

論文

MPU 法に基づく色情報付き陰関数曲面の生成

藤本 大地^{†*} 伊東 拓^{††} 仲田 晋^{†††} 北川 高嗣^{††}
 岡 将史^{††††} 田中 覚^{†††}

Implicit Surface Reconstruction with Color Distribution Using MPU Implicits

Daichi FUJIMOTO^{†*}, Taku ITOH^{††}, Susumu NAKATA^{†††}, Takashi KITAGAWA^{††},
 Masafumi OKA^{††††}, and Satoshi TANAKA^{†††}

あらまし CG (Computer Graphics) や CAD (Computer Aided Design) では、三次元計測装置から得た離散点群から 3D オブジェクト曲面を再構成することがしばしば要求される。この表現法として、近年、陰関数曲面が注目されている。また、離散点群上の色情報を同時に取得できる計測装置も実用化されており、再構成した陰関数曲面上に色情報を付加することが問題となっている。本研究では、同問題の新たな表現法として、MPU (Multi-level Partition of Unity) 法に基づく色関数の生成手法を提案する。本提案法では、小領域で定義される区分的な色関数を構成し、重み関数を介してそれらを滑らかに結合することで全領域で定義される色関数を構成する。また、小領域セル内の色情報の状態により色関数の構成法を適応的に変えることで色境界の表現も実現される。

キーワード コンピュータグラフィックス, 形状表現, 陰関数曲面, 色情報

1. ま え が き

CG (Computer Graphics) や CAD (Computer Aided Design) において、歴史的建造物や芸術作品のデジタル化、構造解析のメッシュ作成などの過程で 3D レーザスキャナなどの三次元計測装置から得た離散点群から 3D オブジェクトを再構成することはしばしば要求される。また、近年の計測装置の高精度化やコンピュータ資源の増大により、離散点群の規模は増加傾向にある。従来、離散点群から 3D オブジェクトを再構成する問題の表現法には、ポリゴンメッシュやパラメトリック曲面など 3D オブジェクト表面の情報

を直接もつ陽的な方法が用いられてきた。一方、曲面上の点群データからの形状モデリングでは、陰関数曲面による表現法が近年注目されている [1]。

陰関数曲面による 3D オブジェクトの表現法ではオブジェクトは単一の陰関数で表される。この表現はオブジェクトを面ではなくソリッドとして扱うため、CSG (Constructive Solid Geometry) やモーフィング、内外判定、衝突判定などが容易という特徴をもつ [2], [3]。

離散点群から陰関数曲面を求める方法として、RBF (Radial Basis Functions) による表現法が広く用いられてきた [2]~[6]。同法では、陰関数曲面は RBF と多項式の線形結合によって構成される。同法で構成される陰関数は補間性に優れ、2D 画像補修にも応用されている [7]。しかしながら、RBF と多項式の結合係数を得るために、元数が離散点数に依存する連立一次方程式を解く必要がある。したがって、問題の大規模化に伴い求解に多大な時間を必要とするばかりでなく、同連立一次方程式の係数行列をメモリに格納することも困難となる。

これに対し、陰関数を MPU (Multi-level Partition of Unity) 法によって構成する方法が提案されている [8]。MPU 法は、細分化された小領域に対して最小

[†] 筑波大学大学院理工学研究科, つくば市
 Master's Program in Science and Engineering, University
 of Tsukuba, Tsukuba-shi, 305-8573 Japan

^{††} 筑波大学大学院システム情報工学研究科, つくば市
 Graduate School of Systems and Information Engineering,
 University of Tsukuba, Tsukuba-shi, 305-8573 Japan

^{†††} 立命館大学情報理工学部, 草津市
 College of Information Science and Engineering,
 Ritsumeikan University, Kusatsu-shi, 525-8577 Japan

^{††††} 立命館大学大学院理工学研究科, 草津市
 Graduate School of Science and Engineering, Ritsumeikan
 University, Kusatsu-shi, 525-8577 Japan

* 現在, (株) ナナオ所属

二乗法による二次関数のフィッティングを行うことで局所的に陰関数を構成し、その後、小領域セルで構成した陰関数を Partition of Unity [9] と呼ばれる重み関数を介して結合することで全領域で定義される陰関数が構成される。MPU 法では RBF を用いた表現法に見られる大規模な連立一次方程式を解く必要がなく、特に大規模な離散点群を対象とした問題に対して陰関数を高速に構成できるという利点をもつ。

一方、近年の三次元計測装置の発展により、離散点群と同時に各点の色情報が得られる計測装置もあり、再構成した陰関数曲面上に色情報を付加することが新たな問題となっている。同問題に対し、Tobor らは RGB 表色系の各刺激 Red, Green, Blue それぞれに対する色関数を構成する方法を提案している [10], [11]。同法は、2D 画像修復 [7] の三次元問題への応用とみなすことができる。しかしながら、同法では色境界の表現といった色情報特有の問題は考慮されていない。

本研究では、陰関数曲面上に色情報を付加する方法として、MPU 法に基づく色関数の構成手法を提案する。本提案法では、MPU 法に基づく小領域セルでローカル色関数を構成し、各ローカル色関数を重み関数を介して結合することで全領域で定義される色関数を構成する。また、小領域セル内の色の分布に応じて色関数の構成を変えることで適応的な色の表現、例えば色が不連続的に変化する様子を再現することができる。なお、色の境界面の構成法には RBF に基づく方法を採用する。またこのとき、RGB 表色系の代わりに L*a*b*表色系を用いることで、色境界の抽出が容易となる。

本論文の構成は、まず、2. で離散点群から陰関数曲面を再構成する方法として RBF を用いた方法と MPU 法を示し、3. で本研究で提案する MPU 法に基づく色関数について述べる。更に、本研究で提案する色関数が陰関数曲面上に色情報を与える問題に適用可能であることを 4. で示し、最後に、5. で結論を述べる。

2. 陰関数曲面

本章では、離散点から陰関数曲面

$$f(\mathbf{x}) = 0, \quad \mathbf{x} = [x, y, z]^T \in \mathbb{R}^3 \quad (1)$$

を構成する方法として、RBF を用いる方法と MPU 法について概説する。ただし、 $f(\mathbf{x})$ は曲面の内側で正、外側で負の値をとるものとする。本研究では、MPU 法に基づく方法で色関数を構成し、RBF を用いた方

法で色境界の境界面を構成するが、これについては 3. で述べる。

2.1 RBF を用いた陰関数曲面の生成

曲面上の N 個の離散点 $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N\}$ と曲面内部または外部の $M - N$ 個の制約点 $\{\mathbf{p}_{N+1}, \mathbf{p}_{N+2}, \dots, \mathbf{p}_M\}$ が与えられたとき、陰関数 $f(\mathbf{x})$ を以下のように構成することができる [3], [5]。

$$f(\mathbf{x}) = \sum_{i=1}^M \lambda_i \psi(\|\mathbf{x} - \mathbf{p}_i\|_2) + q_0 + q_1x + q_2y + q_3z. \quad (2)$$

ただし、 $\psi(\cdot)$ は RBF (Radial Basis Functions) であり、Gaussian, Thin-plate Spline [2]~[7], CSRBF [6], [12] などが用いられる。未知の係数 $\boldsymbol{\lambda} \equiv [\lambda_1, \lambda_2, \dots, \lambda_M]^T$ と $\mathbf{q} \equiv [q_0, q_1, q_2, q_3]^T$ は次の連立一次方程式の解として得られる。

$$\begin{bmatrix} B & P \\ P^T & O \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{q} \end{bmatrix} = \begin{bmatrix} \mathbf{h} \\ \mathbf{0} \end{bmatrix}. \quad (3)$$

ここで、 $B \in \mathbb{R}^{M \times M}$ の第 (i, j) 要素は $\psi(\|\mathbf{p}_i - \mathbf{p}_j\|_2)$ 、 $P \in \mathbb{R}^{M \times 4}$ の第 i 行は $[1, x_i, y_i, z_i]$ である。また、右辺ベクトル $\mathbf{h} = [h_1, h_2, \dots, h_M]^T \in \mathbb{R}^M$ の第 i 要素 h_i は、 \mathbf{p}_i が曲面上の点であれば 0、曲面の内側であれば正の値、外側であれば負の値として与える。

本論文で提案する色境界を含む色関数はこの RBF を用いた方法を応用することで実現される (3.2.2)。

2.2 MPU 法による陰関数曲面の生成

N 個の離散点 $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N\}$ と対応する単位法線ベクトル $\{\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_N\}$ が与えられているものとする。

MPU 法では、まず、離散点 $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N\}$ を対角長が 1 の立方体セル内に正規化し、このセルを再帰的に 8 分割することで複数の小領域セルに分割する。分割後の全セル数を n とする。このとき、陰関数 $f(\mathbf{x})$ は次のように構成される。

$$f(\mathbf{x}) = \sum_{k=1}^n \phi^{(k)}(\mathbf{x}) Q^{(k)}(\mathbf{x}).$$

$Q^{(k)}(\mathbf{x})$ は第 k 小領域の曲面を近似する陰関数であり、以下、ローカル形状関数と呼ぶ。これは最小二乗法により二次関数をフィッティングすることで決定される。また、 $\phi^{(k)}(\mathbf{x})$ は重み関数であり、第 k 小領域セルの中心座標 $\mathbf{o}^{(k)}$ と重み関数のサポート半径 $R^{(k)}$

及び二次の B スプライン $b_2(t)$ により次のように定義される。

$$\phi^{(k)}(\mathbf{x}) \equiv \frac{\omega^{(k)}(\mathbf{x})}{\sum_{j=1}^n \omega^{(j)}(\mathbf{x})},$$

$$\omega^{(k)}(\mathbf{x}) \equiv b_2\left(\frac{3\|\mathbf{x} - \mathbf{o}^{(k)}\|_2}{2R^{(k)}}\right).$$

これにより、ローカル形状関数 $Q^{(k)}(\mathbf{x})$ は滑らかに結合される。ここで、サポート半径 $R^{(k)}$ はセルの中心 $\mathbf{o}^{(k)}$ から半径 $R^{(k)}$ の領域内に離散点が一定数以上含まれるように、次のように決定される。まず、第 k 小領域セルの対角長 $d^{(k)}$ に対し、初期値を $R^{(k)} = \alpha d^{(k)}$ とする。 α はパラメータであり、ここでは $\alpha = 0.75$ とする。次に $\hat{R}^{(k)} = R^{(k)}$ とおき、半径 $\hat{R}^{(k)}$ の領域に含まれる離散点数が N_{\min} に満たないときはこれを満たすまで $\hat{R}^{(k)} = \hat{R}^{(k)} + \lambda R^{(k)}$ として半径を繰り返し拡大する。ただし、 λ は $\hat{R}^{(k)}$ の増分を決定するパラメータである。

以上が MPU 法における陰関数曲面の生成法である。本章ではこれを応用し、領域内の色の分布を表す関数を生成する方法について示す。

3. MPU 法に基づく色関数

本研究では、曲面上の色情報を表す関数を MPU 法に基づいて構成する方法を提案する。同法では、離散点群が存在する空間を再帰的に 8 分割した各小領域セルに対して局所的に定義される色関数（ローカル色関数）を構成し、その後すべてのローカル色関数を結合することで一つの色関数を構成する。また、ローカル色関数はセル内の色情報によって適応的に構成を変える。すなわち、色情報の変化が小さい場合は最小二乗法によりフィッティングした二次関数で構成し、色情報が不連続的に変化する場合は RBF を用いた関数で構成することで、色境界をもつ点群や大規模な点群への適用が可能となる。

3.1 MPU 法に基づく色関数の生成

三次元空間内の領域 Ω 内に異なる N 個の離散点 $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N\}$ と対応する色情報 $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_N\}$ が与えられたとき、陰関数曲面上に色情報を与える関数を構成することを考える。ただし、 \mathbf{c}_i は $L^*a^*b^*$ 表色系の各刺激 L^* , a^* , b^* を成分とするベクトルとし、 $\mathbf{c}_i = [L_i, a_i, b_i]^T$ で表す。また、 L^* , a^* , b^* の値域は、それぞれ $[0, 100]$, $[-127, 128]$, $[-127, 128]$ である。本研究で提案する

色関数は、 $L^*a^*b^*$ 表色系の各刺激 L^* , a^* , b^* に対応する色関数 $f_L(\mathbf{x})$, $f_a(\mathbf{x})$, $f_b(\mathbf{x})$ を構成することで、陰関数曲面上の任意の場所に色情報を付加するものである。

本研究で $L^*a^*b^*$ 表色系を用いているのは、より自然に色境界を検出するためである。ここでの色境界とは、曲面上での色の変化が不連続となる場所のことであり、これを表現するためには色関数を不連続関数として構成する必要がある。以下では $f_L(\mathbf{x})$ の構成法のみを示すが、 $f_a(\mathbf{x})$, $f_b(\mathbf{x})$ も同様である。

色関数は MPU 法と同様に、まず、点群を対角長が 1 の立方体セルに正規化し、このセルに対し、八分木法に基づく細分化を行うことで小領域に分割する。次に、各小領域セル内の離散点と色情報から最小二乗法と RBF に基づく方法により局所的に定義される色関数（ローカル色関数）を構成する。ただし、ローカル色関数が所望の精度を満たさない場合はセルを細分化し、分割されたセルに対してローカル色関数を構成することを再帰的に繰り返す。最後に、各領域で得られたローカル色関数を重み関数を介して結合することで、領域 Ω 内で定義される色関数を構成する。

第 k 番目の領域のローカル色関数を $Q_L^{(k)}(\mathbf{x})$ とし、全体の色関数 $f_L(\mathbf{x})$ を

$$f_L(\mathbf{x}) = \sum_{k=1}^n \phi^{(k)}(\mathbf{x}) Q_L^{(k)}(\mathbf{x})$$

で表す。ただし、 n は全セル数、 $\mathbf{o}^{(k)}$, $R^{(k)}$, $\phi^{(k)}(\mathbf{x})$ はそれぞれ、MPU 法と同様に、第 k 番目のセルの中心、サポート半径、重み関数である。

セル分割の収束判定は、

$$\max_{\|\mathbf{p}_i - \mathbf{c}^{(k)}\|_2 < \alpha d^{(k)}} \frac{|Q_L^{(k)}(\mathbf{p}_i) - L_i|}{r_L} \cdot |\phi^{(k)}(\mathbf{p}_i)| < \varepsilon_{\text{col}} \quad (4)$$

で行う。ただし、 r_L は L^* の値域幅であり、 $r_L = 100$ である。同様に、 a^* , b^* の値域幅 r_a , r_b は、それぞれ $r_a = 255$, $r_b = 255$ である。(4) では関数値と目的値の差分を値域幅 r_L で割って相対的な値を求めた後、 $\phi^{(k)}(\mathbf{x})$ による重み付けを行っている。(4) を満たさなければ、セルを分割し、分割されたセルでローカル色関数を構成する。(4) を満たせば $Q_L^{(k)}$ をこのセルのローカル色関数として採用する。すなわち、色関数の精度はしきい値 ε_{col} で決定される。

3.2 ローカル色関数

ローカル色関数 $Q_L^{(k)}$ は領域内の色データに応じて Type(A) : 色境界をもたないローカル色関数と Type(B) : 色境界をもつローカル色関数に分類する.

以下では, k 番目のセルの中にある離散点を $\mathcal{P}^{(k)}$, $\mathcal{P}^{(k)}$ 内の離散点の数を $N^{(k)}$ とする. また, Type(A) と Type(B) のどちらを用いるかは以下の判定で決定する. まず, 領域内の離散点の数 $N^{(k)}$ が, $N^{(k)} > N_{\max}$ ならば色境界を評価するための点が多すぎるとしてセルを分割する. ただし, パラメータ N_{\max} を大きくすれば, 色境界でより安定した色関数を構成することができるが, 内部で解く連立一次方程式の規模の増大に直結するため, 演算量の増加につながる. また, N_{\max} を小さくしすぎると, 多くのローカル色関数セルを生成するため, ローカル色関数の構成に多くの時間やメモリが必要となる.

$N^{(k)} \leq N_{\max}$ ならば, 色境界が存在するかどうかを

$$\max_{\mathbf{p}_i, \mathbf{p}_j \in \mathcal{P}^{(k)}} \|\mathbf{c}_i - \mathbf{c}_j\|_2 > \gamma \quad (5)$$

で評価する. (5) は領域内の色情報同士の色差がしきい値 γ よりも大きければ, この領域内に色境界が存在するとして Type(B) を適用し, (5) を満たさない場合は Type(A) を適用する. 処理の単純化のため, 色境界の評価は L^* , \mathbf{a}^* , \mathbf{b}^* で共通にしている. 以上より, 色境界を表現可能な色関数を構成することができる. 以下では, Type(A) と Type(B) それぞれのローカル色関数の構成法を示す.

3.2.1 ローカル色関数 Type(A)

$N^{(k)} \leq N_{\max}$ かつ (5) を満たさない場合, Type(A) として以下の方法でローカル色関数を構成する.

ここではまず, 領域の中心 $\mathbf{o}^{(k)}$ 上の法線 $\mathbf{w}^{(k)}$ を,

$$\mathbf{w}^{(k)} \equiv \frac{\sum_{i=1}^{N^{(k)}} \omega^{(k)}(\mathbf{p}_{\sigma(i)}) \mathbf{n}_{\sigma(i)}}{\left\| \sum_{i=1}^{N^{(k)}} \omega^{(k)}(\mathbf{p}_{\sigma(i)}) \mathbf{n}_{\sigma(i)} \right\|_2}$$

で定義する. ただし, $\sigma(i)$ は $\mathcal{P}^{(k)}$ に属する i 番目の離散点の全体離散点番号を表す. 次に, $\mathbf{o}^{(k)}$ を原点とするローカル座標 (u, v, w) を導入する. ただし, w を $\mathbf{w}^{(k)}$ の正の向きとし, (u, v) 平面は $\mathbf{w}^{(k)}$ と直交するものとする (図 1(a) 参照). また, (u, v, w) は座標変換された \mathbf{x} の座標である. ここで (u, v) 平面は局所的に曲面を近似するものとみなせる.

ローカル形状関数は x, y, z に関する二次関数

$$Q_L^{(k)}(\mathbf{x}) = w - \mathbf{t}^T A \mathbf{t}, \quad \mathbf{t} = [u, v, 1]^T \quad (6)$$

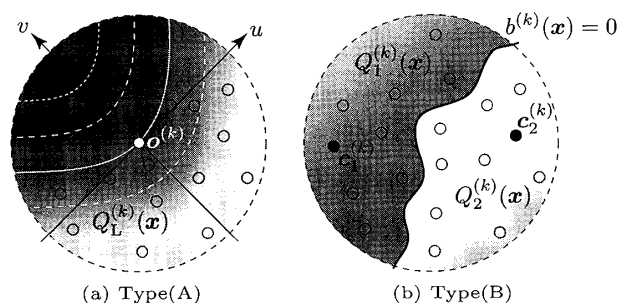


図 1 ローカル色関数

Fig. 1 Local color functions.

で表す, ただし, $A \in \mathbb{R}^{3 \times 3}$ は対称行列であり, 二次関数の係数を表す. このとき, 未知の係数 A は,

$$\sum_{i=1}^{N^{(k)}} \omega^{(k)}(\mathbf{p}_{\sigma(i)}) (Q_L^{(k)}(\mathbf{p}_{\sigma(i)}) - L_{\sigma(i)})^2$$

を最小化することで決定する.

3.2.2 ローカル色関数 Type(B)

ローカル色関数は, 計算量が少ないことから, 基本的に Type(A) を用いて二次式で生成する. しかしながら, Type(A) で生成される連続な二次式では表現できない色分布も存在する. すなわち, Type(A) で得られるローカル色関数では色境界を表現することはできない. 一般に, 色の分布は連続的とは限らず, ある場所を境に色情報が不連続的に変化することはしばしば生じる. このような色境界に対し, 区分的なローカル色関数を以下のように構成することを考える.

まず, 図 1(b) に示すように, 領域内の離散点を色情報に基づく二つのクラスタに分け, 各々のクラスタに対し, RBF に基づく色関数 $Q_1^{(k)}(\mathbf{x})$, $Q_2^{(k)}(\mathbf{x})$ を構成する. 次に, 色境界を表す陰関数曲面 (色境界関数) $b^{(k)}(\mathbf{x}) = 0$ を RBF に基づく方法により構成する. この色境界関数は第 1 クラスタの領域で正, 第 2 クラスタの領域で負, 色境界で零となるように決定する. すなわち, 色境界関数が零以上ならば第 1 クラスタで構成した色関数を, 色境界関数が負ならば第 2 クラスタで構成した色関数を採用することで, 各々のクラスタで構成した色関数を結合することができる.

各クラスタでのローカル色関数 $Q_1^{(k)}(\mathbf{x})$, $Q_2^{(k)}(\mathbf{x})$ を求めるために, まず,

$$\|\mathbf{c}_1^{(k)} - \mathbf{c}_2^{(k)}\|_2 = \max_{i,j} \|\mathbf{c}_{\sigma(i)} - \mathbf{c}_{\sigma(j)}\|_2$$

となる $\mathbf{c}_1^{(k)}$, $\mathbf{c}_2^{(k)}$ を探す. 次に, 領域内の点に対し

$\mathbf{c}_1^{(k)}$ と $\mathbf{c}_2^{(k)}$ に関する色差によるクラスタリングを行う。すなわち, $i = 1, 2, \dots, N^{(k)}$ に対し,

$$\|\mathbf{c}_1^{(k)} - \mathbf{c}_{\sigma(i)}\|_2 \leq \|\mathbf{c}_2^{(k)} - \mathbf{c}_{\sigma(i)}\|_2 \quad (7)$$

を満たすならば $\mathbf{p}_{\sigma(i)} \in \mathcal{P}_1^{(k)}$ とし, (7) を満たさなければ $\mathbf{p}_{\sigma(i)} \in \mathcal{P}_2^{(k)}$ とする。ただし, $\mathcal{P}_1^{(k)}$, $\mathcal{P}_2^{(k)}$ はそれぞれ, $\mathbf{c}_1^{(k)}$, $\mathbf{c}_2^{(k)}$ に対応するクラスタに属する離散点である。各クラスタに対応するローカル色関数 $Q_1^{(k)}(\mathbf{x})$, $Q_2^{(k)}(\mathbf{x})$ の構成方法に差はないため, 以下では $Q_1^{(k)}(\mathbf{x})$ の構成法のみを示す。

$Q_1^{(k)}(\mathbf{x})$ は,

$$Q_1^{(k)}(\mathbf{x}) = \sum_{i=1}^{N_1^{(k)}} \lambda_{(1,i)} \psi(\|\mathbf{x} - \mathbf{p}_{\sigma(1,i)}\|_2) + q_{(1,0)} + q_{(1,1)}x + q_{(1,2)}y + q_{(1,3)}z, \quad (8)$$

$$\psi(r) = \left(1 - \frac{r}{R}\right)_+^2 \quad (9)$$

によって構成する。ただし, $N_1^{(k)}$ は $\mathcal{P}_1^{(k)}$ に属する離散点の数, $\sigma(1,i)$ は $\mathcal{P}_1^{(k)}$ に属する i 番目の離散点の全体離散点番号とし, R は RBF のサポート半径である。

このとき, $\boldsymbol{\lambda}_1 \equiv [\lambda_{(1,1)}, \lambda_{(1,2)}, \dots, \lambda_{(1,N_1^{(k)})}]^T$ と $\mathbf{q}_1 \equiv [q_{(1,0)}, q_{(1,1)}, q_{(1,2)}, q_{(1,3)}]^T$ を未知数として (3) と同様に連立一次方程式

$$\begin{bmatrix} B_1 & P_1 \\ P_1^T & O \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_1 \\ \mathbf{q}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{0} \end{bmatrix} \quad (10)$$

を得る。ただし, $B_1 \in \mathbb{R}^{N_1^{(k)} \times N_1^{(k)}}$ の (i,j) 要素は $B_{1,ij} \equiv \psi(\|\mathbf{p}_{\sigma(1,i)} - \mathbf{p}_{\sigma(1,j)}\|_2)$ であり, $P_1 \in \mathbb{R}^{N_1^{(k)} \times 4}$ の i 行目は $P_{1,i} \equiv [1, x_{\sigma(1,i)}, y_{\sigma(1,i)}, z_{\sigma(1,i)}]$ とする。また, $\mathbf{h}_1 \equiv [L_{\sigma(1,1)}, L_{\sigma(1,2)}, \dots, L_{\sigma(1,N_1^{(k)})}]^T$ で定義する。(10) を解けば, 各クラスタの色関数を構成することができる。

(8) で構成したローカル色関数 $Q_1^{(k)}(\mathbf{x})$, $Q_2^{(k)}(\mathbf{x})$ は, 色境界関数

$$b^{(k)}(\mathbf{x}) = \sum_{i=1}^{N^{(k)}} \lambda_i \psi(\|\mathbf{x} - \mathbf{p}_{\sigma(i)}\|_2) + q_0 + q_1x + q_2y + q_3z, \quad (11)$$

で結合する。この色関数 $b^{(k)}(\mathbf{x})$ も RBF に基づいて構成することを考える。すなわち, $\boldsymbol{\lambda} \equiv$

$[\lambda_1, \lambda_2, \dots, \lambda_{N^{(k)}}]^T$ と $\mathbf{q} \equiv [q_0, q_1, q_2, q_3]^T$ を未知数とし, $B \in \mathbb{R}^{N^{(k)} \times N^{(k)}}$ の (i,j) 要素を $B_{ij} \equiv \psi(\|\mathbf{p}_{\sigma(i)} - \mathbf{p}_{\sigma(j)}\|_2)$, $P \in \mathbb{R}^{N^{(k)} \times 4}$ の i 行目を $P_i \equiv [1, x_{\sigma(i)}, y_{\sigma(i)}, z_{\sigma(i)}]$, $\mathbf{h} \in \mathbb{R}^{N^{(k)}}$ を $\mathbf{h} \equiv [h_1, h_2, \dots, h_{N^{(k)}}]^T$ で定義すれば, 以下の連立一次方程式を得る。

$$\begin{bmatrix} B & P \\ P^T & O \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{q} \end{bmatrix} = \begin{bmatrix} \mathbf{h} \\ \mathbf{0} \end{bmatrix}. \quad (12)$$

ただし, 右辺ベクトル \mathbf{h} は以下のように定義する。

$$h_i = \begin{cases} 1 & ; \mathbf{p}_{\sigma(i)} \in \mathcal{P}_1^{(k)} \\ -1 & ; \mathbf{p}_{\sigma(i)} \in \mathcal{P}_2^{(k)} \end{cases}.$$

(12) を解くことで, $b^{(k)}(\mathbf{x})$ が生成され, このとき, $b^{(k)}(\mathbf{x}) = 0$ が色境界を表す (図 1(b) 参照)。最後に,

$$Q_L^{(k)}(\mathbf{x}) = \begin{cases} Q_1^{(k)}(\mathbf{x}) & ; b^{(k)}(\mathbf{x}) \geq 0 \\ Q_2^{(k)}(\mathbf{x}) & ; b^{(k)}(\mathbf{x}) < 0 \end{cases}$$

によって各クラスタで構成した色関数を結合する。ただし, Type(B) で構成されるローカル色関数は $\mathcal{P}^{(k)}$ 上で所望の精度を満たすと考えられるため, (4) による精度の判定やセルの分割を行う必要はない。これは RBF を用いた補間が, 与えられた節点上において厳密な補間であるためである。

4. 数値実験

本章では, 陰関数曲面上に色情報を付加する問題に対し, 本研究で提案する色関数生成手法の有効性を数値実験によって検証する。本章を通し, 陰関数曲面は MPU 法で構成するものとし, MPU 法と色関数の共通のパラメータは文献 [8] と同様に, $\alpha = 0.75$, $\lambda = 0.1$, $N_{\min} = 15$ とする。また, 陰関数曲面の収束判定のためのしきい値を $\varepsilon_0 = 1 \times 10^{-4}$ とし, 色関数の収束判定のしきい値を $\varepsilon_{\text{col}} = 1 \times 10^{-3}$ とする。更に, 色境界を検出するためのパラメータを $\gamma = 40.0$ とする。また, 特に断らない限り, $N_{\max} = 200$ とし, RBF のサポート半径を $R = 0.4R^{(k)}$ とする。なお, (10), (12) の解法には, 係数行列の構造と対称性を考慮した直接法 [13], [14] を用いた。

数値実験を通して, 曲面上に色を与えるときには, $f_L(\mathbf{x})$, $f_a(\mathbf{x})$, $f_b(\mathbf{x})$ からそれぞれ得られた L^* , a^* , b^* の値を, R, G, B にそれぞれ変換したものを与えた。ただし, R, G, B の値域はすべて $[0, 1]$ である。

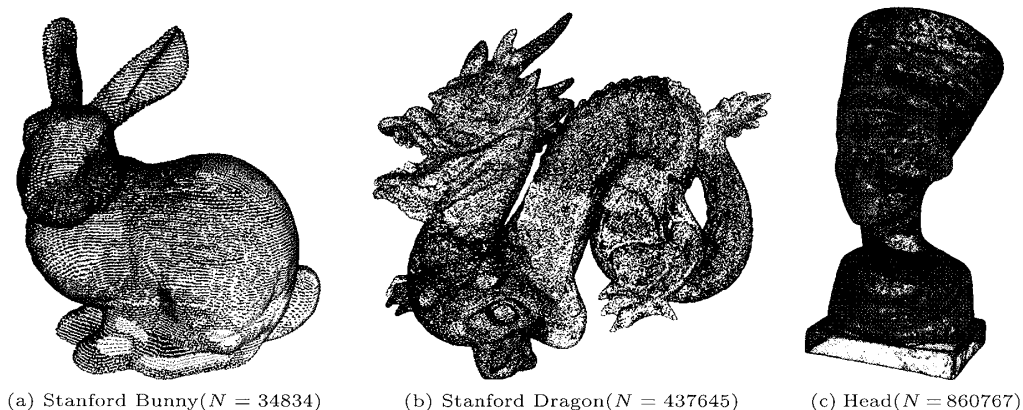


図 2 色情報付き離散点データ
Fig. 2 Point datasets with color distributions.

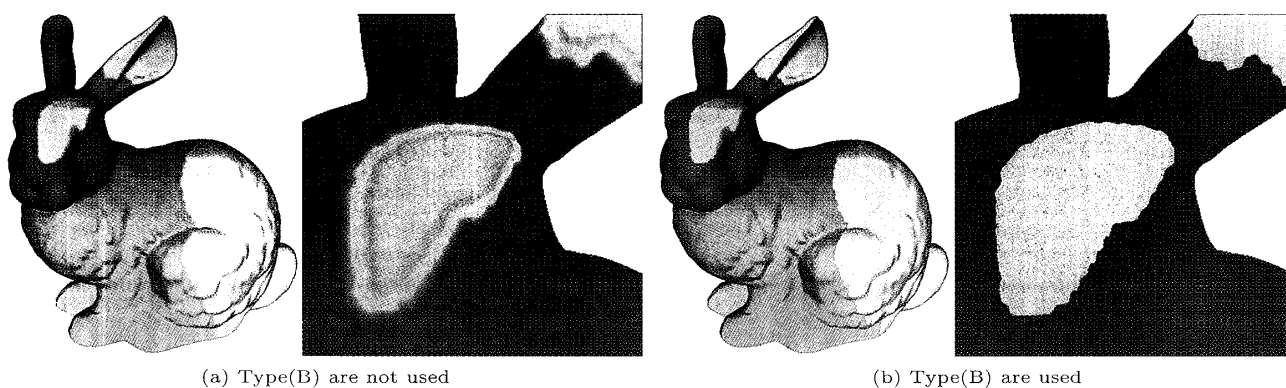


図 3 ローカル色関数 Type(A) と Type(B) の比較
Fig. 3 Comparison of the local color functions Type(A) and Type(B).

また、誤差計算をする際も、変換後の値を用いた。

本章では色情報をもつ離散点として、Stanford Bunny, Stanford Dragon, Head を用いる (図 2 参照)。また、実験環境は、CPU: Intel Pentium4 2.8 GHz, メモリ: 2.0 GByte, OS: Red Hat Linux 9 (Kernel: 2.4.20), コンパイラ: g++ Ver. 3.2.2 である。

4.1 色境界における色関数

ここでは、まず、ローカル色関数 Type(A) のみで得られる色関数と Type(B) も用いて得られる色関数を比較し、色境界の再現性を検証する。図 3 に Type(A) のみで構成した色関数と色境界に Type(B) を適用した場合の Stanford Bunny の再構成結果を示す。ただし、各図の左側に全体の再構成結果を示し、右側に再構成結果の一部の色情報のみを拡大表示している。図 3(a) では滑らかな色関数を構成できているが、色境界で色関数が不鮮明となっている。一方、図 3(b)

表 1 Type(A) のみで色関数を生成したときの CPU 時間と絶対誤差の平均 ($N_{\max} = 2000$)

Table 1 CPU time and average of absolute error using only Type(A) for each model.

離散点群	$t_{\text{col}}[\text{s}]$	e_{col}
Stanford Bunny	1.7	9.4×10^{-3}
Stanford Dragon	81.6	1.9×10^{-2}
Head	93.2	1.2×10^{-2}

表 2 Type(A) と Type(B) を用いて色関数を生成したときの CPU 時間と絶対誤差の平均

Table 2 CPU time and average of absolute error using Type(A) and Type(B) for each model.

離散点群	$t_{\text{rec}}[\text{s}]$	$t_{\text{col}}[\text{s}]$	e_{col}
Stanford Bunny	35.2	6.7	4.2×10^{-4}
Stanford Dragon	79.7	85.8	6.8×10^{-4}
Head	95.1	184.0	2.8×10^{-3}

では、色境界に Type(B) を用いることで、色境界を鮮明に表現できることが確認される。

次に、Type(A) のみを用いたとき及び Type(A) と

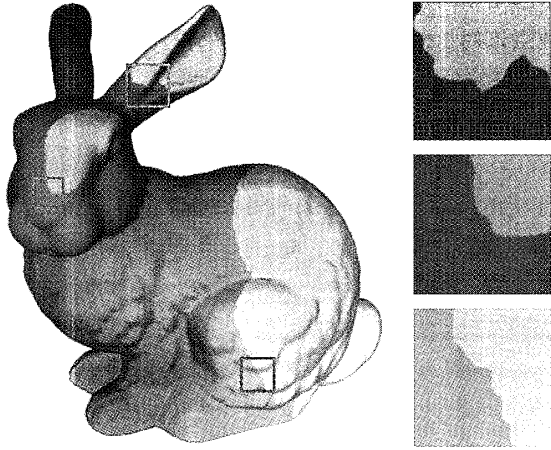


図 4 Stanford Bunny の再構成結果
Fig. 4 Reconstruction of the Stanford Bunny model.



図 5 Stanford Dragon の再構成結果
Fig. 5 Reconstruction of the Stanford Dragon model.



図 6 Head の再構成結果
Fig. 6 Reconstruction of the Head model.

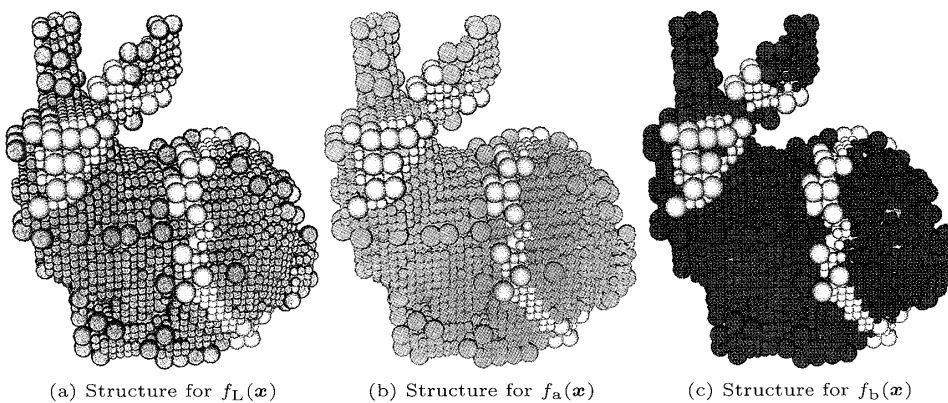


図 7 Stanford Bunny の各色関数における Type(A) 及び Type(B) の分布
Fig. 7 Structure of Type(A) and Type(B) for each color function.

Type(B) を用いたときの色関数生成に要した CPU 時間 t_{col} と絶対誤差の平均 e_{col} を、表 1 及び表 2 に示す。ただし、Type(A) のみを用いる場合には、 N_{max} を大きめにとっても計算量への影響は小さいため、 $N_{max} = 2000$ とした。また、表 2 には t_{rec} として、陰関数生成に要した CPU 時間も示したが、これは色関数生成には影響を及ぼさず、Type(A) のみを用いて色関数を生成したときも、全く同じ CPU 時間となる。 e_{col} を求める際には、まず関数 $f_L(\mathbf{x})$, $f_a(\mathbf{x})$, $f_b(\mathbf{x})$ から得られた値を R, G, B に変換した。その後、各節点上の L^* , a^* , b^* の値も R, G, B に変換し、関数から得られた R, G, B の値と節点上の R, G, B の値のそれぞれの差分の絶対値の和を求めた。全節点においてこの操作を行い、絶対値の和の合計を平均したものが、 e_{col} である。表 1 と表 2 の結果を比較すると、まず、Type(A) のみを用いたときの方が CPU 時間は短い。これは、Type(A) と Type(B) を用いる場合に

深く領域分割をしたあとで Type(A) または Type(B) でローカル色関数を生成するが, Type(A) のみを使う場合には, Type(A) と Type(B) を使う場合と比較して比較的浅い領域分割の段階でローカル色関数を生成できているからである. また, Type(B) で現れるような連立一次方程式を解く必要がないことが CPU 時間の減少につながっていると考えられる. 一方, 計算精度は Type(A) と Type(B) を用いたときの方がよい. これは, Type(A) のみを用いた場合に色境界を表すことができなため, 色境界付近における精度が悪くなっていると考えられる. したがって, 計算精度の面からも色境界がある場合には Type(B) も使用して再構成すべきである.

次に, Stanford Bunny と Stanford Dragon の再構成結果と色境界の拡大図をそれぞれ, 図 4, 図 5 に示

す. 同図より, 十分に拡大した状態であっても色境界を鮮明に表現できていることが分かる. また, 一般的な色情報をもつデータ (Head) から得た陰関数曲面上に本研究で提案する色関数によって色情報を与えた結果を図 6 に示す. 同図より, 一般的な色情報をもった離散点に対し, 十分な再構成結果が得られることが確認される.

ここで, Stanford Bunny 及び Head の各色関数における Type(A) と Type(B) の分布図をそれぞれ図 7 及び図 8 に示す. 両図の (a), (b), (c) がそれぞれ $f_L(x)$, $f_a(x)$, $f_b(x)$ における Type(A) と Type(B) の分布図である. Type(A) 及び Type(B) の領域は両方とも球で表されており, 実際の領域のサポート半径の半分の大きさで描いた. ただし, 図中の白球が Type(B) の領域である. また, 赤球, 緑球, 青球の領域は, それぞれ $f_L(x)$, $f_a(x)$, $f_b(x)$ における Type(A) の領

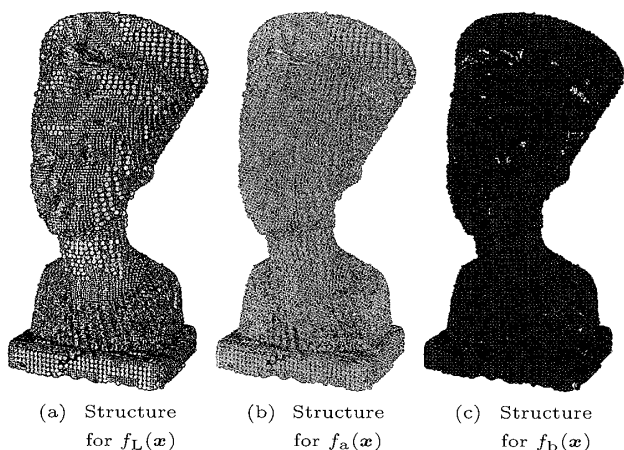


図 8 Head の各色関数における Type(A) 及び Type(B) の分布

Fig. 8 Structure of Type(A) and Type(B) for each color function.

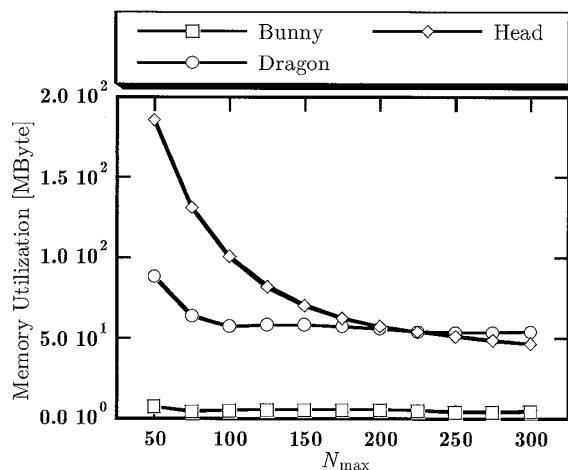


図 9 メモリ使用量の N_{max} 依存性
Fig. 9 Relation between N_{max} and memory utilization for each model.

表 3 領域数の N_{max} 依存性
Table 3 Relation between N_{max} and the number of areas for each model.

N_{max}	Bunny		Dragon		Head	
	Type(A)	Type(B)	Type(A)	Type(B)	Type(A)	Type(B)
50	103301	3291	1243076	33804	3456945	1758
75	48492	2016	748250	25554	2413835	2169
100	44367	2067	555378	23007	1822223	2582
125	40608	2088	475836	22053	1449534	2906
150	35628	1965	398713	20646	1203459	3155
175	31050	1839	340200	18882	1037000	3406
200	25392	1680	289671	17079	918312	3619
225	19782	1431	246444	15408	822572	3795
250	14148	1143	214203	14280	737249	3984
275	12543	1089	189504	13485	664466	4086
300	11880	1101	169164	12846	602927	4179

表 4 CPU 時間 [s] の N_{max} 依存性
Table 4 Relation between N_{max} and CPU time for each model.

N_{max}	Bunny	Dragon	Head
50	7.6	108.0	313.5
75	5.3	86.3	257.3
100	5.1	78.8	227.4
125	5.0	77.1	206.0
150	5.1	78.0	194.9
175	5.4	80.7	187.0
200	6.7	85.8	184.0
225	8.1	94.4	185.3
250	11.7	109.4	193.6
275	12.7	131.4	209.3
300	14.4	167.7	232.2

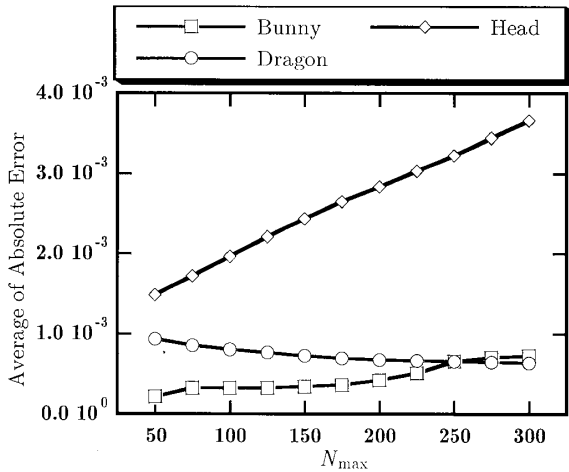


図 10 絶対誤差の平均の N_{max} 依存性
 Fig. 10 Relation between N_{max} and average of absolute error for each model.

域である。

4.2 定量的評価

本節では、 N_{max} の値を変化させたときの、必要メモリ、CPU 時間、絶対誤差の平均の比較を行う。ただし、必要メモリを正確に量るのは難しいため、Type(A) と Type(B) の数から概算した。Type(A) で最低限格納しておく必要のある情報は、Type 識別情報、サポート半径 $R^{(k)}$ 、中心座標 $(o_x^{(k)}, o_y^{(k)}, o_z^{(k)})$ 、法線情報 $(n_x^{(k)}, n_y^{(k)}, n_z^{(k)})$ 、二次式の係数六つである。簡単化のため、これらの情報を 32 ビット整数型または単精度浮動小数点型で格納するとする。32 ビット整数型と単精度浮動小数点型は、一般に一つの情報ごとに 4 バイト使用する。したがって、 $14 \times 4 = 56$ バイトが Type(A) の個数分使用される。一方、Type(B) で最低限格納しておく必要のある情報は、まず Type(A)

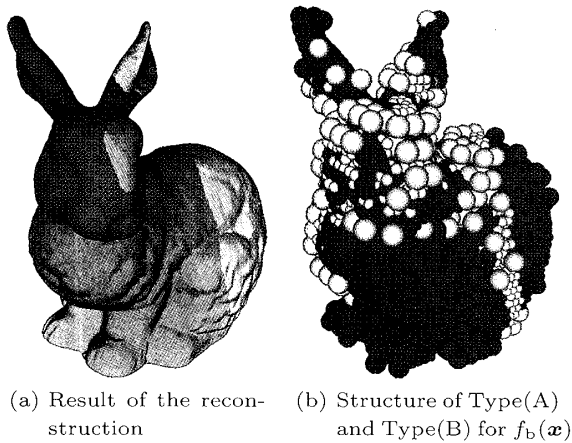


図 11 ノイズを含んだときの再構成結果
 Fig. 11 Reconstruction of the Stanford Bunny model with noise.

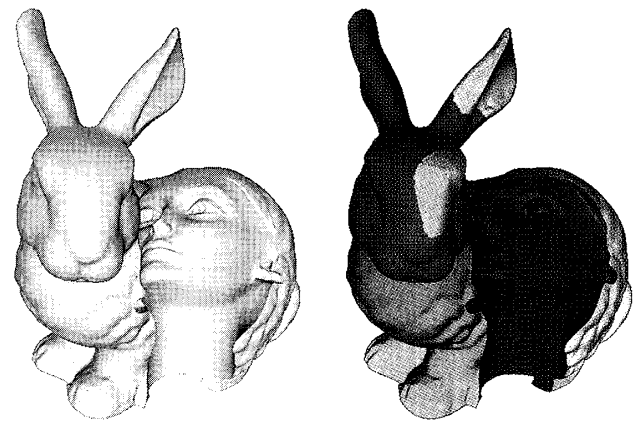


図 13 CSG への適用例
 Fig. 13 CSG operations.

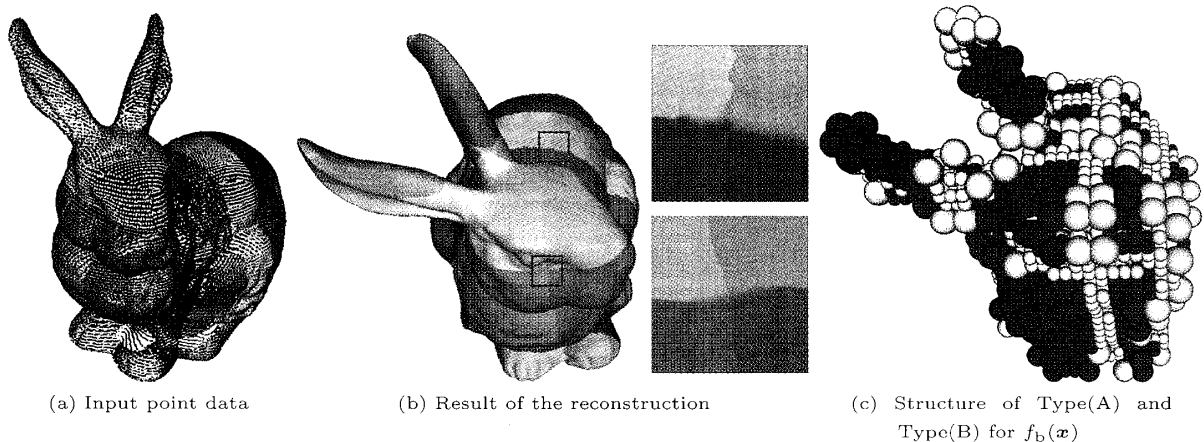


図 12 色の異なる領域が三つ以上混在する場合の再構成結果
 Fig. 12 Reconstruction of color distributions for the reason which consist of more than different 3 colors.

の二次式の係数六つ以外は同様に格納する必要がある。これらに加えて、連立一次方程式の解と連立一次方程式を作るときに使用した節点のインデックスを格納する必要がある。複数の連立一次方程式を解く必要があり、最大で $(2N_{\max} + 12)$ 個の解と N_{\max} 個のインデックスを格納することになる。簡単のため、連立一次方程式の解とインデックスは最大数でカウントすると、 $(8 + (2N_{\max} + 12) + N_{\max}) \times 4 = 12N_{\max} + 80$ バイトが Type(B) の個数分使用される。

表 3 に Type 別領域数の N_{\max} 依存性を離散点群ごとに示した。また、同表のデータから、上述の概算法に従ってメモリ使用量を算出した結果を図 9 に示した。図 9 より、 N_{\max} が小さいとき、特に Head と Dragon のメモリ使用量が増加している。また、 N_{\max} を大きくしていくと、メモリ使用量は徐々に減少していき、途中からほとんど一定になった。

一方、CPU 時間の N_{\max} 依存性を表 4 に示した。CPU 時間も N_{\max} を小さくしすぎたとき増加してしまう。これは表 3 から分かるように、 N_{\max} を小さくしたとき、Type(A) の領域が増加してしまうからである。その一方で、 N_{\max} を大きめにとっても CPU 時間は増加してしまう。これは Type(B) のローカル色関数を生成する際に解く必要のある連立一次方程式の規模が N_{\max} の増加に伴い大きくなってしまうことが原因である。

図 10 には、絶対誤差の平均の N_{\max} 依存性を示した。同図より N_{\max} の変化に伴って誤差の値に変化はあるものの、いずれも 10^{-2} 未満である。

これらの結果をまとめると、メモリ使用量と CPU 時間を抑えるために、適切に N_{\max} の値を選ぶ必要があることが分かる。今回の例で具体的な範囲を挙げると、 $[150, 225]$ 程度の値に N_{\max} を定めるとよい。

4.3 ノイズ入りデータにおける色関数

本節では、ノイズ入りデータから色関数を生成する。Stanford Bunny の色情報にホワイトノイズが混入した場合の再構成結果を図 11(a) に示す。また、図 11(b) に $f_b(\mathbf{x})$ の Type(A) と Type(B) の分布図を示す。 $f_L(\mathbf{x})$ と $f_a(\mathbf{x})$ の分布図は示さなかったが、 $f_b(\mathbf{x})$ とほぼ同様の分布である。図 11(a) より、再構成結果にノイズの影響が現れていることが分かる。また、図 11(b) より、ノイズによって本来境界ではない場所で、Type(B) が選択されていることが分かる。すなわち、ノイズが大きい場合、境界ではないはずのところが境界とみなされてしまう。

4.4 3色以上の境界が存在する場合の色関数

本節では、色分布が複雑な場合として 3 色以上の色境界をもつ場合を考える。3 色以上の色境界が存在する色情報をもった Stanford Bunny を図 12(a) に示す。図 12(b) はその再構成結果であり、同図の右側には 3 色以上の色境界が存在する領域のうち、2 個所の拡大図も示した。また、 $f_b(\mathbf{x})$ の Type(A) と Type(B) の分布図を図 12(c) に示した。 $f_L(\mathbf{x})$ と $f_a(\mathbf{x})$ の分布図は示さなかったが、 $f_b(\mathbf{x})$ とほぼ同様の分布である。図 12(c) より、図 12(b) の拡大図の部分では、Type(B) が選択されていることが分かる。また、同拡大図を見ると、二つの拡大図においてともに縦方向の境界は鮮明に表現できているが、それ以外の境界はぼやけてしまっていることが分かる。これは Type(B) において、クラスタの数が二つしかないことが原因である。したがって、3 色以上の色境界が存在するとき、境界として鮮明に表せるのは 1 本だけであり、全境界を鮮明に表現することはできない。すなわち、本提案法で鮮明に境界を表せるのは 2 色の境界までである。

4.5 CSG への応用

本節では、陰関数曲面の特徴の一つである CSG が、本研究で提案する色関数をもつ曲面モデルに対しても適用できることを示す。まず、図 13 に Stanford Bunny と Head の差として定義されるモデルを示す。同図より、CSG 操作に対し、正しく色情報を付加できていることが確認される。

5. むすび

本研究では、色情報をもった離散点群から陰関数曲面上の色情報を与える関数を生成する方法として、MPU 法に基づく色関数生成手法を提案した。また、色情報特有の問題である色境界の表現方法を示し、本提案法が大規模な問題や 2 色の色境界をもつモデルに対して適用可能であることを実験的に示した。結論を以下にまとめる。

(1) 色境界では、セル内の離散点を色情報によって二つのクラスタに分け、各々のクラスタで色関数を構成し、RBF を用いた陰関数構成法に基づく色境界関数によって各クラスタで生成した色関数を結合することで、色境界を表現することができる。

(2) 大規模な離散点群や一般の色情報をもった離散点群に対し、提案手法を用いることで十分な再構成結果を得ることができる。ただし、色境界を鮮明に表せるのは 2 色の境界だけであり、3 色以上の境界が存

在する場合、ぼやけてしまう境界が存在する。

(3) 提案手法によって生成された色情報付き陰関数曲面モデルは CSG 操作も適用可能である。

以上より、本研究で提案する色関数は陰関数の零等値面上に色情報を付加する問題に対し十分適用可能であるといえる。しかしながら、ノイズ入りのデータへの対応、3色以上の境界が存在するときの鮮明な色境界生成は今後の課題である。また、本研究で提案する色関数は陰関数の零等値面のみで拘束しているため、陰関数曲面の特徴の一つであるオフセット操作 [8] に適さず、今後解決すべき課題の一つである。

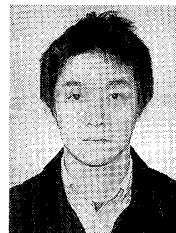
謝辞 実験データを提供して頂きました以下の方々
に深く感謝致します。Stanford 3D Scanning Repository (Stanford Bunny, Stanford Dragon), INUS Technology (Head).

文 献

- [1] J. Bloomenthal, C. Bajaj, J. Blinn, M.P. Cani-Gascuel, A. Rockwood, B. Wyvill, and G. Wyvill, Introduction to Implicit Surfaces, Morgan Kaufmann Publishers, San Francisco, 1997.
- [2] G. Turk and J.F. O'Brien, "Shape transformation using variational implicit functions," Computer Graphics Proceedings, Annual Conference Series (SIGGRAPH 99), vol.33, pp.335-342, 1999.
- [3] G. Turk, H.Q. Dinh, J.F. O'Brien, and G. Yngve, "Implicit surfaces that interpolate," Proc. Shape Modelling International, pp.62-73, 2001.
- [4] V.V. Savchenko, A.A. Pasko, O.G. Okunev, and T.L. Kunii, "Function representation of solids reconstructed from scattered surface points and contours," Computer Graphics Forum, vol.14, no.4, pp.181-188, 1995.
- [5] G. Turk and J.F. O'Brien, "Modelling with implicit surfaces that interpolate," ACM Trans. Graphics, vol.21, no.4, pp.855-873, 2002.
- [6] N. Kojekine, I. Hagiwara, and V. Savchenko, "Software tools using CSRBFs for processing scattered data," Int. J. Comput. Graph., vol.27, no.2, pp.311-319, 2003.
- [7] V. Savchenko, N. Kojekine, and H. Unno, "A practical image retouching method," CW '02: Proc. First International Symposium on Cyber Worlds, pp.480-487, IEEE Computer Society, 2002.
- [8] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel, "Multi-level partition of unity implicits," ACM Trans. Graphics, vol.22, no.3, pp.463-470, 2003.
- [9] R. Franke and G. Nielson, "Smooth interpolation of large sets of scattered data," International Journal of Numerical Methods in Engineering, vol.15, no.11, pp.1691-1704, 1980.

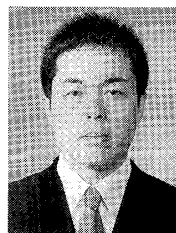
- [10] I. Tobor, P. Reuter, and C. Schlick, "Efficient reconstruction of large scattered geometric datasets using the partition of unity and radial basis functions," WSCG (Winter School of Computer Graphics), vol.12, no.3, pp.467-474, 2004.
- [11] I. Tobor, P. Reuter, and C. Schlick, "Multiresolution reconstruction of implicit surfaces with attributes from large unorganized point sets," Proc. Shape Modelling International, pp.19-30, 2004.
- [12] H. Wendland, "Piecewise polynomial, positive definite and compactly supported radial basis functions of minimal degree," Advances in Computational Mathematics, vol.4, pp.389-396, 1995.
- [13] A. George and J. W-H Liu, Computer Solution of Large Sparse Positive Definite Systems, Prentice-Hall Inc., Englewood Cliffs, NJ, 1981.
- [14] G.H. Golub and C.F. Van Loan, Matrix computations (3rd ed.), Johns Hopkins University Press, Baltimore and London, 1996.

(平成 17 年 4 月 26 日受付, 8 月 23 日再受付)



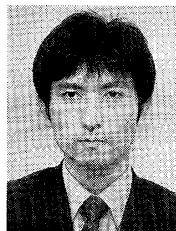
藤本 大地

2003 山形大・工・電子情報工卒。2005 筑波大学院修士課程理工学研究科了。現在、(株)ナナオ勤務。専門分野、数値解析。



伊東 拓 (学生員)

2001 山形大・工・電子情報工卒。2003 筑波大学院修士課程理工学研究科了。現在、同大学院博士課程システム情報工学研究科在学中。専門分野、数値解析。日本応用数学会、情報処理学会各会員。



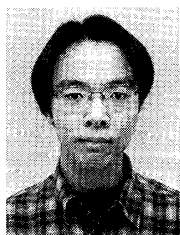
仲田 晋

1996 筑波大・第三学群情報学類卒。1998 同大学院修士課程理工学研究科了。2001 同大学院博士課程理工学研究科了、博士(工学)。同年東京工業大学大学院理工学研究科非常勤研究員。2002 立命館大学理工学部講師。2005 同大学情報理工学部助教授、現在に至る。専門分野、数値解析、逆問題。日本応用数学会、情報処理学会各会員。



北川 高嗣

1978 名大・工卒。1983 同大大学院工学研究科情報工学専攻博士後期課程中退。その後、スタンフォード大学計算機科学科客員研究員、愛媛大学理学部数学科助手、講師、筑波大学電子・情報工学系助教授、教授を経て、現在、同大学システム情報工学研究科教授。工学博士。専門分野、数値解析、逆問題、マルチメディア情報システム、日本応用数学会、情報処理学会各会員。



岡 将史

2004 立命館大・理工・情報卒。現在、同大大学院理工学研究科修士課程在学中。専門分野、コンピュータグラフィックス。



田中 覚 (正員)

1982 早大・理工・応用物理卒。1984 同大大学院理工学研究科物理学及应用物理学専攻博士前期課程了。1987 同大学院理工学研究科物理学及应用物理学専攻博士後期課程了。博士(理学)。1992 福井大学工学部助手。1996 同講師。2000 同助教授。2002 立命館大学工学部教授。2004 立命館大学情報理工学部教授。現在に至る。専門分野、コンピュータグラフィックス、情報処理学会、Eurographics Association 各会員。