# 文字図形にたいする細線化アルゴリズムの提案と評価

5D－2

清水道夫　福田宏*　中村義作**
長野県短期大学　静岡県立大学*　東海大学**

## 1 はじめに

2値図形の細線化は文字認識や図形認識の前処理に使われる一技法であり、様々なアルゴリズムが提案されてきた。しかし、いずれの手法も線分の縮退、消失、ひげの出現、交差部での歪みなどのマイナス要素が生じやすい性質をもっている。

本報告では Hilditch の古典的な細線化アルゴリズム[1]を改良し、より安定した出力の得られるアルゴリズムを提案する。さらに、文字図形に対して特に有効な2種類のマスクを付加する。そして、このアルゴリズムを評価するために、実際の 432 種類の文字フォントに対して細線化を行い品質を調査した。その結果、既提案の代表的なアルゴリズム[1-4]よりも高い性能が得られることがわかった。

## 2 連結数

アルゴリズムを記述するために必要な用語を定義する。処理対象は画面正方格子上の 0 か 1 の値をとる画素（2値図形）である。8近傍点とは、図1に示すように注目画素に接する8つの画素である。以下8近傍点に番号をつけ、その値を $\chi_k (k = 1,2,3,\cdots,8)$ で表す。

A Thinning Algorithm for Digital Figures of Characters, Michio SHIMIZU, Hiroshi FUKUDA* and Gisaku NAKAMURA**, Department of Liberal Arts, Nagano Prefectural College,8-49-7 Miwa Nagano-Shi, 380-8525 JAPAN,*School of Administration and Informatics, University of Shizuoka,52-1 Yada Shizuoka-Shi, 422-8526 JAPAN, ** Research Institute of Education, Tokai University,2-28-4 Tomigaya Shibuya-Ku, Tokyo-To, 151-8677 JAPAN

注目画素の連結数 $N_c$は、$\bar{x}_k = 1 - x_k$ として

| | | |
|---|---|---|
| 4 | 3 | 2 |
| 5 | 0 | 1 |
| 6 | 7 | 8 |

図1　8近傍点

次式で定義される[5]。ただし、添え字は $k = 7$ の時、$k + 2 = 1$ とする。

$$N_c = \sum_{k=1,3,5,7} \bar{x}_k (1 - \bar{x}_{k+1} \bar{x}_{k+2})$$

## 3 並列型 Hilditch アルゴリズム

細線化アルゴリズムは逐次型と並列型に分けられる。逐次型は画面左上から右下へ向かって1行づつ画素を走査しその順に削除する方法であり、並列型は全ての画素を同時に調べて削除する方法である。

古典的な逐次型アルゴリズムである Hilditch の方法は、連結数を用いた簡明なアルゴリズムであるが、細線化結果が等方性に欠けるという逐次型特有の欠点をもつ。本報告では、Hilditch のアルゴリズムを並列型に置き換えることでのその欠点を克服し、これをアルゴリズムの骨格とする。

並列型の処理は、心線を消失させないためにいくつかのサイクルに処理を分割する必要がある。4サイクルの処理とは、並列処理を $c$=1,3,5,7 の4つに分け、各サイクル $c$ では $\chi_c = 0$ の画素のみを細線化処理の対象とする。つまり、文字図形の右側面、上側面、左側面、下側面の順に

処理される。

並列型 Hilditch 細線化アルゴリズムは、連結数 $N_c = 1$ である境界点を前処理で検索し、その点のみを４サイクルの並列処理で削除する。削除する点は各サイクル c において以下の条件を満たす点である。

(1) $\sum_{k=1}^{8} x_k > 1$：端点でないこと

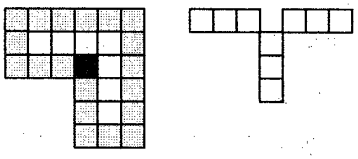
(2) $N_c = 1$： 境界点であること

文字図形に対してこのアルゴリズムを用いて細線化を行うと、実際に Hilditch の方法を等方化、安定化した細線が出力される。

## 4 マスク

本アルゴリズムは、並列化したことによって、逐次型では不可能なマスク処理を付加することが可能となる。マスク処理は例外的な点の処理を行うための手法で、既にいくつかの並列型細線化アルゴリズムに適用されている。

まず L 字や T 字の角を保護するために内部点マスクを導入する。図２(a)の灰色で示す部分は、



(a)　　(b)

**図２ マスクの処理対象点**

前処理で検索される削除処理の対象画素であるが、黒色で示す内側の角の画素を含まない。そこで、この画素を内部点マスクで検出し削除処理の対象に加える。内部点マスクによってＬ字、T 字、V 字などの内側の角を正しく細線化できる。

一方、細線が生成される最終段階において、図２(b)に示す T 字の中心画素は $N_c = 1$ の境界点なので削除されてしまう。しかし、視覚的には保存されたほうがよいので、外部点マスクを導入してこの画素を削除対象外の点とする。

マスクは大局的な図形情報を取り扱えるように、3×3 の８近傍点を拡張し、5×5 の大きさのパターンで構成されている。

なお、Hilditch のアルゴリズムも３で提案した並列型 Hilditch アルゴリズムも完全８連結[5]の細線を生成する。ところが、マスクの導入により、生成される細線は不完全８連結[3]となる。

## 5 評価

３で述べたアルゴリズムは、並列化の効果でヒゲが生じにくい安定した細線を出力する。さらに、４のマスクの導入によって交差点や曲がり角での歪みが防止される。これらの効果を既提案のアルゴリズムと比較検討するために、432 個の様々な種類・大きさの文字フォントに対して、実際に細線化をおこなう。細線化の品質は定量的には測定できないが、７個の細線品質評価項目[3]について視覚的な５段階評価を行った。その結果、ここで提案したアルゴリズムは、処理時間はそれほど増加せずに、期待通りほぼ全ての項目に対して既提案のアルゴリズムを上回る安定した性能を発揮することがわかった。

## 6 おわりに

本アルゴリズムは文字図形だけでなく、一般の２値画像データの細線化処理にも適用可能であると考えられる。ただし、アルファベット“X”のように、一定の交差角度を持つ２直線にたいしては、交差部分の処理について改善の余地が残されている。

## 文 献

[1] C. Judich Hilditch: Machine Intelligence 4, (Edinburgh Univ. Press), 403-420 (1969).
[2] E. S. Deutsch: C. ACM 15, 827-837 (1972).
[3] 田村秀行:電子技術総合研究所報告,第 835 号, 26-64 (1983).
[4] 鶴岡他: 信学技報, PRL78-47,41-49 (1978).
[5] 横井他: 信学論 56-D, 662-669 (1973).