



**HAL**  
open science

# Complicial thinning for three-dimensional digital images: recursive definition of reductive points and their topological characteristics

Yukiko Kenmochi, Kenji Ushigusa

## ► To cite this version:

Yukiko Kenmochi, Kenji Ushigusa. Complicial thinning for three-dimensional digital images: recursive definition of reductive points and their topological characteristics. Meeting on Image Recognition and Understanding (MIRU'05), Jul 2005, Awaji, Hyogo, Japan. pp.120-127. hal-00622237

**HAL Id: hal-00622237**

**<https://hal.science/hal-00622237>**

Submitted on 11 Jan 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# 複体を用いた 3次元デジタル画像の細線化

## 削除可能点の再帰定義とその位相的性質

剣持雪子<sup>†</sup> 牛草研二<sup>††</sup>

<sup>†</sup> 仏国立科学研究センターガスパーモンジュ研究所

<sup>††</sup> 岡山大学工学部情報工学科

あらまし 細線化とは、3次元2値画像中の対象領域の位相を保存しつつ、対象領域の点を順に削っていく処理である。その際、点の削除可能性の定義は細線化処理の要となる。これまでに提案された定義のほとんどは点の連結性に基づいており、空間に位相や次元が定義されていないため、細線化の過程・結果において次元などの位相的性質に関する議論が曖昧であった。本稿では、複体の概念を用いて離散空間に位相を導入することにより、それらの議論が可能な細線化手法を提案する。具体的には、点の削除可能性を再帰的に定義し、削除可能点の位相的性質を明らかにすることで、再帰呼出しのない点の削除可能性の判定方法を提案する。

キーワード 細線化, 削除可能点, 多面体的複体, 位相

## Complicial thinning for three-dimensional digital images

### Recursive definition of reductive points and their topological characteristics

Yukiko KENMOCHI<sup>†</sup> and Kenji USHIGUSA<sup>††</sup>

<sup>†</sup> Gaspard-Monge Institute, CNRS/University of Marne-la-Vallée/ESIEE, France

<sup>††</sup> Department of Information Technology, Okayama University, Japan

**Abstract** Thinning is an image operation whose goal is to reduce object points in a "topology-preserving" way. In the process, therefore, the definition of reductive points plays an important role. Most of conventional definitions are based on the concept of point connectivities and they lack topological tools, especially for three-dimensional digital images. In this paper, we introduce a new notion of thinning based on polyhedral complexes which give a topology in the discrete space and the recursive definition of reductive points. We also clarify their topological characteristics which provide an effective thinning algorithm.

**Key words** thinning, reductive points, polyhedral complex, topology

#### 1. ま え が き

2次元もしくは3次元物体の離散形状データの次元の変換については、これまでに多くの研究が重ねられてきたが、3次元物体の次元変換に関しては、とり得る形状パターンが膨大な数になることもあり、2次元ほどの研究成果は挙がっていない。しかし、例えば医療画像において、血管や気管支などの木構造を解析するために線構造を取り出すという前処理として1次元への次元変換処理、すなわち細線化が必要となる。また、血管の壁や脳膜などの面構造を解析するための前処理として2次元への次元変換処理、すなわち薄面化が必要となる [3], [9]。

本研究の目的は、3次元2値画像に対して、その位相的性質を保存しつつ次元変換を行ない、物体固有の形状特徴を抽出す

ることである。本稿では今後このような処理を総称として（結果として面構造が得られた場合にも）細線化と呼ぶ。具体的には3次元2値画像中の対象領域の位相を保存しつつ、対象領域の点を順に削っていく。したがって、細線化においては、点の削除可能性の定義、および判定法が処理の要となる。これらの削除可能点は単純点、あるいは消去可能点と呼ばれ [1], [3], [9]、それらの定義は主に点の連結性に基づいており公理的に与えられている。

ところで、3次元デジタル画像の細線化では、その結果として1点、線構造、面構造の3種類の次元が得られる。これらの次元の混在に対応するため、得られた結果を次元ごとに分類する方法 [6]、あるいは得られる次元をあらかじめ限定して処理する方法 [8], [9] などの工夫が考えられてきた。それらの多くの

方法は、多様体や単純な曲線や曲面の交差には対応出来るものの、複数の曲線や曲面が複雑に交わった場合には対応が難しいという共通の問題を抱えている [8]. また、公理的方法がとられていることが多く、理論的にすべての場合にうまくいくことが証明されているわけではない。その原因として、これまでに提案された細線化手法の多くが点の連結性に基づいた方法であるため、空間に位相、次元が定義されておらず、上記の次元の混在に関する問題を解決するには道具不足であったことがあげられる。この問題を解決するには、離散空間に位相を導入し、さらに 1 次元曲線や 2 次元曲面を離散空間でどのように表現するのか、という基本的な問題をまず解決しなければならない<sup>(注1)</sup>。

そこで本稿では、複体を用いて離散空間に位相を導入することによって、純粋に元の物体の位相のみを考慮した細線化を考える。そして、そのような細線化では、削除可能点の定義は形状の次元には依存しないこと、細線化の結果は形状固有のものが得られること、すなわち、ある次元の（あるいは部分的に異なる次元を持つ）複体<sup>(注2)</sup>が得られることを確認する。具体的には、まず文献 [5] によって提案された離散多面体的複体を用いて離散空間に位相を導入し、それを用いることによって、点の削除可能性を再帰的に定義できることを示す。そして、文献 [7] によって提案された点の位相的性質による分類方法を利用して、削除可能点の位相的性質を明らかにすることで、再帰呼出しのない、効率的な点の削除可能性の判定方法を提案する。

## 2. 諸定義

### 2.1 離散多面体的複体の定義

本研究では、離散多面体的複体の概念を用いて 3 次元離散形状データ (3 次元 2 値画像) の細線化を行なう。3 次元ユークリッド空間において整数値のみの座標を持つ格子点の集合を離散空間とよび、 $\mathcal{Z}^3$  で表す。離散多面体的複体は、以下の手順で  $\mathcal{Z}^3$  において定義される [5], [7].

[定義 1] 頂点が  $\mathcal{Z}^3$  の点である単位立方体を考える。単位立方体の各頂点が 0 か 1 のいずれかの値を持つとき、1 の値を持つ点の集合から作られる凸包を離散凸多面体と定義する。また、凸包が  $n$  次元凸多面体 ( $n = 0, 1, 2, 3$ ) のとき、 $n$  次元離散凸多面体とよぶ。

上記の定義から、離散凸多面体の各頂点は格子点であり、かつ隣接する頂点は 26 近傍の位置関係にあることが分かる。ここで、 $\mathbf{x} \in \mathcal{Z}^3$  の 26 近傍は以下のように定義できる。

$$\mathbf{N}_{26}(\mathbf{x}) = \{\mathbf{y} \in \mathcal{Z}^3 : \|\mathbf{x} - \mathbf{y}\|^2 \leq 3\}.$$

そのため、上記定義の離散凸多面体を 26 近傍離散凸多面体とよぶこともある。本稿では、26 近傍のみを扱うため、離散凸多面体と 26 近傍離散凸多面体を同一視してよい。

[定義 2]  $r$  次元離散凸多面体  $\sigma$  の面とは、 $\sigma$  の境界部分に含まれる  $s$  次元離散凸多面体 ( $s < r$ ) をさす。また、 $\sigma$  のすべての面の集合を  $face(\sigma)$  と書く。

[定義 3] 離散多面体的複体  $\mathbf{K}$  とは、離散凸多面体の有限集

(注1): 2 次元画像の細線化では、結果が 1 点、あるいは線構造になり、それらは点の連結性があれば表現することができた。



図 1 3 次元離散凸多面体の例

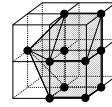


図 2 3 次元離散複体の例

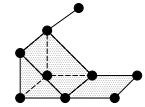


図 3 齊次元でない離散複体の例

合で以下の条件を満たすものをいう。

- (1)  $\emptyset \in \mathbf{K}$ ;
- (2) もし  $\sigma \in \mathbf{K}$  ならば、 $face(\sigma) \subset \mathbf{K}$ ;
- (3) もし  $\sigma, \tau \in \mathbf{K}$  ならば、 $\sigma \cap \tau = \bigcup_{a \in face(\sigma) \cap face(\tau)} a$ .

[定義 4]  $\mathbf{K}$  を離散多面体的複体とする。このとき、 $\mathbf{K}$  の次元は  $\mathbf{K}$  に属する離散凸多面体の中で最も次元の高いものと同じとし、 $dim(\mathbf{K})$  で表す。

直観的には、 $\mathcal{Z}^3$  上の点を結ぶことで作成される  $n$  次元物体が  $n$  次元離散多面体的複体である。離散凸多面体はその構成要素に該当する。そして、面は、その要素を形作るさらに小さな構成要素となる。例えば、図 1 の 3 次元離散凸多面体に含まれる面は、0 次元の面が 6 個、1 次元の面が 9 個、2 次元の面が 5 個である。以降は離散多面体的複体を離散複体と略記する。

### 2.2 離散多面体的複体における数学的道具

ここでは離散多面体的複体におけるいくつかの数学的道具を与える。

[定義 5]  $n$  次元離散複体  $\mathbf{K}$  において、任意の  $s$  次元離散凸多面体  $\sigma \in \mathbf{K} (s < n)$  に対して、

$$\sigma \in face(\tau)$$

となる  $n$  次元離散凸多面体  $\tau \in \mathbf{K}$  が存在するならば、 $\mathbf{K}$  は齊次元であるという。

図 2 で例に挙げた離散複体の次元は 3 次元であり、かつ齊次元である。齊次元でない離散複体とは、例えば図 3 のような複体のことである。以下の組み合わせの閉包、および連結の定義は、離散複体の位相的な情報を取り扱うために必要である。

[定義 6]  $\mathbf{K}$  を離散複体とする。このとき、 $\mathbf{K}$  の部分集合  $\mathbf{K}_0 \subset \mathbf{K}$  の組合せの閉包を次のように定義する。

$$Cl(\mathbf{K}_0) = \mathbf{K}_0 \cup \left( \bigcup_{a \in \mathbf{K}_0} face(a) \right).$$

[定義 7]  $\mathbf{K}$  を離散複体、 $\mathbf{K}$  中の任意の 2 つの離散凸多面体を  $\sigma, \tau$  とする。このとき、以下の条件を満たす要素列  $\sigma = a_1, a_2, \dots, a_n = \tau$  が定義できれば、 $\mathbf{K}$  は連結であるという。

- (1)  $a_i \in \mathbf{K} (i = 1, 2, \dots, n)$ ;
- (2)  $Cl(\{a_i\}) \cap Cl(\{a_{i+1}\}) \neq \emptyset (i = 1, 2, \dots, n-1)$ .

[定義 8] 離散複体  $\mathbf{K}$  に含まれる離散凸多面体の頂点の和集合を  $\mathbf{K}$  の離散点集合と呼び、 $dps(\mathbf{K})$  と表す。

離散複体  $\mathbf{K}$  の離散点集合  $dps(\mathbf{K})$  の各点において、その位相的な性質を調べるために、さらに次のような定義が必要となる。

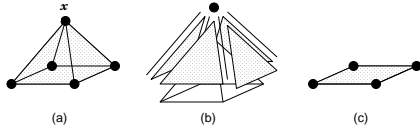


図4 (a) 離散複体  $\mathbf{K}$ , (b)  $x \in dps(\mathbf{K})$  のスター, (c)  $x$  のリンク

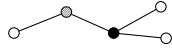


図5 1次元離散複体の点の分類: スターが線状な点 (グレー), 半線状な点 (白), そのいずれでもない点 (黒)

[定義9] 離散複体  $\mathbf{K}$  における, ある点  $x \in dps(\mathbf{K})$  のスターを次のように定義する.

$$star(x) = \{\sigma \in \mathbf{K} : x \in \sigma\}.$$

[定義10] ある点  $x \in dps(\mathbf{K})$  のリンクを次のように定義する.

$$link(x) = \bigcup_{\sigma \in star(x)} face(\sigma) \setminus star(x).$$

ここで, リンクは離散複体だが, スターは必ずしも離散複体とはならないことに注意する. 特に, スターとリンクを考える離散複体  $\mathbf{K}$  を明記したいときには, それぞれ  $star(x : \mathbf{K})$ ,  $link(x : \mathbf{K})$  と書く. 図4に, スターとリンクの例を示す.

[定義11]  $\mathbf{K}$  を離散複体,  $x$  を  $dps(\mathbf{K})$  に含まれる点とする. このとき,  $star(x : \mathbf{K})$  の次元は,  $star(x : \mathbf{K})$  に属する離散凸多面体の次元の中で最も高いものと等しく,  $dim(star(x : \mathbf{K}))$  で表す.

### 2.3 離散多面体的複体の各点の性質

前節で定義した数学的道具を用いて, 離散複体  $\mathbf{K}$  の離散点集合  $dps(\mathbf{K})$  の各点における位相的な性質を調べるため, 以下のような定義を与える.

[定義12] 離散複体  $\mathbf{K}$  において点  $x \in dps(\mathbf{K})$  の  $link(x)$  がふたつの0次元離散凸多面体からなるとき,  $star(x)$  は線状であるという.

[定義13] 離散複体  $\mathbf{K}$  において点  $x \in dps(\mathbf{K})$  の  $link(x)$  がひとつの0次元離散凸多面体からなるとき,  $star(x)$  は半線状であるという.

線状, 半線状の例を図5に示す.

次に線状, 半線状の定義を用いて,  $\mathcal{Z}^3$  における曲線を定義する.

[定義14]  $\mathbf{K}$  を連結な1次元斉次元離散複体とする. このとき,  $dps(\mathbf{K})$  の各点のスターが線状か半線状であり, かつ, 少なくとも1つ以上の半線状の点が存在するならば,  $\mathbf{K}$  を端点をもつ離散曲線とよぶ.

[定義15]  $\mathbf{K}$  を連結な1次元斉次元離散複体とする. このとき,  $dps(\mathbf{K})$  のすべての点のスターが線状なら,  $\mathbf{K}$  を離散閉曲線とよぶ.

離散曲線の定義を用いて, スターが2次元のときの性質を定義する.

[定義16] 離散複体  $\mathbf{K}$  のある点  $x \in dps(\mathbf{K})$  において  $link(x)$  がひとつの離散閉曲線ならば,  $star(x)$  は輪状であるという.

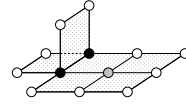


図6 2次元離散複体の点の分類: スターが輪状な点 (グレー), 半輪状な点 (白), そのいずれでもない点 (黒)

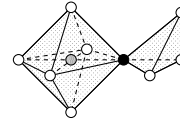


図7 3次元離散複体の点の分類: スターが球状な点 (グレー), 半球状な点 (白), そのいずれでもない点 (黒)

[定義17] 離散複体  $\mathbf{K}$  のある点  $x \in dps(\mathbf{K})$  において  $link(x)$  がひとつの端点をもつ離散曲線なら,  $star(x)$  は半輪状であるという.

輪状, 半輪状の例を図6に示す.

輪状, 半輪状を用いて, 離散空間  $\mathcal{Z}^3$  における曲面を次のように定義する.

[定義18]  $\mathbf{K}$  を連結な2次元の斉次元離散複体とする. このとき,  $dps(\mathbf{K})$  の各点のスターが輪状か半輪状になり, かつ, スターが半輪状である点が少なくともひとつ以上存在するならば,  $\mathbf{K}$  を縁をもつ離散曲面とよぶ.

[定義19]  $\mathbf{K}$  を連結な2次元斉次元離散複体とする. このとき,  $dps(\mathbf{K})$  に含まれるすべての点のスターが輪状ならば,  $\mathbf{K}$  を離散閉曲面とよぶ.

離散曲線と以下の組合せ境界の定義を用いて, スターが3次元のときの性質を定義する.

[定義20]  $\mathbf{K}$  を  $n$ 次元斉次元離散複体とする.  $\mathbf{K}$  に含まれる  $n-1$ 次元の離散凸多面体うち, たゞひとつの  $n$ 次元離散凸多面体の面になっているものの集合を  $\mathbf{H}$  とする. このとき,  $\mathbf{K}$  の組合せ境界  $\partial\mathbf{K}$  を次のように定義する.

$$\partial\mathbf{K} = Cl(\mathbf{H}).$$

[定義21] 離散複体  $\mathbf{K}$  のある点  $x \in dps(\mathbf{K})$  において  $link(x)$  が離散閉曲面ならば,  $star(x)$  は球状であるという.

[定義22] 離散複体  $\mathbf{K}$  のある点  $x \in dps(\mathbf{K})$  において  $link(x)$  が縁をもつ離散曲面で, かつその組合せ境界  $\partial link(x)$  が連結なら,  $star(x)$  は半球状であるという.

球状, 半球状の例を図7に示す.

### 2.4 離散複体の生成

本稿では, 以下のように定義される3次元2値画像を細線化処理の入力とする.

[定義23] 離散空間  $\mathcal{Z}^3$  の各格子点に0または1の値を割り当てた空間を3次元2値画像と定義する.

入力の3次元2値画像のうち1の値を持つ点を1点とよぶことにすると, この1点の集合が厳密には細線化処理を施す点集合となる. しかし, 点集合そのものには細線化に必要な位相の概念がないので, この点集合から離散複体を生成することにより, 点集合に位相を導入することにする. 導入方法を簡単に

説明すると、 $\mathcal{Z}^3$  の各単位立方格子（8 個の格子点を頂点とする単位立方体）ごとに、定義 1 により離散凸多面体をまず生成し、 $\mathcal{Z}^3$  におけるすべての離散凸多面体の和集合を離散多面体的複体とする。方法の詳細と、離散凸多面体の和集合が離散多面体的複体の定義 3 の 3 つの条件を満たしていることの証明に関しては、文献 [5] を参照されたい。ここでは、同文献より、さらに以下の重要な定理を引用する。

[定理 1] 上記のような生成方法では、1-点集合  $\mathbf{V} \subset \mathcal{Z}^3$  から必ず一意に離散複体  $\mathbf{K}(\mathbf{V})$  が得られる。

## 2.5 離散複体を用いた点の分類

2.3 節の離散複体の位相的な性質に基づいて、3 次元 2 値画像中の各 1-点を、以下のそれぞれの条件を満たす 12 タイプに分類することができる [7]。

- (1)  $\dim(\text{star}(\mathbf{x})) = 0$ ;
- (2)  $\text{star}(\mathbf{x})$  が線状;
- (3)  $\text{star}(\mathbf{x})$  が半線状;
- (4)  $\dim(\text{star}(\mathbf{x})) = 1$ , かつ  $\text{star}(\mathbf{x})$  が線状でも半線状でもない;
- (5)  $\text{star}(\mathbf{x})$  が輪状;
- (6)  $\text{star}(\mathbf{x})$  が半輪状;
- (7)  $\dim(\text{star}(\mathbf{x})) = 2$ , かつ  $Cl(\text{star}(\mathbf{x}))$  が斉次元, かつ  $\text{star}(\mathbf{x})$  が輪状でも半輪状でもない;
- (8)  $\dim(\text{star}(\mathbf{x})) = 2$ , かつ  $Cl(\text{star}(\mathbf{x}))$  が非斉次元;
- (9)  $\text{star}(\mathbf{x})$  が球状;
- (10)  $\text{star}(\mathbf{x})$  が半球状;
- (11)  $\dim(\text{star}(\mathbf{x})) = 3$ , かつ  $Cl(\text{star}(\mathbf{x}))$  が斉次元, かつ  $\text{star}(\mathbf{x})$  が球状でも半球状でもない;
- (12)  $\dim(\text{star}(\mathbf{x})) = 3$ , かつ  $Cl(\text{star}(\mathbf{x}))$  が非斉次元;

## 3. 細線化

### 3.1 細線化アルゴリズム

細線化を実現するアルゴリズムは非常に単純で、以下のようになる。ただし、削除可能点の定義については次節で定義する。[アルゴリズム 1] (細線化)

入力: 3 次元 2 値画像の 1-点集合  $\mathbf{V}$

出力: 3 次元 2 値画像の 1-点集合  $\mathbf{V}$

1.  $\mathbf{V}$  から削除可能点集合  $\mathbf{P}$  を得る;

2. **while** ( $\mathbf{P} \neq \emptyset$ ) **do**

2.1. 点  $\mathbf{x} \in \mathbf{P}$  を選択;

2.2.  $\mathbf{x}$  の値を 1 から 0 に書き換え ( $\mathbf{V}$  の書き換え);

2.3.  $\mathbf{x}$  の近傍の 1-点の削除可能性をチェックし、削除可能でなくなった点集合を  $\mathbf{I}(\mathbf{x})$ , あらたに削除可能になった点集合を  $\mathbf{D}(\mathbf{x})$  とし,  $\mathbf{P} \leftarrow \mathbf{P} \setminus \mathbf{I}(\mathbf{x}) \cup \mathbf{D}(\mathbf{x})$ ;

3. **return**  $\mathbf{V}$ .

ここで、入力 2 値画像が 3 次元ならば、細線化の最終結果  $\mathbf{V}$  に対応する離散複体  $\mathbf{K}$  の次元は 2 次元以下となる。特に、 $\mathbf{K}$  が 2 次元斉次元離散複体になるとき、その処理を細線化の代わりに薄面化とよぶこともある。しかし、本稿では最終結果の離散複体  $\mathbf{K}$  の次元や斉次元性によらず、上記アルゴリズムによる処理を細線化とよぶ。なぜなら、削除可能点の定義は  $\mathbf{K}$  の

次元や斉次元性によらず、また細線化後の  $\mathbf{K}$  の次元や斉次元性を決めるのは入力  $\mathbf{V}$  の位相であってユーザではないからである。言い替えると、細線化結果は  $\mathbf{V}$  の位相、すなわち  $\mathbf{K}$  によって決まる。

以上のことから、アルゴリズム 1 の細線化結果は、位相的な見方をすると ( $\mathbf{K}$  の次元や斉次元性などの位相的性質)、ステップ 1 の点  $\mathbf{x} \in \mathbf{P}$  の選択方法によらないことが分かる。そこで、本稿ではステップ 1 の詳細については議論せず、後述の実験では  $\mathcal{Z}^3$  の 3 つの座標軸を単純に順次探索していくことにより  $\mathbf{x}$  の選択を行うこととする。

ところで、上述のアルゴリズムは、細線化のみならず骨格化でも同様に適用される。骨格化の場合、背景からの距離などの幾何学計量が 1-点の削除に考慮されるため、ステップ 1 の 1-点  $\mathbf{x}$  の選択方法にも幾何学計量を考慮する必要がある。もちろん、削除可能点の定義にも幾何学計量が考慮されることになる。

### 3.2 削除可能点の再帰定義

形状の位相を考慮した細線化を実現するためには、1-点集合  $\mathbf{V}$  の位相的性質を変化させないように 1-点を削除していかなければならない。そのような 1-点を削除可能点とよび、離散複体の収縮可能性とともに、以下のように再帰的に定義する。

[定義 24] 1-点集合  $\mathbf{V}$  に対応する離散複体を  $\mathbf{K}(\mathbf{V})$  としたとき、削除可能な点と収縮可能な離散複体を以下のように再帰的に定義する。

- (1)  $link(\mathbf{x} : \mathbf{K}(\mathbf{V}))$  が収縮可能ならば点  $\mathbf{x} \in \mathbf{V}$  は削除可能;
- (2)  $\mathbf{V}$  がひとつの点からなるならば、 $\mathbf{K}(\mathbf{V})$  は収縮可能;
- (3) 削除可能な点  $\mathbf{y} \in \mathbf{V}$  が存在し、かつ  $\mathbf{K}(\mathbf{V} \setminus \{\mathbf{y}\})$  が収縮可能ならば、 $\mathbf{K}(\mathbf{V})$  は収縮可能。

上記のような削除可能点の再帰定義の似たような試みは文献 [2], [4] にも見られ、収縮可能の概念も扱われている [2]。しかし、上記の定義のように明確に再帰的に表現されているわけではなく、また本稿の離散複体とは異なる位相空間が扱われている [2]。ここで、削除可能な点の定義には収縮可能な離散複体が必要であり、逆に収縮可能な離散複体の定義には削除可能な点が必要であることから、2 つの定義を分離することは難しい。

### 3.3 削除可能点のタイプ導出

前節において、細線化における削除可能点が再帰的に定義された。また、前章において、細線化の対象である 1-点集合の各 1-点は、離散複体を用いることによって 12 タイプのいずれかに分類できることを示した。以上の流れから、削除可能点か、果してどのタイプの点であるか調べることは自然であろう。この調査は、処理の高速化の観点からも重要である。なぜなら、削除可能点の定義が再帰的であるため、そのままプログラムに実装した場合、処理速度に期待が持てないからだ。削除可能点がどのタイプに属するのかあらかじめ分かっていたら、処理速度の向上が見込める。

以下では、削除可能点のタイプを導出していく。最終的に定理 2 を得るために、まず補題を準備する。

[補題 1] タイプ 1 の点は削除可能ではない。

(証明) タイプ 1 の点  $\mathbf{x}$  は、その周囲  $\mathbf{N}_{26}(\mathbf{x})$  に 1-点が存在し

ない点である。したがって、 $link(x)$  は空集合となり、定義 24 より収縮可能ではないので、 $x$  は削除可能ではない。

[補題 2] タイプ 2 の点は削除可能ではない。

(証明) タイプ 2 の点  $x$  は、その定義から  $link(x)$  が 2 つの 0 次元離散凸多面体からなり、 $dps(link(x))$  の 2 つの点はそれぞれ削除可能点にはなり得ないので、 $link(x)$  は収縮可能にならず、したがって  $x$  は削除可能ではない。

[補題 3] タイプ 3 の点は削除可能である。

(証明) タイプ 3 の点  $x$  は、その定義から  $link(x)$  がひとつの点からなるので、 $link(x)$  が収縮可能となり、 $x$  は削除可能となる。

[補題 4] タイプ 4 の点は削除可能でない。

(証明) タイプ 4 の点  $x$  は、その定義から  $link(x)$  に 3 つ以上の連結成分が含まれる。したがって、 $link(x)$  が収縮可能とならず、 $x$  は削除可能でない。

[補題 5] タイプ 5 の点は削除可能でない。

(証明) タイプ 5 の点  $x$  は、 $link(x)$  にタイプ 2 の点しか含まない。そして、それらの点はどれも削除可能点にはなりえないので、 $link(x)$  が収縮可能とならず、したがって、 $x$  は削除可能でない。

[補題 6] タイプ 6 の点は削除可能である。

(証明) タイプ 6 の点  $x$  は、その定義から  $link(x)$  は端点を持つ離散曲線となる。すなわち、 $dps(link(x))$  の各点のタイプは 2 あるいは 3 となり、さらにタイプ 3 の点は少なくとも 1 つ以上存在する。タイプ 3 の点は削除可能であり、また離散曲線の端点に相当する。つまり、 $link(x)$  の細線化は離散曲線の端点を順に削除していく操作となり（細線化の過程で、タイプ 2 の点が順にタイプ 3 と変化していく）、最終的には  $dps(link(x))$  はひとつの点となる。したがって、 $link(x)$  は収縮可能であり、 $x$  は削除可能である。

[補題 7] タイプ 7 の点は削除可能でない。

(証明) タイプ 7 の点  $x$  は、その定義から、 $link(x)$  が互いに連結しない 2 つ以上の離散曲線からなる。各離散曲線が、削除可能点を持ち、収縮可能である可能性はあるが、それぞれの離散曲線の細線化結果が再び連結することはあり得ない。したがって、 $link(x)$  は収縮可能ではないので、 $x$  は削除可能でない。

[補題 8] タイプ 8 の点は削除可能でない。

(証明) タイプ 8 の点  $x$  は、その定義から、 $link(x)$  が互いに連結しない 2 つ以上の 1 次元以下の離散複体からなる。それぞれの離散複体が削除可能点を持ち、収縮可能である可能性はあるが、タイプ 7 同様、それぞれの離散複体の細線化結果が再び連結することはあり得ない。したがって、 $link(x)$  は収縮可能とならず、 $x$  は削除可能でない。

[補題 9] タイプ 9 の点は削除可能でない。

(証明) タイプ 9 の点  $x$  は、その定義から、 $link(x)$  にタイプ 5 の点しか含まず、それらは削除可能ではない。したがって、 $link(x)$  は収縮可能ではなく、 $x$  は削除可能でない。

[補題 10] タイプ 10 の点は削除可能である。

(証明) タイプ 10 の点  $x$  は、その定義から  $link(x)$  は端を持つ離散曲面となる。すなわち、 $dps(link(x))$  の各点のタイプは

5 あるいは 6 となり、さらにタイプ 6 の点は少なくとも 1 つ以上存在する。タイプ 6 の点は削除可能であり、また離散曲面の端の点に相当する。つまり、 $link(x)$  の細線化は離散曲面の端を順に削除していく操作となり（細線化の過程で、タイプ 5 の点が順にタイプ 6 と変化していく）、最終的には  $dps(link(x))$  はひとつの点となる。したがって、 $link(x)$  は収縮可能であり、 $x$  は削除可能である。

タイプ 11 と 12 に対しては、以下の追加の分類を必要とする。

[定義 25] タイプ 11 の点をまず以下のように 2 つに分類し、

(11a)  $link(x)$  が連結；

(11b)  $link(x)$  が非連結、

そのうちタイプ 11a の点をさらに以下の 2 つに詳しく分類する。

(11a-a)  $link(x)$  が収縮可能；

(11a-b)  $link(x)$  が非収縮可能。

[定義 26] タイプ 12 の点  $x$  をまず以下のように 2 つに分類し、

(12a)  $link(x)$  が連結；

(12b)  $link(x)$  が非連結、

そのうちタイプ 12a の点をさらに以下の 2 つに詳しく分類する。

(12a-a)  $link(x)$  が収縮可能；

(12a-b)  $link(x)$  が非収縮可能。

タイプ 11 と 12 の追加分類は、 $link(x)$  の収縮可能性、すなわち  $link(x)$  に位相的な穴が存在するか否かによっている。具体的な例を図 8 に示す。図 8 では、削除可能であるタイプ 3 と 6 の点（補題 3, 6 より）を逐次的に削除することで、リンクの細線化を実現している。

上記の定義より、以下の補題が成り立つことは明らかである。

[補題 11] タイプ 11a-a, 12a-a の点は削除可能であり、タイプ 11a-b, 12a-b の点は削除可能ではない。

[補題 12] タイプ 11b, 12b の点は削除可能でない。

(証明) タイプ 11b, 12b の点  $x$  は、その定義から、 $link(x)$  が連結していない。したがってタイプ 7, 8 と同様、それぞれの連結成分の細線化結果が再び連結することはあり得ない。したがって、 $link(x)$  は収縮可能とならず、 $x$  は削除可能でない。

以上の補題より、次の定理を得る。

[定理 2] 細線化における削除可能な点のタイプは 3, 6, 10,

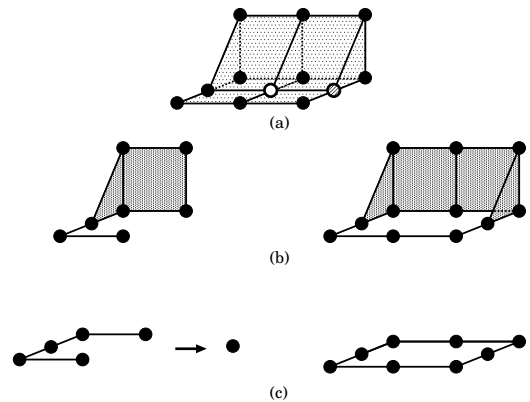


図 8 (a) 離散複体におけるタイプ 12a-a の点 (白) と 12a-b の点 (グレー), (b) それぞれのリンク (左: タイプ 12a-a, 右: 12a-b), (c) リンクの細線化過程と最終結果の形状の相違

11a-a, 12a-a である。

上記の定理で挙げられた削除可能なタイプの点  $x$  はすべて、 $link(x)$  が連結しており、その細線化結果がひとつの点となる。すなわち、これらの点は削除されたとしても局所的に、また画像全体に対しても、その位相構造を変化させない点であると言える。

### 3.4 アルゴリズム 1 の結果と考察

定理 2 を利用することで、アルゴリズム 1 のステップ 1 と 2.3 における点の削除可能性の判定は、離散複体における点のタイプ判定となり、これは  $3 \times 3 \times 3$  の局所領域で実現可能である [7]。

まず、アルゴリズム 1 を適用する 2 つの入力画像を示す。

入力画像 1： 内部に空洞を持つ球（図 9 はその離散複体表現で、図 10 ではその内部構造を見るために同一の離散複体をワイヤーフレーム表現している）；

入力画像 2： 連続した立方格子物体（図 13）。

そして、それら入力画像 1, 2 にアルゴリズム 1 を適用した結果を離散複体表現したものを、それぞれ図 11（図 12 はそのワイヤーフレーム表現）、14 に示す。

入力画像 1 の形状特徴は、内部に空洞を含むことであり、したがって、アルゴリズム 1 の細線化の結果は離散閉曲面になることが望ましい（すなわち、薄面化）。図 11, 12 の結果を観察すると、一見薄面化されているようにも見えるが、よく見ると厳密には離散閉曲面になっていない。図 15, 16 に、離散閉曲

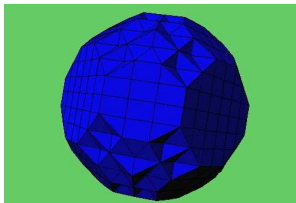


図 9 入力画像 1 の離散複体表現

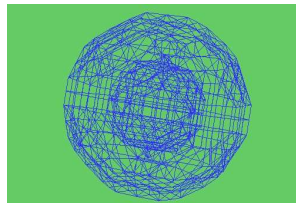


図 10 図 9 のワイヤーフレーム表示

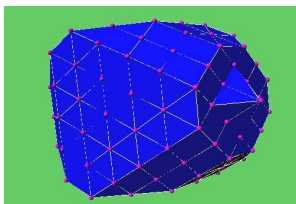


図 11 入力画像 1 へのアルゴリズム 1 の適用結果

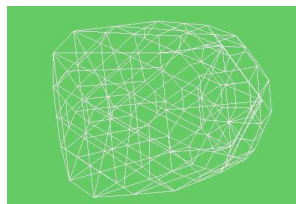


図 12 図 11 のワイヤーフレーム表示

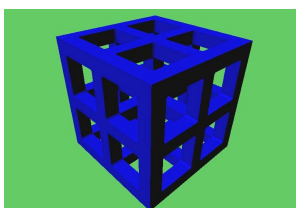


図 13 入力画像 2 の離散複体表現

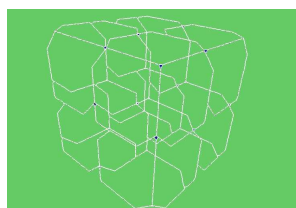


図 14 入力画像 2 へのアルゴリズム 1 の適用結果

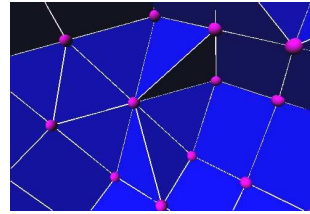


図 15 図 11 の離散閉曲面でない部分の拡大図

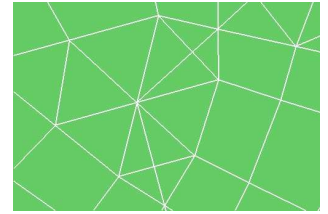


図 16 図 15 のワイヤーフレーム表示

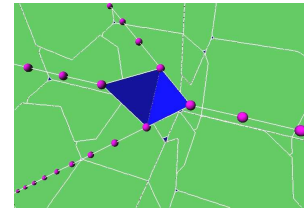


図 17 図 14 の離散曲線でない部分の拡大図

面となっていない部分の拡大図を示す。それらの図より、3次元離散凸多面体が部分的に含まれているのが分かる。その3次元離散凸多面体のいずれの頂点も削除可能点とならないことは明らかである。

入力画像 2 へのアルゴリズム 1 の細線化結果は格子状の離散曲線となることが予想される。図 14 の結果からは、一見細線化処理が行われたように見えるが、この場合も離散複体は1次元ではなく、実は2次元になっている。図 17 に2次元離散凸多面体が含まれている部分の拡大図を示す。

上記の結果のいずれの場合も、削除可能点は存在しないことが分かる。つまり、削除できる点がないため、部分的に離散複体の次元を落とすことができなくなってしまっているのである。その原因を突き詰めていくと、定理 1 の点集合  $V$  に対する離散複体  $K(V)$  の一意性に到達する。 $K(V)$  の  $V$  に対する一意性を維持する限り、 $K(V)$  の次元の削減は望めない。本研究では、より良い細線化結果を得るために、この一意性を捨て、点の削除によってではなく、余分な1次元以上の離散凸多面体を  $K(V)$  より削除することによって  $K(V)$  の次元を補正し、問題に対処する。

## 4. 細線化結果の次元補正

本章ではアルゴリズム 1 の結果から得られる離散複体  $K$  から、余分な1次元以上の離散凸多面体を消去し、 $K$  の次元を補正するアルゴリズムを示す。

### 4.1 2次元補正アルゴリズム

入力画像 1 へのアルゴリズム 1 の適用結果（図 11）のように、離散複体  $K$  の大部分が2次元で局所的に3次元を含んでいる場合、余分な3次元離散凸多面体を消去して薄面化を実現する。このとき、3次元離散凸多面体の面のいくつかも同時に消去し、最終的な結果を位相的に正しくする。

具体的には、以下のようなアルゴリズムになる。ここで、離散複体  $K$  に含まれる  $n$  次元離散凸多面体の集合を  $poly(n, K)$  と表し、同様に、離散凸多面体  $u$  の面  $face(u)$  のうち  $n$  次元



の面の集合を  $face(n, u)$  と表す。また、 $type(x)$  は、点  $x$  のタイプを返す関数とする。さらに、集合  $\mathbf{A}$  に対して  $|\mathbf{A}|$  はその集合  $\mathbf{A}$  の要素数を表す。

[アルゴリズム 2] (2次元補正)

入力： アルゴリズム 1 の出力  $\mathbf{V}$  の 3次元離散複体  $\mathbf{K}$

出力： 余計な離散凸多面体を切削した 2次元離散複体  $\mathbf{K}$

1.  $poly(3, \mathbf{K})$  を求める;
2. **for** 各  $u \in poly(3, \mathbf{K})$ 
  - 2.1. **if**  $|\{v \in face(2, u) : z \in face(1, v)\}| = 2$  となる  $z \in face(1, u)$  がひとつでも存在する
    - then**  $\mathbf{K} \leftarrow \mathbf{K} \setminus (\{u, z\} \cup \{v \in face(2, u) : z \in face(1, v)\})$ ;
    - else**  $|\{w \in poly(3, \mathbf{K}) : v \in face(2, w)\}| = 1$  となる  $v \in face(2, u)$  を選び、 $\mathbf{K} \leftarrow \mathbf{K} \setminus \{u, v\}$ ;
3. **return**  $\mathbf{K}$ .

図 18, 19 にアルゴリズム 2 の適用前後の例を示す。

次元補正方法のアイデアは以下のとおりである。まず、3次元離散凸多面体  $u$  の 1次元面  $z$  のうち、それが離散複体  $\mathbf{K}$  に属するただ 2つの 2次元離散凸多面体  $v_1, v_2$  の面となっているものを探し、あれば図 20 のように  $u, z, v_1, v_2$  を  $\mathbf{K}$  より削除する。そのような 1次元面  $z$  がいない場合、すなわちいずれの  $z$  も  $u$  以外の 3次元離散凸多面体の面にもなっている場合、図 21 のように  $u$  のみの面となっている 2次元離散凸多面体  $v$  を探し、 $u, v$  を  $\mathbf{K}$  より削除する。

図 18 の例においては、図 20 の削除が適用されて図 19 のように次元が補正されていることがわかる。

#### 4.2 1次元補正アルゴリズム

入力画像 2 へのアルゴリズム 1 の適用結果 (図 14) のように、離散複体  $\mathbf{K}$  の大部分が 1次元で局所的に 2次元を含んでいる場合、余分な 2次元離散凸多面体を消去して完全な細線化を実現する。2次元補正アルゴリズムと同様、2次元離散凸多面体とそのいくつかの面を同時に消去し、最終的な結果を位相的に正しくする。具体的には、以下のようなアルゴリズムになる。

[アルゴリズム 3] (1次元補正)

入力： アルゴリズム 1 の出力  $\mathbf{V}$  の 2次元離散複体  $\mathbf{K}$

出力： 余計な離散凸多面体を削除した 1次元離散複体  $\mathbf{K}$

1.  $poly(2, \mathbf{K})$  を求める;
2. **for** 各  $u \in poly(2, \mathbf{K})$ 
  - $|\{w \in poly(2, \mathbf{K}) : v \in face(1, w)\}| = 1$  となる  $v \in face(1, u)$  を選び、 $\mathbf{K} \leftarrow \mathbf{K} \setminus \{u, v\}$ ;
3. **return**  $\mathbf{K}$ .

図 22, 23 にアルゴリズム 3 の適用前後の例を示す。

次元補正方法のアイデアは、2次元補正と同様、離散複体  $\mathbf{K}$  中に現われる 2次元離散凸多面体  $u$  に着目し、図 24 のように  $u$  と同時にその一つの 1次元面  $z$  を削除する。その際、 $z$  が  $u$  以外の  $\mathbf{K}$  の 2次元離散凸多面体の面にならないようにする。

#### 4.3 次元補正を考慮した細線化処理

これまでに述べたアルゴリズムをまとめ、入力画像依存の自動的な細線化アルゴリズムを以下に示す。

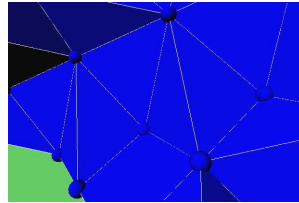


図 18 2次元補正適用前

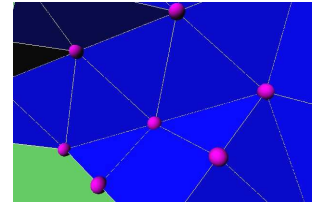


図 19 2次元補正適用後



図 20 1, 2, 3次元離散凸多面体削除による2次元補正

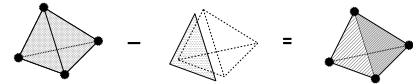


図 21 2, 3次元離散凸多面体削除による2次元補正

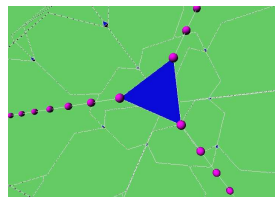


図 22 1次元補正適用前

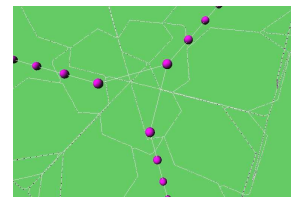


図 23 1次元補正適用後



図 24 1, 2次元離散凸多面体削除による1次元補正

[アルゴリズム 4] (次元補正を考慮した細線化)

入力： 3次元 2値画像  $\mathbf{V}$

出力： 細線化後の  $\mathbf{V}$  に対応した離散複体  $\mathbf{K}(\mathbf{V})$

1.  $\mathbf{V}$  にアルゴリズム 1 を適用;
1.  $\mathbf{V}$  に対応する離散複体  $\mathbf{K}(\mathbf{V})$  を作成;
2. **if**  $\mathbf{K}(\mathbf{V})$  に 3次元離散凸多面体が存在する
  - then**  $\mathbf{K}(\mathbf{V})$  にアルゴリズム 2, 3 をこの順番で適用;
  - else**  $\mathbf{K}(\mathbf{V})$  にアルゴリズム 3 を適用;
4. **return**  $\mathbf{K}(\mathbf{V})$ .

## 5. 実験結果

アルゴリズム 4 を前章の入力画像 2 (図 13) に適用した結果を図 25 に示す。次元補正のおかげで、余計な 2次元離散凸多面体がすべてきれいに消去できているのが分かる。

さらに、入力画像として以下のものを準備する。

入力画像 3： ドーナツ型の物体 (図 26)

入力画像 4： その背景画像 (0 と 1 の値を反転させた画像)

図 27 は、ドーナツ型の物体の細線化結果である。物体の形状特徴として円状の線構造が得られている。実はこの入力画像の場合は、次元補正を適用する必要がなく、アルゴリズム 1 の結果とアルゴリズム 4 が同一となった。

次に、図 26 の背景画像へのアルゴリズム 1 の適用結果を見



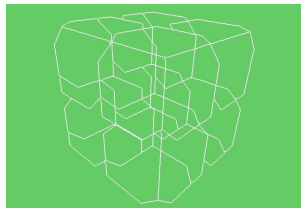


図 25 入力画像 2 (図 13) へのアルゴリズム 4 の適用結果

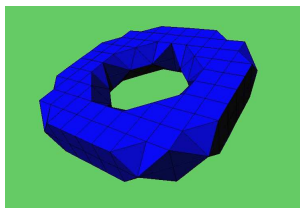


図 26 ドーナツ型の物体

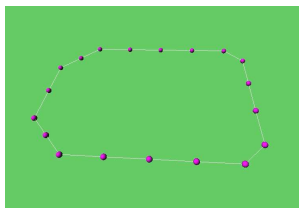


図 27 図 26 の細線化結果

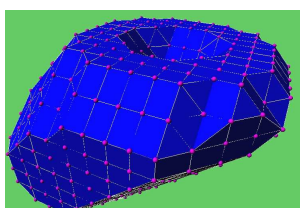


図 28 図 26 の背景へのアルゴリズム 1 の適用結果

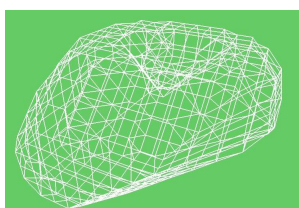


図 29 図 28 のワイヤーフレーム表示

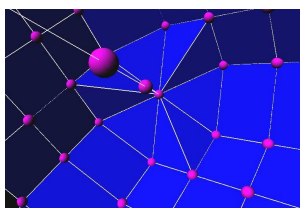


図 30 図 28 の拡大図 (次元補正前)

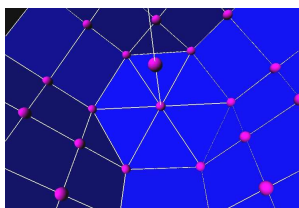


図 31 図 28 の拡大図 (次元補正後)

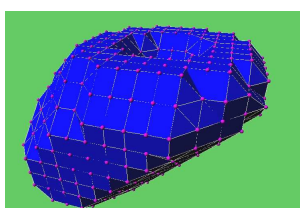


図 32 図 26 の背景へのアルゴリズム 4 の適用結果

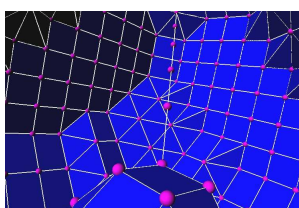


図 33 図 32 の内部

てみると、図 28 (図 29 はそのワイヤーフレーム表現) のようになった。外側は 2 次元の閉じた曲面になっており、内部に 1 次元の曲線分がひとつあるのが見える。この外側の離散閉曲面と内部の離散曲線の接続部分に着目してみると (図 30)、余計な 3 次元離散凸多面体が離散閉曲面に張りついているのが分かる。これにアルゴリズム 4 の次元補正を施すと、図 31 の結果が得られる。図 32, 33 にも、もう少し大域的な、図 26 の背景画像へのアルゴリズム 4 の適用結果を示す。最終結果は、内部に 1 次元離散曲線を持つ 2 次元離散閉曲面 (1 次元離散曲線の

2 端点は 2 次元離散閉曲面に含まれている) となっている。

入力画像 3 と 4 のそれぞれの細線化結果 (図 27 と図 32, 33) を見比べると、それらは互いを補完する関係にあることがわかる。また本研究で提案した細線化アルゴリズムでは、次元の異なる形状特徴が混同するような物体に対しても細線化が行えることを実証している。

## 6. あとがき

本稿では、複体の概念を用いた新しい細線化方法を構築し、処理の要である削除可能点の再帰的な定義を与えた。また、離散複体を用いられているため、各点の位相的性質を明らかにすることができ、それらの性質を利用した効率的な点の削除可能性の判定方法を提案した。さらに、複体の斉次元性を得るために、その次元を補正する方法も示した。

文献 [9] では、多面体ではなく単体を用いた、本稿と類似したアプローチが見られるが、位相保存の定義が異なるため、同一の細線化結果が得られるかどうか今後調べてみる必要がある。また、医療などの実際の応用分野では、位相だけではなく、物体の概形を連想させるような構造 (例えば中心線など) をその幾何学計量から抽出する次元変換が求められており [9]、その要求に答えるためには、位相だけでなく幾何学も考慮した、より柔軟な細線化アルゴリズムの提案を目指す必要がある (本稿の細線化手法では、血管などのループ構造のない木構造は最終的に 1 点になる)。例えば、境界からの距離などの幾何学計量などを用いて、概形保持のため削除してはいけない保存点 (例えば木構造の端点) を設定する方法、削除する点の順番を決定する方法、などを考えていく必要があるだろう。

## 文 献

- [1] G. Bertrand, "Simple points, topological numbers and geodesic neighborhoods in cubic grids," *Pattern Recognition Letters*, Vol. 15, pp. 1003-1011, 1994.
- [2] G. Bertrand, "New notions for discrete topology," in LNCS 1568: Discrete Geometry for Computer Imagery, Proceedings of DGCI'99, G. Bertrand, M. Couprie, L. Perrotin (Eds.), pp. 218-228, 1999.
- [3] T. Y. Kong, A. Rosenfeld, "Digital topology: Introduction and survey," *Computer Vision, Graphics, and Image Processing*, Vol. 48, No. 3, pp. 357-393, 1989.
- [4] T. Y. Kong, "Topology-Preserving Deletion of 1's from 2-, 3-, and 4-Dimensional Binary Images," in LNCS 1347: Discrete Geometry for Computer Imagery, Proceedings of DGCI'97, E. Ahronovitz, C. Fiorio (Eds.), pp. 3-18, 1997.
- [5] Y. Kenmochi, A. Imiya, "Combinatorial boundary tracking of a 3D lattice point set," *Memories of the Faculty of Engineering, Okayama University*, Vol. 38, No. 2, pp. 73-90, 2004.
- [6] G. Malandain, G. Bertrand, N. Ayache, "Topological segmentation of discrete surfaces," *International Journal of Computer Vision*, Vol. 10, No. 2, pp. 183-197, 1993.
- [7] 野村祐介, 剣持雪子, "組合せ構造を用いた離散曲面の局所パターンの数え上げ", 画像の認識・理解シンポジウム (MIRU2004), 2004.
- [8] S. Svensson, G. Sanniti di Baja, "Simplifying curve skeletons in volume images," *Computer Vision and Image Understanding*, Vol. 90, pp. 242-257, 2003.
- [9] 鳥脇純一郎, 3次元デジタル画像処理, 昭晃堂, 東京, 2003.