850. The number of rays cast to sample each hemisphere is $N = 4000$. The order of HSH used is $n = 10$. Except for the back wall, all objects are Lambertian. Direct lighting and first bounce reflection for the back wall were used to generate the images. The timings given here do not include direct illumination computation. Image 9(b) took 25.8 sec. to render[†] with 1121 radiance cache samples. Images 9(a) and (c) took 11.2 and 10.3 seconds to render since only 250 and 198 additional radiance cache samples were needed. Memory consumption of the radiance cache was 1.5 MB. The timings compare well with 260 seconds rendering time for the images of approximately the same visual quality generated with MC importance sampling.

7. Conclusion

This paper described a new basis for hemispherical data representation. This basis is ideal for the representation of low-frequency hemispherical functions. These functions are often encountered in many applications where the interaction between a point on a surface and its environment needs to be computed. We demonstrated its use for the following representative sets of applications: efficient representation of BRDFs, real-time environment map rendering of non-diffuse surfaces and efficient global-illumination computation. As

shown, this basis provides a more compact and accurate way of representing hemispherical functions than spherical harmonics.

References

[CMS87]   CABRAL B., MAX N., SPRINGMEYER R.: Bidirectional reflection functions from surface bump maps. In *Proceedings of SIGGRAPH* (1987), ACM Press, pp. 273–281.

[Dut03]   DUTRE P.: Global illumination compendium. http://www.cs.kuleuven.ac.be/ phil/GI/.

[DvGNK99]   DANA K. J., VAN GINNEKEN B., NAYAR S. K., KOENDERINK J. J.: Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics 18*, 1 (1999), 1–34.

[HS99]   HEIDRICH W., SEIDEL H.-P.: Realistic, hardware-accelerated shading and lighting. In *Proceedings of SIGGRAPH* (1999), ACM Press/Addison-Wesley Publishing Co., pp. 171–178.

[IR96]   IVANIC J., RUEDENBERG K.: Rotation matrices for real spherical harmonics. direct determination by recursion. *J. Phys. Chem. 100*, 15 (1996), 6342–6347.

[KGPB04]   KŘIVÁNEK J., GAUTRON P., PATTANAIK S., BOUATOUCH K.: *Radiance Caching for Efficient Global Illumination Computation*. Tech. Rep. 1623, IRISA, Rennes, France, June 2004.

[KSS02]   KAUTZ J., SLOAN P.-P., SNYDER J.: Fast, arbitrary BRDF shading for low-frequency lighting using spherical harmonics. In *Proceedings of the 13th*

[†] The timings in this section were measured on a 2.26GHz Pentium 4 with 512MB RAM running Windows XP.

Eurographics workshop on Rendering (2002), Eurographics Association, pp. 291–296.

[KvDS96] KOENDERINK J., VAN DOORN A., STAVRIDI M.: Bidirectional reflection distribution function expressed in terms of surface scattering modes. *ECCV B* (1996), 28–39.

[Mak96] MAKHOTKIN O. A.: Analysis of radiative transfer between surfaces by hemispherical harmonics. *Journal of Quantitative Spectroscopy and Radiative Transfer 56*, 6 (1996), 869–879.

[Pho75] PHONG B. T.: Illumination for computer generated pictures. *Commun. ACM 18*, 6 (1975), 311–317.

[RH02] RAMAMOORTHI R., HANRAHAN P.: Frequency space environment map rendering. In *Proceedings of SIGGRAPH* (2002), ACM Press, pp. 517–526.

[SAWG91] SILLION F. X., ARVO J. R., WESTIN S. H., GREENBERG D. P.: A global illumination solution for general reflectance distributions. In *Proceedings of SIGGRAPH* (1991), ACM Press, pp. 187–196.

[SHHS03] SLOAN P.-P., HALL J., HART J., SNYDER J.: Clustered principal components for precomputed radiance transfer. *ACM Trans. Graph. 22*, 3 (2003), 382–391.

[SKS02] SLOAN P.-P., KAUTZ J., SNYDER J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Trans. Graph. 21*, 3 (2002), 527–536.

[Sme98] SMELOV V. V.: On completeness of semispherical harmonics system. *Siberian Journal of Mathematics 1*, 4 (1998), 391–395.

[SS95] SCHRÖDER P., SWELDENS W.: Spherical wavelets: efficiently representing functions on the sphere. In *Proceedings of SIGGRAPH* (1995), ACM Press, pp. 161–172.

[Sze75] SZEGÖ G.: *Orthogonal polynomials*, 4 ed. American Mathematical Society, Providence, Rhode Island, 1975, p. page 29.

[War92] WARD G. J.: Measuring and modeling anisotropic reflection. In *Proceedings of SIGGRAPH* (1992), ACM Press, pp. 265–272.

[War94] WARD G. J.: The radiance lighting simulation and rendering system. In *Proceedings of SIGGRAPH* (1994), ACM Press, pp. 459–472.

[WAT92] WESTIN S. H., ARVO J. R., TORRANCE K. E.: Predicting reflectance functions from complex surfaces. In *Proceedings of SIGGRAPH* (1992), ACM Press, pp. 255–264.

[WC92] WYANT J. C., CREATH K.: Basic wavefront aberration theory for optical metrology. In *Applied optics and Optical Engineering, Vol XI* (1992), Academic Press, Inc., pp. 27–39.

[Wei04] WEISSTEIN E.: *World of Mathematics: A Wolfram Web Resource*. http://mathworld.wolfram.com/LegendrePolynomial.html, 2004.

[WH92] WARD G. J., HECKBERT P. S.: Irradiance gradients. In *Proceedings of 2nd Eurographics Workshop on Rendering* (Bristol, 1992).

[WRC88] WARD G. J., RUBINSTEIN F. M., CLEAR R. D.: A ray tracing solution for diffuse interreflection. In *Proceedings of SIGGRAPH* (1988), ACM Press, pp. 85–92.

Appendix A: HSH Function Plots

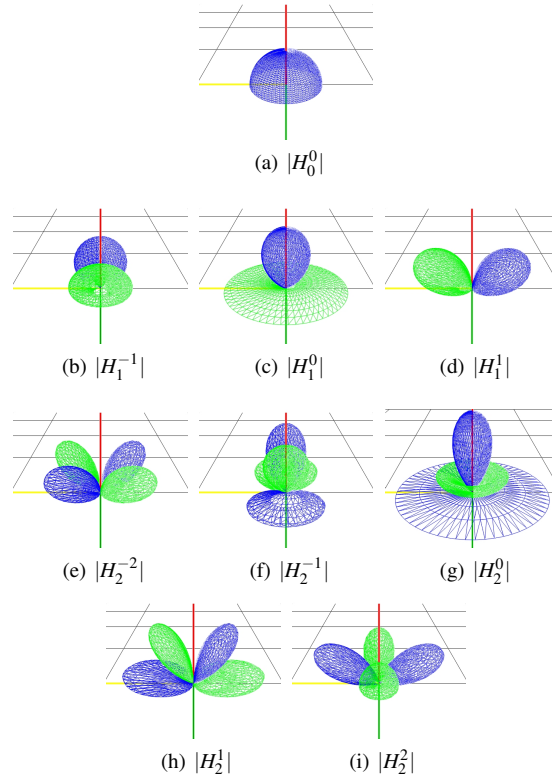In Figure 10 we show the shape of the first few HSH functions.

Figure 10: *Function plots of H_l^m for l from 0 to 2.*

Appendix B: Associated Legendre Polynomials

The ALPs are defined as

$$P_l^m(x) = (-1)^m(1-x^2)^{m/2}\frac{d^m}{dx^m}P_l(x) \quad (25)$$

$$= \frac{(-1)^m}{2^l l!}(1-x^2)^{m/2}\frac{d^{m+l}}{dx^{m+l}}(x^2-1)^l \quad (26)$$

where the $P_l(x)$ are the unassociated Legendre Polynomials. These polynomials can be computed efficiently using the following recurrence relations :

$$(l-m)P_l^m(x) = x(2l-1)P_{l-1}^m(x) \quad (27)$$
$$- (l+m-1)P_{l-2}^m(x)$$
$$P_m^m(x) = (-1)^m(2m-1)!!(1-x^2)^{m/2} \quad (28)$$
$$P_{m+1}^m(x) = x(2m+1)P_m^m(x) \quad (29)$$

where $n!!$ is the double factorial of n, i.e. the product of all positive integers less than or equal to n with same parity as n.