Goniometric Diagram Mapping for Hemisphere

Vlastimil Havran Kirill Dmitriev Hans-Peter Seidel

MPI Informatik, Saarbrücken, Germany

Abstract

Describing the light intensity over the hemisphere using a goniometric diagram is a common practice in the lighting industry and is prescribed for instance by IESNA (Illuminating Engineering Society of North America) standards. Goniometric diagram specifies the spatial distribution of the emitted power via a hemispherical surface subdivided by meridians and parallels. Similar tabulated representations are extensively used for complex bidirectional reflectance distribution functions (BRDF) that are difficult to approximate with analytical models. We present an approximative bicontinuous mapping from the unit square to a goniometric diagram on the hemisphere. This mapping has low distortion and the error of the approximation is low. The proposed mapping algorithm is obtained as a composition of four mappings. We outline its use for importance sampling of light sources described by goniometric diagrams and for the representation of BRDF.

Note: *This is a preliminary version of the text that should appear in Eurographics Short Presentations 2003. For a complete version of the text and other details, please, contact the first author or/and access the Eurographics Digital Library:* http://www.eg.org/EG/DL.

1. Introduction

Surface parameterization hence *inverse mapping* from 3D to 2D and *forward mapping* from 2D to 3D is one of the key concepts in computer graphics. Mapping from an initial 2D rectangular domain $u,v \in \langle 0,1 \rangle^2$ to a sphere or hemisphere is used at least in two computer graphics applications: description of light scattering intensity for point light sources and simple area light sources and for description of bidirectional reflectance distribution functions (BRDF).

Let us recall traditional spherical mapping from 2D to 3D unit sphere surface that was used in computer graphics already in sixties and seventies:

$$\begin{aligned} x &= \sin\theta \cdot \cos\phi \\ y &= \sin\theta \cdot \sin\phi \\ z &= \cos\theta. \end{aligned} \tag{1}$$

The meaning of angles is depicted for the hemisphere in Figure1 (a). We further assume that the spherical mapping is anchored at center point O and oriented by normal $\vec{N}$. The mapping described by the formulas above is highly non-linear, i.e. the ratio of the fractional surface area in 2D domain and of the corresponding fractional surface area in 3D varies significantly. This property can be described mathematically using the determinant of the Jacobian matrix of the mapping function. However, further on we will use a less formal and more intuitive formulation of this property. Let us assume that the samples are uniformly distributed over a 2D domain, as it is common for 2D texture texels in the raster image or for a uniform random number generator over the 2D unit square. Further on we stick with the terminology used in probability and sampling: we write that the *probability density* of samples in a 2D domain is constant when the distribution of samples in 2D is uniform. After mapping from 2D to 3D using the formulas above, the density of samples on the 3D sphere surface is the highest at the poles ($\theta = 0$ and $\theta = \pi$), and the lowest along the equator ($\theta = \pi/2$).

A simple spherical mapping from 2D to 3D can be extended in order to prescribe the probability density at some *key-points* located on the sphere surface. *Goniometric diagrams* (also called photometric web diagrams[1, 2] in the photometry literature) were introduced in computer graphics by Verbeck and Greenberg[12]. The sphere surface is subdivided

by M meridians ($\phi = const$) and N parallels ($\theta = \text{const}$), see Figure 1 (b). At the key-points given by intersections of parallels and meridians the luminous intensity $[candela = lm/sr]$ of a luminaire or lamp (simplified as a point) is measured. The specification of the goniometric diagram then assumes that the luminous flux is bilinearly interpolated with respect to the θ and ϕ angles between the adjacent key-points on the goniometric diagram. Surface regions of the goniometric diagram given by the closest four key-points in both angle directions are called *goniometric patches*. The luminous intensity integrated over the sphere surface, i.e. over all goniometric patches, gives the total power emitted by the light source. The probability density used above is given by the normalized luminous intensity. We further use for sake of clarity only the term *probability density* or simply *density*.

Another common use of goniometric diagrams is for bidirectional reflectance distribution function (BRDF) defined over the hemisphere. The BRDF specifies for each incoming angle of an incident light ray direction the *transfer probability density* for all outgoing directions divided by the cosine of the incoming angle direction. The use of the BRDF in the rendering equation actually corresponds to the computation of the transfer probability density from BRDF. In practice, measured BRDF of real materials are represented by a set of goniometric diagram at several key-points. Thus we can represent by goniometric diagrams either BRDFs or directly transfer probability densities. Usually, the BRDF data measured on goniometric diagrams is approximated by some analytical formula or compressed/approximated using different schemes, for example wavelets[6].

In this paper, we present an approximative mapping from the 2D unit square to the 3D unit hemisphere. The density of the proposed mapping in the 2D domain is naturally described in the 3D domain by a goniometric diagram over the hemisphere. This is advantageous since the description of the density is given in the application domain. This mapping is bijective (unique), bicontinuous (preserving adjacency), and with a low distortion controlled by the goniometric diagram. The theoretical maximum error of the approximation at the worst case is 4.2%, but for typical goniometric diagrams less than 0.5%.

The mapping is given as a composition of four mappings. In brief, for the forward mapping we use a mapping from the unit square to a 2D disk originally described by Shirley and Chiu[10], advocated for ray tracing by Shirley[8]. In the next step we perform two algorithmic mappings on 2D disk to achieve the density prescribed by the goniometric diagram (the diagram is mapped from the hemisphere to a 2D disk as a preprocess). Finally, we map samples from the 2D disk to the 3D hemisphere using a mapping that preserves the density of samples.

An important property of our mapping algorithm is that it does not use numerical iteration; in other words, the computation of forward and inverse mapping is analytical and hence computed using a constant number of arithmetic operations. However, we have to apply binary search to locate the corresponding goniometric patch for a given input. In the worst case, the time complexity of search is $\mathcal{O}(logM + logN)$.

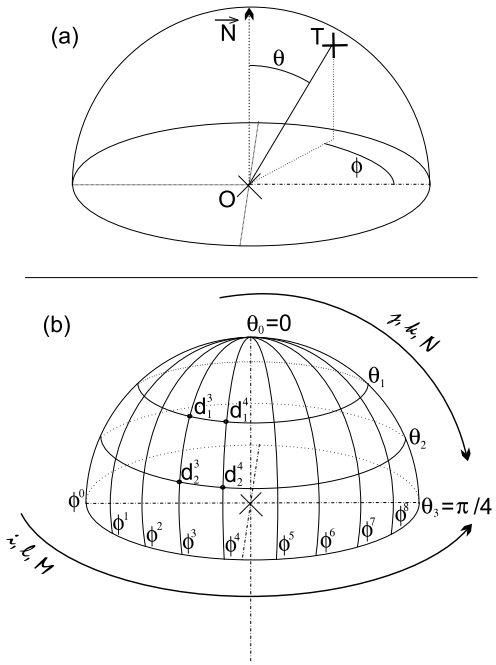

Figure 1: *(a) Hemisphere parameterization by two angles θ and ϕ. (b) Goniometric diagram - values d_1^3, d_1^4, d_2^3, and d_2^4 specify a probability density at corners of goniometric patch. Symbols i, l, and M are used in relation to ϕ, symbols j, k, and N are used in relation to θ in the paper.*

This article is structured as follows. In Section 2 we briefly overview the mapping algorithm from composed of four different mappings. In Section 3 we describe the four mappings in detail. In Section 4 we discuss the properties of the inverse mapping. In Section 5 we discuss briefly the inverse mapping, memory requirements, and real execution speed of the mapping algorithm. In Section 6 we outline the application of the proposed mapping algorithm to BRDF and sampling light sources described by goniometric diagram. Section 7 concludes the paper with prospects for future work.

2. Hemisphere Algorithm Overview

In this section we describe the forward mapping from a unit square to the surface of a hemisphere.The mapping is composed of four mappings, see Figure 2 for the mapping algorithm overview.

In a first step, the M_B mapping maps point (u, v) from the 2D unit square to the 2D unit disk using the method of concentric maps (Shirley and Chiu[10]). In a second step, the M_C

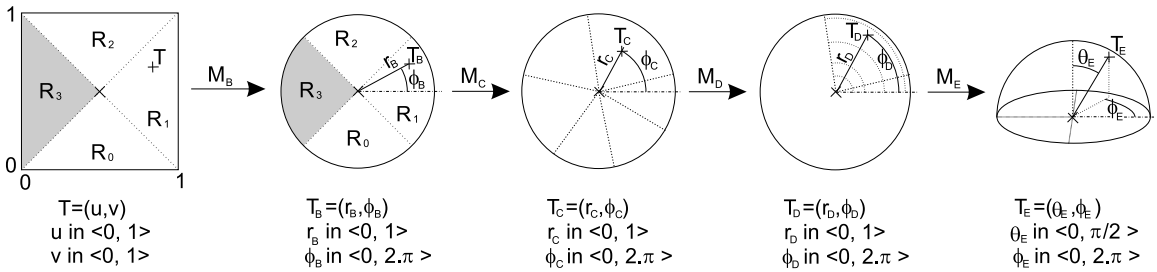

Figure 2: *Mapping Algorithm Overview.*

mapping changes the angle ϕ_B to ϕ_C and radius r_B to r_C in order to 'stretch' the uniform density on the disk to the density prescribed by the goniometric diagram when projected to a disk. In a third step, M_D changes the radius r_C to r_D in order to match the density to the value prescribed by bilinear interpolation in every goniometric patch projected from the hemisphere to the disk. In the last step, M_E maps the point coordinates from the 2D disk to the 3D hemisphere so that the density is preserved on the 3D hemisphere surface.

3. Mappings Description

In this section, we describe the mappings M_B, M_C, M_D, and M_E in the order that seems to us the most logical for understanding the whole mapping algorithm. We discuss the necessary properties of mapping, but formal and strict mathematical proofs are omitted in order to keep the paper easily readable.

Our mapping algorithm works in two stages. In *preprocessing*, the auxiliary arrays and variables are precomputed from the probability densities of the goniometric diagram. During *execution*, we use the precomputed values to compute the required forward and inverse mappings as fast as possible.

3.1. Mapping M_B

The first mapping from a unit square ($u,v \in \langle 0,1 \rangle^2$) to a unit disk (radius r and angle ϕ) is the easiest to describe. It has previously been published in detail by Shirley and Chiu[10, 11]. The authors have proved that it preserves adjacency and fractional area unlike the traditional polar mapping ($r = u$ and $\phi = 2 \cdot \pi \cdot v$). M_B also exhibits low distortion when mapping a shape from a unit square to a disk. Briefly, the principle of M_B mapping is outlined in Figure 2, left. The unit square is subdivided into four triangles and these triangles are separately mapped to the four adjacent sectors of a circle covering the unit disk. The density on a square is scaled by a factor of $1/\pi$ for the density on a disk that exactly corresponds to the ratio of the surface areas of unit square and unit disk. We have modified this algorithm in order to provide mapping to the range of $\phi \in \langle 0, 2 \cdot \pi \rangle$ unlike the original algorithm formulation given in paper[10] that maps to the range $\phi \in \langle -\pi/4, 7/4 \cdot \pi \rangle$.

3.2. Mapping M_E

This mapping is performed at the last stage. It maps the point from 2D unit disk to the surface of the hemisphere in 3D. This mapping has been described also by Shirley and Chiu[10] and originally by Shirley[9]. The mapping is as follows:

$$\phi_E = \phi_D \quad (2)$$

$$\theta_E = \arccos(1 - r_D^2) \quad (3)$$

It is relatively easy to prove that this mapping also keeps fractional area from 2D disk to 3D surface hemisphere with a factor of $1/2.0$. Unfortunately, this mapping cannot provide at the same time the linearity of θ_E with respect to the radius r_D. This linearity would be desirable when using goniometric diagram since the probability density on the sphere surface is linearly interpolated with respect to ϕ_E and θ_E angles. Our algorithm actually computes, during preprocessing, the radius r_D for prescribed θ angles during preprocessing using mapping M_E^{-1}. For these key θ-angles the density from 2D to 3D is mapped exactly. Given the angle θ ($\theta \in \langle \theta_1, \theta_2 \rangle$), the correct density $d_C(\theta)$ on surface of hemisphere is:

$$d_C(\theta) = d_1 + (\theta - \theta_1)/(\theta_2 - \theta_1) \cdot (d_2 - d_1) \quad (4)$$

However, mapping of θ to the radius r by Eq.3 gives us the density $d_R(\theta)$ that is linearly interpolated on the 2D disk:

$$\begin{aligned} \theta_1 &\mapsto r_1 = \sqrt{1 - \cos\theta_1} \\ \theta_2 &\mapsto r_2 = \sqrt{1 - \cos\theta_2} \\ d_R(\theta) &= d_1 + (d_2 - d_1) \cdot (\sqrt{1 - cos\theta})/(r_2 - r_1) \end{aligned} \quad (5)$$

The absolute error of density is then $e_A(\theta) = (d_R(\theta) - d_C(\theta))/Max(d_1, d_2)$, and the relative error is $e_R(\theta) = e_A(\theta)/d_C(\theta)$. The maximum absolute error is $e_A(\theta = 0.895354) = 4.2\%$ and the maximum relative error is $e_R(\theta = \pi/2) = 21.5\%$ for $N = 2$ parallels (pole and equator). For these errors the corresponding input is $\theta_1 = 0$, $d_1 = 1.0$, $\theta_2 = \pi/2$, and $d_2 = 0.0$.

The error of density mapping for M_E is due to the fact that the fractional area is kept unmodified from 2D to 3D. This is a required property for other stages of the mapping. For $N = 2$ the error of mapping is depicted in Figure 3. Such crude goniometric diagrams are not used in practice and typical errors for $N > 2$ are much lower.

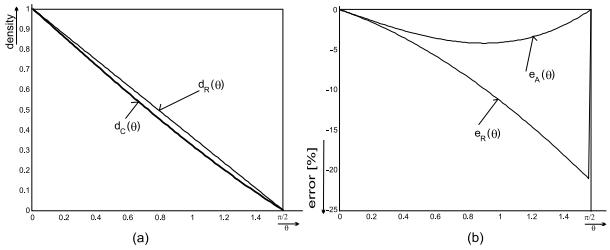

Figure 3: *Mapping M_E. (a) The correct and real density function with respect to angle θ. (b) The absolute and relative error [%] of density with respect to θ.*

In practice, the error due to M_E does not represent any problem for most applications: for a typical number $N = 10$ of equally placed parallels on the hemisphere, the maximum error of angle mapping is $e_A = 0.9\%$ and $e_R = 3.6\%$ which is negligible for most applications. Moreover the error can be easily compensated using a correction factor for the value of samples corresponding to the relative error e_R.

3.3. Mapping M_D

The mapping M_D changes only the radius on 2D disk in order to obtain the density prescribed by the goniometric diagram. For the ϕ angle, the mapping is trivial: $\phi_D = \phi_C$. For the radius, the forward mapping M_D requires a binary search and solving an integral equation on the fly.

In preprocessing, the θ angles describing parallels of the hemisphere are mapped (nearly projected) to the 2D disk, using the inverse mapping M_E^{-1}:

$$r_D = \sqrt{1-\theta_E^2} \tag{6}$$

We keep the density from the 3D hemisphere surface to the 2D disk thanks to the property of the mapping M_E, creating concentric circles on the disk for parallels on the hemisphere surface. Let us call a region on the hemisphere surface for which holds $\theta \in \langle 0, \pi/2 \rangle$ and $\phi \in \langle \phi_i, \phi_{i+1} \rangle$ an *hemispherical digon*. This hemi-spherical digon is mapped to the 2D disk as *a sector of the disk*.

M_D for prescribed ϕ_i

First, we show the mapping M_D for the discrete values ϕ_i prescribed by the goniometric diagram along the meridian i. An illustration of this mapping is given in Figure 4. Let us assume that for a given ϕ_i the result of the mapping M_C gives us the constant density s. The mapping M_D must preserve the integral of probability density s on the disk before M_D so that it is equal to the integral of the prescribed probability density d_j on the disk after M_D. For sake of clarity let us denote radii y_j (corresponding to r_C in Figure 2) for the constant density s for given parallel j. Further, let us denote prescribed radii x_j (corresponding to r_D in Figure 2) for the prescribed density d_j. Let us note that d_j corresponds to d_j^i

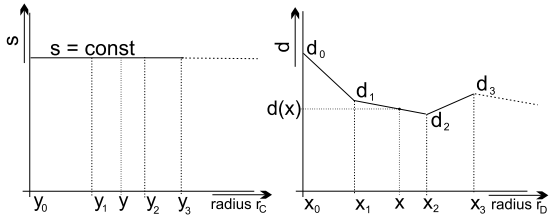

Figure 4: *Mapping M_D. (a) Constant density given as result of M_C. (b) Required density given by translation from the hemisphere to 2D disk.*

on the i-th meridian at the key-point (i, j) of the goniometric diagram. The density is linearly interpolated between two key-radii corresponding to two adjacent key-points on the goniometric diagram with respect to the radius:

$$d(x) = (d_{j+1} - d_j) \cdot \frac{x - x_j}{x_{j+1} - x_j} + d_j, \quad x \in \langle x_j, x_{j+1} \rangle \tag{7}$$

Then the mapping M_D from x_{j+1} to y_{j+1} for a particular y_{j+1} (thus θ_{j+1} on the boundary of a sector of the disk) verifies the following integral equation:

$$\int_{x=x_j}^{x_{j+1}} d(x) \cdot (2 \cdot \pi \cdot x) \cdot dx = \int_{y=y_j}^{y_{j+1}} s \cdot (2 \cdot \pi \cdot y) \cdot dy \tag{8}$$

Let us assume that x_j, x_{j+1}, and y_j are known. Solving the integral equation 8 gives us the formula for y_{j+1}:

$$\begin{aligned} (y_{j+1})^2 &= \frac{d_{j+1}}{s} \cdot (x_{j+1}^2 - x_j^2) + \\ &\quad \frac{d_j - d_{j+1}}{s \cdot (x_{j+1} - x_j)} \cdot (\frac{1}{3} x_{j+1}^3 - x_{j+1} \cdot x_j^2 + \frac{2}{3} \cdot x_j^3) + y_j^2 \end{aligned} \tag{9}$$

We use Eq. 9 in preprocessing to precompute the values of y_j for each j, $j \in \langle 0, N \rangle$, where N is the number of θ-angles for the hemisphere (number of polars). The initial condition is given at the center of the disk (pole of the hemisphere): $y_0 = x_0 = 0$.

For the forward mapping M_D the problem is similar. We want to get such X ($X \in \langle x_j, x_{j+1} \rangle$) for input Y ($Y \in \langle y_j, y_{j+1} \rangle$) for which holds the integral equation:

$$\int_{x=x_j}^{X} d(x) \cdot (2 \cdot \pi \cdot x) \cdot dx = \int_{y=y_j}^{Y} s \cdot (2 \cdot \pi \cdot y) \cdot dy \tag{10}$$

We derive that for linear interpolation of density $d(x)$ between two positions x_j and x_{j+1}, the problem is reformulated as a cubic equation:

$$\begin{aligned} X^3 + B_j . X^2 + C_J(Y) &= 0 \\ B_j &= -\frac{3}{2} \cdot (d_{j+1} \cdot \frac{x_{j+1} - x_j}{d_j - d_{j+1}} + x_{j+1}) \\ C_J(Y) &= \frac{3}{2} \cdot [(s \cdot (Y^2 - y_j^2) + (d_{j+1} \cdot x_j^2)) \cdot \frac{x_{j+1} - x_j}{d_j - d_{j+1}} \\ &\quad + x_{j+1} \cdot x_j^2 - \frac{2}{3} \cdot x_j^3] \end{aligned} \tag{11}$$

This cubic equation is solved analytically using the Cardan

formulas. In general the cubic equation has three roots. However, based on the knowledge of mutual relation between d_j and d_{j+1}, we can quickly compute one of the two real roots for which holds: $X \in \langle x_j, x_{j+1} \rangle$.

A particular case $d_j = d_{j+1}$ must be solved separately by the following equation:

$$X = \sqrt{\frac{s}{d_j} \cdot (Y^2 - y_j^2) + x_j^2}. \tag{12}$$

During the forward mapping M_D given an input $Y = r_c$, we first locate such index j for which holds $Y \in \langle y_j, y_{j+1} \rangle$. We use binary search which leads to a time complexity of $\mathcal{O}(\log N)$. A faster searching technique called *interpolation-binary search*[7], that achieves expected time complexity $\mathcal{O}(\log\log N)$ and worst time complexity $\mathcal{O}(\log N)$, can be used for higher values of N.

M_D for arbitrary ϕ

We have shown above the M_D mapping for some prescribed ϕ^i, $i \in \langle 0, M-1 \rangle$, where M is the number of meridians on the hemisphere surface. Hence M corresponds to the number of hemispherical digons and to the number of sectors of a disk. Below, we detail the mapping M_D for $\phi^i(\alpha) \in \langle \phi^i, \phi^{i+1} \rangle$, where $\phi(\alpha) = \phi^i \cdot (1-\alpha) + \phi^{i+1} \cdot \alpha$ and $\alpha \in \langle 0, 1 \rangle$.

Assuming $x_0 = y_0 = 0$ we rewrite formula 9 in a non-recursive way:

$$\begin{aligned} (y_j)^2 &= \frac{1}{s} \sum_{k=0}^{j-1} (d_{k+1}) \cdot (x_{k+1}^2 - x_k^2) + \\ &\quad \frac{d_k - d_{k+1}}{(x_{k+1} - x_k)} \cdot (\frac{1}{3} x_{k+1}^3 - x_{k+1} \cdot x_k^2 + \frac{2}{3} \cdot x_k^3) \end{aligned} \tag{13}$$

For the left boundary of a sector of the disk i (thus ϕ^i) the formula can be rewritten to:

$$(y_j^i)^2 = \frac{1}{s_i} \sum_{k=0}^{j-1} (d_{k+1}^i) \cdot F_k + (d_k^i - d_{k+1}^i) \cdot G_k, \tag{14}$$

where density d_k^i is prescribed by the goniometric diagram (see Figure 1 (b)) and

$$F_k = (x_{k+1}^2 - x_k^2) \tag{15}$$

$$G_k = \frac{1}{(x_{k+1} - x_k)} \cdot (\frac{1}{3} x_{k+1}^3 - x_{k+1} \cdot x_k^2 + \frac{2}{3} \cdot x_k^3) \tag{16}$$

For angle $\phi^i(\alpha)$ we can therefore derive the following formula using linear interpolation between the left boundary i and the right boundary $i+1$ of the disk sector:

$$\begin{aligned} (y_j^i(\alpha))^2 = \sum_{k=0}^{j-1} & F_k \cdot \quad (d_{k+1}^i \cdot \frac{1-\alpha}{s^i} + d_{k+1}^{i+1} \cdot \frac{\alpha}{s^{i+1}}) + \\ & G_k \cdot \quad ((d_k^i - d_{k+1}^i) \cdot \frac{1-\alpha}{s^i} + (d_k^{i+1} - d_{k+1}^{i+1}) \cdot \frac{\alpha}{s^{i+1}}) \end{aligned} \tag{17}$$

We can rewrite the formula above as:

$$(y_j(\alpha))^2 = (1-\alpha) \cdot \frac{1}{s^i} \cdot U(i,j) + \alpha \cdot \frac{1}{s^{i+1}} \cdot U(i+1,j), \tag{18}$$

where

$$U(i,j) = \sum_{k=0}^{j-1} F_k \cdot d_{k+1}^i + (d_k^i - d_{k+1}^i) \cdot G_k. \tag{19}$$

In the preprocessing we precompute values $\frac{1}{s^i} \cdot U(i,j)$ in an auxiliary array of size $(M \times N)$. Given $Y^i(\alpha)$ and $\phi^i(\alpha)$, and thus α and i, as a result of the M_C mapping described below, we perform a binary search for a given input value $(Y^i(\alpha))^2$ to get the index j for which holds:

$$(1-\alpha) \cdot \frac{1}{s^i} \cdot U(i,j) + \alpha \cdot \frac{1}{s^{i+1}} \cdot U(i+1,j) \le (Y^i(\alpha))^2 \quad \text{and}$$

$$(Y^i(\alpha))^2 \le (1-\alpha) \cdot \frac{1}{s^i} \cdot U(i,j+1) + \alpha \cdot \frac{1}{s^{i+1}} \cdot U(i+1,j+1)$$

The binary search also takes $O(\log N)$ steps in the worst case and always succeeds to find out the corresponding index j.

3.4. Mapping M_C

The mapping M_C also works exclusively on the disk and it computes both radius r_C from r_B and angle ϕ_C from ϕ_B. The input for M_C is the constant density for all points on the disk as the result of the mapping M_B. The need for the mapping M_C follows from description of M_D mapping. The probability density integrated over the boundary of a sector of the disk for a given ϕ^i can be different for each i. This integral corresponds to $\sqrt{U(i,N-1)}$ using the terminology above. Informally speaking, on the input of M_C we have the constant density s_{const} and by changing the ϕ angle we want to get the precomputed density s^i for a given i. There is one index $i_{\max}$ for which holds: $U(i_{\max}, N-1) \ge U(l, N-1), \quad l \in \langle 0, M-1 \rangle, \quad l \ne i_{\max}$. For this particular angle $\phi^{i_{\max}}$ we want to keep the differential of the angle ϕ around $\phi^{i_{\max}}$. For any other angle ϕ, we want to stretch ϕ in a non-linear way to get the prescribed density s^i for ϕ that is used as the input of M_D. The mapping M_C is schematically depicted in Figure 5.

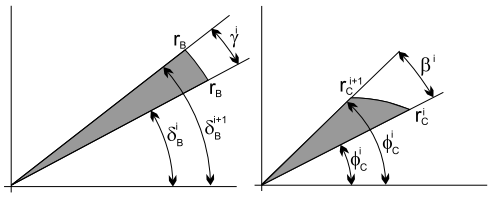

Figure 5: *Mapping M_C.*

Mathematically, we can formulate M_C for a whole sector of the disk i for the unknown variable γ^i and input $\beta^i = \phi^{i+1} - \phi^i$ as follows:

$$\int_{\phi=0}^{\beta^i} \int_{r=0}^{r_u} r \cdot d\phi \cdot dr = \int_{\phi=0}^{\gamma^i} \int_{r=0}^{\sqrt{\frac{1}{s^{i_{\max}}} \cdot U(i_{\max}, N-1)}} r \cdot d\phi \cdot dr, \tag{20}$$

where

$$r_u = \sqrt{(1 - \frac{\phi}{\beta^i}) \cdot \frac{1}{s^i} \cdot U(i, N-1) + \frac{\phi}{\beta^i} \cdot \frac{1}{s^{i+1}} \cdot U(i+1, N-1)}.$$

Note that on the left side of equation in the double integral we integrate over ϕ which is used at the same time as the upper boundary of the inner integral. Solving this integral equation we get:

$$\gamma^i = \frac{\beta^i \cdot s^{i_{\max}}}{U(i_{\max}, N-1)} \cdot \left[\frac{U(i,N-1)}{s^i} + \frac{1}{2} \cdot \left(\frac{U(i+1,N-1)}{s^{i+1}} - \frac{U(i,N-1)}{s^i}\right)\right] \quad (21)$$

Based on the known values β^i we precompute during the preprocessing M values γ^i. For the binary search we need to store in a δ-array the cumulative sum of γ^i, so $\delta^i = \sum_{l=0}^{i-1} \gamma^l$. Obviously, $\delta^0 = 0$.

During execution stage for forward mapping M_C, we get ϕ_B within the range $\langle 0, 2\cdot\pi\rangle$ as result of M_B. We compute $\phi'_B = \phi_B \cdot \delta^{M-1}/(2\cdot\pi)$. Then we perform binary search in the δ array to locate the index i for which holds:

$$\delta^i \le \phi'_B \quad \text{and} \quad \phi'_B \le \delta^{i+1} \quad (22)$$

After finding the index i and the values δ^i and δ^{i+1}, we compute $\gamma^i = \delta^{i+1} - \delta^i$ and $\gamma(\phi_B) = \phi'_B - \delta^i$. Our mapping problem is very similar to the Eq. 20 used in the preprocessing above, but we want to get the angle $\beta^i.\varepsilon$, where $\varepsilon \in \langle 0,1\rangle$ is the only unknown variable:

$$\int_{\phi=0}^{\beta^i\cdot\varepsilon} \int_{r=0}^{r_u} r \cdot d\phi \cdot dr = \int_{\phi=0}^{\gamma(\phi_B)} \int_{r=0}^{\sqrt{\frac{1}{s^{i_{\max}}} \cdot U(i_{\max}, N-1)}} \cdot r \cdot d\phi \cdot dr \quad (23)$$

Solving the integral equation leads to the following quadratic equation:

$$(\varepsilon)^2 \cdot \left[\frac{\beta^i}{2} \cdot \left(\frac{1}{s^{i+1}} \cdot U(i+1,N-1) - \frac{1}{s^i} \cdot U(i,N-1)\right)\right] + \varepsilon \cdot \left[\frac{\beta^i}{s^i} \cdot U(i,N-1)\right] - \gamma(\phi_B) \cdot \frac{1}{s^{i_{\max}}} \cdot U(i_{\max}, N-1) = 0 \quad (24)$$

A solution always exists such that $\varepsilon \in \langle 0,1\rangle$. So the mapping of angle ϕ_B to ϕ_C is then finalized:

$$\phi_C = \phi^i + \varepsilon \cdot (\phi^{i+1} - \phi^i) \quad (25)$$

Since we change the angle ϕ in a non-linear way, we compensate for this by changing the radius. We introduce an *effective density* $s^i(\varepsilon)$ as:

$$s^i(\varepsilon) = \frac{U(i_{\max}, N-1) \cdot s^{i_{\max}}}{(1-\varepsilon)\cdot U(i,N-1) + \varepsilon \cdot U(i+1,N-1)} \quad (26)$$

The mapping M_C for radius is finally:

$$r_C = \frac{r_B}{\sqrt{s^i(\varepsilon)}} \quad (27)$$

Notice that the whole mapping M_C is required to avoid "rejection of samples" in the "radius direction". After mapping M_C in M_D we use for all formulas $s^i = s_{\text{const}} = 1$ for all $i \in \langle 0, M\rangle$. Note that it holds: $r_C \ge r_B$ and $\gamma^i \le \beta^i$ ($\delta^{M-1} \le 2\cdot\pi$).

4. Inverse Mapping

The inverse mapping from the 3D hemisphere surface to the 2D unit square is nearly the inverse algorithm of the forward mapping described above. Mapping M_B^{-1} and M_E^{-1} are given by inverse algorithms. However in a chain of mappings for the hemisphere, mapping M_C^{-1} is performed directly after mapping M_E^{-1} and then is followed by mapping M_D^{-1}. This is necessary since the hemi-spherical digon i must be located first for both forward and inverse mapping. The formulae used in the integral equations for M_C and M_D are greatly simplified in the inverse mapping case: there is neither cubic nor quadratic equation. We need to compute two $\sqrt{}$ operations, otherwise we need only operations $+$, $-$, $\times$, $/$, and if.

5. Results

In this section we discuss memory requirements, verification of the implementation, and actual speed of mapping.

5.1. Memory Cost

In order to perform forward and inverse mapping we need to store the original densities d^i_j and auxiliary array $U(i,j)$, both arrays of size $M \times N$. Further we need an array of radii after mapping from the hemisphere to the 2D disk of size $M+1$ and the array γ of size $M+1$ for mapping M_C. We also need to store the array ϕ of size M that keeps the original ϕ angles for meridians. The original values of θ_j are not required. In total, for an hemisphere having M meridians and N parallels we need to store $2.M.N + 3.M + 2$ floating-point values.

5.2. Implementation Verification

We have implemented proposed forward and inverse mapping algorithm in ANSI C++. We have verified the correctness of forward mapping by comparison for application in global illumination for a goniometric light source for a few goniometric diagrams. The reference solution for forward mapping was provided by rejection sampling.

Note that the rejection sampling method generates the required probability distribution, but in general cannot be considered as unique forward mapping from 2D to 3D space since many generated samples are rejected. The second and major problem of rejection sampling is that it does not preserve adjacency of samples from 2D domain to 3D domain. The bicontinuity of mapping is a desirable property, particularly in ray tracing and global illumination algorithms. The samples taken after reflection can form coherent groups of rays that increases the performance of rendering algorithms. It is even required by some particular rendering algorithms[4].

The correctness of the inverse mapping has been checked with respect to the forward mapping algorithm. The error

of our method on the intensity has been always below our theoretical expectations of 4.2%, and typically less than 1%.

5.3. Algorithm Speed

On our rather non-optimized source code we have measured the speed of the mapping algorithm for $N = 8$, $M = 11$. The time complexity of the mapping algorithm depends only weakly on $N + M$ since the binary search is of logarithmic nature. Most of the computation time is thus spent in computations of formulas. We have measured the mapping speed of 6.7×10^6 samples. The forward mapping allows us to map 332×10^3 samples per second. The inverse mapping is according to our expectations faster and it maps $1,702 \times 10^3$ samples per second. The timings include also the time to generate sample using a 2D QMC Halton generation with base 2 and 3. For measurement we have used a PC with CPU Intel Pentium 4 at 2.6 GHz.

5.4. Mapping Visualization

We have performed a few visualizations of our mapping for different goniometric diagrams. The visualization for a 2D disk from hemisphere is shown in Figure 6 (before mapping M_E), where one patch intentionally has zero power.

6. Applications

In this section we discuss a couple of native applications of the new mapping described by goniometric diagram mapping for the hemisphere.

6.1. Description of Light Sources

Description of light emittance restricted to a hemisphere is a native application of goniometric diagram[1]. Using forward goniometric diagram mapping, we can easily perform importance sampling from a light source described by goniometric diagram without rejection sampling. The bicontinuous nature of our mapping is important if we want to keep the adjacency of samples from 2D to 3D. The samples in 2D can be generated by some MC or QMC random generator. The global illumination method where the use of proposed mapping is highly advantageous, was published by Dmitriev et al.[4]. Although goniometric diagram is in general described for a full sphere, many light sources in practice are described only by goniometric diagrams over hemisphere. Often, a goniometric diagram is assigned to linear or area light sources where each point on the surface of the light source has the same spatial distribution of emitted power. For such light sources the goniometric diagram described by hemisphere only is natural.

The error caused by non-linearity of angle with respect to radius is not perceivable for typical goniometric diagram. In case we want to obtain the exact solution, we correct the non-linearity by changing the photons energy by a correction factor so that the luminous energy corresponds completely to bilinear interpolation.

6.2. Representation of BRDF

The BRDF representation has been addressed in the past by many papers, most of them try to approximate the measured data by some analytical model. However, the most natural and exact is to use the original measured data obtained for the goniometric diagram. For details of BRDF measurement, see paper by Dana et al.[3]. For simplicity, let us assume an isotropic BRDF; value of BRDF is the same when rotating material along normal at given point of a surface. For every incident light direction described by angle with respect to surface normal we measure reflected light on goniometric diagram. For importance sampling of BRDF we compute the input angle θ_I, we search for two neighboring goniometric diagrams given by index n for angles $\theta_I[n]$ and $\theta_I[n+1]$. It holds $\theta_I[n] < \theta_I < \theta_I[n+1]$. The coefficient of linear interpolation is then $\alpha = \frac{\theta_I - \theta_I[n]}{\theta_I[n+1] - \theta_I[n]}$. Given a random value (u, v) on a unit square we simply generate two outgoing directions for the two goniometric diagrams and we interpolate the resulting outgoing direction using α.

7. Conclusions and Future Work

In this paper we have presented a new fast forward and inverse mapping algorithm for goniometric diagrams over the hemisphere with a time complexity $\mathcal{O}(\log M + \log N)$ for M meridians and N parallels. To our knowledge, it is the first algorithm of this kind. The mapping algorithm is numerically robust and all formulas in the algorithm are computed analytically, and as a consequence requires a constant number of arithmetic operations. We believe the proposed mapping algorithm is applicable to a wide range of problems in computer graphics, including rendering and modeling.

In our outgoing work we want to extend the mapping presented in this paper for a hemisphere to the whole sphere. This extension would allow its use in interesting applications such as texture mapping over weakly convex objects[5], specification of bidirectional scattering distribution function, and several other applications in global illumination and image based rendering.

Acknowledgments

We would like to thank Karol Myszkowski, Jaroslav Křivánek, and the anonymous reviewers for their comments on the preliminary version of this paper. The authors are grateful to Cyrille Damez for proofreading almost final version of the paper. This work was supported partly by the European Community within the scope of the RealReflect project IST-2001-34744 'Realtime visualization of complex reflectance behavior in virtual prototyping".

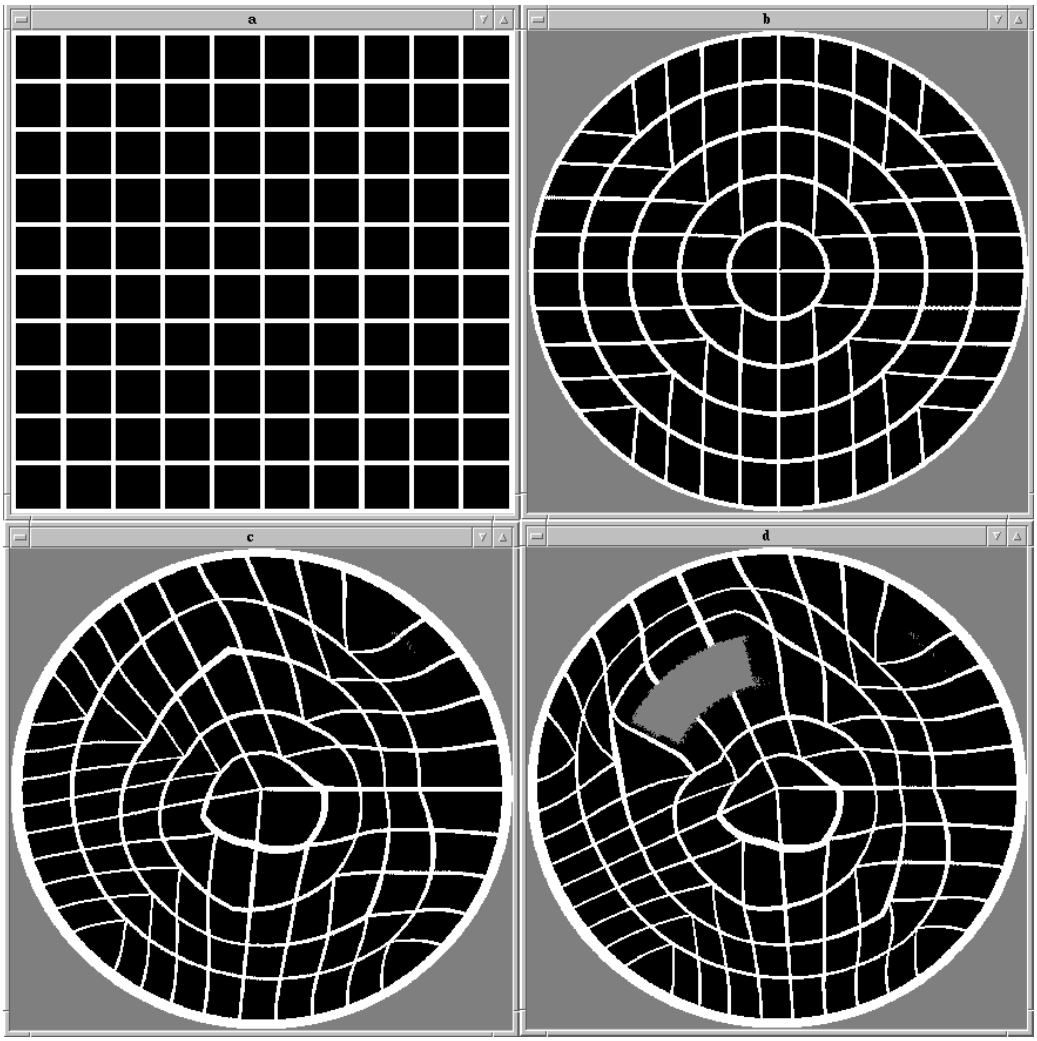

Figure 6: *Visualization for goniometric diagram from 2D unit square to a circle as a result of mapping M_D for $N = 6$ and $M = 7$. (a) The samples in the 2D square with the underlying grid for visualization purposes. (b) Result of M_D for the goniometric diagram with constant density. (c) Result of M_D for the goniometric diagram with density generated randomly. (d) Result for M_D for input density similar to (c), where one goniometric patch is redefined by four key points with zero density.*

References

1. ANSI/IES. *Nomenclature and Definitions for Illuminating Engineering.* ANSI/IES RP-16-96. New York, NY:Illuminating Engineering Society of North America, 1996. 1, 7
2. IESNA LM-75-01. *Goniophometer Types and Photometric Coordinates* New York, NY:Illuminating Engineering Society of North America, 2001. 1
3. K.J. Dana, B.V. Ginneken, S.K.Nayar, J.J.Koendrink Reflectance and Texture of Real-World Surfaces. *ACM Transaction on Graphics*, **18**(1):1–34, 1999. 7
4. K. Dmitriev, S. Brabec, K. Myszkowski, and H.-P. Seidel. Interactive Global Illumination Using Selective Photon Tracing. *Rendering Techniques 2002*, pp. 25–37, 2002. 6, 7
5. N. Green. Applications of world projections. In *Proceedings of Graphics Interface'86*, pp. 108–114, 1986. 7
6. P. Lalonde and A. Fournier. Generating Reflected Directions from BRDF Data. *Computer Graphics Forum (Eurographics'97 Proc.)*, **16**(3):293–300, 1997. 2
7. N. Santoro and J. Sidney Interpolation - Binary Search. *Inform. Process. Letters*, **20**(4):179–181, 1985. 5
8. P. Shirley. A Ray Tracing Method for Illumination Calculation in Diffuse-Specular Scenes. In Proceedings of *Graphics Interface'90*, pp. 205–212. May 1990. 2
9. P. Shirley. Nonuniform Random Point Sets via Warping. In *Graphics Gems III*, edited by D. Kirk, pp. 80–83. San Diego: Academic Press, 1992. 3
10. P. Shirley and K. Chiu. A Low Distortion Map Between Disk and Square. *Journal of Graphics Tools*,**2**(3):45–52, 1997. 2, 3
11. P. Shirley. *Realistic Ray Tracing*. A.K. Peters, 2000. 3
12. C.P. Verbeck, D. Greenberg. A Comprehensive Light-Source Description for Computer Graphics. *IEEE CG&A*, **4**(7):66–75, July 1984. 1