

補間処理手法の簡略化による折れ線グラフ画像の高速認識

安達 則男 大町 真一郎 阿曾 弘具

東北大学大学院工学研究科 〒980-8579 宮城県仙台市青葉区荒巻字青葉 05

E-mail: adachi@aso.ecei.tohoku.ac.jp, {machi, aso}@ecei.tohoku.ac.jp

あらまし 文書画像の認識・理解に関する研究はその多くがテキスト部分を対象としたものであるが、文書画像中のグラフ画像も多く情報を抱えている。グラフ画像を認識し、理解することが可能となれば電子化された文書をより効率的に利用できるようになる。文書画像中のグラフ画像を対象とした研究の一つに実線、点線、破線および一点鎖線で構成させた折れ線グラフを認識するものがあるが、速度の点で問題があった。本論文ではグラフ認識の高速化手法を提案し、実験によりその有効性を示す。

キーワード グラフ認識, 文書認識, 連結成分, 追跡経路の統合・補間

High-Speed Recognition of Line Graph Images In Documents by Simplified Interpolation Technique

Norio ADACHI Shinichiro OMACHI and Hirotomo ASO

Graduate School of Engineering, Tohoku University Aoba 05, Aramaki, Aoba-ku, Sendai-shi, Miyagi 980-8579 Japan

E-mail: adachi@aso.ecei.tohoku.ac.jp, {machi, aso}@ecei.tohoku.ac.jp

Abstract Most documents recognition researches are for text area. However graph images on documents contain many information. Graph information makes it possible to make use more efficiently of the electronic documents. We propose hi-speed recognition technique for line, dotted line, dashed line and one-point chain line graph images on documents, and show effectiveness by experiments.

Keyword Recognition of line graph image, Document recognition, Connected components, Fusion and interpolation between tracing routes

1. まえがき

文書画像の認識・理解に関する研究は主にそのテキスト部分を対象とし、多くの成果を残している。しかし、文書画像中のグラフ画像も多く情報を含んでおり、グラフ画像を単に画像ではなく個々のデータとして認識することが可能になれば、電子化されたグラフを再利用したり、ユーザーの目的により任意のレイアウトに変更して出力することなどが可能になる。

このような理由から、文書画像中のグラフ認識を目的とした研究が行われている。横倉らは棒グラフを対象とし、グラフ要素間の構成規則をもとに様々な表現形式の棒グラフを認識する手法を提案している[1]。Leeらは、マーカの存在する実線の折れ線グラフを認識対象とし、折れ線を形成しているマーカをプレートマッチングで認識し、そのマーカの場所を軸のラベルと比較することで折れ線グラフを認識している[2]。布田らは、折れ線パターンが実線や点線、一点鎖線などの折れ線グラフを対象としている[3]。折れ線の線種パターンを連結成分の黒画素数と連結成分間の距

離を特徴として表す手法を用いている。折れ線グラフとしての特徴を利用することにより線を追跡し、各線種パターンに対応する折れ線を認識しているが追跡に膨大な時間がかかる点が問題であった。

本文では、実線、点線、破線そして一点鎖線の存在するグラフ画像を対象として、布田らの行ったマーカのない折れ線グラフを認識する手法をもとに、処理時間を短縮させる高速認識手法を提案する。グラフ画像において、違う線種パターンが重なりあう折れ線グラフでは連結成分の追跡が途切れてしまい、そこで同じ線種パターンを探し、それら二つの追跡経路を結合する。このとき利用する手法が同種線種パターンの追跡経路の補間であり、従来の手法ではあらゆる可能性を考慮していたために処理時間がかかっていた。提案する補間処理手法では、結合するとき追跡経路間に存在する画素の位置データなどから特徴を抽出し、その特徴をもとに最適な連結追跡間の連結筋道をたてて、それにより処理時間の短縮をはかる。以下、2.では本論文で提案するグラフ画像認識および簡略化した補間処

理手法を述べ、3.では認識実験により提案手法の評価を行う。4.は結論である。

2. 提案手法

本章では、まず認識対象とするグラフ画像、グラフ認識手法の概要を述べ、提案手法を示す。

2.1. グラフ認識手法

2.1.1. 認識対象グラフ

以下の条件を満たす折れ線グラフ画像を認識対象とする。文書画像中からのグラフ画像の切り出しや傾き補正は完了しているものとする。折れ線に限らず曲線で構成された線グラフも認識対象とするが、簡単のためにすべて折れ線グラフと記述する。

- (ア) マーカが表示されていない。
- (イ) グラフ領域が x 軸 y 軸を含む長方形枠で囲まれている。
- (ウ) グラフ上の折れ線は x 軸方向に戻らない。
- (エ) 凡例領域が一箇所に集まっている。
- (オ) 線種パターンは、実線、点線、破線、一点鎖線とする。複数種類の点線パターン、破線パターン、一点鎖線パターンがあってもよいが、線種パターンと折れ線は 1 対 1 に対応している。
- (カ) 目盛線は等間隔とする（ログスケールは対象外）。

2.1.2. グラフ認識手法の流れ

本論文でのグラフ認識手法の流れを図 1 に示す。文書画像中のグラフ領域を切り出し、折れ線を追跡する。最終的な目的は折れ線グラフ各点の数値データを得ることである。

- 前処理
 - 入力した画像をあらかじめ細線化[4]し、水平方向、垂直方向に走査し、黒画素の累積数から軸の検出を行う。その後 3 回の領域分割を行う。
 - (ア) 軸内領域と軸外領域の分割
 - (イ) 軸内領域でグラフ領域と凡例領域の分割
 - (ウ) 凡例領域で凡例と線種パターンの分割
- 目盛線検出
 - 軸内領域より目盛線候補を抽出する。候補は軸に接触しており、比較的少ない画素で構成される連結成分となる。そして目盛線候補から等間隔で並んでいる連結成分を検出することで目盛線を検出する。
- 追跡処理
 - 領域分割により抽出されたグラフ領域に対して、連結成分追跡を行うことにより折れ線を抽出し、折れ線に対応する座標を出力する。この処理については 2.2 において詳しく述べる。
- 文字領域に対する処理
 - 領域分割処理により抽出された文字領域から一文字ごとの連結成分を切り出し、文字認識処理を実

行し、文字列を生成する。

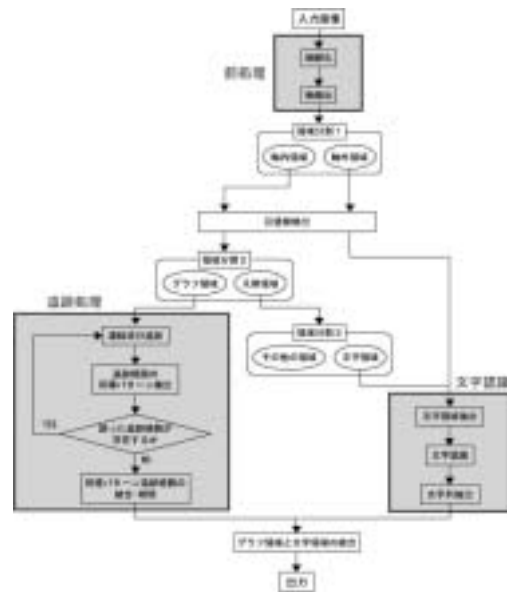


図 1 認識手法の流れ

2.2. 連結成分追跡手法

マーカの存在しない折れ線グラフを認識するために、グラフ領域内に存在する線種パターンを用いて折れ線を認識し、それらを区別する。まずは実線で表された折れ線を検出し、その後、実線以外の線種パターンを検出する。以下に各処理について述べる。

2.2.1. 実線の検出

実線を表す連結成分は、グラフ領域内の他の連結成分と比べて十分に長い。そこでグラフ領域内の連結成分のうち、 n_{max} より大きい連結成分を、実線を表す連結成分として抽出する。

2.2.2. 連結成分追跡

次に、点線や一点鎖線といった比較的小さな連結成分で構成されている線種パターンを抽出する。連結成分を追跡し、その追跡経路上の連結成分の特徴から線種パターンを抽出する。連結成分同士を正確な追跡経路で追跡するために以下の手順で処理を行う。

1. 追跡距離の正規化

連結成分同士を追跡する際、連結成分間の距離を追跡する基準として用いる。この基準が単純にユークリッド距離では異なる線種パターンが近接している場合は誤った追跡を行ってしまうため、いずれの線種パターンでもそれらを構成している連結成分間の距離は一定の値を持っているという特徴から、追跡すべき距離を抽出し、それにもとづき追跡ができるよう距離の正規化をはかる。

2. 追跡距離と追跡方向の融合

連結成分からの追跡方向を 4 分割し、図 2 のように追跡距離と追跡方向を融合させる。これにより、

いくつかの条件を満たす追跡対象を追跡することができるため、作成される追跡経路は、異なる線種パターンが交差した点線や実線により線種パターンが遮へいされている部分で切断された追跡経路となる。つまり、可能な限り誤追跡を避けた追跡経路となる。

さらに、作成された追跡経路が持つ線種パターンを抽出し、それをもとに追跡経路から誤追跡を行った部分経路を検出し、それを削除する。

上記の手順を実行した後の追跡経路は、正解となる部分追跡経路が誤追跡経路よりも長くなっている。これにより、追跡経路上に存在する連結成分のパターンを抽出する際、多数決処理により正解の線種パターンを抽出することが可能となる。

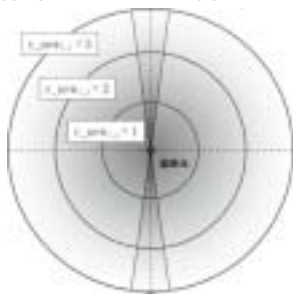


図 2 追跡距離と追跡方向の融合

2.2.3. 追跡経路の線種パターンの抽出

追跡経路の線種パターンを抽出するため、追跡経路上の連結成分間の図 3 で示す特徴を 3 次元ベクトル $(n_i, n_j, r_{i,j})$ として定義する。ここで n_i は追跡経路上の各連結成分 i を構成する黒画素数、 $r_{i,j}$ は連結成分 i と連結成分 j の間のユークリッド距離である。追跡経路上の連結成分間の特徴から、追跡経路がもつ線種パターンを抽出する手法として、投票を用いる [5][6]。追跡経路 k 上のそれぞれの連結成分 i, j 間の特徴 $(n_i, n_j, r_{i,j})$ について、3 次元の特徴空間で、中心 $(n_i, n_j, r_{i,j})$ と半径 $(\delta_1, \delta_2, \delta_3)$ (これは誤差を考慮したものである) で決まる超楕円内に投票し、最も高い投票数を得た点を追跡経路 k が持つ線種パターン $(n_{patk1}, n_{patk2}, r_{pat1,2})$ として抽出する。投票されなかった部分追跡経路は誤追跡経路となる。パターン投票時の半径 $(\delta_1, \delta_2, \delta_3)$ はグラフ画像が持っている量子化誤差から求める。定数 a, b, c を用いて以下のように定義する。

$$\begin{aligned} \delta_1 &= a + \frac{n_i}{b} \\ \delta_2 &= a + \frac{n_j}{b} \\ \delta_3 &= a + \frac{r_{i,j}}{c} \end{aligned} \quad (1)$$

a, b, c の値は予備実験より求める。

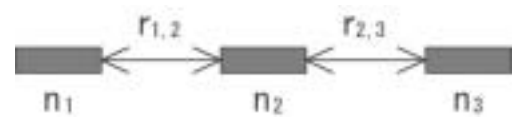


図 3 連結成分間の特徴

2.2.4. 追跡経路の統合

2.2.2 で作成された追跡経路は、実線との重なりや他の線種パターンとの交点において切断された、途切れた追跡経路となっている。追跡経路の統合では、この途切れた追跡経路を各追跡経路の線種パターンを用いることによって統合する。その後、統合された追跡経路を補間し、折れ線の形状を復元する。

各追跡経路において抽出された線種パターンについて 2.2.2 で示した 3 次元の特徴空間に投票し、投票数の多い点からグラフ領域内に存在する線種パターンとする。この処理により、すべての追跡経路線種パターンから、グラフ領域内に存在するすべての線種パターンとその数を検出することができる。

2.3. 補間処理

本節では、同じ線種パターンを持つ追跡経路について、追跡経路間を補間し、各追跡経路を結合する手法を説明する。追跡経路の統合処理により、各追跡経路が持つ線種パターンは既知となっており、各追跡経路が属する折れ線とその線種パターンも既知である。したがって、追跡経路が分断されている場合でも次に連結成分が存在するはずの位置間での距離が推定できる。そこで、追跡経路の線種パターンに基づいて、最適な経路を探索し評価関数によって補間経路を決定する。

布田らが提案した補間処理手法 [1] では、探索手法では DP マッチングを使い、追跡経路間の補間を実現している。また、ビームサーチにより枝刈りを行ってはいたが、全補間点数が大きいほど、グラフ領域内の追跡経路上で切断数が多いほど処理に時間がかかってしまっていた。

2.3.1. 簡略化した補間処理手法

従来法では、図 4 で示すように、グラフ領域内での途切れた追跡経路元から、次の途切れた追跡経路先までの空間全てを走査し補間候補先の黒画素を抽出している。追跡経路先までに数点を抽出、追跡し最適な経路を探索している。しかし、これでは 2.3. で述べたように多大な処理時間を要してしまう。

簡略化した補間処理手法では、処理時間短縮のために、グラフ領域内の途切れた追跡経路元と途切れた追跡経路先の間を補間するために以下の二つの条件を満たす点のみを探索する。

1. 実線および他種線種パターン上の点である

2. 追跡経路元と追跡経路先を結ぶ線分から大きく離れていない

条件1.は、追跡経路が途切れるのが、実線もしくは他の線種パターンで表される経路に接触あるいは交差している場合に限定されると仮定して導かれたものである。

また、それまでの追跡経路からの余弦関数を加えることで、急激な変化を抑えている。補間候補先を探索にあたり、用いる評価関数を以下に定義する。

$$P = \max_k \sum_{i=1}^N f(x_{i-1}^k, x_i^k) / N \quad (3)$$

$$f(x_{i-1}^k, x_i^k) = \cos \theta_i^k \quad (x_i^k \text{が黒画素の場合})$$

$$\text{拘束条件: } \|x_i^k - x_{i-1}^k\| = m_i^k$$

ただし、 x_i は補間経路 k 上の i 番目の補間点、 N は補間に要した全補間点数、 θ_i は元の補間経路と続く補間経路とがなす角度であり、 m_i は次の式に従う。

$$m_i^k = \begin{cases} n_{\text{patk1}} (\|x_{i-3}^k - x_{i-2}^k\| = n_{\text{patk2}}) \\ n_{\text{patk2}} (\|x_{i-3}^k - x_{i-2}^k\| = n_{\text{patk1}}) \\ n_{\text{patk1,2}} (\|x_{i-3}^k - x_{i-2}^k\| = n_{\text{patk1,2}}) \end{cases}$$

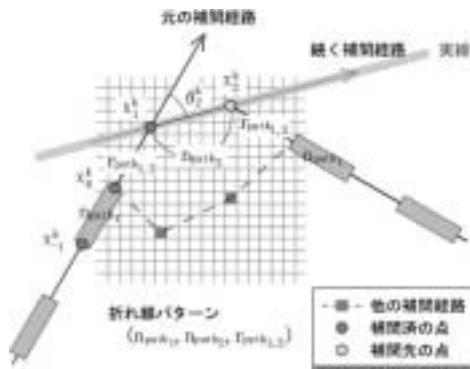


図4 追跡経路の補間

3. 認識実験

3.1. 実験条件

本論文で提案した簡略化した補間処理手法を用いた折れ線グラフ認識手法が有効に働くこと、従来手法より高速化されていることを確認するための認識実験を行った。全ての入力画像は、各文献から 400dpi で取り込んだ。各画像に対して、提案手法を適用して得られた折れ線グラフの数値情報（座標情報）を gnuplot によってプロットし、出力画像とした。実験に先立ち、予備実験を行い、 n_{max} , r_{max} , および $a \sim c$ を以下のように設定した。

$$\begin{aligned} n_{\text{max}} &= \text{Cols}/12 \\ r_{\text{max}} &= \text{Cols}/15 \\ a &= 2 \\ b &= 10 \\ c &= 20 \end{aligned}$$

ただし、Cols は入力画像の水平方向の画素数である。したがって n_{max} と r_{max} は画像の解像度に依存しない。 $a \sim c$ の値は 400dpi の場合の値として設定するが、他の解像度の場合は応じて適切に選ぶ必要がある。今回は 400dpi の設定でうまく連結できない画像においては $a \sim c$ の値を変更している。

3.2. 実験結果と考察

3.2.1. 実験結果

線種パターン抽出処理の評価

提案した折れ線グラフ認識手法で、線種パターンの表現法よりグラフ中の線種パターンが抽出されることを確認するために図5の画像を入力とした。図6の出力画像から4つの折れ線が異なるパターンとして出力されていることがわかる。

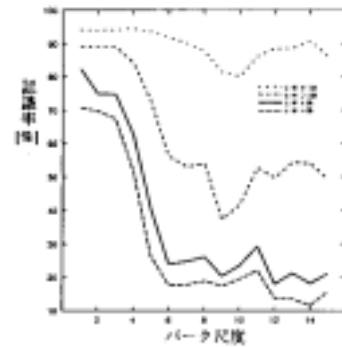


図5 入力画像 1(ppm 画像 496 pixel x 502 pixel)

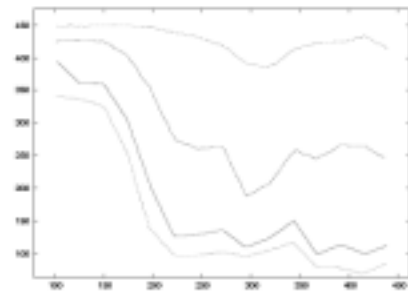


図6 出力画像 1

経路の補間処理の評価

次に、分断された折れ線の経路を補間することにより、同一の折れ線が特定できることを確認するための実験を行った。図7はグラフ領域内の線種パターンが破線の折れ線が実線によって分断されている折れ線グラフである。この場合、追跡経路が分断されているため、それらの追跡経路を線種パターンをもとに正しく統合し、その追跡経路間を補間する必要がある。結果

を見ると、実線により切断されている追跡経路が正しく統合され、その追跡経路が正しく補間されていることがわかる。

図9は比較的大きな連結成分から構成されている折れ線であり、一つの折れ線が実線によって大部分が遮断されている折れ線グラフ画像である。この場合も同種線種パターン追跡経路の統合と補間が有効に働き、大きく間があいている追跡経路間を正確に補間できていることがわかる。

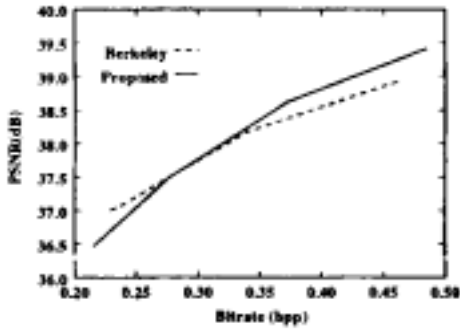


図7 入力画像 2(ppm 画像 525 pixel × 370 pixel)

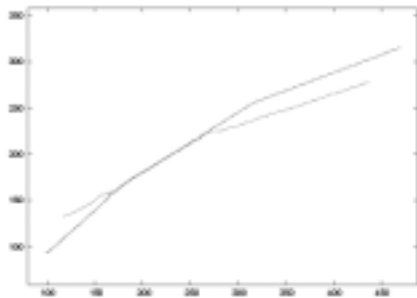


図8 出力画像 2

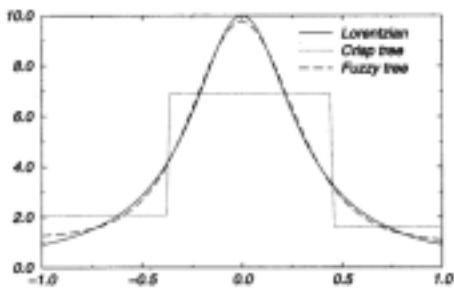


図9 入力画像 3(ppm 画像 1224 pixel × 816 pixel)

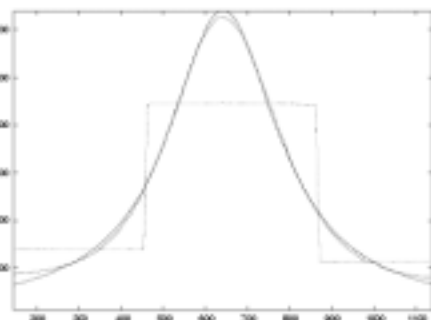


図10 出力画像 3

3.2.2. 処理時間

各入力画像の認識に要した計算時間を表1に示す。使用計算機は、Intel Pentium 800MHzである。表より図5以外において処理時間が従来手法の半分以下になっていることがわかる。従来手法では処理時間のほとんどが追跡経路の補間で占められていたが、補間経路を全走査で処理するよりも、補間経路先を追跡経路間の特徴を用いることで候補を限定することができたため半分に抑えることができた。グラフ領域内の追跡経路が、実線や他種線種パターンを持つ追跡経路と交差、接触することの多いグラフ画像では、簡略化した補間処理手法を用いることで大幅な処理時間の短縮が実現できることがわかる。しかし、交差や接触がない、もしくは少ない画像においては処理時間の変化はほとんどないことが確認された。

表1 各認識実験に要した計算時間

入力画像	図5	図7	図9
補間全走査手法	1.76	68.91	1643.93
簡略補間手法	2.42	15.3	738.2

入力画像	図11	図13
補間全走査手法	29.14	61.26
簡略補間手法	10.15	20.04

3.2.3. 考察

以上の実験結果から、提案手法が従来手法に比べ処理時間の短縮できたことが確認された。しかし、折れ線同士が線種パターンを持つ間隔以上に近接している場合や折れ線が他の折れ線と重なっている場合などは従来手法と同様に、本手法でも正しく認識できなかった。その例を図11に示す。出力をみると一部欠けている。これはグラフ領域内の点線と一点鎖線が重なっているため、その経路を持つ折れ線パターンとは異なったものとなってしまう、いずれの折れ線にも属さない追跡経路として扱われてしまったからである。

また、従来手法では追跡経路の補間処理として、補間候補を周り全ての画素を対象として全走査し、評価関数により最も当てはまる補間経路を探索していたために、誤った補間経路を探索することは少なかったが、簡略化した補間処理手法では、補間候補を追跡経路の間の特徴より探索し、また連結成分のベクトルも考慮しているため、逆に誤った経路が生成されてしまうこ

とある．その例を図 13 に示す．図 13 では入力画像の x 軸の目盛りが 0.4 の部分が軸に接しているのに対し，出力画像では先の追跡経路のベクトルが合わなかったために，誤った追跡経路を辿っている．

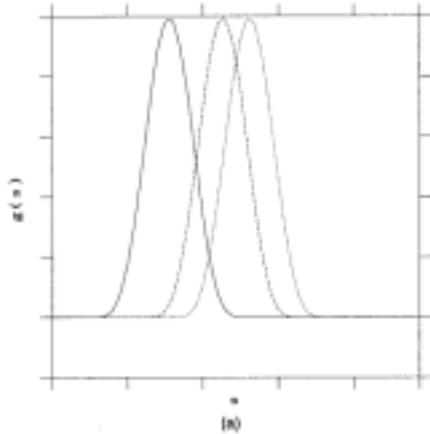


図 11 入力画像 4(ppm 画像 1208 pixel × 1170 pixel)

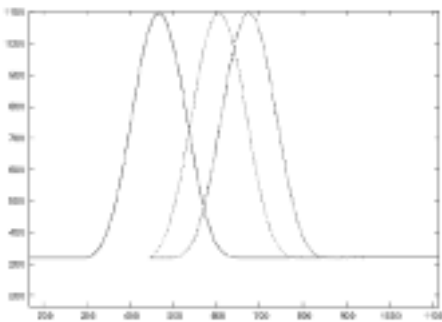


図 12 出力画像 4

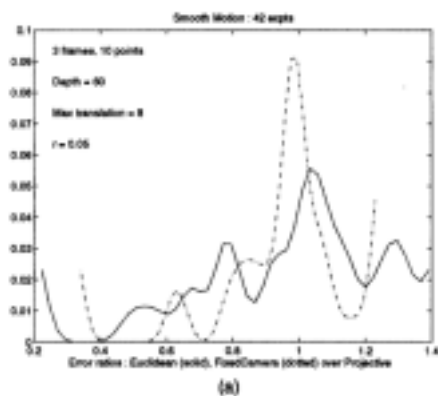


図 13 入力画像 5(ppm 画像 1192 pixel × 1040 pixel)

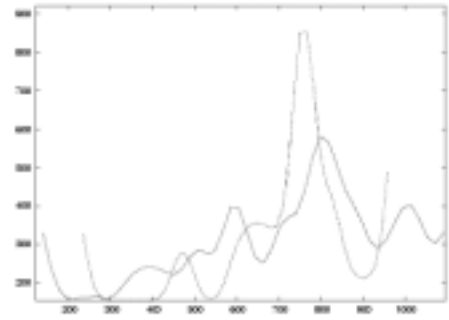


図 14 出力画像 5

4. 結論

本論文では，マーカの存在しない折れ線グラフ画像を認識対象とした簡略化した補間処理手法を用いた折れ線グラフ認識手法を提案した．提案手法により，従来手法よりほとんどの画像において処理時間を半分以下に抑えることができることが確認できた．ただ，追跡経路上の連結成分間の 3 次元ベクトルの特徴で，投票時の半径($\delta_1, \delta_2, \delta_3$)は，予備実験により一応の値を設定することはできるが，入力する画像によっては半径の大きさを変更させなくてはならないものがある．自動化した折れ線グラフ認識のためには，半径の自動取得の検討が今後の課題である．

文献

- [1] 横倉直子，渡邊豊英，“レイアウト構造知識を用いた棒グラフの認識，” 情処学論，vol.40,no.7,pp.2954-2966,July 1999.
- [2] M.H. Lee, N. Babaguchi, and T. Kitahachi, “Symbolization and presentation of graph images for intelligent communication of document images,” Proc. ACCV’95, vol.3, pp.680-684, 1995.
- [3] 布田寿康，大町真一郎，阿曾弘具 “連結成分追跡による文書画像中の折れ線グラフの認識，” 信学論 (D-)，vol.J86-D- ， no.6, pp.825-835, June 2003.
- [4] C.J. Hilditch, “Linear skeleton from square cupboards,” in Machine Intelligence, vol.6, pp.403-733, Edinburgh Univ. Press, 1969.
- [5] R.O. Duda and P.E. Hart, “Use of the Hough transformation to detect lines and curves in pictures,” Commun. ACM, vol.15, pp.11-15, 1972.
- [6] F. O’Gorman and M. Clowes, “Finding picture edges through collinearity of feature points,” IEEE Trans. Comput., vol.C-25, no.4, pp.449-456, 1976.