Parallel Monte Carlo Sampling Scheme for Sphere and Hemisphere*

I.T. Dimov[1], A.A. Penzov[2], and S.S. Stoilova[3]

[1] Institute for Parallel Processing, Bulgarian Academy of Sciences
Acad. G. Bonchev Str., bl. 25 A, 1113 Sofia, Bulgaria and ACET Centre, University of Reading Whiteknights, PO Box 217, Reading, RG6 6AH, UK
I.T.Dimov@reading.ac.uk

[2] Institute for Parallel Processing, Bulgarian Academy of Sciences,
Acad. G. Bonchev Str., bl. 25 A, 1113 Sofia, Bulgaria
apenzov@parallel.bas.bg

[3] Institute of Mathematics and Informatics, Bulgarian Academy of Sciences,
Acad. G. Bonchev Str., bl. 8, 1113 Sofia, Bulgaria
stoilova@math.bas.bg

Abstract. The sampling of certain solid angle is a fundamental operation in realistic image synthesis, where the rendering equation describing the light propagation in closed domains is solved. Monte Carlo methods for solving the rendering equation use sampling of the solid angle subtended by unit hemisphere or unit sphere in order to perform the numerical integration of the rendering equation.

In this work we consider the problem for generation of uniformly distributed random samples over hemisphere and sphere. Our aim is to construct and study the parallel sampling scheme for hemisphere and sphere. First we apply the symmetry property for partitioning of hemisphere and sphere. The domain of solid angle subtended by a hemisphere is divided into a number of equal sub-domains. Each sub-domain represents solid angle subtended by orthogonal spherical triangle with fixed vertices and computable parameters. Then we introduce two new algorithms for sampling of orthogonal spherical triangles.

Both algorithms are based on a transformation of the unit square. Similarly to the Arvo's algorithm for sampling of arbitrary spherical triangle the suggested algorithms accommodate the stratified sampling. We derive the necessary transformations for the algorithms. The first sampling algorithm generates a sample by mapping of the unit square onto orthogonal spherical triangle. The second algorithm directly compute the unit radius vector of a sampling point inside to the orthogonal spherical triangle. The sampling of total hemisphere and sphere is performed in parallel for all sub-domains simultaneously by using the symmetry property of partitioning. The applicability of the corresponding parallel sampling scheme for Monte Carlo and Quasi-Monte Carlo solving of rendering equation is discussed.

* Supported by the Ministry of Education and Science of Bulgaria under Grand No. I1405/04 and by FP6 INCO Grand 016639/2005 Project BIS-21++.

T. Boyanov et al. (Eds.): NMA 2006, LNCS 4310, pp. 148–155, 2007.

1 Introduction

The main task in the area of computer graphics is realistic image synthesis. For creation of photorealistic images the solution of a Fredholm type integral equation must be found. This integral equation is called rendering equation and it is formulate first by Kajiya in [3]. The rendering equation describes the light propagation in closed domains called frequently scenes (see Fig. 1).

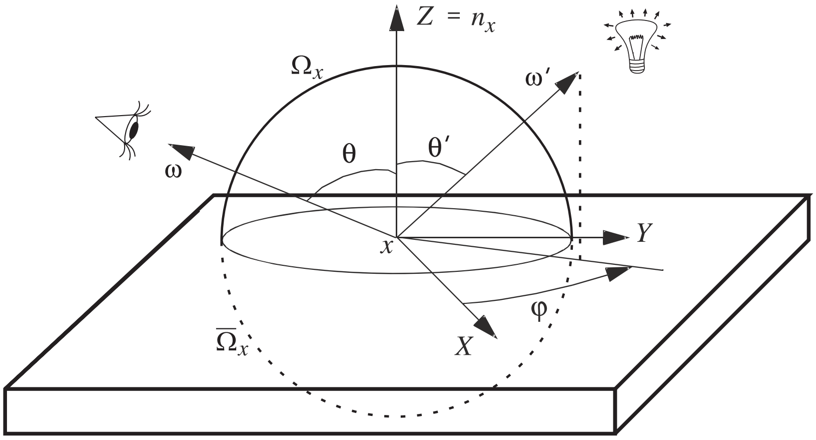

Fig. 1. The geometry for the rendering equation

The radiance L, leaving from a point x on the surface of the scene in direction $\omega \in \Omega_x$, where Ω_x is the hemisphere in point x, is the sum of the self radiating light source radiance L^e and all reflected radiance:

$$L(x,\omega) = L^e(x,\omega) + \int\limits_{\Omega_x} L(h(x,\omega'), -\omega') f_r(-\omega', x, \omega) \cos\theta' d\omega'.$$

Here $h(x,\omega')$ is the first point that is hit when shooting a ray from x into direction ω'. The radiance L^e has non-zero value if the considered point x is a point from solid light source. Therefore, the reflected radiance in direction ω is an integral of the radiance incoming from all points, which can be seen through the hemisphere Ω_x in point x attenuated by the surface BRDF (Bidirectional Reflectance Distribution Function) $f_r(-\omega', x, \omega)$ and the projection $\cos\theta'$. The angle θ' is the angle between surface normal in x and the direction ω'. The law for energy conservation holds, i.e.: $\int_{\Omega_x} f_r(-\omega', x, \omega) \cos\theta' d\omega' < 1$, because a real scene always reflects less light than it receives from the light sources due to light absorption of the objects.

When the point x is a point from a transparent object the transmitted light component must be added to the rendering equation. This component estimates the total light transmitted trough the object and incoming to the point x from all directions opposite to the hemisphere Ω_x. The transmitted light in direction

ω is an integral similar to the the reflected radiance integral where the domain of integration is the hemisphere $\overline{\Omega}_x$ in point x and BRDF is substituted by the surface BTDF (Bidirectional Transmittance Distribution Function) [2]. In this case the integration domain for solving the rendering equation is a sphere $\Omega^{(x)}$ in point x, where $\Omega^{(x)} = \Omega_x \bigcup \overline{\Omega}_x$.

Applying Monte Carlo methods for solving the rendering equation, we must sample the solid angle subtended by unit hemisphere or unit sphere in order to perform the numerical integration of the rendering equation. The sampling of certain solid angle is a fundamental operation in realistic image synthesis, which requires generating directions over the solid angle. To generate sampling directions for numerical integration of the rendering equation it is enough to generate points over unit hemisphere or unit sphere.

A good survey of different sampling algorithms for unit sphere and unit hemisphere is given by Philip Dutré in [2]. Some of them generate the sampling points directly over the hemisphere and sphere. Others first find points uniformly on the main disk, and then project them on the hemisphere or the sphere. Deterministic sampling methods for spheres are proposed in [7] and applied in robotics, where the regularity of Platonic solids is exploited. Arvo [1] suggests a sampling algorithm for arbitrary spherical triangle and Urena [6] shows an adaptive sampling method for spherical triangles.

Further in this paper we consider the parallel samples generation over sphere and hemisphere for Monte Carlo solving of rendering equation.

2 Partitioning of Sphere and Hemisphere

Consider hemisphere and sphere with center in the origin of a Descartes coordinate system. Similarly to the Bresenham algorithm [5] for raster display of circle we apply the symmetry property for partitioning of hemisphere and sphere. It is obvious that the coordinate planes partition the hemisphere into 4 equal areas and the sphere into 8 equal areas. The partitioning of each one area into sub-domains can be continued by the three bisector planes. One can see that the bisector planes to the dihedral angles $(\overrightarrow{X}, \overrightarrow{Y})$, $(\overrightarrow{X}, \overrightarrow{Z})$ and $(\overrightarrow{Z}, \overrightarrow{Y})$, partition each area into 6 equal sub-domains. In Fig. 2 we show the partitioning of the area with positive coordinate values of X, Y and Z into 6 equal sub-domains.

As described above we can partition the hemisphere into 24 and respectively the sphere into 48 equal sub-domains. Something more, due to the planes of partitioning each sub-domain is symmetric to all others. The symmetric property allows us to calculate in parallel the coordinates of the symmetric points. For example to calculate the coordinates of the point P_3 we consecutively multiply the coordinates of the point $P_0(x_0, y_0, z_0)$ by two matrix of symmetry:

$$P_0(x_0, y_0, z_0) * \begin{pmatrix} 0\,0\,1 \\ 0\,1\,0 \\ 1\,0\,0 \end{pmatrix} * \begin{pmatrix} 1\,0\,0 \\ 0\,0\,1 \\ 0\,1\,0 \end{pmatrix} = P_2(z_0, y_0, x_0) * \begin{pmatrix} 1\,0\,0 \\ 0\,0\,1 \\ 0\,1\,0 \end{pmatrix} = P_3(z_0, x_0, y_0).$$

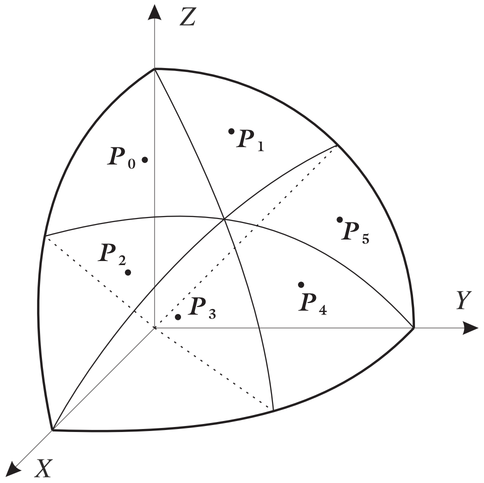

Fig. 2. Partition of a spherical area into 6 sub-domains

Table 1. Parallel Sample Coordinates Calculation of Hemisphere

$P_0(x_0, y_0, z_0)$	$P_0'(-x_0, y_0, z_0)$	$P_0''(-x_0, -y_0, z_0)$	$P_0'''(x_0, -y_0, z_0)$
$P_1(y_0, x_0, z_0)$	$P_1'(-y_0, x_0, z_0)$	$P_1''(-y_0, -x_0, z_0)$	$P_1'''(y_0, -x_0, z_0)$
$P_2(z_0, y_0, x_0)$	$P_2'(-z_0, y_0, x_0)$	$P_2''(-z_0, -y_0, x_0)$	$P_2'''(z_0, -y_0, x_0)$
$P_3(z_0, x_0, y_0)$	$P_3'(-z_0, x_0, y_0)$	$P_3''(-z_0, -x_0, y_0)$	$P_3'''(z_0, -x_0, y_0)$
$P_4(x_0, z_0, y_0)$	$P_4'(-x_0, z_0, y_0)$	$P_4''(-x_0, -z_0, y_0)$	$P_4'''(x_0, -z_0, y_0)$
$P_5(y_0, z_0, x_0)$	$P_5'(-y_0, z_0, x_0)$	$P_5''(-y_0, -z_0, x_0)$	$P_5'''(y_0, -z_0, x_0)$

Table 2. Parallel Sample Coordinates Calculation of Sphere

$P_0(x_0, y_0, z_0)$	$P_0'(-x_0, y_0, z_0)$	$P_0''(-x_0, -y_0, z_0)$	$P_0'''(x_0, -y_0, z_0)$
$P_1(y_0, x_0, z_0)$	$P_1'(-y_0, x_0, z_0)$	$P_1''(-y_0, -x_0, z_0)$	$P_1'''(y_0, -x_0, z_0)$
$P_2(z_0, y_0, x_0)$	$P_2'(-z_0, y_0, x_0)$	$P_2''(-z_0, -y_0, x_0)$	$P_2'''(z_0, -y_0, x_0)$
$P_3(z_0, x_0, y_0)$	$P_3'(-z_0, x_0, y_0)$	$P_3''(-z_0, -x_0, y_0)$	$P_3'''(z_0, -x_0, y_0)$
$P_4(x_0, z_0, y_0)$	$P_4'(-x_0, z_0, y_0)$	$P_4''(-x_0, -z_0, y_0)$	$P_4'''(x_0, -z_0, y_0)$
$P_5(y_0, z_0, x_0)$	$P_5'(-y_0, z_0, x_0)$	$P_5''(-y_0, -z_0, x_0)$	$P_5'''(y_0, -z_0, x_0)$
$\overline{P}_0(x_0, y_0, -z_0)$	$\overline{P}_0'(-x_0, y_0, -z_0)$	$\overline{P}_0''(-x_0, -y_0, -z_0)$	$\overline{P}_0'''(x_0, -y_0, -z_0)$
$\overline{P}_1(y_0, x_0, -z_0)$	$\overline{P}_1'(-y_0, x_0, -z_0)$	$\overline{P}_1''(-y_0, -x_0, -z_0)$	$\overline{P}_1'''(y_0, -x_0, -z_0)$
$\overline{P}_2(z_0, y_0, -x_0)$	$\overline{P}_2'(-z_0, y_0, -x_0)$	$\overline{P}_2''(-z_0, -y_0, -x_0)$	$\overline{P}_2'''(z_0, -y_0, -x_0)$
$\overline{P}_3(z_0, x_0, -y_0)$	$\overline{P}_3'(-z_0, x_0, -y_0)$	$\overline{P}_3''(-z_0, -x_0, -y_0)$	$\overline{P}_3'''(z_0, -x_0, -y_0)$
$\overline{P}_4(x_0, z_0, -y_0)$	$\overline{P}_4'(-x_0, z_0, -y_0)$	$\overline{P}_4''(-x_0, -z_0, -y_0)$	$\overline{P}_4'''(x_0, -z_0, -y_0)$
$\overline{P}_5(y_0, z_0, -x_0)$	$\overline{P}_5'(-y_0, z_0, -x_0)$	$\overline{P}_5''(-y_0, -z_0, -x_0)$	$\overline{P}_5'''(y_0, -z_0, -x_0)$

Therefore, sampling the hemisphere and calculating the coordinates of a sampling point $P_0(x_0, y_0, z_0)$ from a given sub-domain, we can calculate in parallel the other sampling point coordinates for the hemisphere in accordance to the Table 1.

The coordinate of symmetric points when we sampling the sphere can be calculated in a same way. The Table 2. represents the parallel coordinates calculations for the sphere. Note that the marked with $\overline{P}^{(\cdot)}_{(\cdot)}$ points in Table 2. are the same as the respective points presented in Table 1. and only differ in negative sign of the Z coordinate.

This kind of partitioning allows to sample only one sub-domain and to calculate in parallel all other samples for the hemisphere or sphere. Since the symmetry is identity the generation of uniformly distributed random samples in a sub-domain leads to the uniform distribution of all samples in the hemisphere and sphere.

3 Algorithms for Parallel Sampling Scheme

In this section we consider the problem for sampling a sub-domain in the terms of hemisphere and sphere partitioning, described in the previous section. Each sub-domain represents solid angle subtended by orthogonal spherical triangle with fixed vertices and computable parameters. In order to generate uniformly distributed random samples of the sub-domain we propose two algorithms.

3.1 Algorithm 1

This algorithm is very similar to the Arvo's [1] algorithm for sampling of arbitrary spherical triangle. Let us consider the solid angle subtended by the spherical triangle $\triangle ABC$ shown in Fig. 3.

Due to the partitioning planes, we can observe, that the arcs $\widehat{AB}$, $\widehat{BC}$ and $\widehat{AC}$ are the arcs of the main (central) circles. The angle γ is equal to $\frac{\pi}{2}$ and the length of the arc $\widehat{BC}$ is $\frac{\pi}{4}$. One can write the following identities:

$$\cos \widehat{AB} = \cos \widehat{BC} \cdot \cos \widehat{AC} + \sin \widehat{BC} \cdot \sin \widehat{AC} \cdot \cos \gamma$$

$$\cos \beta = -\cos \gamma \cdot \cos \alpha + \sin \gamma \cdot \sin \alpha \cdot \cos \widehat{AC}$$

$$\frac{\sin \widehat{BC}}{\sin \alpha} = \frac{\sin \widehat{AB}}{\sin \gamma}.$$

The first two expressions are spherical cosine law for the arc $\widehat{AB}$ and for the angle β, as well the third is spherical sine law. By definition the angle β is the angle between tangents to the spherical arcs in the point B which is equal to the dihedral angle between the partitioning planes. Therefore, the angle β is

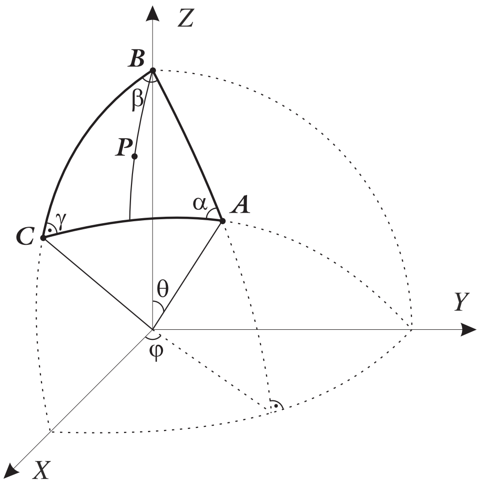

Fig. 3. Sampling point generation by Algorithm 1

always equal to the angle φ. Since the arc $\widehat{AB}$ is arc from the central circle in the orthogonal spherical triangle, it equals to θ, we can write the following equality:

$$\tan\widehat{AB} = \frac{1}{\cos\beta} \Rightarrow \tan\theta = \frac{1}{\cos\varphi}.$$

Similarly to the strategy presented in [1] we attempt to generate a sample by mapping of the unit square onto orthogonal spherical triangle. With other words we seek a bijection $F(u,v) : [0,1]^2 \to \triangle ABC$, where u and v are random variables uniformly distributed in $[0,1]$. Now we introduce the following transformation:

$$\varphi = \frac{u\pi}{4} \quad \text{and} \quad \theta = \arctan\frac{v}{\cos\varphi} = \arctan\frac{v}{\cos\frac{u\pi}{4}},$$

where $u, v \in [0,1]$; $\varphi \in [0, \frac{\pi}{4}]$ and $\theta \in [0, \arctan\frac{1}{\cos\varphi}]$. The algorithm can be described as:

```
▷ Generate Random Variables:( real $u$ , real $v$)
▷ Calculate angles : $\varphi = \frac{u\pi}{4}$    and    $\theta = \arctan\frac{v}{\cos\frac{u\pi}{4}}$
▷ Calculate the sampling point coordinates:
  $P_x = \cos\varphi \cdot \sin\theta$,    $P_y = \sin\varphi \cdot \sin\theta$    *and*    $P_z = \cos\theta$
▷ Return Sampling Point:    $P(P_x, P_y, P_z)$.
```

3.2 Algorithm 2

This algorithm tries to compute directly the unit radius vector of a sampling point inside to the orthogonal spherical triangle. Consider a point P inside

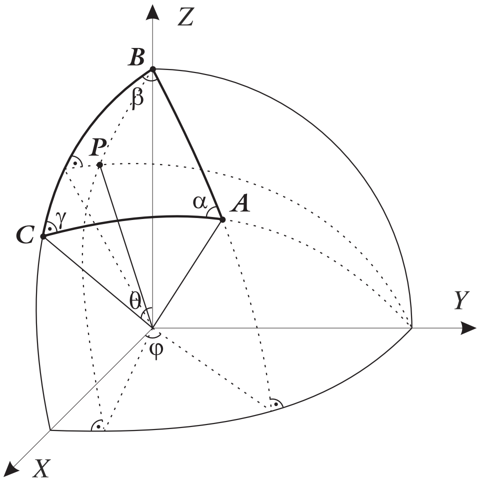

Fig. 4. Sampling point generation by Algorithm 2

for the spherical triangle $\triangle ABC$ shown in Fig. 4. The coordinates of an arbitrary sampling point P could be calculated by finding the intersection point of spherical triangle $\triangle ABC$ with the two coordinate planes having normal vectors $N_x(1,0,0)$ and $N_y(0,1,0)$, and rotated respectively on angle φ and angle θ, where $\theta, \varphi \in [0, \frac{\pi}{4}]$. It is clear that when $\theta = \varphi = \frac{\pi}{4}$, the point $P \equiv A$ and when $\theta = \varphi = 0$, the point $P \equiv B$.

The rotations of the coordinate planes are defined by the matrices of rotation. Let, $R_{y\theta}$ is the matrix of rotation on angle θ around the axis Y and $R_{z\varphi}$ is the matrix of rotation on angle θ around the axis Z:

$$R_{y\theta} = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \qquad \textit{and} \qquad R_{z\varphi} = \begin{pmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Applying the rotations to the normal vectors $N_x(1,0,0)$ and $N_y(0,1,0)$ we calculate the normal vectors N'_x and N'_y to the rotated coordinate planes:

$$N'_x = N_x R_{y\theta} = (\cos\theta, 0, -\sin\theta) \qquad \textit{and} \qquad N'_y = N_y R_{z\varphi} = (-\sin\varphi, \cos\varphi, 0).$$

The normalized vector product of the vectors N'_x and N'_y compute the coordinate of point P as:

$$P = \frac{N'_x \times N'_y}{\left|N'_x \times N'_y\right|} \qquad \textit{with} \qquad |P| = 1$$

where $N'_x \times N'_y = (\sin\theta\cos\varphi, \sin\theta\sin\varphi, \cos\theta\cos\varphi)$ and

$$\left|N'_x \times N'_y\right| = \sqrt{\sin^2\theta + \cos^2\theta\cos^2\varphi} = \sqrt{\cos^2\varphi + \sin^2\theta\sin^2\varphi}.$$

The algorithm can be described as:

```
▷ Generate Random Variables:( real $u$ , real $v$)
▷ Calculate angles : $\varphi = \frac{u\pi}{4}$ and $\theta = \frac{v\pi}{4}$
▷ Calculate the sampling point coordinates:
  $P_x = \frac{\sin\theta \cdot \cos\varphi}{\sqrt{\sin^2\theta + \cos^2\theta\cos^2\varphi}}$, $P_y = \frac{\sin\theta \cdot \sin\varphi}{\sqrt{\sin^2\theta + \cos^2\theta\cos^2\varphi}}$
  and $P_z = \frac{\cos\theta \cdot \cos\varphi}{\sqrt{\sin^2\theta + \cos^2\theta\cos^2\varphi}}$
▷ Return Sampling Point: $P(P_x, P_y, P_z)$.
```

4 Conclusion

The presented parallel sampling scheme for Monte Carlo solving of the rendering equation uses partitioning of the hemisphere or sphere by a natural way. The sphere or hemisphere is divided into equal sub-domains of orthogonal spherical triangles by applying the symmetry property. The advantages of this approach lie in the parallel computations. Sampling only one sub-domain, the sampling points over hemisphere or sphere are calculated in parallel. This approach is suitable for realization over parallel (MIMD, multiple instruction - multiple data) architectures and implementation on Grid infrastructures. The proposed algorithms for sampling of orthogonal spherical triangles accommodate the stratified sampling. Instead of using random variables u and v for sampling point generation we can apply low discrepancy sequences as shown in [4]. This fact leads to parallel Quasi-Monte Carlo approach for solving of the rendering equation, which is a subject of future study and research.

References

1. Arvo, James, Stratifed sampling of spherical triangles. In: Computer Graphics Proceedings, Annual Conference Series, ACM Siggraph, pp. 437–438, (1995).
2. Dutré, Philip, Global Illumination Compendium, Script of September 29 2003, http://www.cs.kuleuven.ac.be/∼phil/GI/TotalCompendium.pdf
3. Kajiya, J. T., The Rendering Equation, Computer Graphics, **vol. 20**, No. 4, pp. 143–150, Proceedings of SIGGRAPH‘86, (1986).
4. Keller, Alexander, Quasi-Monte Carlo Methods in Computer Graphics: The Global Illumination Problem, Lectures in Applied Mathematics, **vol. 32**, pp. 455–469, (1996).
5. Rogers, D. F., Procedural Elements for Computer Graphics. McGraw-Hill Inc, (1985).
6. Urena, Carlos, Computation of Irradiance from Triangles by Adaptive Sampling, Computer Graphics Forum, **Vol. 19, No. 2**, pp. 165–171, (2000).
7. Yershova, Anna and LaValle, Steven M., Deterministic Sampling Methods for Spheres and $SO(3)$, Robotics and Automation, 2004. IEEE Proceedings of ICRA’04, **vol.4**, pp. 3974–3980, (2004).