

Low Discrepancy 数列のコンピュータグラフィックスへの適用

土井 章 男[†] 山口 美 春^{†,☆} 千葉 則 茂[†]

本論文では、準モンテカルロ法で用いられている low discrepancy 数列をコンピュータグラフィックス (CG) における 2 次元および 3 次元サンプリング問題に適用し、その考察を行った。low discrepancy 数列としては、Sobol 数列、GFSR 数列、Good Lattice Points, Hammersley Points を取り上げ、計算時間、ディスクレパンシー (discrepancy)、RMS 誤差を用いて、各数列の有利な点や discrepancy による評価の問題点を明らかにした。また、ラジオシティ法などで用いられる 3 次元サンプリングにおいて、low discrepancy 数列を適用したサンプリング法を提案し、従来手法である乱数を用いた方法に比べて、最大 47 パーセント以上の速度向上を可能にした。

Low Discrepancy Sequence for Computer Graphics Applications

AKIO DOI,[†] MIHARU YAMAGUCHI^{†,☆} and NORISHIGE CHIBA[†]

In this paper, we applied low discrepancy sequences to the field of 2D and 3D sampling problems of image synthesis of Computer Graphics (CG). As low discrepancy sequences, we selected Sobol's sequences, GFSR sequences, Good Lattice Points, and Hammersley Points. We evaluated these sequences in computational time, discrepancy, and RMS error, and show the effectiveness and the problems of low discrepancy sequences. We also proposed a 3D sampling method based on low discrepancy sequences, and improve the performance more than 47 percents in comparison with the previous method.

1. はじめに

複雑な数値積分を解く手法としてモンテカルロ (Monte-Carlo) 法が有名であり、その中では乱数が重要な役割を担っている。この乱数を乱数よりも一様に分布する数列 (low discrepancy 数列) に置き換えて、モンテカルロ法を高速にした手法が準モンテカルロ (Quasi-Monte-Carlo) 法である。ディスクレパンシー (discrepancy) とは、乱数の一様な分布を表す 1 つの尺度であり、その値が小さいほどその乱数は均一であるといえる。Discrepancy の小さな数列を low discrepancy 数列と呼び、代表的なものとして、Halton 数列^{7),13),18)}, Sobol 数列¹⁵⁾, GFSR 数列^{11),19)}, Good Lattice Points¹²⁾, Hammersley Points^{8),12)} などがある。

コンピュータグラフィックス (以後、CG と略す) の分野では、一般に low discrepancy 数列よりも乱数が用いられている。2 次元空間でのサンプリングでは、

画像生成時のエイリアスを防ぐために、jittered sampling^{5),6)} が使用される。3 次元空間でのサンプリングでは、jittered sampling の拡張として、stratified sampling が提案されている¹⁾。どちらの方法もサンプリングする領域内 (stratified sampling の場合は球面もしくは球面三角形) でサンプル数が等しくなるように決定される。

CG における low discrepancy 数列の適用としては、Niederreiter が理論的にディスクレパンシーの小さな Good Lattice Points と Hammersley Points を提案し、これらの数列が CG に有効であると示唆した¹²⁾。しかしながら、Niederreiter は実際の画像生成には適用しておらず、また、誤差評価も行っていない。最近では、画素のサンプリング⁹⁾、大域照明問題¹⁰⁾、面光源の輝度計算¹⁴⁾ に low discrepancy 数列を適用した例があるが、どの low discrepancy 数列が有効であるかは明確ではない。

本論文では、代表的な CG アプリケーションへの適用に際して、最も適した low discrepancy 数列を発見し、理論的にディスクレパンシーの小さい数列が必ずしも CG アプリケーションに有効でないことを実験的に示す。また、low discrepancy 数列を用いた 3 次元

[†] 岩手大学工学部

Faculty of Engineering, Iwate University

[☆] 現在、株式会社アルファ・システムズ

Presently with Alpha Systems

サンプリング法を提案し、本手法が従来の乱数を用いた方法に比べて、高速かつ良好なサンプリング特性を持っていることを示す。

本文の構成は、2章で画像生成におけるエリアシングと従来の乱数を用いたサンプリング手法、そしてディスクレパンシイについて説明する。3章では、Sobol 数列、GFSR 数列、Good Lattice Points, Hammer-sley Points の概略について述べる。4章では、各 low discrepancy 数列と CG で最もよく使用されている jittered sampling⁶⁾を比較検討し、計算時間、ディスクレパンシイ、精度 (RMS 誤差) の3点から2次元サンプリング結果を評価する。5章では、low discrepancy 数列を用いた適応的3次元サンプリング法とルックアップテーブル方式による low discrepancy 数列生成法について述べる。6章は結論である。

2. 画像生成におけるエリアシングとディスクレパンシイ

2.1 画像生成におけるエリアシング

画像生成においてエリアシングが発生するのは、直線、多角形、色、輝度、時間などが連続的であるのに対して、その画像生成結果がサンプリング不足のため離散的になるためである。これを防ぐ方法 (アンチエイリアシング) として、よく使用されているのがスーパーサンプリングである⁴⁾。スーパーサンプリングでは、サンプリング領域に対してサンプリング数を増加させ、各サンプルの平均、もしくは重み付けした平均により計算する。

画素領域に対する従来のサンプリング手法としては、N-rooks sampling, poisson disk sampling, jittered sampling 等がある¹⁶⁾。これらの方法に共通することは、均一にサンプル点を分布させるため、サンプル点の位置制約を付けている点と乱数を用いている点である。N-rooks sampling では、あらかじめ区切られた区画で、各サンプル点がお互いの行および列の区画に入らないよう制約している。Poisson disk sampling では、各サンプル点間の距離がある一定値以上になることを保証するため、一定値以上になるまでサンプリングを繰り返す。そのため、サンプリング点が増加すると計算効率が悪い。Jittered sampling では、まず、画素領域をサンプル数で区画化し、各区画内の位置は乱数により決定する。そのため、各区画内に必ずサンプル点が存在する。Jittered sampling は、上記サンプリング法の中で最も効率が良く、一般に使用されている。Jittered sampling の N が 16, 64 の場合の分布を図1に示す。

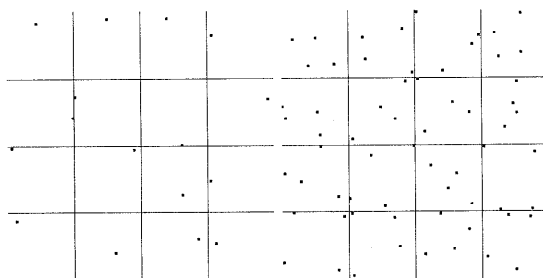


図1 Jittered sampling ($N = 16, 64$)

Fig. 1 Jittered sampling ($N = 16, 64$).

2.2 ディスクレパンシイ

S 次元空間の単位立方体 $I^s = [0, 1]^s$ に分布する点の集合 $x^1, x^2, \dots, x^N$ を評価する1つの尺度がディスクレパンシイである。

$x^1, x^2, \dots, x^N \in I^s$ を満たす点の集合と部分集合 $G \subset I^s$ が与えられたとき、 $x^i \in G$ を満たす点の数を数える関数 $S_N(G)$ を用意する。次に任意の点 $x = (x_1, x_2, \dots, x_s) \in I^s$ に対して、式 (1) で区切られる s 次元空間内の領域を G_x で表す。

$$G_x = [0, x_1] \times [0, x_2] \times \dots \times [0, x_s] \quad (1)$$

このとき、ディスクレパンシイは以下の式で定義される²⁾。

$$D^*(x^1, x^2, \dots, x^N) = \sup |S_N(G_x) - Nx_1x_2 \dots x_s|. \quad (2)$$

本ディスクレパンシイの定義では、領域 G_x が各軸に垂直な区画になるため、関数 $S_N(G)$ の計算は容易である。Dobkin ら²⁰⁾は任意軸の区画を与えるディスクレパンシイを定義し、その計算方法について述べているが、関数 $S_N(G)$ の計算は複雑になる。

2次元の簡単な例を示すと、総数 1000 点 ($N = 1000$) のサンプリング点のうち、4等分された区画の左下の領域 ($x_1 = 0.5, x_2 = 0.5$) に 245 点入っていたとすると、ディスクレパンシイ $D^*(0.5, 0.5)$ は $\sup |245 - 1000 * 0.5 * 0.5|$ と計算される。

我々はディスクレパンシイを計算するうえで、次の方法を用いた。

- (1) 生成された数列を面積1の領域に分布させる。
- (2) 領域内で、乱数 a, b により、左下隅を $(0, 0)$ 、右上隅を (a, b) とする長方形の区画を決定する。
- (3) 区画の中の点 n を数え、その場合のディスクレパンシイ $\frac{n}{N} - a * b$ を計算する。

この方法を適当な回数、繰り返し、ディスクレパンシイにはその平均を用いた。この方法以外にも数列を発生させるたびにディスクレパンシイを計算し、全体の算術平均を求める方法もあるが、今回の評価には

用いなかった。これは、Good Lattice Points および Hammersley Points の生成順序が領域を端から順に埋めていくためである。

3. Low discrepancy 数列

数値積分でよく使用されている代表的な low discrepancy 数列には、Sobol 数列、Halton 数列、GFSR 数列、Good Lattice Points、Hammersley Points などがある。ここでは、Neiderreiter らが CG アプリケーションに有効であると提案している low discrepancy 数列の Good Lattice Points と Hammersley Points、そして Sobol 数列と GFSR 数列 (Halton 数列は、Hammersley 数列の部分数列である) を取り上げる。サンプル点のばらつき度合としては、Good Lattice Points、Hammersley Points、Sobol 数列、GFSR 数列の順に前者ほど規則的である。

3.1 Sobol 数列

Sobol 数列¹⁵⁾は、0 から 1 の間の j 番目の実数 X_j を長さ w の特別な 2 進数配列 $V_i, i = 1, 2, \dots, w$ を用、各 X_j の i 番目のビットがゼロでないという条件のときのみ、元の数列と 2 進数配列の要素 V_i の XOR (排他的 OR) をとる。以降、各数列 V_i は連続的に生成される。Sobol 数列の N が 16, 64 の場合の分布を図 2 に示す。

3.2 GFSR 数列

L ビット GFSR (Generalized feedback shift register) 数列 $u_i, i = 1, 2, \dots$ は、以下のように定義される^{11), 19)}。

$$\begin{aligned} u_i &= \sum_{j=1}^L b_{dj+i} 2^{-j} = 0.a_0 a_1 \dots a_{k-1} \\ &= b_{d+i} \left(\frac{1}{2}\right)^1 + b_{2d+i} \left(\frac{1}{2}\right)^2 + \dots \end{aligned} \quad (3)$$

ここで、 $b_j, j = 1, 2, \dots$ は、a linear feedback shift register と呼ばれるビット列である。Lewis と Payne¹¹⁾は、 r 次元の d の値は $100r$ より大きくすべきであると述べている。全体の数列は r より小さな s を用いて連続的に生成される。

$$u_i = u_{i-r+s} \text{ XOR } u_{i-r} \quad (4)$$

GFSR 数列の N が 16, 64 の場合の分布を図 3 に示す。

3.3 Good Lattice Points

Good Lattice Points¹²⁾は、優良格子法とも呼ばれ、サンプリングする個数 $N (\geq 2)$ と適当な整数 $g (1 \leq g < N)$ を用いて、

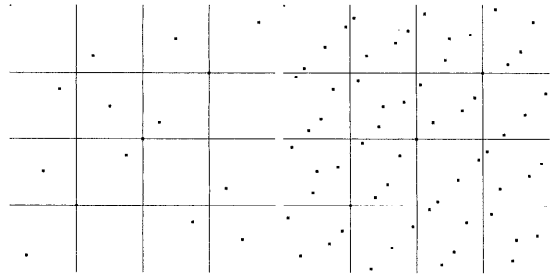


図 2 Sobol 数列 ($N = 16, 64$)

Fig. 2 Sobol's sequences ($N = 16, 64$).

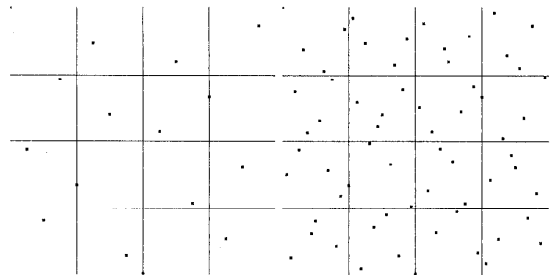


図 3 GFSR 数列 ($N = 16, 64$)

Fig. 3 GFSR sequences ($N = 16, 64$).

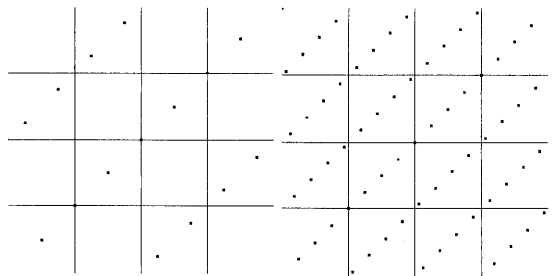


図 4 Good Lattice Points ($N = 16, g = 7$), ($N = 64, g = 13$)

Fig. 4 Good Lattice Points ($N = 16, g = 7$), ($N = 64, g = 13$).

$$\begin{aligned} p_n &= \left(\frac{n}{N}, \left[\frac{ng}{N} \right] \right) \\ &\text{for } n = 0, 1, \dots, N-1. \end{aligned} \quad (5)$$

このとき、 $[t]$ は、実数 t から、 $(0 < [t] < 1)$ への変換を意味する。この点列は N-rooks sampling となる。Good Lattice Points の N が 16, 64 の場合の分布を図 4 に示す。

3.4 Hammersley Points

Hammersley Points^{8), 12)}は、整数値 $n (> 0)$ のバイナリ表現 $n = a_{k-1} a_{k-2} \dots a_1 a_0$ のビット列を反転させた、式 (6) のビット反転関数 $\Phi(n)$ を定義する。

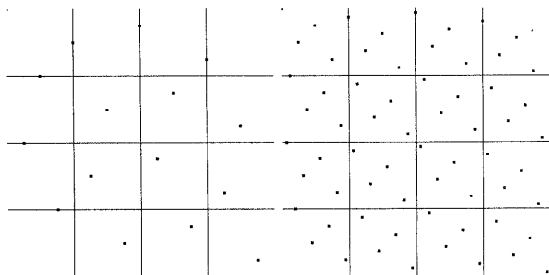


図5 Hammersley Points ($N = 16, 64$)
Fig. 5 Hammersley Points ($N = 16, 64$).

$$\begin{aligned}\Phi(n) &= 0.a_0a_1\dots a_{k-1} \\ &= a_0\left(\frac{1}{2}\right)^1 + a_1\left(\frac{1}{2}\right)^2 + \dots\end{aligned}\quad (6)$$

この結果、 $\Phi(n)$ は、 $0 \leq \Phi(n) < 1$ を満たし、次式の Hammersley 点列が定義される。 $\Phi(n)$ のみを用いたものが Halton 数列である。

$$\begin{aligned}p_n &= \left(\frac{n}{N}, \Phi(n)\right) \\ \text{for } n &= 0, 1, \dots, N-1.\end{aligned}\quad (7)$$

Hammersley Points の N が 16, 64 の場合の分布を図 5 に示す。

4. 評価

4.1 計算時間

各サンプリング法の生成効率を評価するため、我々は 6 種類のアルゴリズム、Sobol 数列、GFSR 数列、Good Lattice Points、Hammersley Points、乱数による方法、jittered sampling を C 言語で実装し、HP9000/770 (PA-RISC@100 MHz) HP-UX B10.01 上の UNIX time 機能を用いて計算時間を計測した。Sobol 数列、GFSR 数列、Good Lattice Points および Hammersley Points の実装には、それぞれ、文献 7), 11), 12), 15) を参考にした。乱数による方法および jittered sampling には HP/UX 上の rand() 関数を用いた。乱数による方法は、画素領域のサンプル点を独立に乱数のみで決定し、poisson disk sampling のようなサンプル点の取り直しは行っていない。表 1, 表 2 は、 512×512 画素に対する斜め格子パターンおよびサーキュラーゾーンプレート (circular zone plate) (図 6) を生成する際の CPU 時間の平均値である。斜め格子パターンの線の厚みは画素幅の半分である。

各列は 1 画素あたりのサンプリング数 (16 サンプル, 64 サンプル, 256 サンプル) を表し、数列発生時間と図形評価時間の両方を含む。両パターンにおけ

表 1 512×512 画素に対する計算時間 (斜め格子パターン) (秒)

Table 1 Computational time for 512×512 pixels (diagonal pattern) (sec.).

1 画素あたりのサンプル数	(16)	(64)	(256)
Sobol 数列	29.60	117.77	470.07
GFSR 数列	84.12	348.82	1445.78
Good Lattice Points	34.92	138.88	555.27
Hammersley Points	48.82	193.94	773.92
rand() (HP/UX)	43.56	173.43	693.82
Jittered Sampling	82.67	327.87	1308.39

表 2 512×512 画素に対する計算時間 (サーキュラーゾーンプレート) (秒)

Table 2 Computational time for 512×512 pixels (circular zone plate) (sec.).

1 画素あたりのサンプル数	(16)	(64)	(256)
Sobol 数列	7.95	31.08	123.49
GFSR 数列	62.45	258.95	1080.48
Good Lattice Points	13.41	52.91	210.78
Hammersley Points	33.95	220.47	1223.81
rand() (HP/UX)	22.05	101.43	349.52
Jittered Sampling	27.25	107.89	428.11

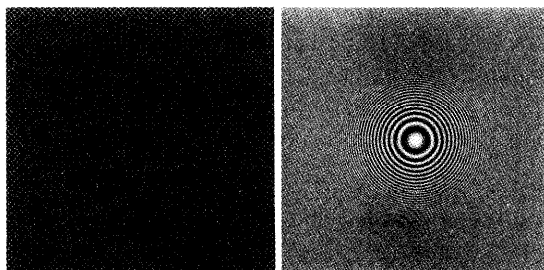


図6 斜め格子パターンとサーキュラーゾーンプレート
Fig. 6 Diagonal pattern and circular zone plate.

る計算時間の違いは、各パターンの関数値計算による差である。我々の実験では、両パターンで Sobol 数列が最も速く、2 番目が Good Lattice Points であった。その理由として、Sobol 数列の生成方法が、条件分岐と XOR (排他的 OR) のみの計算であり、Good Lattice Points では、単純な割り算のみであるためである。

4.2 ディスクレパンシによる評価

CG における 2 次元サンプリング問題において、Shirley¹⁶⁾ はディスクレパンシによる評価の有効性を報告している。表 3 は各 low discrepancy 数列のディスクレパンシを 256×256 回計算した平均値である。ディスクレパンシでの評価では、Sobol 数列と GFSR 数列が最も良い結果を示している。Good Lattice Points と Hammersley Points は規則的に空間を埋めていき、理論的にもそのディスクレパンシは小

表3 256 × 256 回計算したディスクリパンシーの平均値
Table 3 The average of discrepancy calculated 256 × 256 times.

1 画素あたりのサンプル数	(16)	(64)	(256)
Sobol 数列	0.027519	0.007785	0.002139
GFSR 数列	0.027524	0.007887	0.002575
Good Lattice Points	0.042400	0.011252	0.003330
Hammersley Points	0.043744	0.013054	0.003513
rand() (HP/UX)	0.087642	0.060689	0.053967
Jittered Sampling	0.041541	0.015237	0.005717

表4 RMS 誤差 (斜め格子パターン)
Table 4 The RMS error in comparison with jittered sampling (diagonal pattern).

1 画素あたりのサンプル数	(16)	(64)	(256)
Sobol 数列	0.045855	0.020917	0.008507
GFSR 数列	0.037369	0.012186	0.005494
Good Lattice Points	0.051985	0.023824	0.013595
Hammersley Points	0.046697	0.019056	0.008477
rand() (HP/UX)	0.067935	0.034250	0.017520
Jittered Sampling	0.036483	0.013496	0.000000

さい。しかしながら、本実験のようにサンプリング数が少ない場合では、ディスクリパンシーが小さくならない。そして、Good Lattice Points や Hammersley Points のような規則正しいサンプリング点の場合、サンプリング数が少ないとモアレなどが生じやすいので注意が必要である。

4.3 精 度

Jittered sampling で 1 画素あたり 256 点サンプリングした画像 (ここでは、これを基準画像と呼ぶ) を用いて、他のサンプリング方法で作成した画像と比較した。基準画像に用いたテストパターンは、図 6 の斜め格子とサーキュラーゾーンプレートである。比較方法は画素ごとに差を求めて、その差の 2 乗和 (RMS 誤差) を計算する (式 (8))。この場合、各画素の値が相対的に基準画像の画素に近いほど、そのサンプル法の RMS 誤差は小さくなる。

$$\text{RMS Error} = \sqrt{\frac{\left(\sum (V_i^{\text{jittered}} - V_i^{\text{others}})^2\right)}{\text{sample num}}} \quad (8)$$

ここで、 V_i^{jittered} と V_i^{others} は、jittered sampling と他のサンプリング法で計算した画素の値である。表 4 と表 5 は、格子方向と斜め方向パターンの RMS 誤差の平均値である。両パターンで最も良い結果を示したのは GFSR 数列と Sobol 数列であり、ディスクリパンシーの評価ともよく似た結果になった。各 RMS 誤差は、サンプル数 n が増加するにつれて、 $\sqrt{n}$ に比例して減少している。

表5 RMS 誤差 (サーキュラーゾーンプレート)
Table 5 The RMS error in comparison with jittered sampling (circular zone plate).

1 画素あたりのサンプル数	(16)	(64)	(256)
Sobol 数列	2.687021	0.814257	0.521840
GFSR 数列	2.778503	0.543694	0.317586
Good Lattice Points	3.417050	0.666778	0.355223
Hammersley Points	6.493344	2.045505	0.697995
rand() (HP/UX)	12.397014	6.184552	3.117691
Jittered Sampling	4.009379	1.069483	0.000000

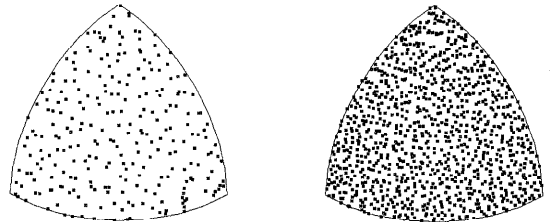


図7 Jittered sampling による 3 次元サンプリング
($N = 256, 1024$)

Fig. 7 3D sampling of jittered sampling ($N = 256, 1024$).

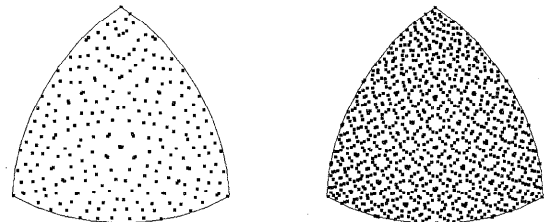


図8 Sobol 数列による 3 次元サンプリング ($N = 256, 1024$)
Fig. 8 3D sampling of Sobol's sequences ($N = 256, 1024$).

5. Low Discrepancy 数列を用いた適応的 3 次元サンプリング

5.1 3 次元サンプリング

モンテカルロ法を用いる輝度計算¹⁷⁾や鏡面反射を考慮したラジオシティ法では、球面上でのサンプリングが必要になる。Arvo¹⁾は、2 個の乱数を用いた球面上の 3 点から成る球面三角形内をサンプリングする方法を提案した。本論文では、Arvo の方法に対して、low discrepancy 数列を用いて、アルゴリズムの改良を試みる。本アルゴリズムの利点は、サンプリング数を増加させたとき、サンプル点が重複しないことが保証されていることと計算時間が乱数より速い点である。

図 7 は、Arvo の方法により生成したものであり、図 8、図 9、図 10、図 11 は、それぞれ、Arvo の方法の乱数を Sobol 数列、GFSR 数列、Good Lattice Points、Hammersley Points に置き換えたものであ

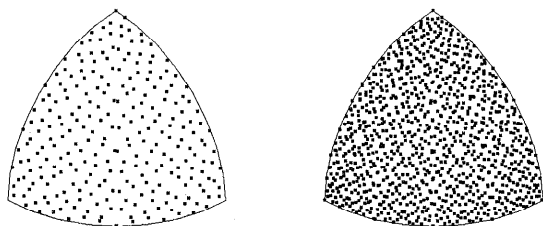


図9 GFSR 数列による3次元サンプリング ($N = 256, 1024$)
Fig. 9 3D sampling of GFSR sequences ($N = 256, 1024$).

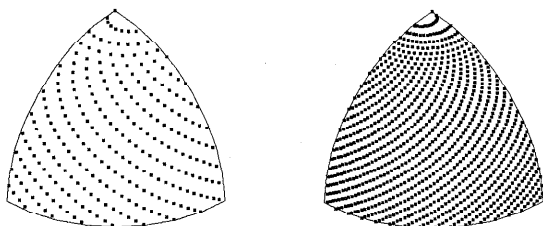


図10 Good Lattice Points による3次元サンプリング
($N = 256, 1024$)
Fig. 10 3D sampling of Good Lattice Points
($N = 256, 1024$).

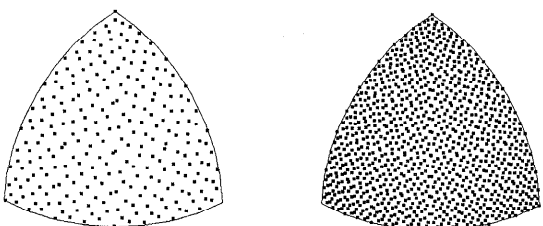


図11 Hammersley Points による3次元サンプリング
($N = 256, 1024$)
Fig. 11 3D sampling of Hammersley Points
($N = 256, 1024$).

る。サンプリング点を増加させることにより、各サンプリング法特有のパターンが生成されるのは興味深い。計算時間に関しては、表1、表2に示した計算時間に比例し、乱数 rand() と Sobol 数列を比べると約47パーセント以上、速度が向上する。一般にラジオシティ法では、適応的3次元サンプリングの計算は全体の中で大きな割合を占めるので、この速度向上は全体の処理速度に大きく利いてくる。

また、レイトレーシング法で画素ごとにサンプリング数を変更させたり、面光源の輝度を必要な精度までサンプリングしたい場合、生成された low discrepancy 数列の計算を最初からやり直すのでは効率が悪い。このような場合には、Sobol 数列や GFSR 数列を用いることにより、あらかじめ区画された領域を守りながら、サンプル点を増加させることができる。図12は、

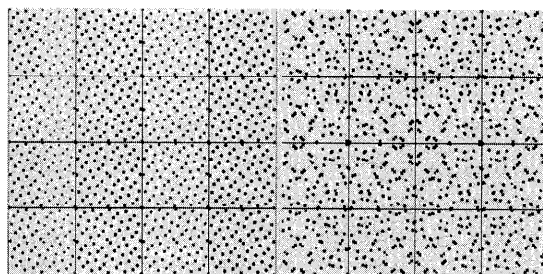


図12 適応的サンプリング (Hammersley Points と GFSR 数列)

Fig. 12 Adaptive sampling (Hammersley Points and GFSR sequences).

表6 ルックアップテーブル方式を用いた計算時間 (GFSR 数列)
Table 6 Computational time for look-up table approach for 512×512 pixels (GFSR sequences) (sec.).

1画素あたりのサンプル数	16	64	256
斜め格子パターン	27.75	110.35	440.71
サンキュラ・ゾーンパターン	5.92	23.02	91.32

Hammersley Points と GFSR 数列で 4×4 区画内で順にサンプル点を増加させた例である。サンプル点は50個ずつ色を変更している。GFSR 数列は順次区画を横断しながら、サンプリング空間を埋めていく。Good Lattice Points や Hammersley Points はサンプル数により発生する位置が変わるため、このような場合には向かない。この性質を利用することにより、必要な精度まで任意の回数、サンプリングを続けることができる。

5.2 ルックアップテーブル方式を用いた Low Discrepancy 数列

Low discrepancy 数列は決定性ルールにより生成されるので、前処理によりあらかじめ数列をルックアップテーブルに格納しておき、表引きのみで数値を参照することができる。表6は、ルックアップテーブル方式による GFSR 数列の各パターンの計算時間 (サンプル数は1画素あたり16点で 512×512 画素計算) であり、ルックアップテーブルからの表引き時間と図形評価時間の総和である。ルックアップテーブルに使用したメモリ容量は約32Kバイトである。本方式の利点はどんな数列にも適用できる点と、前処理さえ終了すれば、表引きの計算時間がたいへん短い点である。メモリ面でのトレードオフはあるが、最近のワークステーションレベルのコンピュータではほとんど問題ないと思われる。また、別の利点として、適応的サンプリングに向いていない Good Lattice Points や Hammersley Points の数列でも、いったん数列をルックアップテーブルに格納してしまえば、表引きの順序で簡単に適応

的サンプルを行うことができる。

表します。

6. おわりに

本研究では、準モンテカルロ法で使用されている low discrepancy 数列 (Sobol 数列, GFSR 数列, Good Lattice Points, Hammersley Points) と従来から使用されてきた乱数を用いたサンプリング法 (jittered sampling) を比較し、その有効性を検証した。一般に low discrepancy 数列は計算効率が良く、均一にサンプリングできるため、CG で使用されているほとんどの乱数計算に使用できると思われる。また、乱数を用いた方法に比べてサンプル点の重複を避けることができる。

Sobol 数列は計算効率とばらつき具合の点でバランスのとれた数列である。GFSR 数列は各点のばらつき具合が今回テストした数列の中で一番良いが、我々の実装ではその計算効率は良くなかった。これはサンプル点生成のために行列計算による方法¹⁹⁾を用いているからであるが、より効率の良い実装方法の開発が必要と思われる。また、Sobol 数列と GFSR 数列の生成パターンは、jittered sampling の生成パターンに近く、サンプル数を適応的に変更したい場合にも有効である。Good Lattice Points はアルゴリズムが簡単でインプリメントが容易であり処理速度も速い。ただし、Good Lattice Points と Hammersley Points は周期性が強いため使用には注意が必要である。Good Lattice Points や Hammersley Points は理論的にディスクレパンシイが小さいことが証明されているが、サンプル数の少ない画像生成における 2 次元のアンチエイリアスにはあまり向かないと思われる。

今後の研究として、より高速で実用的な low discrepancy 数列を発生させるアルゴリズムとディスクレパンシイの評価方法の開発があげられる。Dobkin の提案しているディスクレパンシイ評価方法²⁰⁾はその 1 つの解決案である。また、今回は各数列を評価するうえで、よく CG で使用されている jittered sampling によるサンプリング結果と比較したが、各数列の理論的な誤差のオーダに関する議論も必要かと思われる。

謝辞 本研究をまとめるにあたり、研究施設を使わせていただいたニューヨーク州立大学 Stony Brook 校 Ari E. Kaufman 教授および研究室の皆様へ感謝いたします。また、本研究を始める機会を与えて下さった日本アイ・ビー・エム (株) 東京基礎研究所の手塚集氏に感謝いたします。

本研究は、一部、(財) 実吉奨学金 (Saneyoshi Scholarship Foundation) の助成を受けた。ここに謝意を

参考文献

- 1) Arvo, J.: Stratified Sampling of Spherical Triangles, *SIGGRAPH '95*, pp.437-438 (1985).
- 2) Bratley, P. and Fox, B.L.: ALGORITHM 659 Implementing Sobol's Quasirandom Sequence Generator, *ACM Trans. Math. Soft.*, Vol.14, No.1, pp.88-100 (1988).
- 3) Baratley, P., Fox, B.L. and Niederreiter, H.: Implementation and Tests of Low-discrepancy Sequences, *ACM Trans. Modeling and Computer Simulation*, Vol.2, No.3, pp.195-213 (1992).
- 4) Fy, van Dam, Feiner and Hughes: *Computer Graphics, Principles and Practice*, 2nd Edition, pp.617-647, Addison-Wesley (1990).
- 5) Cook, R.L., Porter, T. and Carpenter, L.: Distributed Ray Tracing, *SIGGRAPH '84*, Vol.3, No.18, pp.137-145 (1984).
- 6) Cook, R.L.: Stochastic Sampling in Computer Graphics, *SIGGRAPH '86*, Vol.5, No.1, pp.51-72 (1986).
- 7) Halton, J. and Smith, G.: Radical-inverse Quasi-random Point Sequence [G5], *Comm. ACM*, Vol.7, No.12, pp.701-702 (1964).
- 8) Halton, J. and Zerebma, S.: The Extreme and L2 Discrepancies of Same Plane Sets, *Monatsh. Math.*, Vol.73, pp.316-328 (1969).
- 9) Heinrich, S. and Keller, A.: Quasi-Monte Carlo Methods in Computer Graphics Part I: The QMC-Buffer, Technical Report, 242/94, University of Kaiserslautern, Germany (1994).
- 10) Heinrich, S. and Keller, A.: Quasi-Monte Carlo Methods in Computer Graphics Part II: The Radiance Equation, *Technical Report, 243/94*, University of Kaiserslautern, Germany (1994).
- 11) Lewis, T.G. and Payne, W.H.: Generalized Feedback Shift Register Pseudorandom Number Algorithms, *J. ACM*, Vol.20, No.3, pp.456-468 (1973).
- 12) Niederreiter, H.: Quasirandom Sampling in Computer Graphics, *Image Processing in Medicine, Remote Sensing and Visualization of Information*, Latvian Academy of Sciences, Riga, pp.29-34 (1992).
- 13) Niederreiter, H.: *Random Number Generation and Quasi-Monte Carlo Methods*, CBMS-NSF SIAM, Philadelphia, PA (1992).
- 14) Ohbuchi, R. and Aono, M.: Quasi-Monte-Carlo Rendering Using Low-discrepancy Sequence, *IPSS Sig. of CG and CAD*, 81-16, pp.91-96 (1996).

- 15) Press, W., Teukolsky, S., Vetterling, W. and Flannery, B.: *Numerical Recipes in C*, Second Edition, Cambridge Univ. Press (1992).
- 16) Shirley, P.: Discrepancy as a Quality Measure for Sampling Distributions, *EUROGRAPHICS '91*, pp.183-194 (1991).
- 17) Shirley, P., Wade, B., Hubbard, P.M., Zareski, D., Walter, B. and Greenberg, D.: Global Illumination via Density-Estimation, *Proc. Eurographics Workshop '95 (Rendering Techniques '95)*, pp.219-230 (1995).
- 18) Tezuka, S.: Polynomial Arithmetic Analogue of Halton Sequences, *ACM Trans. Modeling and Computer Simulation*, Vol.3, No.2, pp.99-107 (1993).
- 19) Tezuka, S.: The k -dimensional Distribution of Combined GFSR Sequences, *Mathematics of Computation*, Vol.63, No.206, pp.809-817 (1994).
- 20) Dobkin, D., Eppstein, D. and Mitchell, D.: Computing the Discrepancy with Applications to Supersampling Patterns, *ACM Trans. Computer Graphics*, Vol.15, No.4, pp.354-376 (1996).

(平成 8 年 11 月 18 日受付)

(平成 9 年 5 月 8 日採録)



土井 章男 (正会員)

昭和 33 年生。昭和 57 年神戸大学大学院工学研究科土木工学専攻修士課程修了。同年日本アイ・ビー・エム (株) 東京サイエンティフィック・センター (後の東京基礎研究所) 入社。平成 7 年より岩手大学工学部情報工学科講師, 平成 8~9 年にかけて New York 州立大学 Stony Brook 校客員教授。工学博士。コンピュータグラフィックスによる数値データの可視化技術や画像生成手法の研究に従事。著書に「3 次元グラフィックスの基礎と応用」(共著) など。ACM, IEEE, 画像電子学会, 各会員。



山口 美春

昭和 48 年生。平成 7 年岩手大学工学部情報工学科卒業。同年 (株) アルファ・システムズ入社。コンピュータグラフィックス, サンプル理論, 乱数, レイトレーシング法, 隠面消去アルゴリズム等に興味を持つ。



千葉 則茂 (正会員)

昭和 50 年岩手大学工学部電気工学科卒業。同年より昭和 53 年まで (株) 日本ビジネスコンサルタント (現 (株) 日立情報システムズ) に勤務。昭和 59 年東北大学大学院博士後期課程修了。工博。同年同大学工学部助手。昭和 61 年仙台電波高専助教授。昭和 62 年岩手大学工学部助教授, 現在教授。著書に「離散数学」(共著), 「レイトレーシング CG 入門」(共著), 「C アルゴリズム全科」(共著) など。アルゴリズム, コンピュータグラフィックス, 形の科学に興味を持つ。電子情報通信学会, 画像電子学会, ACM, CGS, IEEE, など各会員。