

論文

CSRBF 法による表面修復

Using CSRBF for surface retouching

カジョキン ニキタ (東京工業大学大学院理工学研究科) [非会員]

谷口 由紀 (東京工業大学大学院理工学研究科) [非会員]

萩原 一郎 (東京工業大学大学院理工学研究科)  
[正会員#199507539]

サブチェンコ ウラディミール (法政大学情報科学部) [非会員]

#### 連絡先

萩原 一郎

〒152-8552 東京都目黒区大岡山 2-12-1

東京工業大学 大学院理工学研究科 機械物理工学専攻

TEL: (03)5734-3555 FAX: (03)5734-3555

email: hagiwara@mech.titech.ac.jp

## 概要

画像上の傷の修復問題に CSRBF 法を適用する．ここでは，著者らが開発した高速アルゴリズムに基づく CSRBF 法を用いて，画像上についた傷を修復するための手法の新たな開発を行う．本報で開発した手法を用いると，従来の画像の傷修復アルゴリズムではできなかった広範囲に及ぶ傷や大きな傷の修復を，短時間に精度よく行うことができる．また，本報では，テクスチャ模様のような色の変化が大きい画像修復に適している新たな LCSRBF 法や，多角形的な傷の修復問題に対する新しい手法の開発も行い，実際の使用例を提示してこれらの有効性について述べる．

## Abstract

We employ the method of compactly supported radial basis functions(CSRBF) for a problem of fulfillment of damaged surface areas. In this paper we propose a new technique based on the method of CSRBF for retouching surface of damaged areas. By using our method, we can retouch disconnected large areas. It is enough speed and accuracy for practical use. We also develop a new method of LCSRBF(local compactly supported radial basis functions) for image retouching of textured images which have a sharp change in the color and a new technique for surface retouching of polygonal objects with missing or damaged areas in this paper. We show these efficacy by giving experimental results.

## 1. はじめに

画像修復アルゴリズムには、1996年に Hirani and Totsuka<sup>1)</sup>が提案したクローニング法によるアルゴリズムと、2000年に Bertalmio and Sapiro<sup>2)</sup>が提案した偏微分方程式に基づく image-inpainting アルゴリズムが知られている。前者はテクスチャを張り付けることで画像の修復をするのに対し、後者はテクスチャを扱わずに行う。また、2001年には、Oliveira, Bowen, McKenna and Chang<sup>3)</sup>により RGB に基づく修復アルゴリズムが提案された。この手法は、入力画像を RGB の三原色に分けて扱い、これを用いて傷の修復を行うアルゴリズムである。これらのアルゴリズムは、現在、一般的に使われている手法であるが、小さな傷の修復に対してのみ有効であり、傷が大きくなると用いることはできないという欠点がある。また、小さな傷でも、広範囲に散らばっている場合や、多数の傷がある場合には適用することはできない。これに対し、大きな傷の修復問題については、最近の Yamauchi, Haber and Seidel<sup>4)</sup>が提案したテクスチャ合成に基づく手法や Drori, Cohen-Or and Yeshurun<sup>5)</sup>によるイメージ補完に基づく手法が有効であるが、これらの手法は複雑な手続きに伴い計算時間がかかるため、傷が小さい場合には適していない。さらに、テクスチャ模様上の傷の修復については、Hirani らのクローニング法によるアルゴリズムが使われているが、この手法は小さな傷についてでさえも冗長な作業が強いられ計算コストがかかるといった問題点がある。このように画像上についた傷の修復を行うには、実用化の面で多くの問題点があり、さらなる改良が必要である。

そこで本研究では、画像の関数化を行い、それを利用して画像上の傷の修復を行うことを目的とする。物体の関数表現の代表的方法として、1995年に Savchenko, Pasko, Okunev and Kunii<sup>6)</sup>が提案した RBF (radial basis functions) 法がある。この手法はある種のスプライン関数や球関数に基づいており、画像の再構成に対して高レベルの実現性を備えている。そのため、コンピュータグラフィックスの手段として有効である。しかし、この手法は非常に計算時間がかかり、簡単な物体にしか適用できないという欠点がある。従って、ユーザーが使用目的に合わせた高速化を行わない限り、実際に利用することは困難である。2001年に Morse, Yoo, Rheingans, Chen and Subramanian<sup>7)</sup>や Carr, Mitchell, Beaton, Cherrie, Fright, McCallum and Evans<sup>8)</sup>はアルゴリズムを効率的に求める手法の開発を行い RBF 法の高速化を試みたが、多大なメモリが必要であ

り、十分なものではなかった。この他にもいくつかの方法が提案されたが<sup>9),10)</sup>、どれも実用化のレベルには達しておらず、RBF 法は実際にはあまり活用されていない状況にある。これに対し、1995年に Wendland<sup>11)</sup> は基底関数をコンパクトにした CSRBF (compactly supported radial basis functions) 法を提案し、これを用いて物体の関数化を行った。この手法では、コンパクトな台を持つ放射状の基底関数を使って再構成を行うため、RBF 法よりも計算速度は速く、適用範囲は広いという特徴を持つ。しかし、計算時間は実用化するのに十分な速さではない。

そこで、著者等は過去の研究において、CSRBF 法を用いた新たな高速アルゴリズムの開発を行った<sup>12),13)</sup>。これは計算上構成される補間行列 (SLAE 行列) を、バンド性を持つ対角行列に作り替えるためのアルゴリズムで、必要なパラメータを高速に計算することができる。このような高速アルゴリズムに基づく CSRBF 法は実用化するのに十分な速さがあり、大きな入力データの処理にも有効である。さらに著者等が提案したアルゴリズムは、データの格納のためのメモリが少なく済む、という利点も備えている。最近では、Ohtake, Belyaev and Seidel<sup>14)</sup> が多重スケール技法により SLAE 行列をいくつかの小さな行列に分割する手法を開発しているが、これは入力データが極めて大きい場合 (数百万点) に対して有効であり、ここで扱うような数千点から構成される画像に用いると1つの SLAE 行列を解くよりも計算時間がかかってしまう。ここでは、このような著者等が開発した高速アルゴリズムに基づく CSRBF 法を画像上の傷の修復問題に適用し、実用化するのに十分な速さである手法の開発を行う。また、テクスチャ模様のような色の変化が激しい画像に対し、CSRBF 法を局所的に繰り返して用いる新たな LCSRBF 法の開発も行う。さらに、考古学的な土器に多く見られるような多角形的な傷の修復問題に対する新たな手法の開発も行う。本報では、各手法で用いるアルゴリズムを開発し、実際に適用した例を示して、これらの方法の有効性を示す。

## 2. 解析手段

### 2.1 高速アルゴリズムに基づく CSRBF 法による表面再構成

CSRBF (compactly supported radial basis functions) 法は、表面を再構成する技術のうちの一つであり、離散点  $P(x, y, z)$  から陰関数  $f(P)$  を生成する手法である。Savchenko らが提案した RBF 法における基底関数に対し、Wendland はコ

コンパクトな台を持つ放射状の基底関数 (compactly supported radial function) ;

$$\phi(P_i, P_j) = \begin{cases} \left(1 - \frac{r(P_i, P_j)}{r_0}\right)^2 & 0 \leq r(P_i, P_j) \leq r_0 \\ 0 & r(P_i, P_j) > r_0, \end{cases} \quad (1)$$

を用いて物体の関数化を行った．ここで  $r(P_i, P_j)$  は任意の 2 点  $P_i, P_j$  間の距離 ( $r(P_i, P_j) = \|P_i - P_j\|$ ) を示し，点  $P_i$  を中心とした半径  $r_0$  の球内全ての点との距離を求めている．実際に数値計算する時には，画像の 1 辺を長さ 1 に規格化して行うため，本報で用いている  $r_0$  は規格化後の値である．

このような基底関数に対し，陰関数  $f(P)$  は

$$f(P) = \sum_{j=1}^N \lambda_j \phi(P, P_j) + p(P), \quad (2)$$

$$p(P) = \nu_0 + \nu_1 x + \nu_2 y + \nu_3 z, \quad (3)$$

で与えられる．ここでの重み  $\{\lambda_j (1 \leq j \leq N), \nu_0, \nu_1, \nu_2, \nu_3\}$  は，式 (2) の右辺に式 (4) のような制限  $h_i$  を与え，これを満たすように求めている．

$$h_i = \sum_{j=1}^N \lambda_j \phi(P_i, P_j) + p(P_i). \quad (4)$$

ここでは，制限  $h_i (1 \leq i \leq N)$  として点  $P_i$  における輝度値  $0 \sim 255$  を  $0 \sim 1$  に正規化した値を与え，重み  $\{\lambda_j (1 \leq j \leq N), \nu_0, \nu_1, \nu_2, \nu_3\}$  を求めるためには，次のような行列表現を用いている．式 (4) を

$$H = A\Lambda, \quad (5)$$

としたとき，行列  $A$  は以下のようにになっている．

$$A = \left( \begin{array}{cccc|cccc} 1 & \phi(P_1, P_2) & \dots & \phi(P_1, P_N) & 1 & P_1^x & P_1^y & P_1^z \\ \phi(P_2, P_1) & 1 & \dots & \phi(P_2, P_N) & 1 & P_2^x & P_2^y & P_2^z \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi(P_N, P_1) & \phi(P_N, P_2) & \dots & 1 & 1 & P_N^x & P_N^y & P_N^z \\ \hline 1 & 1 & \dots & 1 & 0 & 0 & 0 & 0 \\ P_1^x & P_2^x & \dots & P_N^x & 0 & 0 & 0 & 0 \\ P_1^y & P_2^y & \dots & P_N^y & 0 & 0 & 0 & 0 \\ P_1^z & P_2^z & \dots & P_N^z & 0 & 0 & 0 & 0 \end{array} \right), \quad (6)$$

ここで， $H = {}^t(h_1, h_2, \dots, h_N, 0, 0, 0, 0)$ ， $\Lambda = {}^t(\lambda_1, \lambda_2, \dots, \lambda_N, \nu_0, \nu_1, \nu_2, \nu_3)$  である．このとき，行列  $A$  を

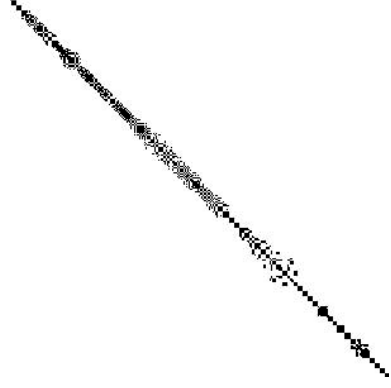


図 1 バンド性を持つ対角行列.

Fig. 1 A band-diagonal matrix.

$$A = \begin{pmatrix} B & C \\ {}^tC & D \end{pmatrix}, \quad (7)$$

と書き換えると,  $B$  は  $N \times N$ ,  $D$  は  $4 \times 4$  の平方行列,  $C$  は  $N \times 4$  の行列になり, 行列  $A$  は 3つのブロックから構成される補間行列 (SLAE 行列) になる. このときの重みの計算に当たっては, 基底関数がコンパクトではない RBF 法では行列  $A$  (SLAE 行列) が密になり  $O(N^3)$  の計算量が必要であるのに対し, CSRBF 法では SLAE 行列が疎行列になり計算量を大幅に削減することができる.

しかし, このような CSRBF 法における陰関数  $f(P)$  の決定において, 重み  $\{\lambda_j (1 \leq j \leq N), \nu_0, \nu_1, \nu_2, \nu_3\}$  の計算は実用化するのに十分な速度ではない. そこで, 著者等は重み計算の中で, 疎行列の SLAE 行列からバンド性を持つ対角行列 (図 1) を作成する手法の開発を行った<sup>12),13)</sup>. これは, 離散点から成る初期データ列の並び換えをして, 効率よく SLAE 行列のバンド化を行うものである. 空のリストを 1つ用意し, 点列の並び換えを行うアルゴリズムを次に示す.

#### 点列の並び換えのアルゴリズム

1. 初期データ列の初めの点をリストに加える.
2. リストに加えた点に対し, 初期データ列の中から近傍点を探してリストに加える.
3. リストに加えた点は初期データ列から除く.
4. リストの最後の点まで近傍点の探索を繰り返す.
5. 初期データ列にある初めの点をリストに加える.

**表 1** 高速のアルゴリズムの使用例.

Table 1 An application of our algorithm in this paper.

| ステップ | リスト         | 初期データ       |
|------|-------------|-------------|
| 1    | -           | 1,2,3,4,5,6 |
| 2    | 1,3,6       | 2,4,5       |
| 3    | 1,3,6,4     | 2,5         |
| 4    | 1,3,6,4     | 2,5         |
| 5    | 1,3,6,4     | 2,5         |
| 6    | 1,3,6,4,2   | 5           |
| 7    | 1,3,6,4,2,5 | -           |

**表 2** バンド性を持つ対角行列の生成例.

Table 2 A generated diagonal matrix with band structure.

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
|   | 1 | 3 | 6 | 4 | 2 | 5 |
| 1 | 1 | × | × | 0 | 0 | 0 |
| 3 | × | 1 | 0 | × | 0 | 0 |
| 6 | × | 0 | 1 | 0 | 0 | 0 |
| 4 | 0 | × | 0 | 1 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 1 | × |
| 5 | 0 | 0 | 0 | 0 | × | 1 |

6. 2 ~ 5 をくり返す.

このようなアルゴリズムを用いた点列の並び換えの簡単な例を示す. 立方体内に  $\{1, 2, 3, 4, 5, 6\}$  の 6 個の離散点があり, この 6 つの点それぞれに対し, 特定の半径  $r_0$  の球内における近傍点が, 次のように与えられているとする. 点 1 の近傍点: 1, 3, 6, 点 2 の近傍点: 2, 5, 点 3 の近傍点: 3, 1, 4, 点 4 の近傍点: 3, 4, 点 5 の近傍点: 2, 5, 点 6 の近傍点: 1, 6. このような 6 つの離散点から成る初期データ列に対し, 著者等が考案した点列並び換えのアルゴリズムを用いると, 与えられた点列は**表 1** のようになり, **表 2** のようなバンド性を持つ対角行列 (SLAE 行列) が生成される. ここで, 表中の  $\times$  は非ゼロであることを示す.

実際に,  $N = 1487$  の離散点列から成る “頭” の絵の再構成表示に著者等が開発した高速アルゴリズムに基づく CSRBF 法を用いたところ, 点列の並び換え

は0.03 秒, SLAE 行列の計算は 0.05 秒と非常に速く処理することができるようになった (図は略). また, このときのデータの格納に必要なメモリは1,675,800 bytes (約 1MiB) であり, 従来の手法の約 1/8 で済むという利点もある. 従って, 著者等が開発した高速アルゴリズムは, 実用化するのに十分な要素を備えているといえる. ここでは, SLAE 行列の解法として, George and Liu<sup>15)</sup> が提案した Block Gauss 解法と Cholesky 分解を併用し, ソフトウェアツールの開発を行った.

## 2.2 画像上の傷の修復手段

本研究では, 著者等が開発した高速アルゴリズムに基づく CSRBF 法を, 画像の修復問題に適用する. 適用例については次章で示す. ここでは, 2次元画像上の傷の修復, テクスチャ模様上の傷の修復, 複雑な多角形的な傷の修復に, 著者等の CSRBF 法を適用する方法について述べる.

### 2.2.1 2次元画像上の傷の修復手法

2次元画像上の傷がついている部分を  $WR$  で表すとする. 傷の部分 ( $WR$ ) を少しだけ広げた領域を  $W+$  とし,  $W+$  から  $WR$  を引いた領域を  $WL$  で表す. このとき, 領域  $W+$  はユーザーが初期値として与える必要が生じるが, これについては Adobe Photoshop GUI interface 等のソフトウェアを用いることで簡単に決めることができる. このように領域  $W+$  を与え, 領域  $WL$  について CSRBF 法を適用して画像の修復を行う. つまり, 領域  $WL$  に CSRBF 法を適用して関数  $f(P)$  を求め, 傷のついた部分を補間して修復を行う. このときの手順をまとめると次のようになる.

#### 修復手順

1. 画像上の傷の部分 ( $WR$ ) を少し広げた領域  $W+$  を決める.
2. 領域  $WL$  に対して著者等の CSRBF 法を適用し, 関数  $f(P)$  を求める.
3. 関数  $f(P)$  を用いて傷の部分 ( $WR$ ) の補間を行い, 領域  $W+$  を修復する.

### 2.2.2 LCSRBF 法

本報で CSRBF 法を多数の画像修復問題に適用する. 結論を先に述べると, この手法は, 自然画像や人物画像のような, 色の変化が比較的滑らかに起こるような画像上の傷の修復に対して有効である. これは, CSRBF 法は画像上の離散点から滑らかな関数  $f(P)$  を求めて, 表面再構成を行うからである. 一方, テクスチャ模様のように色の変化がシャープに起こるような画像に対し, 滑らかな関数表示を行う CSRBF 法をそのまま用いることは, 精度の面で適さない.

そこで、本研究では、CSRBF 法を局所的に繰り返して用いる新たな LC-SRBF (local compactly supported radial basis functions) 法を提案する。これは、CSRBF 法を局所的に用いて関数表現を行うものである。定式化で用いる基底関数と方程式は CSRBF 法の場合と同じであるが、適用方法が異なる。ここでは、例えば、画像サイズが  $128 \times 128$  ピクセルのテクスチャ模様上に、左下から右上への直線的な傷が 1 本ある場合の修復問題を考える（実際の適用例は次章で示す）。直線的な傷が始まる左下のピクセルを“傷の始点”（あるいは、“傷の 1 番目の点”）、始点に対して最も近いピクセルを“傷の 2 番目の点”、始点を除いて 2 番目の点に最も近いピクセルを“3 番目の点”とし、これを繰り返して傷が終わる右上のピクセルを“傷の終点”とする。こうしたとき、テクスチャ模様上の直線的な傷を、始点から終点までの各点毎に、次のようなアルゴリズムで修復する。

#### テクスチャ模様上の傷の修復アルゴリズム

- “傷の始点” の修復を、始点を中心とする半径  $r_t$  (これについては後述する) を調節することにより得られた周囲の点から関数  $f(P)$  を求めて行う。
- 次に、“傷の 2 番目の点” について同じ  $r_t$  を用いて周囲の点を求め、これらの点を用いて新たな関数を作成して修復をする。2 番目の点を修復する際には、1 番目の点は修復後の値を用いる。
- “傷の終点” までこれを繰り返す。

各点を修復する際に用いる周囲の点には、修復する点を中心とする半径  $r_t = M/N_t$  内にあるピクセルで半分以上含まれるものを採用している。ここで  $N_t$  は画像サイズを示し（この例では  $N_t = 128$ ）、 $M$  は 1 から  $N_t$  の値を初期値としてユーザーが与えるものである。 $M$  の値は大き過ぎると、計算に時間がかかり、精度の良い結果は得られない（修復後の画像は滲んでしまう）。最適な  $M$  の求め方は未だ開発されていないが、多数のテクスチャ模様上の傷の修復に対し、様々な  $M$  を用いて数値計算を行った結果、計算コストと得られる画像精度の両方を満たす値として、ここでは  $M = 3$  を用いている。

このように LCSRBF 法は、傷の始点から終点において CSRBF 法を繰り返して用い、傷の 2 番目の点以降は、その前点の修復結果を反映させていることが最大の特徴である。このような LCSRBF 法を用いることで、CSRBF 法では困難であるテクスチャ模様上の傷の修復が精度よく行えるようになる。

### 2.2.3 多角形の傷の修復方法

本研究では、多角形の傷を修復するためのアルゴリズムを新たに開発する。ここで扱う傷は、例えば、考古学的な土器に多く見られるような欠けた部分である。このときの傷は多角形となり、非常に複雑である。このような多角形の傷に対する修復問題は、過去にほとんど研究されていない未解決分野である。本報では、このような多角形の傷の修復を次の三段階に分けて行う。

#### 多角形の傷の修復方法

Step 1. 傷の輪郭抽出。

Step 2. 輪郭の縫い合わせ。

Step 3. 空間スペースマッピング法に基づくスムージング。

各ステップでの手続きは次の通りである。

**Step 1. 傷の輪郭抽出：**ここでは、物体についての多角形の傷の輪郭を全て見つけ出す手続きを次のように行う。

1. 物体上の傷の部分で境界になり得る全ての辺（境界辺）を見つ找出す。
  2. 1. で得られた境界辺から閉じた折線を次のようにして構成する。
    - 2.1 1. で得られた全ての境界辺の中から1つの境界辺  $e_1$  を選ぶ。このとき、境界辺  $e_1$  の始点を  $v_1$ 、終点を  $v_2$  とする。
    - 2.2 次に、 $v_2$  を始点とするような境界辺を見つけて  $e_2$  とし、境界辺  $e_2$  の終点を  $v_3$  とする。
    - 2.3 この手続きを頂点  $v_1$  に戻るまで繰り返し、1つの閉折線を構成する。
- このとき、それぞれの境界辺は1つの閉折線にだけ含まれるとする。従って、同じ境界辺を重複して2つ以上の閉折線に用いることはできない。ここでは、有限個の頂点において得られた閉折線が交わることはあり得るし、閉折線の組み立ては複数考えられる。
3. 2. で得られた閉折線の中で交点を持つものは、交点で結合して1つの閉折線に作りなおす。この手続きは、交点がなくなるまで繰り返して行う。

例えば、上記の輪郭抽出の手続きの1, 2において、2つの閉折線  $L_1 = \langle v_1, v_2, \dots, v_n \rangle$ ,  $L_2 = \langle w_1, w_2, \dots, w_m \rangle$  が構成されたとする（図2の左図）。閉折線  $L_1$  と  $L_2$  は2つの交点を持ち、そのうち1つは、閉折線  $L_1$  で頂点  $v_{p1}$ 、閉折線  $L_2$  で頂点  $w_{p2}$  となっている。このとき、この

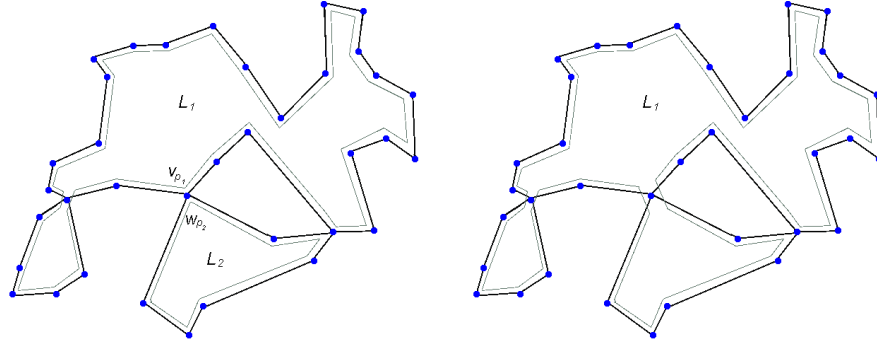


図2 左図は2つの閉折線  $L_1, L_2$  が交点を持つ場合. 右図は結合した結果.

Fig.2 The left figure is closed polygonal lines  $L_1, L_2$ . These have intersection point  $v_{p1}, w_{p2}$ . The right one shows the union of these two closed polygonal lines.

交点において手続きの3より,  $L_1$  と  $L_2$  を結合し, 1つの閉折線  $L = \langle v_1, \dots, v_{p1}, w_{p2+1}, \dots, w_m, w_1, \dots, w_{p2}, w_{p1+1}, \dots, v_n \rangle$  が構成される (図2の右図). このように閉折線を交点において結合することで自己交差を含む1つの閉折線を作成することができ, 交点を1つ減らすことが可能である.

**Step 2. 輪郭の縫い合わせ:** ここでは, 閉折線で表される多角形の傷を三角形分割して縫い合わせて修復を行う. 物体上の傷の輪郭抽出において, 閉折線  $L = \langle v_1, v_2, \dots, v_n \rangle$  が得られたとする. このとき, 閉折線  $L$  上には結び目がないとする. ここでは, 次の条件を満たすように閉折線  $L$  を  $n - 2$  個の三角形に分割する.  $T$  は閉折線  $L$  の三角形分割とする.

1. 三角形分割  $T$  の頂点は 閉折線  $L$  の頂点である.
2.  $T$  の辺は適当な共通部分を持つ. つまり, どの2つの三角形も重なることはないが, ある頂点かある辺で隣接することはあり得る.
3.  $T$  の境界は  $L$  に一致する.

上記のような条件において考えられる三角形分割の数は莫大にある. 例えば  $n$  個の点  $v_1, v_2, \dots, v_n$  があるとき,  $m = n - 2$  本の対角線に対し  $2^{m+1}(2m+1)!!/(m+2)!, m \geq 1$  個の三角形分割の可能性がある. このような膨大な数の中から最適な三角形分割を選ぶ必要が生じるが, そのためのアルゴリズムはアプローチが悪いと計算コストが上り, さほど多くない点データの計算でさえも行うことはできなくなってしまう.

そこで著者等は最適な三角形分割を得るために, 1992年に Cormen, Lieser-

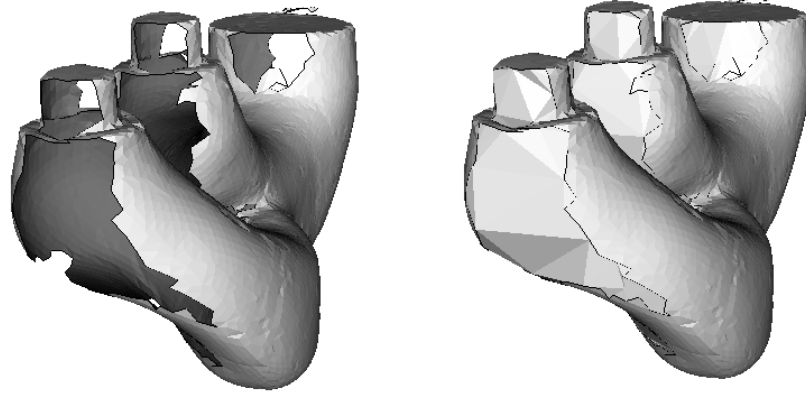


図3 左図は原画像. 右図は, 最適な三角形分割によって再構成した結果.

Fig. 3 The left is original figure. The right one shows the reconstruction of missing data based on optimal triangulation of a bounding polygon.

son and Rivest<sup>16)</sup> が提案した作業全体を部分的に分けて検証を行うためのアルゴリズムを用いた. この手法を用いて, 例えば, ある閉折線に対して得られた1つの三角形分割 ( $T_{1n}$ ) を, 1つの三角形 ( $\triangle_{1kn}$ ) で繋ぐように2つの三角形分割の部分 ( $T_{1k}, T_{kn}$ ) に分け, 分かれたそれぞれの三角形分割  $T_{1k}, T_{kn}$  が最適なものであるかどうか調べる. 実際に, ここでは, 三角形  $\triangle_{1kn}$  の法線ベクトルを  $\vec{n}_{1kn}$ , 辺  $v_1v_k$  を含み三角形  $\triangle_{1kn}$  に隣接する面の法線ベクトルを  $\vec{n}_{1k}$ , 辺  $v_kv_n$  を含み三角形  $\triangle_{1kn}$  に隣接する面の法線ベクトルを  $\vec{n}_{kn}$ , 辺  $v_nv_1$  を含み三角形  $\triangle_{1kn}$  に隣接する面の法線ベクトルを  $\vec{n}_n$  としたとき, ベクトル  $\vec{n}_{1kn}$  とその周囲のベクトル  $\vec{n}_{1k}, \vec{n}_{kn}, \vec{n}_n$  とのスカラ乗法の和を最小にするものを最適な三角形分割としている. つまり,

$$\begin{aligned} \sum(T_{1n}) = \min_{1 \leq k \leq n} & (\sum(T_{1k}) + \sum(T_{kn}) + (\vec{n}_{1k}, \vec{n}_{1kn}) \\ & + (\vec{n}_{kn}, \vec{n}_{1kn}) + (\vec{n}_n, \vec{n}_{1kn})) \end{aligned} \quad (8)$$

となるような  $T_{1n}$  を見つけている.

この手法は, 得られた三角形の集合が2つ以上の三角形分割に含まれるとき, その面集合における検証が反復されるのをできる限り避けるように作られているため, 計算時間を削減することができる. 実際に, 多角形の傷を三角形分割して再構成した結果を図3に示す.

Step 3. スムージング法: 上記のような手続きで行った物体上についた多角形

の傷の修復結果は、断面的であり、原物とは程遠い。実際の物体は滑らかで、修復部分がここで得られたような角ばったものでは不十分である (図3)。そこで本研究では、Step 2. で得られた三角形分割による傷部分に対し、次のようなアルゴリズムで表面の再構成を行い、スムージングをする。ここでは、“2.2.1 2次元画像上の傷の修復手法”と同様に、傷の部分を WR, WR を少し広げた領域を W+, W+ から WR を引いた領域を WL とする。

1. Step 2. で得られた各三角形に対し、次のような手続きによって傷内部の詳細な点データを求める。
  - 1.1 オリジナル物体を三角形分割したときの三角形の平均サイズを求める。
  - 1.2 傷部分における三角形で平均サイズよりも大きい場合は、重心に1点を補い、重心と3頂点を結んで3つの領域に分ける。
  - 1.3 新たに形成された三角形についても手続きの 1.2 を適用する。この手続きは、平均サイズよりも大きい三角形がなくなるまで繰り返す。
2. 領域 WL に対して著者等が新たに開発した CSRBF 法を適用し、関数  $f(P)$  を求める。
3. 関数  $f(P)$  を用いて 1. で得られた傷内部の詳細な点データで補間を行い、領域 WR の修復をする。この際、領域 WR の各点における高さ成分 ( $z_i^s$ ) については、逆写像関数  $z_i^s = f(P) + z_i^d$  で再構成点  $z_i^d$  を求めている。

CSRBF 法は与えられた点データから滑らかな関数化を行い、画像再構成を行う手法であるため、このような画像表面のスムージングに適している。

### 3. 適用例

#### 3.1 2次元画像上の傷の修復例

ここでは 図4 のような古い写真の傷の修復を行う。この写真上には広範囲に渡って大小様々な傷が生じている。従来の手法では、このような広域に傷があるような画像の修復はできない。そこで本研究では、前章の“2.2.1 2次元画像上の傷の修復”で示したアルゴリズムを用いて実際に画像の修復を行う。

図4の原画像上にある白い線や斑点が傷の部分である。本報で述べたアルゴリズムに基づくと、このような傷の部分は WR で表される。この傷の近傍を含



図4 左図は広範囲に傷がある古い写真. 真ん中の図は修復に必要な範囲. 右図は修復した結果.

Fig.4 The left figure is an old photograph which has disconnected damaged areas. The middle one shows the selected retouching area. The right one shows a result of restoration.

んだ部分が  $W+$  である. 図4の真ん中の図の黒い部分がそれに相当する. このような  $WR$  と  $W+$  の差の部分 ( $WL$ ) に含まれる離散点について, 著者等が開発した CSRBF 法を用いて修復した結果は 図4の一番右の図である. この結果から, 精度よく画像再構成が行われ, 傷の修復ができていることがわかる.

図4の画像サイズは  $238 \times 342$  24bpp である. この画像上では, 傷がついている部分のピクセル数は 2076 ピクセル, 画像修復の際に用いたピクセル数 (つまり,  $WL$  に含まれるピクセル数) は 3018 ピクセルである. このような画像に対し基底関数の半径を  $r_0 = 0.03$  として実験を行ったところ, 画像上の傷の修復にかかった時間は 0.32 秒であった. この他にも, 黒いひっかき傷があるトンボの絵上の傷の修復, 古い写真上についている折り傷の修復, 画像上に落書きがある場合の修復等, 2次元平面的な画像上の傷の修復に対して多数の実験を行ったが, いずれの場合も精度よく, かつ, 高速に処理することができた (図は略).

また, 本報で開発した画像上の傷の修復法は, 図5のような多数の多角形の合成からできているような画像上の大きな傷の修復にも有効である. 図5の左図は原画像であり, 10526 個の辺と 20835 個の多角形から構成されている. この図上の “IH” の部分が傷であり, これは 930 個の辺から成っている. このよ

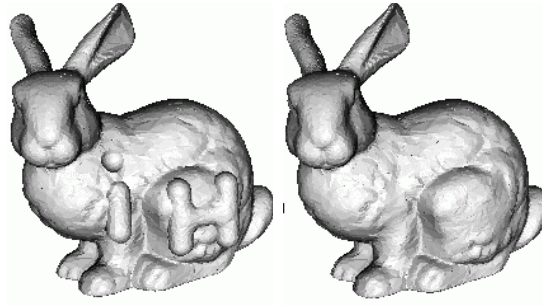


図5 左図は大きな傷があるうさぎの絵. 右図は修復した結果.

Fig. 5 The left figure is a rabbit picture which has large damaged areas. The right one shows a result of restoration.

うな画像に対し,  $r_0 = 0.2$  のサポート半径を用いて修復を行った結果 (図5の右図), 計算時間は1秒であった. また, 著者等の手法は, 既存の装置に簡単に組み込むこともできるため, レーザースキャナでレーザーが当たる部分に生じる窪みを取り除くためのスムージングとしても効果的である (図は略). このように, 本報の画像上の傷の修復方法は, 高速, かつ, 高精度に行うことができ, 適用範囲も広いという利点も備えている.

### 3.2 LCSRBF 法の適用例

テクスチャ模様上の傷の修復に対し, LCSRBF 法と CSRBF 法を適用した結果を示す (図6). 図6の原画像上には, 黒い直線の傷がついている. この傷に対し, 本報で開発した LCSRBF 法を用いたところ, 効率よく修復を行うことができる (図6の二段目の左図). 同じ傷の修復に CSRBF 法を適用すると, 傷の部分が白い筋のように残ってしまい, LCSRBF 法に比べると精度がよくない (図6の二段目右図). これは, LCSRBF 法は, 原物をより忠実に再現するために, 画像の滑らかさを重視する CSRBF 法とは異なり, 1つ前のステップで得られた結果を取り入れて再構成を行う手法であるからである. そのため CSRBF 法と比較すると計算時間はかかるが, CSRBF 法では不十分であった色の変化がシャープな画像上の傷の修復を, LCSRBF 法は精度よく行うことができる. LCSRBF 法と CSRBF 法による修復結果の違いは, ディスプレイ上ではより鮮明である.

### 3.3 多角形の傷の修復例

図7は, 物体上に16個の穴がある場合の傷の修復問題に対し, 本報で開発した手法を適用した結果である. 物体上の傷の部分を自動的に見つけ出し, その

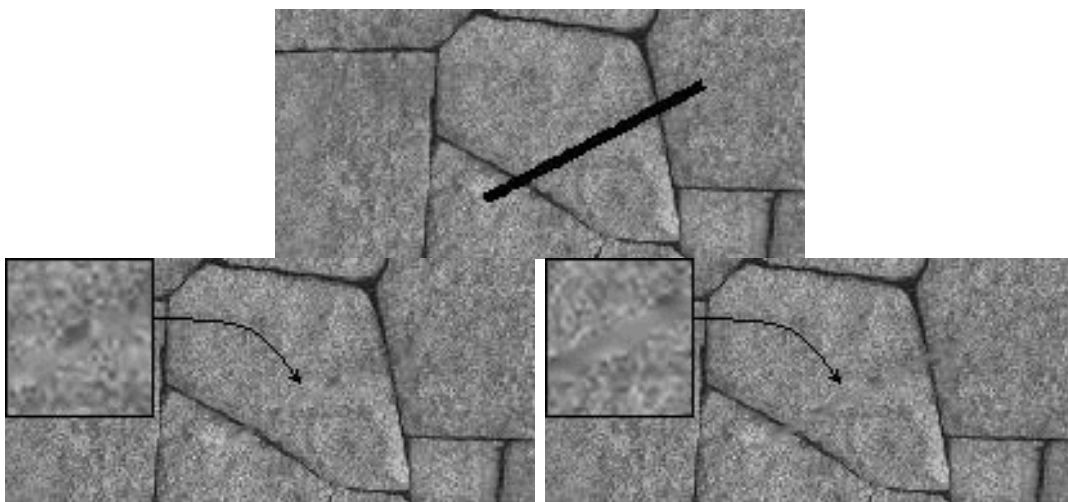


図6 一段目はテクスチャ模様にある傷がある場合の原画像。二段目左図は LCSRBF 法を用いて修復した結果。右図は CSRBF 法を適用した結果。

Fig.6 The first row is a original figure. It is a textured images which have damaged areas. The left figure of second row shows a result of restoration using the method LCSRBF. The right one is a result in the case of the method of CSRBF.

部分の再構成を行っている。原画像は 19467 個の多角形から構成されており、16 個の穴の部分は 372 本の辺から成っている。このような画像上の傷の修復時間は 7.3 秒で、実用化レベルの速さである。また、図 7 の最終的な結果から、精度の面でも十分である。

#### 4. ま と め

本研究では、CSRBF 法を用いて画像を修復するためのアルゴリズムの開発を行い、実際にそれを適用してその有効性を示した。著者等が開発した高速アルゴリズムに基づく CSRBF 法を画像修復問題に用いた結果、従来の手法では不可能であった大きな傷の修復を行うことができるようになった。また、従来の手法は小さな傷でも広範囲に及ぶ場合は修復することができないが、本報で開発したアルゴリズムは、このような散乱する傷も精度よく修復を行うことができる。また、ここで用いているアルゴリズムは高速に計算でき、メモリが少なく済むという好条件も備えているため、実用化するのに十分な性能がある。さらに、本報では、テクスチャ模様のような色の変化が大きい画像上の傷の修

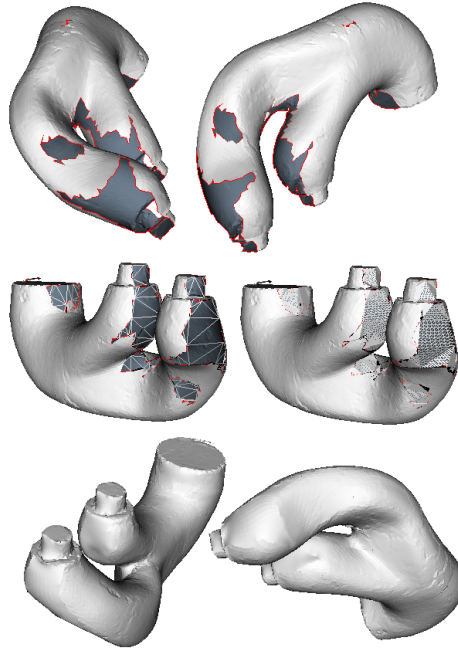


図7 1列目は原画像であり，16個の穴があいている．2列目の左図は多角形の傷を三角形分割し，右図は三角形内部の詳細な点データを求めている．3列目は，CSRBF法を用いてスムージングを行った結果である．

Fig.7 The First row is original model that contains 16 holes. The left figure of second row shows the triangulation of closed polygonal line and the right one shows the data of points inside the hole area. The third row is result of CSRBF extrapolation for missing areas.

復問題に対し，CSRBF法を局所的に繰り返して用いるLCSRBF法を開発しそれを用いることで，従来の手法よりも効率よく修復を行うことができるようになった．また，多角形の傷の修復に対してのアルゴリズムの開発も行い，物体上についた複雑な傷の修復を行うことも可能になった．本報で開発したアルゴリズムは，全て簡単に使用することができ，既存の方法よりも効率的である．今後は，与えられた画像に対してCSRBF法とLCSRBF法をどのように使い分けるべきか，その有効性について検討を行うと共に，多角形の傷の中に傷でない部分を含むような場合（つまり，ドーナツ型の傷）についての手法の開発を行う予定である．

## 参 考 文 献

- 1) Hirani, A.N. and Totsuka, T.: Combining frequency and spatial domain information for fast interactive image noise removal, *SIGGRAPH 96*, pp. 269–276 (1996).
- 2) Bertalmio, M., Sapiro, G., Caselles, V. and Caltagirone, C.: Image Inpainting, *Computer Graphics, SIGGRAPH 2000*, pp. 417–422 (2000).
- 3) Oliveira, M.M., Bowen, B., McKenna, R. and Chang, Y-S.: Fast Digital Image Inpainting, in *Proceedings of the Visualization, Imaging, and Image Processing IASTED Conference*, Marbella, Spain, pp. 261–266 (2001).
- 4) Yamauchi, H., Haber, J. and Seidel, H.-P.: Image Restoration using Multiresolution Texture Synthesis and Image Inpainting, *CGI 2003*, pp.120-125 (2003).
- 5) Drori, I., Cohen-Or, D. and Yeshurun, H.: Fragment-Based Image Completion, *ACM SIGGRAPH 2003*.
- 6) Savchenko, V., Pasko, A., Okunev, O. and Kunii, T.: Function Representation of Solids Reconstructed from Scattered Surface Points and Contours, *Computer Graphics Forum*, Vol. 14, No. 4, pp. 181–188 (1995).
- 7) Morse, B., Yoo, T. S., Rheingans, P., Chen, D. T. and Subramanian, K. R., Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions, in *Proceedings of the Shape Modeling conference*, Genova, Italy, pp. 89–98 (2001).
- 8) Carr, J.C., Mitchell, T.J., Beatson, R.K., Cherrie, J.B., Fright, W.R., McCallum, B.C. and Evans, T.R.: Reconstruction and representation of 3D objects with Radial Basis Functions, *Computer Graphics, SIGGRAPH 2001*, pp. 67–76 (2001).
- 9) Litwinowicz, P. and Williams, L.: Animating Images with Drawing, *Computer Graphics, SIGGRAPH 1994*, pp. 409–412 (1994).
- 10) Turk, G. and O’Brien, J. F.: Shape Transformation Using Variational Implicit Functions, *Computer Graphics, SIGGRAPH 1999*, pp. 335–342 (1999).
- 11) Wendland, H.: Piecewise polynomial, positive defined and compactly sup-

- ported radial functions of minimal degree, *AICM*, Vol.4, pp.389–396 (1995).
- 12) Kojekine, N., Hagiwara, I. and Savchenko, V.: Software Tools Using CSRBFs for Processing Scattered Data, *Computers & Graphics*, pp. 309–317 (2002).
  - 13) Kojekine, N., Savchenko, V. and Hagiwara, I.: Surface Reconstruction Based on Compactly Supported Radial Basis Functions in *Geometric Modeling: Techniques, Applications, Systems and Tools*, M. Sarfraz (editor), Kluwer Science (An extended version of our Hawaii’2002 conference paper, accepted for this book as one of best papers selected from three events: CAGD Symposium in the International Conference on “Information Visualization” (IV’2001), London, UK, Special Session on Geometric Design and Modeling (GDAM) - CGIM’2001, Hawaii. USA, and Graphical Models And Imaging (GMAI) - (CISST’2001), Las Vegas, USA).
  - 14) Ohtake, Y., Belyaev, A. and Seidel, H.-P.: A Multi-scale Approach to 3D Scattered Data Interpolation with Compactly Supported Basis Functions, *Shape Modeling International 2003*, (2003).
  - 15) George, A. and Liu, J.W.H.: Computer Solution of Large Sparse Positive Definite Systems, *Prentice-Hall: Englewood Cliffs, NJ*, (1981).
  - 16) Cormen, T.H., Leiserson, C.E. and Rivest, R.L.: Introduction to Algorithms, *The MIT Press* (1992).