2003－CG－112　(20)
2003／8／19

複雑な陰関数曲面の確率過程的並列サンプリングとその可視化

木村彰徳[†]　田中　覚[†]
仲田　晋[†]　柴田章博[††,†††]

数千から数万項の陰関数で表現される複雑な曲面を確率過程的アルゴリズムによってサンプリングし，点群を生成することで精密かつ高速に可視化する手法を提案する．確率過程的アルゴリズムによるサンプリングは並列処理効率が非常に優れており，並列化した各々のサンプリングは独立に処理することができる．PCクラスタシステムで，実際に並列化したアルゴリズムによって補間点群を高速に生成し可視化した結果について述べる．

Parallelized Stochastic Sampling and Rendering of Intricate Surface Represented with Implicit Function

Akinori Kimura ,[†] Satoshi Tanaka ,[†] Susumu Nakata [†]
and Akihiro Shibata[††,†††]

We propose a procedure for fast sampling and rendering of intricate surface represented with implicit function. A lot of points constrained on the intricate surface are generated by using an algorithm based on stochastic process which is resolved by a Monte Carlo simulation. It is simple to parallelize the algorithm based on stochastic process, becase the stocastic process is independent of each other. We also describe results of fast and precise generation of the points constrained on intricate surface using PC cluster system and rendering its points.

1. はじめに

３Ｄスキャナ（レンジ・センサ）の発達に伴い，物体表面上に与えられた点群をもとにして形状モデリングを行うという需要が増している．しかし，入力データとなる点群（入力点群）は，部分的に充分稠密でない場合が多い．例えば，３Ｄスキャナによる測定の場合には，物体表面の凹凸やレーザ光の照射方向の関係で，どうしても測定点を充分に（あるいは全く）得られない部分が生ずる．このような部分では，当然，レンダリングのためのポリゴン化も不正確になる．点群を基にした形状モデリングでは，このような点密度の粗い（または欠けた）部分の処理に多大な労力と時間を要する．

そこで本報告では，入力点群を通過する陰関数曲面を作り，これを確率微分方程式に基づく並列モンテカルロ・シミュレーションで，高速，一様，かつ稠密にリサンプリングする手法を提案する．本手法は，与えられた入力点群を非線形補間した，精密な補間点群を高速生成するものである．

本報告で提案する手法は，最近我々が提案した確率過程サンプリング法（SSM）[1)~4)]に基づいている．SSMは，ブラウン運動の理論に基づいた，陰関数曲面の高速・一様なサンプリング法である．陰関数曲面上でのブラウン運動を，確率微分方程式を数値的に解くことで

† 立命館大学理工学部情報学科
Department of Computer Science, Ritsumeikan University
†† 高エネルギー加速器研究機構計算科学センター
Computing Research Center, High Energy Accelerator Research Organization
††† Centre for Novel Computing, University of Manchester

生成し，その軌跡をサンプル点群として用いる．計算時間は生成するサンプル点数に比例するので，非常に高速なサンプリングが可能である．しかし，入力点群を通過するように作った陰関数曲面のリサンプリングの場合は，解くべき陰関数（代数方程式）の項数が非常に多い（数千から数万以上）ため，更なる高速化が必須である．

そこで我々は，SSM を並列化することで，高速性を大幅に向上させる．並列化の原理は非常に簡単である．従来の確率過程サンプリング法は，1 粒子のブラウン運動に基づくものであったが，これを，各粒子が独立なブラウン運動をする多粒子系に拡張することで，並列処理を可能にする．各粒子間には相互作用が全く無いので，高い並列化率，すなわち加速率が実現される．

また，本報告の最後では，提案手法によって得られる一様，稠密な補間点群の特長を生かして，ポリゴン化不要の粒子レンダリング[1)]を実行した結果も紹介する．

2. 点群と陰関数曲面

2.1 陰関数曲面化

3D スキャナなどで得た入力点群を非線形補間して陰関数曲面化するためには，Turk と O'Brien の方法[5)]を用いる．基底関数には，次のものを選択する:

$$f_i(\mathbf{q}) = |\mathbf{q} - \mathbf{P}_i|^2 \ln |\mathbf{q} - \mathbf{P}_i| \ . \qquad (1)$$

ここで，$\mathbf{q} = (q_1, q_2, q_3)$ は 3 次元座標，$\mathbf{P}_i (i = 1, \ldots, n)$ は曲面上の n 個の入力点群の位置ベクトル，$\mathbf{P}_{n+i} (i = 1, \ldots, n)$ は $\mathbf{P}_i$ の外側近傍の点の位置ベクトルである．$2n$ 個の点に対して，この基底関数の線形結合:

$$F(\mathbf{q}) = \sum_{i=1}^{2n} \lambda_i f_i(\mathbf{q}) + p(\mathbf{q}), \qquad (2)$$

$$p(\mathbf{q}) = \lambda_{2n+1} q_1 + \lambda_{2n+2} q_2 + \lambda_{2n+3} q_3 + \lambda_{2n+4} \qquad (3)$$

を作り，$F(\mathbf{P}_i) = 0 (i = 1, \ldots, n)$ が成り立つように係数 λ_i を求めることで，n 個の入力点群を通る陰関数曲面の方程式を得る．すなわち n 個の入力点群に対し，関数 $F(\mathbf{q})$ は

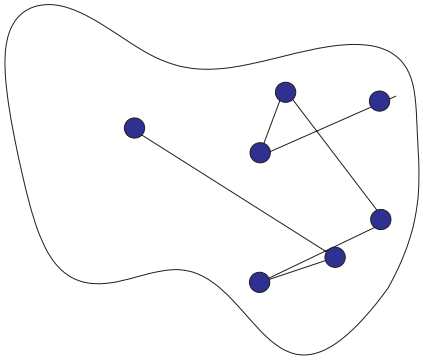

図 1 曲面上に閉じ込められた 1 粒子のブラウン運動.

$2n + 4$ 個の項で構成される．

この陰関数曲面化は，線形補間をするポリゴン化と比較して，より精密な非線形補間をしたことになる．

2.2 陰関数曲面上の点群の生成

この非線形補間によって得た陰関数曲面を正確に可視化するために，SSM を用いてリサンプリングし，陰関数曲面上を高密度に埋め尽くす点群を生成する．

方程式 $F(\mathbf{q}) = 0$ で定義される陰関数曲面を I_F とする．ここで，$F(\mathbf{q})$ はスカラー関数である．以下の確率微分方程式を差分化して数値的に解くことで，I_F 上に一様なサンプル点群を高速生成することが出来る:

$$dq_i(t) = dq_i^{(\mathrm{T})}(t) + dq_i^{(\mathrm{S})}(t) + dq_i^{(\mathrm{N})}(t) \ . \qquad (4)$$

ここで，t は仮想的に導入された時間変数であり，$dq_i^{(\mathrm{T})}$, $dq_i^{(\mathrm{S})}$, $dq_i^{(\mathrm{N})}$ は，それぞれ，

$$dq_i^{(\mathrm{T})} \equiv \sum_{j=1}^{d} P_{ij} dw_j \ , \qquad (5)$$

$$dq_i^{(\mathrm{S})} \equiv -\frac{\alpha}{|\nabla F|^2} \left(\frac{\partial F}{\partial q_i} \right) \mathrm{Tr} \left\{ (\partial^2 F) \cdot P \right\} dt, \qquad (6)$$

$$dq_i^{(\mathrm{N})} \equiv -K\alpha \left(\frac{\partial F}{\partial q_i} \right) \frac{F}{|\nabla F|^2} dt \ , \qquad (7)$$

で定義される．ランダムさを生成するのは，ガウス型のランダム変数 $dw_i(t)$ であり，これは $\langle dw_i(t) \rangle = 0$, $\langle dw_i(t) dw_j(t) \rangle = 2\alpha\delta_{ij} dt$ の統計的性質を満足する．ここで，$\langle \cdot \rangle$ は統計平均を表す期待値記号であり，α は正の定数，δ_{ij} はクロネッカーのデルタである．$dw_i(t)$ に作

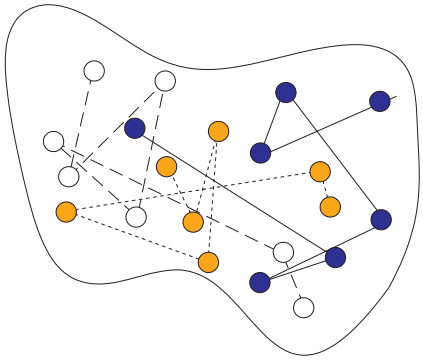

図 2 多粒子系のブラウン運動による補間点群の並列生成.

用している P_{ij} は I_F 上への射影行列である.

確率微分方程式 (4) が生成する確率過程は, 陰関数曲面 I_F 上に閉じ込められたウィーナー過程となる. これは, 直感的には, 曲面上に閉じ込められた微小な 1 粒子のブラウン運動 (不規則運動) に対応する (図 1).

ブラウン運動の軌跡上の点を集めれば, I_F 上の一様な補間点群が得られる. 確率微分方程式 (4) を数値的に解くことを長く続ければ続ける程, 多くの補間点群が得られる. そして, 計算時間は軌跡の長さ, すなわち得られる補間点数に正比例する. このため, 大量の補間点群を生成しても, 計算時間は, ほぼその補間点数に比例する. これが, SSM の高速性の理由であり, また, 大量の補間点群の生成に適している理由でもある.

2.3 確率過程サンプリング法の並列化

Turk と O'Brien の方法で生成した陰関数曲面の方程式は, 入力点の数と同程度の非常に多数の項を持つ. これをリサンプリングするには, どのようなアルゴリズムを用いるにせよ, 長い計算時間を要する. SSM の場合には, その実行速度は項数に反比例して低下する. しかし, SSM は, 並列処理によって大幅な高速化が可能である. したがって, PC クラスタシステムなどのマルチプロセッサ環境によって高速実行が可能となる.

SSM の並列化は極めて簡単である. 通常の SSM は, 前節で述べたように, 1 粒子のブラウン運動に対応する. これを, 図 2 のように, 独立なブラウン運動を行う多粒子系に置き換え, 各粒子が生成する補間点群を単純に重ね合わればよい. 各粒子間には相互作用は全く無いので, 計算の完全並列化が可能である. 粒子それぞれのブラウン運動を異なるプロセッサで計算させれば, プロセッサ数に比例した高速化が可能になる.

表 1 PC クラスタシステムの仕様.

PC cluster system	
Number of hosts	16 compute hosts 1 process manage host 1 file server host, RAID1
Compute Network	Myrinet2000
Manage Network	Ethernet, 100 BASE-TX
System software	Redhat Linux 7.3 with SCore 5.2.0
Specification of a host	
CPU	Intel Xeon 2.4 GHz (Dual CPU)
Main Memory	2 GB (but the file server is 1 GB)

並列計算においては, ひとつの管理プロセスを設定し, 残りのプロセスをリサンプリングを実行する計算プロセスとする. 各プロセスは別々のプロセッサで実行する. 管理プロセスは, リサンプリングを始める前に, 各計算プロセスに別々の初期位置 ($\mathbf{q}(t=0)$) と乱数シードを分配する. これにより, 各計算プロセスごとに独立なブラウン運動が割り当てられる. リサンプリングが開始されると, 各計算プロセスは互いに独立に補間点群を生成し, それらを管理プロセスに転送する. 管理プロセスは, 受け取った補間点群を統合してファイルに保存する (オフラインの可視化). またはレンダリング用のプロセスやネットワークで接続された PC に転送する (オンラインの可視化).

3. 実　　験

3.1 並列処理による点群の生成

並列処理プログラムは MPI (Message Passing Interface) を利用して実装し, PC クラスタシステムで実行する.

PC クラスタシステムは, 表 1 に示すように CPU が Dual Intel Xeon 2.4 GHz, メインメモリ 2GB の計算ホストを 16 台 (32 計算プロセッサ), 同仕様のジョブ管理のためのジョブ管理ホストを 1 台, およびファイルサーバー

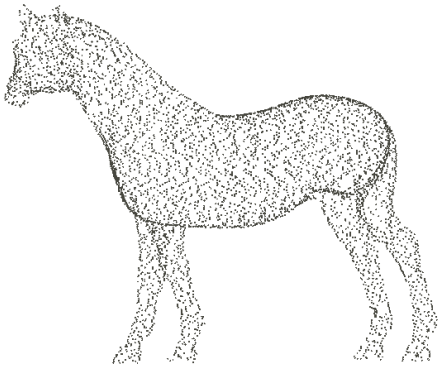

図 3 陰関数曲面化に用いた入力点群 (5,000 点).

ホストの 1 台によって構築されている．各計算ホスト間は Myrinet2000 で接続し，管理ホストとは 100 BASE-TX Ethernet で接続している．OS は Redhat Linux 7.3，PC クラスタシステムソフトウェアに SCore 5.2.0 を用いている．

図 3 のような馬の形状を用いて，リサンプリングの数値実験を行った．図 3 に描かれた 5,000 点の入力点データを陰関数曲面化し，リサンプリングを行い，計算速度とプロセッサ数（並列度）の関係を調べた結果を図 4 に示す．

図 4 に示すグラフの中の点は，6 通り (1,3,7,15,23,31 プロセッサ) の並列度において，計算プロセスで補間点群の位置ベクトルと法線ベクトル☆を計算し，そのデータを管理プロセスに集めてファイルに保存するまでの時間を計測したものである．

図 4 より，理論どおり，プロセッサ数に比例して計算速度が増大していることがわかる．図中には，データ点を通る直線も最小 2 乗法で求めて描き加えてある．

図 5 にリサンプリングによって得られた補間点群（100,000 点）を示す．一様かつ高密度な補間点群が得られていることがわかる．図 3 の入力点群には局所的に低密度な箇所が多数あるが，図 5 の補間点群ではそれらの箇所が密に埋め尽くされていることが分かる．

なお，図 5 で馬の蹄の部分が丸くなっているが，これは足の裏に入力点が無いために陰関数曲面化の過程で曲面に補間されたためである．

☆ 法線ベクトルは補間点における $F(\mathbf{q})$ の勾配から計算できる．補間点を陰影をつけて可視化するのに用いられる．

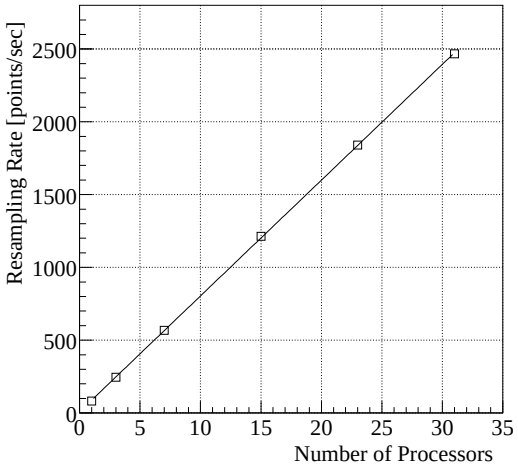

図 4 プロセッサ数と計算速度の関係.

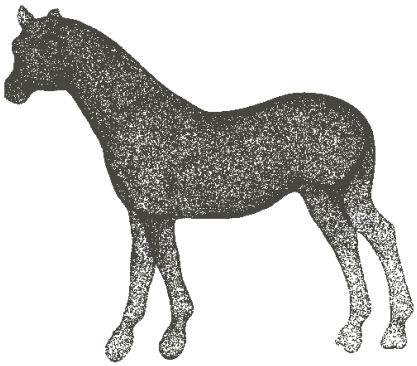

図 5 生成された補間点群（100,000 点）.

図 6 にリサンプリング速度の陰関数の項数依存を示す．項数が 4,000，10,000, 14,000 項の陰関数を用いて 100,000 点の補間点群を生成したときに，リサンプリング速度が陰関数の項数に反比例している．これは，前述のように SSM の計算時間のほとんどが陰関数の各項の計算に占めてられていることを意味している．

3.2 点群を用いた可視化

SSM を用いれば，一様かつ高密度な補間点群（及び各補間点における法線ベクトル）を高速に生成することが出来た．これらの補間点のそれぞれに，その点での陰関数曲面の反

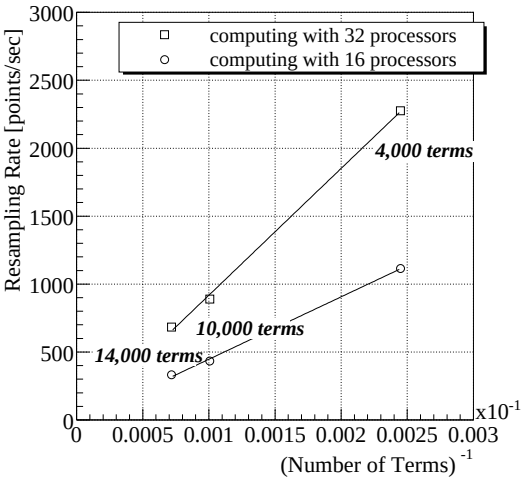

図 6 プロセッサ数と陰関数の項数の関係.

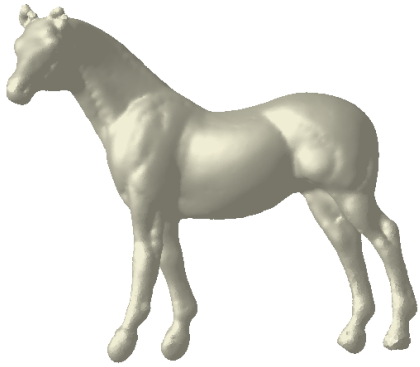

図 7 粒子レンダリングによって可視化した例 1.

射光の色を割り当て，微小な球として可視化すれば，陰関数曲面のレンダリングが可能となる．我々は，これを「粒子レンダリング」[1] と呼んでいる．粒子レンダリングには，次の 2 つの意味がある．

(1) 点群からのポリゴン化を必要としないのでその計算時間を削減できる．

(2) 入力点群を非線形補間した陰関数曲面を，ポリゴン化という線形近似を経ずにそのまま可視化できる．

図 7，8 に，SSM によるリサンプリングによって得られた補間点群を用いて粒子レンダリングした例を示す．このように滑らかかつ筋肉のような凹凸をはっきりとレンダリングできるのは，陰関数化と SSM によるリサンプリングによって，入力点群の位置と線ベクトルの両方の情報が精密かつ稠密に非線形補間

図 8 粒子レンダリングによって可視化した例 2.

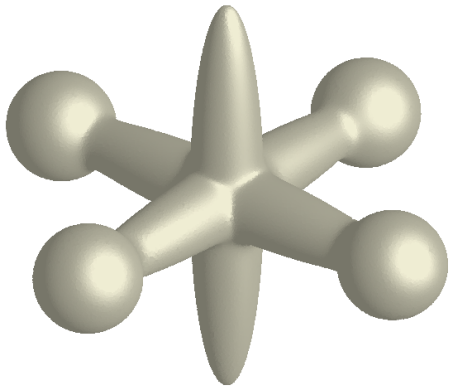

図 9 部品数 7 のメタボール曲面を粒子レンダリングした例 (Jack).

されるからである．

次に，このポリゴン化の過程を省略した粒子レンダリングによって，リアルタイムアニメーションの可能性を調べた．ここでは，図 9 に示す，陰関数で表現した部品数 7 のメタボール曲面 (Jack と呼ぶことにする) のリサンプリング速度によって考察する．

曲面 Jack の並列処理によるプロセッサ数に対するリサンプリング速度を図 10 のグラフに示す．丸の点は，求めた補間点群の位置ベクトルと法線ベクトルのデータをファイルに保存しないときの結果で，四角の点はそれらのデータを保存したときの結果である．

陰関数の項数が少ない場合に，SSM による計算時間に対して，並列処理プログラムの MPI による点群データのネットワーク転送時間及び点群データをファイルに保存する処理時間が相対的に長くなるために，総処理時間に大きく影響して見えている．

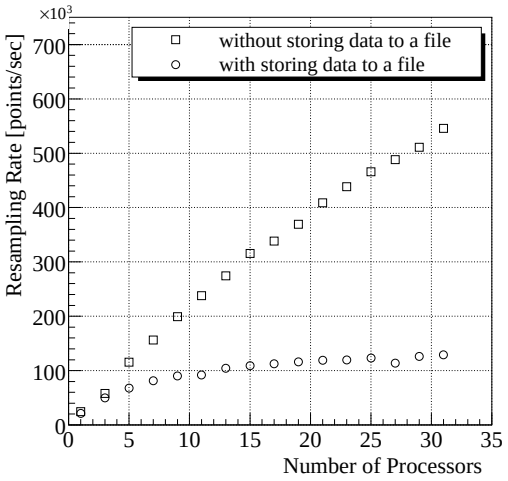

図 10 プロセッサ数と計算速度の関係.

しかしながら，四角の点で示した計算した点群データを保存しない場合と比較すると，ネットワーク転送の時間は比較的影響が少ないことが分かる．そしてこのとき，32 プロセッサ (31 計算プロセス+1 管理プロセス) によるリサンプリングで，1 秒間に約 550,000 点を生成している．つまり，本研究の実験環境において，OpenGL などによるハードウェアレンダリングによって，リサンプリングに対して十分に高速な粒子レンダリングをすることで，1 フレームあたり約 50,000 点の補間点群を生成し，11 fps 程度のアニメーションが可能になる．

4. おわりに

曲面上の非一様なサンプリング点群を Turk と O'Brien の方法を用いて非一様な点群から陰関数曲面を生成し，確率過程サンプリング法 (SSM) を用いてリサンプリングした．このリサンプリングによって，陰関数曲面上に非線形補間した一様で高密度な点群を生成することができた．SSM によるリサンプリングの手法の並列処理アルゴリズムは独立性が高く，PC クラスタを利用した MPI による実装において，高い並列性を示した．

また，この方法で生成した一様で密な点群を粒子レンダリングすることで効率的に可視化できることについて述べた．

参 考 文 献

1) S. Tanaka, A. Shibata, H. Yamamoto, H. Kotsuru, " Generalized Stochastic Sampling Method for Visualization and Investigation of Implicit Surfaces," Computer Graphics Forum 19(3), (2001), 359-367. (Proceedings of Eurographics 2001).

2) S. Tanaka, A. Morisaki, S. Nakata, Y. Fukuda, H. Yamamoto, "Sampling Implicit Surfaces Based on Stochastic Differential Equations with Converging Constraint," Computers & Graphics, 24(3), (2000), 419–431.

3) S. Tanaka, T. Nakamura, M. Ueda, H. Yamamoto, K. Shino, "Application of the Stochastic Sampling Method to Various Implicit Surfaces," Computers & Graphics, 25(3), (2001), 441-448.

4) S. Tanaka, Y. Fukuda, H. Yamamoto, "Stochastic Algorithm for Detecting Intersection of Implicit Surfaces," Computers & Graphics, 24(4), (2000), 523-528.

5) G. Turk, J. F. O'Brien, "Shape Transformation Using Variational Implicit Function," Computer Graphics Proceedings, Annual Conference Series (SIGGRAPH 1999), 335-342.