

ユークリッド距離変換を用いた3次元デジタル画像の薄面化 および細線化の逐次型アルゴリズムとその諸性質

齋藤 豊文[†] 森 健策[†] 鳥脇純一郎[†]

A Sequential Thinning Algorithm for Three Dimensional Digital Pictures Using the Euclidean Distance Transformation and Its Properties

Toyofumi SAITO[†], Kensaku MORI[†], and Jun-ichiro TORIWAKI[†]

あらまし 本論文では、ユークリッド距離変換を用いた3次元デジタル2値画像に対する薄面化および細線化手法を提案し、その性質について述べる。本論文の手法は従来提案されていた手法に対して、特に図形の回転依存性が大幅に改善されている。3次元画像の薄面化／細線化は2次元画像における細線化に対応する基本的な処理であり、図形の構造解析などに利用される。しかし、従来提案されていた手法は図形を主に6方向から順に削っていくというものであり、入力図形が回転した場合、薄面化／細線化結果が大きく変化していた。本手法ではユークリッド距離変換の結果を用いることにより、図形の回転依存性を改善した。

キーワード 距離変換, ユークリッド距離, 3次元, 薄面化, 細線化, スケルトン

1. まえがき

近年の超高速、あるいはスパイラル走査型のX線CT装置の普及によって、立方体ボクセルの3次元配列で定義される3次元(3D)画像を処理する機会が急増している[1], [2]。3D画像処理の基本的手法に関する検討はある程度行われているが、2次元(2D)画像の場合と比べるとまだ極めて不十分である。とりわけ、図形のトポロジカルな性質が関与する処理に関しては3D画像固有の難しさがあって検討が遅れている[3], [4]。そのような処理の一つとして3D画像の薄面化／細線化がある。

3D画像の薄面化／細線化は、3D図形のトポロジカルな性質を保ったまま、また、過度に図形を縮退させることなく原図形を厚さ1の面図形または太さ1の線図形に変換する処理で、2次元画像処理における細線化に対応する画像処理の基本的手法の一つである。それは、2D図形に対する細線化処理と同様、図形の構造解析、記述に不可欠の処理であると考えられる。

2次元図形に対する細線化アルゴリズムは、その処理目的に合わせて数多く考案されてきた[5]。しかし、3D図形の薄面化／細線化の研究は非常に少ない。実際、入力図形に制約をおかず、図形のトポロジーを明

確に保存する^(注1)手法を具体的に提示したのものとしては、Tsaoらのアルゴリズム[6]と川瀬らのアルゴリズム[7]～[9]が挙げられるにとどまる。Tsaoらのアルゴリズムは並列型で、連結性保存の検査のためにチェックプレーンという概念を導入し、6連結、26連結の場合について薄面化アルゴリズムを提案している。一方、川瀬らは逐次型アルゴリズムを提案している。ここでは、連結性の検査には連結指数などを用い、アルゴリズムは6, 18, 18', 26連結のいずれの連結性に対しても適用できる。他の薄面化に関する報告としては[10]～[12]等があるが、これらは主に消去可能性判定の方法について考察しており、薄面化／細線化手順の能力に対する考察はほとんどなされていない。また、トポロジー保存に関する検討はKong, Rosenfeld[13]らの研究にも見られるが、主としてトポロジー保存条件そのものの意味と3D画像特有の難しさを指摘するにとどまり、具体的なアルゴリズムの提案には至っていない。更に、18連結は川瀬ら以外には扱われていない。

また、従来から提案されている薄面化／細線化アルゴリズムはその性質があまり考察されていなかった。特に、原図形の回転に対して薄面化結果が大きく変化するという欠点についてはほとんど言及されていない。

(注1)：図形のトポロジーの保存とは、図形の連結成分、穴、空洞の数およびそれらの位相幾何学的位置関係を変化させないことを言う。

[†] 名古屋大学大学院工学研究科情報工学専攻, 名古屋市
Dept. of Information Engineering, Graduate School of Engineering,
Nagoya University, Nagoya-shi, 464-01 Japan

また、図形の分岐点での状態もあまり調べられていない。3D 画像は2D 画像と比べて近傍の1画素の配置の場合の数が非常に多く、2D 画像の消去可能性判定法と比べて複雑な判定法が必要となり、2D 図形の場合のような簡便な手法の考案は難しい[14]。また、図形形状も複雑となるため、一般的な解決は困難である。具体的応用の場面では基本手法を個別問題の性格に応じて拡張、変形することにより対処することになると考えられる。そのためにも、基本的な性質の良好なアルゴリズムをできるだけ多数開発しておくことは重要である。

本研究では従来提案されていた3D薄面化／細線化手法に比べて図形の回転依存性を大幅に減少させた手法を提案する。このアルゴリズムは、筆者らが従来報告していたアルゴリズムである川瀬らの手法[7]～[9]を改良したもので、ユークリッド距離変換を用いることにより図形の回転依存性を低減している。但し、従来のユークリッド距離変換手法は6近傍距離変換あるいは26近傍距離変換に比べて図形の回転依存性が少ないが、計算時間がかかり、かつ多くの記憶容量を必要とするという問題点があった。これに対し、筆者らは効率の良いユークリッド距離変換の手法を考案し[15]、[16]、この方法を用いて計算時間、記憶容量等をあまり犠牲にすることなく薄面化アルゴリズムに適用できるに至ったものである。更に、同じ考え方にたった細線化の方法を具体的に示し、6, 18, 18', 26連結の4種類の連結性を扱い、かつ、任意の図形に対してトポロジーを保存する薄面化／細線化アルゴリズムを明示する。

本論文では、2.で薄面化、細線化に対する基本的な考え方を述べた後、3.でアルゴリズムを提案する。4., 5.では実験例を示しアルゴリズムの性質について考察を行う。

2. 薄面化、細線化の定義と要求条件

2.1 基礎的定義

ここでは、本論文で必要な画像処理に関する概念や記法の定義を与えておく。

本論文では、大きさ L 行 M 列 N 段の3次元デジタル画像（以下3D画像と書く）(図1)を扱う。第 i 行 j 列 k 段の画素（ボクセル、単に点とも言う）を (i, j, k) で表し、その濃度値が f_{ijk} で与えられる画像を $F = \{f_{ijk}\}$ のように書く。

濃度値が0と1しかとらない画像を2値画像と呼び、値0, 1の画素をそれぞれ、0-画素, 1-画素と呼ぶ。また、1-画素の集合を図形、0-画素の集合を背景と呼ぶ。

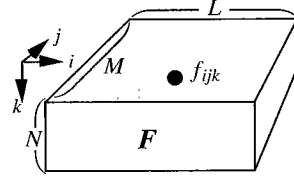


図1 3次元デジタル画像
Fig. 1 Three dimensional digitized picture $F = \{f_{ijk}\}$.

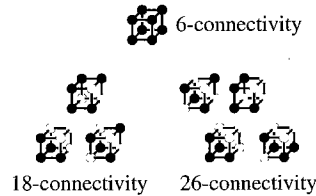


図2 3次元単体
Fig. 2 All possible 3-D simplexes.

本文中で用いるユークリッド距離変換は実際の距離値を2乗した値に変換を行うものを意味し、以下のように定義される[15]、[16]。

[定義1] 2値画像 $F = \{f_{ijk}\}$ の2乗ユークリッド距離変換を $S = \{s_{ijk}\}$ とするとき、 s_{ijk} は入力画像 F の全0-画素から (i, j, k) までの距離値の最小値の2乗である。すなわち、

$$s_{ijk} = \min_{(p,q,r)} \{(i-p)^2 + (j-q)^2 + (k-r)^2\};$$

$$: f_{ijk} = 0, 1 \leq p \leq L, 1 \leq q \leq M, 1 \leq r \leq N\}.$$

以下簡単のため、 F から S を求める処理および画像 S をともに F の距離変換と呼ぶ。

2.2 薄面化、細線化の定義

3次元画像の薄面化／細線化は以下のように定義または意味付けされる。

[定義2] (薄面化) 厚さのある図形を、厚さ1の中心面に変換する。厚さ1の面はそれ以上変形しない。ここで面図形とは、その図形に含まれるすべての画素 x が、以下の条件 i, ii の少なくとも一つを満たす場合を言う。
i. x の $3 \times 3 \times 3$ 近傍に図2の配置（3次元単体）を含まない（図形の厚さが1である）。

ii. x が消去可能でない。

[定義3] (細線化) 厚さ、太さのある図形を太さ1の中心線に変換する。線図形とは、その図形が消去可能画素を含まない場合を言う。

これらは2D 図形の場合の細線化の3D 画像への拡張である。但し、3D 画像の特徴として、線図形に変換す

るか面図形に変換するかの区別が生じる。また3D図形の場合、トポロジー保存条件により、図形によっては(例えば空洞をもつ図形の場合)細線化を行っても線図形に変換できない場合が生じる。よって、両者の区別は必ずしも明確ではないが、厚さ1の帯状の図形をそのままにするか(薄面化)、中心線にまで変形するか(細線化)では両者の区別は明確である(図3)。どちらが良いかは個々の応用によって異なるため、アルゴリズムとしては両者を開発しておくことが望まれる。

2.3 アルゴリズム設計上の要求条件

2D画像の細線化の場合に対応して、以下のような事項が薄面化/細線化に対する要求条件として挙げられる[17], [7]~[9]。

R1. 原図形のトポロジーを保存する。

R2. 交差部のような特殊な部分を除いて、厚さ1の面図形(薄面化)若しくは、太さ1の線図形(細線化)になる。

R3. 薄面化/細線化結果は原図形の中心面/線上に位置する。

R4. 図形の端が過度に縮退しない。

R5. 交差部において自然な形状になる。

R6. 図形の回転に影響されない(原図形を回転したものは、回転前の図形の処理結果を回転したものと等しい、若しくは大きく違わない)。

但し、これらは互いに両立しにくいものも含まれる。また、R1を除いて、どの条件も理論的に厳密に定義することは難しく、従って処理結果の評価はある程度人間の直感的な判断、理解に頼らざるを得ない。これらの点も2D図形に対する細線化の場合と同様である。

2.4 従来の方と本論文の方との関連

薄面化そのものについては[7]~[9]にゆずり、以下に、本論文に関係の深い事項のみをまとめておく。従来から提案されているトポロジー保存条件を満たす方法として、Tsao and Fuの方法(T-F法)[6]、および川瀬らの方法(K-T法)[7]~[9]がある。いずれの方法も6サブサイクルをもち、 $3 \times 3 \times 3$ 近傍を用いた近傍演算に

よる反復型アルゴリズムである。6サブサイクルを用いるのは、図形を前後左右上下の6方向から均等に削っていくためである。原図形のトポロジー保存性はいずれのアルゴリズムも満たしているが、画素の消去可能性判定法は異なる。T-F法では注目画素を含む 3×3 の平面(チェックプレーンと呼ばれる)を用いて消去可能性判定を行い、K-T法では連結指数[18]~[20]などの局所特徴量を用いて行う。T-F法では6, 26連結のみが提案されているのに対してK-T法は6, 18, 18', 26連結のいずれに対しても判定法が示されている(注2, 3)。消去可能条件についてはこのほかにもいくつかの検討がなされている[12]が、いずれの連結性についても適用可能な手法は連結指数を用いた手法のみである。

図形の縮退、ヒゲ(不自然な小さい枝)の発生については、T-F法では縮退が多めでヒゲの発生が少なめ、逆にK-T法では縮退が少なめでヒゲの発生が多めとされている[7], [8]が、入力図形によっては異なる場合もある。図形の回転依存性についてはいずれのアルゴリズムも十分であるとは言えず、原図形を回転した場合の薄面化結果は、回転前の原図形の薄面化結果に対してかなり変化する。特にT-F法は回転によって大きな変化が現れる。細線化についてはT-F法ではそのアルゴリズムが示されているがK-T法では示されていない。

本論文ではK-T法をもとにした薄面化/細線化手法を中心に述べる。K-T法は、トポロジー保存が厳密に証明されていること、6, 18, 18', 26連結のいずれの連結性においてもアルゴリズムが提案されている唯一の方法であることなどから、この方法を生かした薄面化/細線化手法を示してその性質を明確にしておくことは十分に意義があることと考える。

また、本論文では距離変換を組み入れた薄面化/細線化を試みる。距離変換を利用した3D薄面化/細線化アルゴリズムは現在まで提案されていない。2Dの場合には阿部らが距離変換を用いて細線化を高速に行う手法を提案している[21]。3Dの場合も上と同様な手法を開発できる可能性はあるが、ここではむしろユークリッド距離変換を用いることで薄面化/細線化の基本的性質の改善を行うことを目的とする。しかし、従来提案されていたユークリッド距離変換手順では計算時間、必要記憶容量ともに大きい3D薄面化の一部と

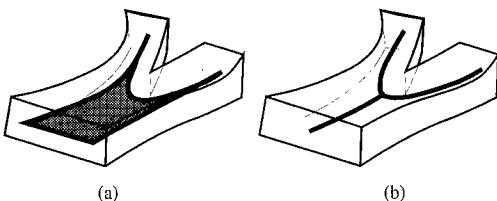


図3 薄面化と細線化

Fig. 3 (a) Surface thinning. (b) Axis thinning.

(注2): 同様の方法が18連結にも適用できることが証明できるというコメントが文献[13]に見られる。

(注3): 18'連結は18連結の処理の対になるものとして、1-画素の消去を伴う変換のみで現れる[3], [18]~[20]。

して使用するのは困難であった。これに対して筆者らが考案した n 次元ユークリッド距離変換のアルゴリズム [15], [16] は、多次元画像に対しても計算時間、記憶容量共に小さく、薄面化処理の一部として使用することも容易であると考えられる。また、ユークリッド距離変換を利用することにより、回転依存性を大幅に改善できることを後に示す。

3. 薄面化／細線化のアルゴリズム

3.1 アルゴリズムの基本的考え方

3D 薄面化アルゴリズムは基本的には 2D の細線化の 3D への拡張であり、以下のような基本的考え方をもとにしている。

(1) 境界画素 (その画素の 6 近傍に 1 個以上の 0-画素が存在する) 中で最小の距離値をもつ画素を抽出し、それらの画素をその近傍の 0-画素の配置によって 9 種類に分類する。

(2) 同一種類の境界画素の消去可能性を判定し、消去可能画素を逐次的に消去 (0-画素に変更) する。

(3) (1) で分類した 9 種類の境界画素の各々について上記 2 の処理を行う (サブサイクル)。

(4) 上記処理 (1) ~ (3) を消去できる画素が存在しなくなるまで繰り返す (メインサイクル)。

また、細線化については文献 [7]~[9] では触れられていないが、以下に示すように薄面化アルゴリズムに若干の変更を施すことで細線化を容易に実現できる。基本的に、細線化と薄面化の違いは端点保存条件のみである。すなわち、薄面化では面の端の画素を保存するのに対して、線の端の画素を保存するように画素の保存条件を若干変更するだけで細線化を実現できる。

なお、厳密に言えば図形内に空洞が存在した場合、トポロジーを保存するという条件のもとでは局所的に線図形に変換できないこともあり、また、複雑な分岐が存在する部分でも線図形とはならない場合が存在する。

3.2 薄面化アルゴリズムの概略

本論文で提案するアルゴリズムの概略を以下に示す。以下のアルゴリズムでは高速化のために作業用のリストを用いている。必要となるリストの大きさは入力図形の境界画素数の 2 倍程度であり、入力画像によって異なる。最悪の場合、入力画像の全画素数と同等の要素数が必要となる。また、図 4 に本節で述べる薄面化アルゴリズムおよび 3.3 で述べる細線化アルゴリズムのフローチャートを示す。

[アルゴリズム 1] (薄面化)

入力画像: $F = \{f_{ijk}\}$: 3 次元配列, 入力 2 値画像. f_{ijk} は画素 (i, j, k) の値 (1 または 0) を示す。

出力画像: $F = \{f_{ijk}\}$: 入力と同じ配列. 薄面化の結果が入る。

境界画素リスト: $list = \{e_n : (i_n, j_n, k_n, l_n)\}$: リストの各要素は、3 次元座標値とラベル値の 4 個の値からなる。

関数: $int(x)$: 引き数 x を整数化する (切捨て)。

[Step 1] (2 乗ユークリッド距離変換)

for all (i, j, k) 's do

$$f_{ijk} \leftarrow \min_{(p, q, r)} \{d_i((i, j, k), (p, q, r))^2$$

$$; f_{pqr(input)} = 0, 1 \leq p \leq L, 1 \leq q \leq M, 1 \leq r \leq N\}$$

enddo

(ユークリッド距離変換アルゴリズムの詳細は文献 [15], [16] を参照)

(Shift distance values to use integers 1~19 as labels)

for all (i, j, k) 's do*

$$\text{if } (f_{ijk} \neq 0) f_{ijk} \leftarrow f_{ijk} + 20$$

enddo

$maxd \leftarrow$ 画像 F 中の 2 乗ユークリッド距離値の最大値

$d \leftarrow$ 画像 F 中の 0 でない 2 乗ユークリッド距離値の最小値

[Step 2] (初期境界画素の検出)

for all (i, j, k) 's do

if $(f_{ijk} > 20$ and f_{ijk} の 6 近傍に 0-画素が存在) then

(i, j, k, f_{ijk}) をリストに加える

$$f_{ijk} \leftarrow 1$$

endif

enddo

$N \leftarrow$ length of the list

(メインサイクル)

[Step 3] (永久保存点の検出 ($l_n \leftarrow 0$ は永久保存点を表す) と境界画素の分類)

for $n = 1$ to N do

if $(l_n \leq d)$ then

if $((i_n, j_n, k_n)$ が消去不可能**) then

e_n をリストから削除 (永久保存点)

else

if $((i_n, j_n, k_n)$ での図形の厚さが 1^{***}) then

```

 $e_n$  をリストから削除 (永久保存点)
else
   $m \leftarrow$  the number of non-0 voxels
  in 26-neighbors of  $(i_n, j_n, k_n)$ 
   $l_n \leftarrow \text{int}(m/3)+7$ 
enddo

```

[Step 4] (画素の消去) (サブサイクル処理)

```

 $N' \leftarrow$  length of the list
for  $\text{bordertype} = 7$  to 15 do
  for  $n = 1$  to  $N'$  do
    if  $(l_n = \text{bordertype})$  then
      if  $((i_n, j_n, k_n)$  が消去不可能) then
         $e_n$  をリストから削除 (永久保存点)
      else
        if  $((i_n, j_n, k_n)$  での図形の厚さが 1) then
           $e_n$  をリストから削除 (永久保存点)
        else
           $f_{i_n, j_n, k_n} \leftarrow 0$  (画素の消去)
           $e_n$  をリストから削除
          for  $((i_n, j_n, k_n)$  の6近傍のすべて) do
            if  $(f_{ijk} > 20)$  then
               $(i_n, j_n, k_n, f_{ijk})$  をリストに加える
              (リストへの画素の追加)
               $f_{ijk} \leftarrow 1$ 
            enddo
          enddo
        enddo
      enddo
    enddo
  enddo

```

[Step 5] (終了条件判定)

```

 $N \leftarrow$  length of the list
 $d \leftarrow$  リスト中の距離値の最小値
(the minimum value in all over 20)
if  $(d < \text{maxd}$  or  $N > 0)$  goto [Step 3]

```

[Step 6] (後処理)

```

for all  $(i, j, k)$ 's do
  if  $(f_{ijk} \neq 0)$  then  $f_{ijk} \leftarrow 1$ 
enddo

```

- * “for all (i, j, k) 's do” は図5のように画像を走査して各画素の処理を行う。
- ** 消去可能性判定については文献[18]~[20]を参照。
- *** “図形の厚さが1”とは、画素 (i, j, k) が3次元単体(3-

D simplex) (図2) に含まれないことである。

各Stepの処理の目的および意図は以下のとおりである。

Step 1 で入力された2値画像に対して距離変換を行

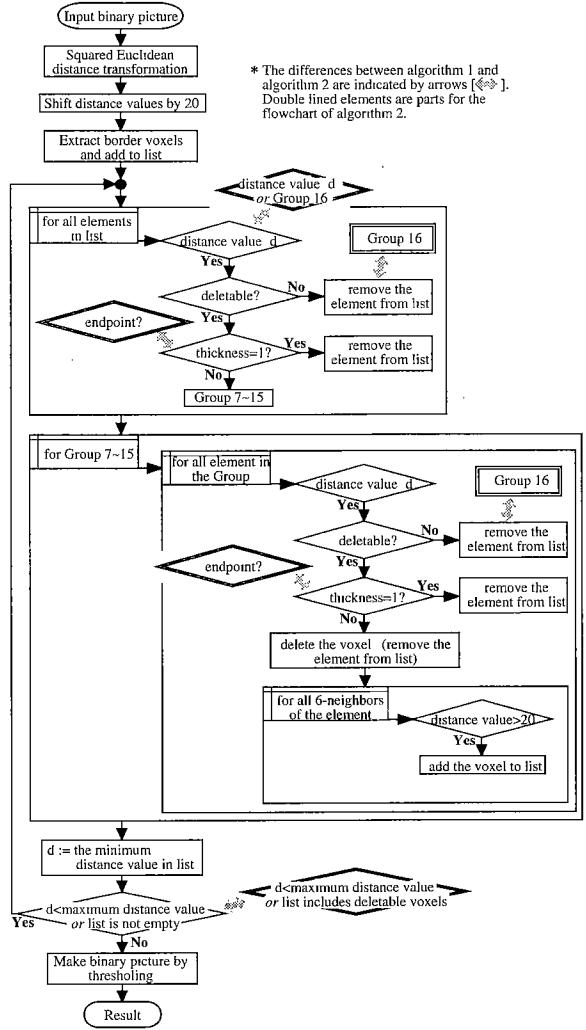


図4 アルゴリズム1, 2のフローチャート

Fig. 4 Flowchart of algorithm 1 and 2.

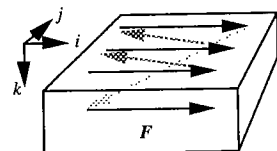


図5 画像の走査方向
Fig. 5 Scanning mode.

い、更にその距離値に20を加える。これは値0~19を境界画素のグループ分けのためのラベル値などとして用いるためである。次に、Step 2ではじめに処理対象となる境界画素リストを作成する。Step 3, 4, 5はメインサイクルであり、Step 3では処理対象となる境界画素の抽出と境界画素のグループ分けを行う。この際、消去不可能画素の検出も行う。処理対象となる境界画素はリスト中の最小距離値をもつ画素である。距離値の小さい画素から順に処理を行うことで、薄面化結果が入力図形の中心に位置することが期待でき(要求R3)、また、図形の回転による影響を少なくできる(要求R6)。境界画素のグループ分けはその画素の26近傍の非0画素の数に基づいて行う。これはヒゲの発生を抑制するためである(要求R4)。また、消去不可能画素はリストから除かれ、永久保存点として以後の処理で消去されることはない。

Step 4では画素の消去、保存処理を逐次的に行う。Step 4の各サブサイクルでは、9種類に分類された中の1種類の画素集合に対してのみ処理を行う。消去不可能画素、厚さ1になったとみなされる画素は永久保存点とし、その後、消去可能な1-画素を消去する。消去の際、消去した画素の6近傍を調べ、リストに登録されたことがない画素が存在すればリストに追加する。追加された画素は次のメインサイクルで処理対象となる。画素の消去は逐次的に行われるため、薄面化結果の図形が入力図形のトポロジーを保存することは明白である(要求R1)。Step 5は終了条件判定である。

画素の消去可能性判定はK-T法と同様の連結指数を用いており、 $3 \times 3 \times 3$ 近傍局所演算により計算される。また、“図形の厚さが1”かどうかの判定は $2 \times 2 \times 2$ 局所パターンマッチングによって決定される[18]~[20]。

上記アルゴリズムをリストを用いずに行うことも可能である。記憶容量が少ない方がよければそのように書き換えればよい。但し、計算時間は非常に大きくなる。

3.3 細線化アルゴリズムの概略

上記アルゴリズム1と同様の考え方で細線化も実現できる。薄面化との違いは、端点処理の部分と消去不可能画素の扱い方にある。薄面化の場合は消去不可能画素は永久保存点と同一であったが、細線化の場合は必ずしも永久保存点とはならない。以下のアルゴリズムでは薄面化と細線化が異なる部分を下線で示す。

[アルゴリズム2] (細線化)

入力画像などの説明はアルゴリズム1と同様である。

[Step 1] (2乗ユークリッド距離変換) アルゴリズム1, Step 1と同様

[Step 2] (初期境界画素の検出) アルゴリズム1, Step 2と同様

(メインサイクル)

[Step 3] (永久保存点の検出 ($l_n \leftarrow 0$ は永久保存点を表す)と境界画素の分類)

for $n = 1$ to N do

if ($l_n \leq d$) then

if ((i_n, j_n, k_n) が δ 消去不可能) then

$l_n \leftarrow 16$ (一時保存点)

else

if ((i_n, j_n, k_n) が δ 端点*) then

e_n をリストから削除 (永久保存点)

else

$m \leftarrow (i_n, j_n, k_n)$ の26近傍に存在する
非0ボクセルの数

$l_n \leftarrow \text{int}(m/3)+7$

enddo

[Step 4] (画素の消去) (サブサイクル処理)

$N' \leftarrow \text{length of the list}$

for $\text{bordertype} = 7$ to 15 do

for $n = 1$ to N' do

if ($l_n = \text{bordertype}$) then

if ((i_n, j_n, k_n) が δ 消去不可能) then

$l_n \leftarrow 16$ (一時保存点)

else

if ((i_n, j_n, k_n) が δ 端点) then

e_n をリストから削除 (永久保存点)

else

$f_{i_n, j_n, k_n} \leftarrow 0$ (画素の消去)

e_n をリストから削除

for ((i_n, j_n, k_n) の6近傍のすべて) do

if ($f_{i_{jk}, j_{jk}, k_{jk}} > 20$) then

$(i_n, j_n, k_n, f_{i_{jk}, j_{jk}, k_{jk}})$ をリストに加える

$f_{i_{jk}, j_{jk}, k_{jk}} \leftarrow 1$

enddo

enddo

enddo

[Step 5] (終了条件判定)

$N \leftarrow$ length of the list

$d \leftarrow$ リスト中の距離値の最小値

(the minimum value in all over 20)

$m \leftarrow$ the number of non-deletable voxels in the list ($l_n = 16$)

if ($d < maxd$ or $m \neq N$) goto [Step 3]

[Step 6] (後処理) アルゴリズム1, Step 6 と同様

* 端点とはその26近傍に非0ボクセルを一つだけもつ画素を言う。

4. 実験

4.1 人工図形に対する実験例

本論文で提案した方法と従来の手法を人工画像に対して適用した例を図6に、また、これらの図形を回転した図形に対する適用例を図7に示す。使用した画像は球を2個結合した画像と、直方体を3個組み合わせた図形である。連結数は26連結を用いた。回転した図形は、図6(a)に示した図形を*ij*平面に沿って5度ずつ回転した後最近傍法を用いて生成した。図8に図形の回転角度と処理後の図形の画素数の関係を示す。

4.2 実画像に対する実験例

本論文で提案した方法を実画像に対して適用した例を図9, 10に示す。図9は人間の手のCT画像をしきい値処理で2値化して得られた2値画像に対して薄面化処理および細線化処理を適用した例、図10は胸部CT画像から気管支内部を抽出した画像[2]に対して細線化処理を適用した例である。なお、図9の画像サイズは $320 \times 320 \times 120$ 画素、処理時間は45秒、図10は $320 \times 320 \times 62$ 画素、処理時間は17秒であった。

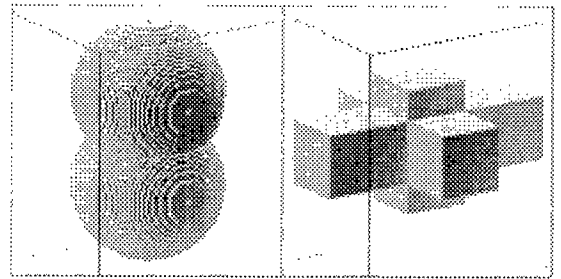
5. 考察

5.1 薄面化および細線化結果の評価

薄面化および細線化の結果について、2.3に述べた要求に照らして検討した結果を要約しておく。

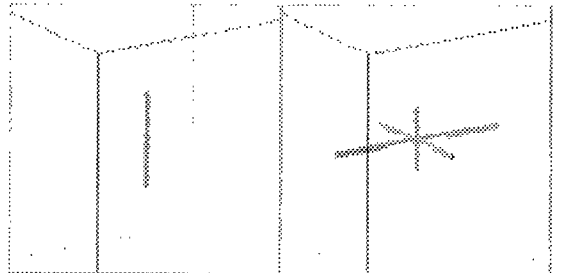
(1) いずれの手法も原図形のトポロジーを保存する。これはもともと消去可能性判定条件から保証されている。

(2) 今回の図形は空洞を含まず、また交差部も比較的単純であるため、いずれの手法も薄面化の場合は厚さ1の面図形に、細線化の場合は太さ1の線図形に



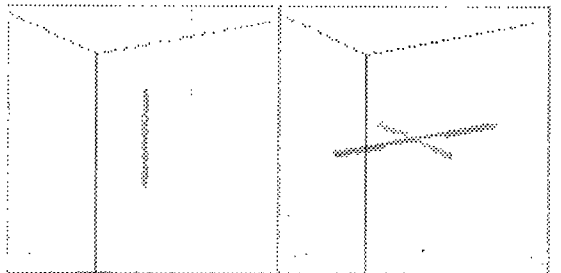
(a) Connected spheres.

(b) Cross.



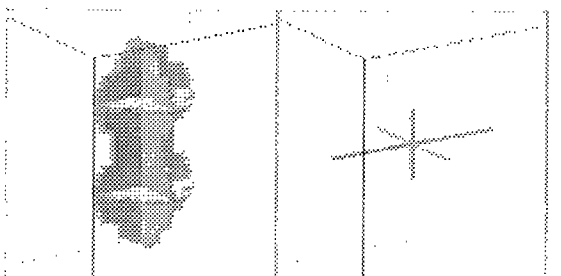
(c) Result of Algorithm 1 for Fig. 6(a).

(d) Result of Algorithm 1 for Fig. 6(b).



(e) Result of K-T method for Fig. 6(a).

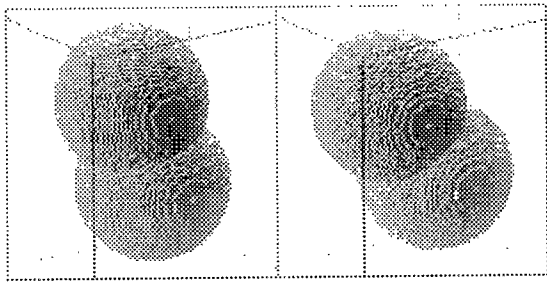
(f) Result of K-T method for Fig. 6(b).



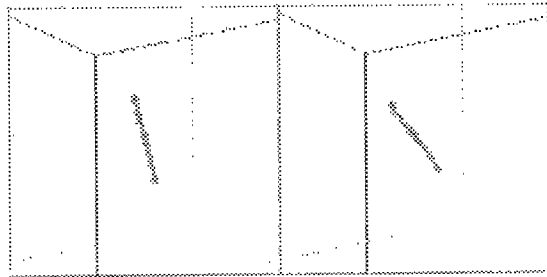
(g) Result of T-F method for Fig. 6(a).

(h) Result of T-F method for Fig. 6(b).

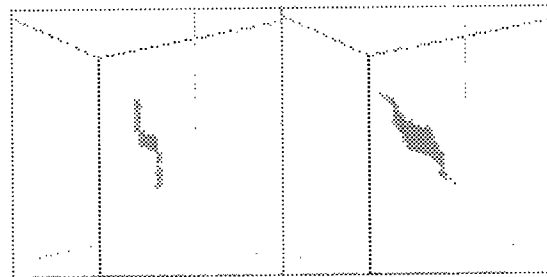
図6 人工図形に対する薄面化の例
Fig. 6 Examples of the results of thinning for artificial figures.



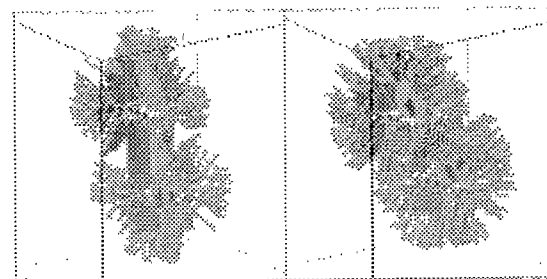
(a) The rotated figure of Fig. 6(a) by 15 degree. (b) The rotated figure of Fig. 6(a) by 35 degree.



(c) The result of Algorithm 1 for (a). (d) The result of Algorithm 1 for (b).



(e) The result of K-T method for (a). (f) The result of K-T method for (b).



(g) The result of T-F method for (a). (h) The result of T-F method for (b).

図7 回転した図形に対する薄面化の例
Fig. 7 The results of thinning for rotated figures.

なっている。

(3) 薄面化結果は回転前の図形に対してはほぼ原図形の中心付近に位置するが、図形が傾いた場合、K-T法では薄面化結果が図形の中心線から外れることがあり、入力図形によっては中心を通らないこともある。これは、画素の消去を6方向から逐次的に行う逐次型アルゴリズムのためである。また、T-F法は図形を回転した場合、多数の画素を残すので中心を通るかどうかは判断できない。本論文の方法の結果は回転の影響はほとんど受けず、極めて良好である。

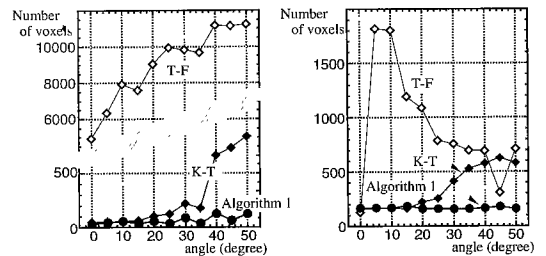
(4) 図形の縮退は、K-T法においてより多く見られる(図6, 7)。例えば3個の直方体の交わった図形に薄面化を実行した例で、最も短い縦の直方体が、薄面化後には縮退により消えている。一方、T-F法は他の二つの手法に比べてより多くの画素を残す傾向があり、ヒゲが多く発生しているのがわかる。本論文の方法は他の方法よりもおおむね良い結果を与える。

(5) 交差部の形状はおおよそ自然な形状であると思われる。

(6) アルゴリズム1の結果は入力図形の回転に対して安定しており、図形を回転させても、薄面化結果は原図形の中心線付近に位置する。これに対して、K-T法、T-F法は図形の回転に対する影響が大きい。

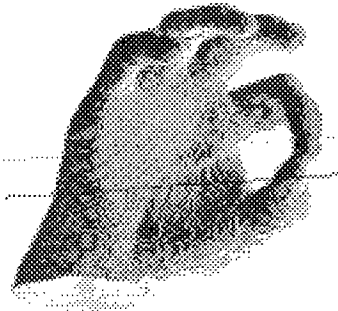
全体として、今回の方法による薄面化/細線化結果は他の手法に比べて良好であり、特に、図形の回転依存性については非常に良好である。これは、ユークリッド距離変換に基づいて消去画素を探索するため、図形の角度が変わっても図形の縁からほぼおなじ速度で図形を削っていくためであると考えられる。

一方、縮退とヒゲの発生は相反する要求であり、本論文のどの方法でも、また、薄面化においても細線化

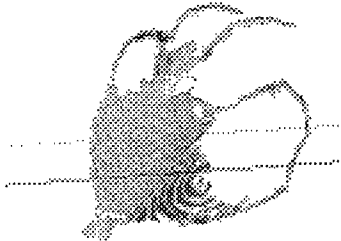


(a) For rotated version of Fig. 6(a). (b) For rotated version of Fig. 6(b).

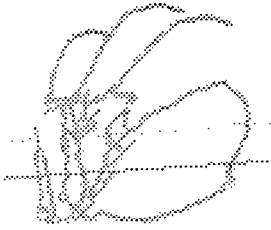
図8 図形の回転に対する薄面化結果の画素数の変化
Fig. 8 Relationship between the number of voxels in the results of thinning and the rotation angle.



(a) Input picture (Hand).

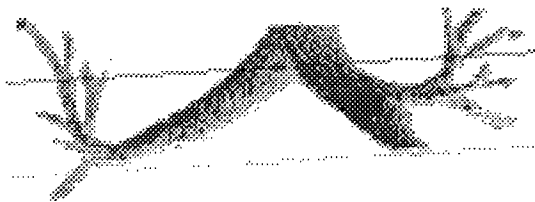


(b) Result of Algorithm 1 (surface thinning) for Fig. 9(a).

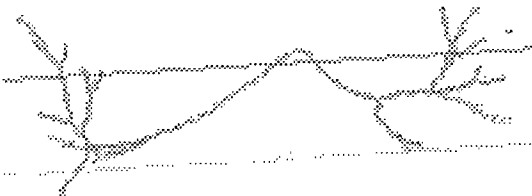


(c) Result of Algorithm 2 (axis thinning) for Fig. 9(a).

図9 CT画像に対する薄面化および細線化の例
Fig. 9 Examples of the results of surface thinning and axis thinning for a CT image.



(a) Input picture (bronchus).



(b) Result of Algorithm 2 for Fig. 10(a).

図10 CT画像に対する細線化の例
Fig. 10 Examples of the results of axis thinning for a CT image.

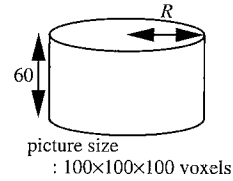


図11 計算時間の評価に用いた図形
Fig. 11 A sample of an input picture for evaluation of computation time.

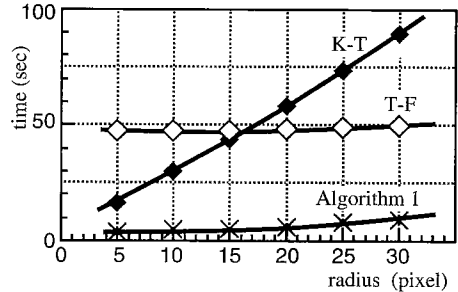


図12 計算時間
Fig. 12 Experimental evaluation of computation time.

においても極めて微妙なトレードオフの関係にある。今回の実験では細線化においてより厳しい問題になるようである。本論文の細線化手法は、図10のような細長い管上の図形に対してはうまく機能するが、入力画像によってはかなりの縮退を生じる場合も見られた。現在のところおそらく個別の問題ごとにヒューリスティックな付帯条件を追加しなくてはならないと思われる。これには骨格線および距離変換画像の尾根線等の利用が考えられるが、その具体化は今後の検討事項である。

5.2 計算時間および記憶容量

アルゴリズム1,2の計算時間は、入力画像の大きさ、図形の大きさ、形に大きく影響を受ける。ここでは入力画像(画素数 $100 \times 100 \times 100$)において、画像の中央に高さ60、半径 R の円柱を置き(図11)、円柱の半径を変えて計算時間を測定した結果を図12に示す。使用した計算機は名古屋大学大型計算機センターFACOM M-1800、プログラムはFORTRANで作成した。

K-T法、アルゴリズム1、アルゴリズム2は同じ傾向を示す。これらはいずれもサブサイクルをもつアルゴリズムを用いており、各画素の消去可能性判定に必要な計算量もそれほど異ならないためである。但し、アルゴリズム1,2は境界画素リストを用いているために他の手法に比べて高速である。T-F法が他のアルゴリズム

と若干異なる傾向にあるのは、消去した画素数が他の方法に比べて非常に少ないためであると考えられる。記憶容量については、アルゴリズム1,2は境界画素リストを用いるため、他の手法に比べてそれだけ余分の記憶容量が必要となる。必要となる境界画素リストの要素数は入力図形の境界画素数の2倍程度である。図9の図形の場合、図形全体は393,500画素からなり要素数は18,500個、図10の場合、図形全体は47,600画素からなり要素数は5,400個であった。

6. む す び

本論文ではユークリッド距離変換を用いることで図形の回転依存性を大幅に低減した3次元2値画像に対する薄面化/細線化アルゴリズムを提案し、その性質を明らかにした。また、人工画像、実画像に本手法を適用し、良好な結果を得た。

本アルゴリズムは従来のアルゴリズムに対して、良好な薄面化/細線化結果を得ることができるため、各応用分野での目的に沿った薄面化/細線化手法を開発する際の基本的アルゴリズムとしても有用であると考えられる。

今後の課題として、縮退と枝の発生を制御する条件を含めて、処理結果の更に詳細な品質評価、高速化、濃淡情報を利用した薄面化などが挙げられる。

謝辞 日ごろ熱心に討論頂く名古屋大学横井茂樹教授、実験に協力頂いた早瀬研二君（現在日立製作所）、佐々木剛君（現在NTT）および研究室の方々に感謝します。また、4.2の試料を提供頂き、実験に際して助言を頂いた藤田保健衛生大学片田和廣博士、安野泰史博士に感謝します。本研究の一部は文部省科研費および厚生省がん研究助成金による。

文 献

- [1] 鳥脇純一郎, 鈴木秀智, “3次元デジタル画像処理とその医用画像への応用,” *Medical Imaging Technology*, vol.8, no.4, pp.411-422, 1990.
- [2] 森 健策, 長谷川純一, 鳥脇純一郎, 安野泰史, 片田和廣, “可変しき値処理と距離変換を用いた3次元胸部CT像からの肺がん候補領域自動抽出手順,” *MEDICAL IMAGING TECHNOLOGY*, vol.12, no.3, pp.216-223, 1994.
- [3] 鳥脇純一郎, 横井茂樹, “3次元デジタル画像処理アルゴリズムの基礎,” *信学論(D)*, vol.J68-D, no.4, pp.426-432, April 1985.
- [4] 鳥脇純一郎, “3次元画像のデジタル処理,” 高木幹雄, 鳥脇純一郎, 田村秀行編: *画像処理アルゴリズムの最新動向*, pp.84-107, 新技術コミュニケーションズ, 1986.
- [5] 田村秀行, “図形の細線化についての比較研究,” *情報処理学会イメージングプロセッシング研資料*, イメージプロセッ

- ング 1-1, 1975.
- [6] Y.F.Tsao and K.S. Fu, “A parallel thinning algorithm for 3-D pictures,” *CGIP*, vol.17, pp.315-331, 1981.
- [7] 川瀬洋一, 横井茂樹, 福村晃夫, 鳥脇純一郎, “3次元デジタル画像の薄面化の一方,” *信学研資(パターン認識と学習研究会)*, PRL82-69, 1983.
- [8] 川瀬洋一, 横井茂樹, 鳥脇純一郎, “3次元デジタル画像の逐次薄面化アルゴリズムについて,” *信学論(D)*, vol.J68-D, no.4, pp.481-488, April 1985.
- [9] J.Toriwaki and S.Yokoi, “Algorithms for skeletonizing three-dimensional digitized binary picture,” *Proc. of SPIE Sym.* no.435, pp.2-9, 1983.
- [10] W.X.Gong and G.Bertrand, “A simple parallel 3D thinning algorithms,” *Proc. of 10th ICPR*, pp.188-190, 1988.
- [11] J.Mukherjee, B.N. Chatterji and P.P.Das, “Thinning of 3-D images using the safe point thinning algorithm (SPTA),” *Pattern Recognition Letters*, vol.10, pp.167-173, 1989.
- [12] T.C.Lee, R.Kashyap, and C.N.Chu, “Building skeleton models via 3-D medial surface/axis thinning algorithms,” *CVGIP*, vol.56, no.6, pp.462-478, 1994.
- [13] T.Y.Kong, A. Rosenfeld, “Survey digital topology : introduction and survey,” *CVGIP*, vol.48, pp.357-393, 1989.
- [14] 鳥脇純一郎, 齋藤豊文, 早瀬研二, “3次元デジタル画像の局所パターン数の計数および連結指数の分布について,” *信学論(D-II)*, vol.J78-D-II, no.7, pp.1080-1090, July 1995.
- [15] 齋藤豊文, 鳥脇純一郎, “3次元デジタル画像に対するユークリッド距離変換,” *信学論(D-II)*, vol.J76-D-II, no.3, pp.445-453, March 1993.
- [16] T.Saito and J.Toriwaki, “New algorithms for Euclidean distance transformation of an n-dimensional digitized picture with applications,” *Pattern Recognition*, vol.27, no.11, pp.1551-1565, 1994.
- [17] 鳥脇純一郎, “画像理解のためのデジタル画像処理 [III],” pp.53-54, 昭晃堂, 1988.
- [18] 米倉達彦, 横井茂樹, 鳥脇純一郎, 福村晃夫, “3次元デジタル空間における図形の連結性とオイラー数,” *信学論(D)*, vol.J65-D, no.1, pp.80-87, Jan. 1982.
- [19] J.Toriwaki, S.Yokoi, T.Yonekura, and T.Fukumura, “Topological properties and topology preserving transformation of a three-dimensional binary pictures,” *Proc. 6th ICPR*, pp.414-419, 1982.
- [20] 米倉達彦, 横井茂樹, 鳥脇純一郎, 福村晃夫, “3次元デジタル空間における1-要素の消去可能性と図形収縮について,” *信学論(D)*, vol.J65-D, no.12, pp.1543-1549, Dec. 1982.
- [21] 鈴木 智, 阿部圭一, “距離変換の結果を利用した二値画像の逐次型細線化,” *信学論(D)*, vol.J68-D, no.4, pp.473-480, April 1985.

(平成7年4月12日受付, 8年3月12日再受付)



齋藤 豊文 (正員)

昭61名大・工・電子卒。平3同大大学院博士課程了。同年4月名古屋大学工学部助手。平7同講師。工博。パターン認識、画像処理に関する研究に従事。日本エム・イー学会会員。



森 健策 (学生員)

平4年3月名大・工・電子卒。平6年3月同大大学院工学研究科博士課程前期課程了。平6年4月より日本學術振興会特別研究員。現在同大大学院工学研究科博士課程後期課程在学中。3次元画像処理およびバーチャルリアリティの医学応用に関する研究に従事。日本医用画像工学会、コンピュータ支援画像診断学会、日本エム・イー学会各会員。



鳥脇純一郎 (正員)

昭37名大・工・電子卒。昭和42年同大大学院博士課程了。同年4月名古屋大学工学部助手。以後、昭45助教授、昭49名古屋大学大型計算機センター助教授、昭55豊橋技術科学大学情報工学系教授、昭58名古屋大学工学部電子工学科教授を経て、昭60年4月より同情報工学科教授。工博。パターン認識、画像処理、グラフィックスおよびそれらの医学情報処理への応用に関する研究に従事。著書「画像理解のためのデジタル画像処理I, II」(昭晃堂), 「パターン認識と画像処理」(朝倉書店), 「認識工学」(コロナ社), 情報処理学会, 日本エム・イー学会, 医療情報学会, 人工知能学会, IEEE各会員。