

原図形の視覚的特徴点を保存する 線分近似法

青山 宏, 河越正弘
(電子技術総合研究所)

線分近似を認識・表示の面からみると、人間のみたときの情報量を減らさないで、データ圧縮することが重要である。そこで本手法では、1)曲率の大きなところでは、角度誤差を優先し、逆に曲率の小さな所では、変位誤差を優先することによって、目で見えた情報を減らさないで、効率的な近似を行った。また、その時に、2)局所線分を考えることによって、corner や spike が保存されるようになった。更に、3)円と直線との滑らかな接続点(接点)が、人間の見た時に不自然にずれる問題に着目し、これを本来の位置に戻す機能を持たせた。

これにより、地図などの自然図形から、記号などの人工図形まで、統一的に適応して近似できるようになった。

A Piecewise Linear Approximation Method
Preserving Visual Feature Points of Original Figures

Hiroshi AOYAMA, Masahiro KAWAGOE

Information Systems Section,
Computer Science Division,
Electrotechnical Laboratory

1-1-4 Umezono, Sakura-mura, Tsukuba Science City, 305 JAPAN

We present a piecewise linear approximation method of planar curves, which can compress the amount of data, without loss of perceptual information. The method has following three merits:

- 1) it makes an efficient approximation by estimating error of both angles and displacement.
- 2) it preserves corners and spikes by local line segments.
- 3) it preserves smooth connecting points between arcs and lines.

The method can be applied to both natural figures and artificial figures, without unexpected transforms.

1. Introduction

近年、画像や図形のパターンデータを計算機で扱う局面が増加してきた。特に、線図形データは、多くの場面で基本的データとして扱われている。

そのため線図形データをデータ圧縮する研究が盛んになされているが、中でも線分近似によるデータ圧縮法は、そのまま図形認識の入力としても利用できる有力な表現手法である。

線分近似法としては、数多くの手法が提案されているが、アルゴリズムの構造（逐次型、一括型）と近似基準（位置ずれ、形状ずれ）の2つの点によって特徴づけられる。

アルゴリズム構造は Fig.1-1 のように、逐次型と一括型に分類できる。逐次型は効率がよいが、近似して行くときの方向によって結果が異なり、改良がされてはいるが [1] コーナーの移動やスパイクの鈍りが起きる。一括型は逆に変形は少ないが、計算効率が悪い [2]。

また近似するときの評価基準には、Fig.1-2 に示すように、位置ずれと形状ずれの2通りの手法があり、その得失は次の通りである。

1) 位置ずれを評価する手法： [3,4]

近似直線と原データの間の距離や面積を評価パラメータとする方法で、誤差の定量的評価が容易なため最適化などの考察が容易。

長所：誤差の定量的評価が容易。工作時の加工精度に直接対応可能。

短所：図形の大きさでパラメータを変える必要がある。形状の変形が生じる。真の直線も傾いてしまう。

2) 形状ずれを評価する手法： [5]

原データの曲率の大きい点で分割する手法や曲がり角検出により分割点を決める手法で、人間が認知に利用する特徴を利用しようとするものである。

長所：人間の着目点に分割点が一致。図の大きさや精度でパラメータを変える必要がない。図形の表示・認識と相性がよい。

短所：誤差の定量的評価が難しい。大きな所で位置ずれが大きくなりすぎる。

我々は、効率的で、見た目の情報を減らさない近似を実現するために、アルゴリズム構造は逐次型とし、かつ変形の少ない近似手法を考えることにした。

このために、位置ずれのパラメータと形状ずれのパラメータを両方とも誤差評価に利用し、細かい所は形状ズレを優先し、大雑把な所は位置ズレを優先させる方式とした。さらに、このとき距離の評価に工夫を加えることにより、スパイクの鈍りやコーナーの移動をかなり防ぐことができた（これを Basic Algorithm と

呼ぶ） [6]。

次に、Fig.1-3 に示すような、もとの図形の滑らかな接点の不自然な移動による意図しない変形を補正するための Extended Algorithm を考える [7]。

ここで、線分近似された図が不自然にゆがんで見える原因を考察すると、

1) 円であれば、その中心が接点で接線に対して、垂直線上にあるはず、という幾何学的関係の喪失と、
2) もとの図にあった平行関係や対称関係などの、大局的な関係の喪失、

の2種類がある。ここでは一番目だつ歪みである、直線と円弧の間の滑らかな接点の位置ずれによる幾何学的関係の喪失を対象として、その補正を行う。

以下、第2章で Basic Algorithm、第3章で Extended Algorithm について述べ、このアルゴリズムの実験結果を、第4章に示す。最後に結論と今後の課題について述べる。



Fig.1-1 アルゴリズムの構造による分類。

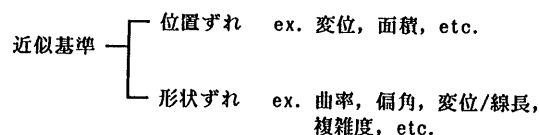


Fig.1-2 近似基準の種類。

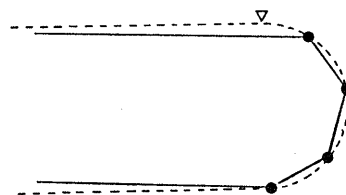


Fig.1-3 滑らかな接点のずれによる変形。

2. Basic Algorithm

2.1 Overview of Algorithm

Basic Algorithm の概要を Fig.2-1 に示す。本アルゴリズムでは、一旦局所線分列に変換することによりデータ圧縮すると共に、逐次的線分近似ではつきものコーナー・スパイクの変形も、距離の補正を併用することにより、避けることができる。

2.2 Algorithm

[Definitions]

- A_i: 格子点列
- B_i: 8方向線分とした時の折れ点
- C_i: 局所線分化した時の折れ点
- D_i: 直線近似した後の折れ点

[Stage 1]

< Procedure A >

- Input: {A_i} 格子点列
- Output: {B_j} 8方向線分とした時の折れ点

格子上の点から成る入力点列 {A_i} を順に8方向線分列として結んだ時の、折れ点および始点終点を {B_j} とする (Fig.2-2中の○印)。

< Procedure B >

- Input: {B_j} 8方向線分とした時の折れ点
- Output: {C_k} 局所線分化した時の折れ点

上のステップで得られる格子上の折れ点列から、直線がデジタル化された場合に出現するパターンを見つけ {B_j} を局所線分列として書き直し、その局所線分列の折れ点列 {C_k} をもとめる。

これによりデータ量の圧縮を行うと共に、stage 2での近似線分あてはめを有効にする。

線分列が以下のルールを満たす場合、直線上に乗っていると見なし局所線分に置き換える (Fig.2-3参照)。これは、Chord Property [9] を簡略化したものである。

- rule1: 2方向は隣接している
- rule2: 片方向の長さはすべて1
- rule3: 対方向の長さは、高々2種類で、隣合っていて、連続するのは一方のみ。
- rule4: rule3の連続の個数は、高々2種類で隣合っていて、続くのは一方のみ。
- rule5: rule4の連続の個数は1種類。
- rule6: 始まりと、終わりは、長さが短くてもよい

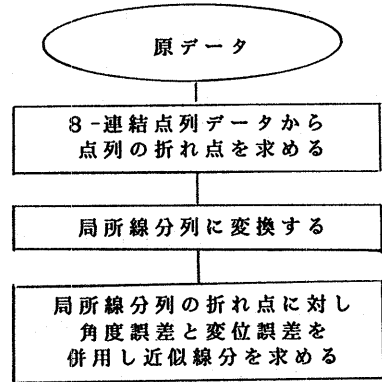


Fig.2-1 Basic Algorithm の概略図。

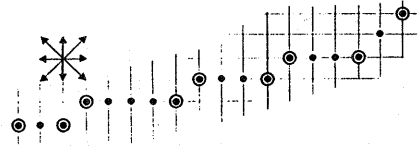


Fig.2-2 格子上点列の折れ点

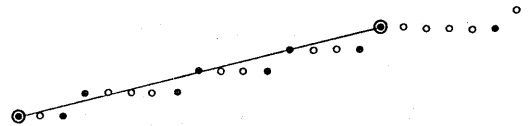


Fig.2-3 局所線分とみなせる点列パターンの例。

[Stage 2]

- Input: {C_k} 局所線分化した時の折れ点
- Output: {D_m} 直線近似した後の折れ点

stage 1の結果に対して近似直線を次頁の Fig.2-4 のように当てはめる。

C_i を始点とする直線 C_iC_j の長さを L_{ij}, その間の全点 $\forall C_k, k \in [i, j]$ と直線との最大偏差を d_{ij} とした時、疑似曲率 $g_{ij} (= d_{ij}/L_{ij})$ と偏位 d 両者を用いて、最大近似直線 C_iC_m を次のように求める。

$\forall j \in [i, m]$ に対して
 $(g_{ij} \leq T1) \wedge (d_{ij} \leq T2)$; 但し T1, T2 は閾値が成立し、かつ $j = m+1$ に対しては上式が成立しない $\exists m$ を求める。

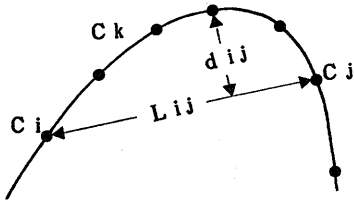


Fig. 2-4 近似直線の当てはめ.

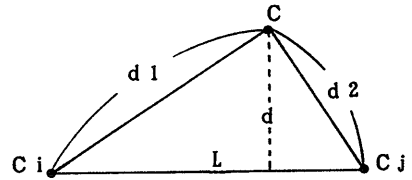


Fig. 2-5 コーナー強調のための加重法.

2.3 コーナー・スパイクを保存する修正 Algorithm

通常、逐次的線分近似では、コーナーがズレたり、スパイクが潰れたりすることがあるので、実際の d, g の計算では、このコーナーやスパイクを保存するために、 d, g に以下の2つの修正を行う。

【修正1】

長い直線を尊重するために、Fig. 2-5 の $d1, d2$ の比で下の式のように、 g を加重・修正する。これは、 d/L の比を $(d/L)^{1/2}$ によって評価するのとはほぼ等しくなる。

原データに対してこれを適用すると線分近似の閾値が近似線分が長くなるに従って大幅に下がることに等しくなるが、本アルゴリズムでは、一旦局所線分化してあるので、無闇に細かくされることが避けられる。

$$g = (d/L) \cdot \text{Max}(d1/d2, d2/d1)^{1/2}$$

【修正2】

Fig. 2-8 のように、点 C から直線 $C_i C_j$ におろした垂線が、 $C_i C_j$ と交わらない場合、下式のように最大変位 d を端点からの距離とすることにより、折れ返りの場合に距離が小さく評価されるのを防ぐ。なお同様の手法は [10] にも提案されている。

$$d = \text{Min}(d1, d2)$$

3. Extended Algorithm

3.1 Overview of Algorithm

ここでは、直線と円の間の滑らかな接続点の保存の問題を考える。Fig. 3-1 にアルゴリズムの概要を示す。

逐次的線分近似では、Fig. 3-2 に示すように、近似線分が直線部分から円部分に入っても線分の垂直方向の変位は、中心角に対して2次の微小量であるが、本手法では、円の中心が利用することにより、1次の微小量である線分の長さ方向の変位に置き換えることにより滑らかな接続点の検出を行う。

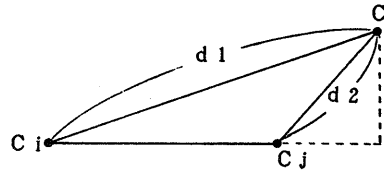


Fig. 2-6 折り返り点の距離の計算法.

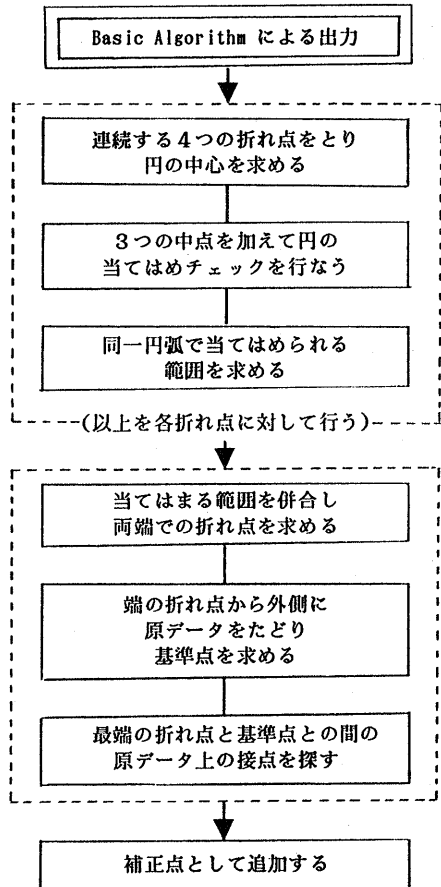


Fig. 3-1 Extended Algorithm の概略図.

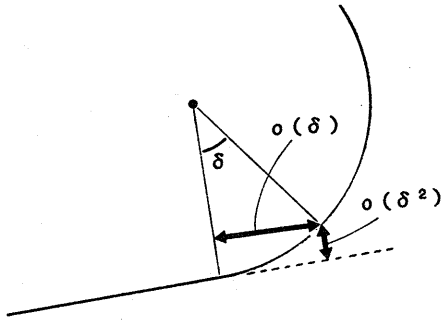


Fig. 3-2 原理図.

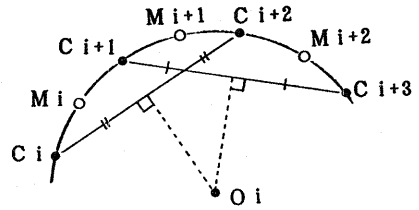


Fig. 3-3 円の検出.

3.2 円の検出とあてはめ

線分近似で得られる折れ点 $C_i, (i=1, 2 \dots n)$, 折れ点 C_i, C_j 間の原データ上での中点を M_i とする。(Fig. 3-3 参照)

- 1) 連続する4つの点 $C_i, C_{i+1}, C_{i+2}, C_{i+3}$ および3つの中点 M_i, M_{i+1}, M_{i+2} に対して, これらが1つの円周上にのるか否かを,

$$\text{Min}(R_u) / \text{Max}(R_u) < T_3 \quad ; \text{但し}$$

O_i : 点 C_i, C_{i+3} が直線 $C_{i+1}C_{i+2}$ に対し互いに同じ側にある時の, 直線 C_iC_{i+2} と直線 $C_{i+1}C_{i+3}$ 両者の垂直二等分線の交点, つまり4点から求めた円の中心,

R_u : 中心 O_i から, 対象としている C 点 M 点までの各距離,

T_3 : 閾値,

によりチェックする。

- 2) 乗る場合は 円周上に乗っているか否かのチェックを, さらに隣合う C 点 M 点へ広げて行い, 1つの円周でカバーできる最大範囲を求め, 中心及び, $\text{Min}(R_u), \text{Max}(R_u)$ を求める。

- 3) つぎに, 各 i のカバー範囲が3点以上重複するものを併合し, Fig. 3-4 に示すようにカバー範囲の両端の折れ点を求め, さらに両端における円の中心と半径をそれぞれ求める。

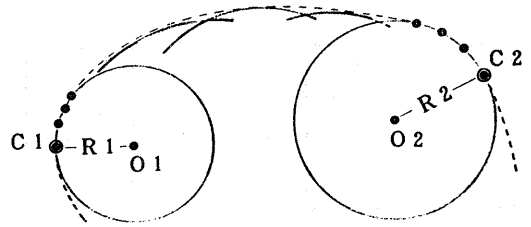


Fig. 3-4 円弧の併合.

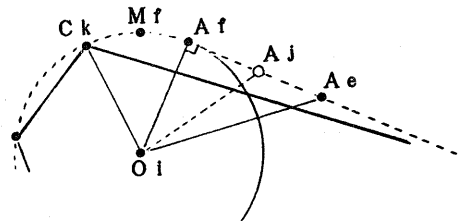


Fig. 3-5 滑らかな接続点の検出.

3.3 滑らかな接続点の位置補正

円周上の最後の折れ点を C_k , その時の円の中心を O_i , O_i から C_k の距離を R とする。

C_k の外側の原データ上の点を A_j 中心 O_i からの距離を R_j とした時, $R_j / R \geq 1 + 2 \cdot T_3$ となる C_j に一番近い原データ上の点 A_e を求めて基準点とする (Fig. 3-5参照)。

この時,

$$\exists A_f \in [C_k, A_e], O A_f \perp A_e A_f,$$

$$\text{Min}(R_{\min}, R_f, R_{M_f}) / \text{Max}(R_{\max}, R_f, R_{M_f}) < T_3$$

(M_f : $C_k A_f$ 間の原データ上での中点)

が成り立てば, 点 A_f を新たな折れ点として追加・補正を行う。

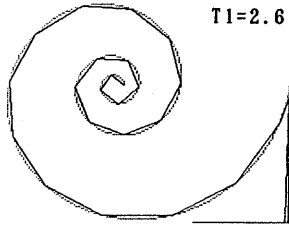


Fig.4-1 変位のみによる線分近似.

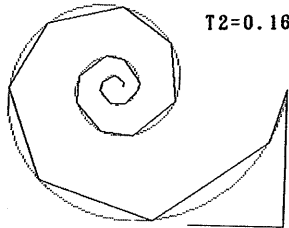


Fig.4-2 変位と線分長の比による近似.

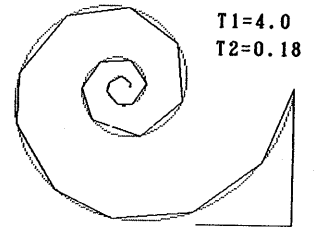


Fig.4-3 変位と比による近似.

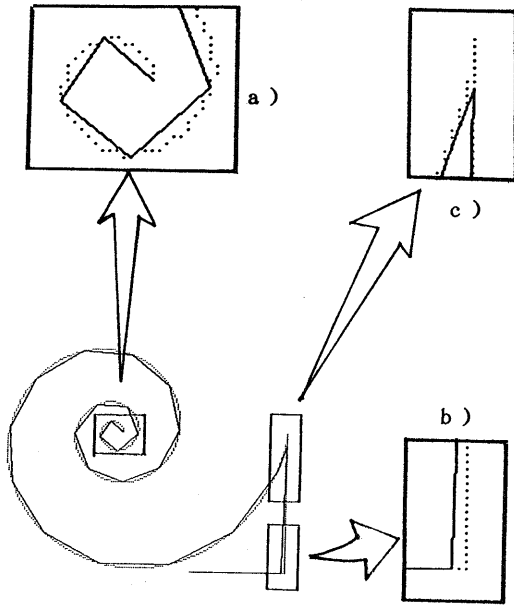


Fig.4-4 距離のみによる近似の問題点.

- a) 曲率の大きな部分が粗く近似される
- b) スパイクが鈍る
- c) コーナーがずれる

4. Experiments and Results

【実験1】 Basic Algorithm の従来手法との比較

Figs.4-1~4-3 は、曲率が大きく変化する螺旋とコーナー・スパイクを含んだテストパターンを同一線分で近似した例である。

Fig.4-1 は、近似線分からの最大偏差一定とした従来手法を適用した例で、曲率の小さな部分で適切に近似しようとすれば螺旋中央部のような曲率の大きな部分が見た目に粗く近似されてしまい (Fig.4-4a)、逆に曲率の大きな部分に合わせると、全体として必要以上に細かく近似されてしまう。またコーナーやスパイクの部分では、ずれ込みや鈍り (Figs.4-4b,4-4c) が生じている。

Fig.4-2 は、Basic Algorithm で近似線分長と最大偏差との比のみを用いて近似した例である。

Fig.4-4a~4-4c で指摘された Fig.4-1 での問題点は解決されている。しかし最大偏差自身に制限を加えていないために曲率の小さな部分では原パターンと近似線分とが離れすぎ不自然な近似となる。

Fig.4-3 は、更に最大偏差に制限を加えた Basic Algorithm による近似例で、上の欠点が解決され、見た目にも自然な結果が得られた。

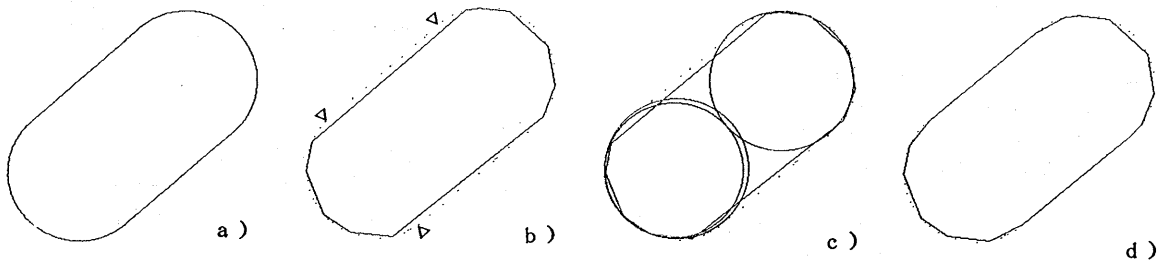


Fig.4-5 Extended Algorithm の適用例 (1) .

a) 原データ, b) Basic Algorithm の出力, 3) 円の当てはめ, 4) Extended Algorithm の出力

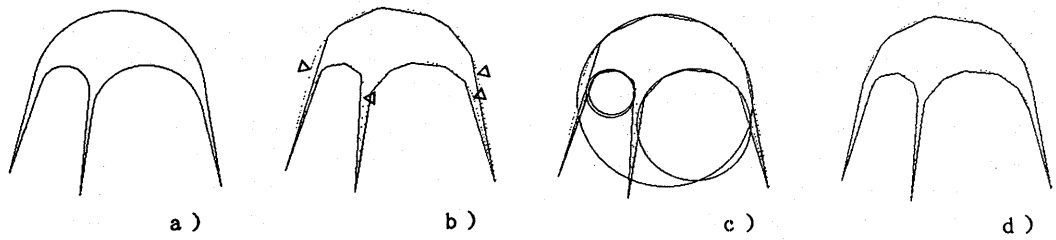


Fig.4-6 Extended Algorithm の適用例 (2).

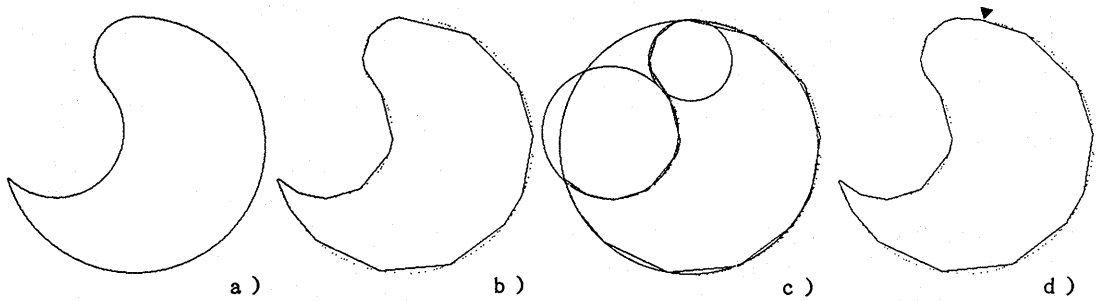


Fig.4-7 Extended Algorithm の適用例 (3).

【実験2】 Extended Algorithm の適用

Fig.4-5 に示す図のような場合 Basic Algorithm の適用後、直線部分が円部分に食い込んで (Fig.4-5b中の矢印部分)、平行線が傾き、見た目の印象が大きく変化してしまう。これに対して、Extended Algorithm を適用することにより補正されている (Fig.4-5d)。

Fig.4-6 も同様の例で、滑らかな接続点が失われたために針の部分の太さが大きく変化してしまった部分が、Extended Algorithm の適用によりもとに戻っている。

Fig.4-7 は、異なる曲率の円弧同士が滑らかにつながっている図形例である。Fig.4-7d の▽印の位置に新たな折れ点が設けられ、より良い近似になっている。

【実験3】 人工図形と自然図形の混在図

人工図形と自然図形が混在する図に本手法を適用した例を Fig.4-8 に示す。このような場合に本近似手法は大変適していると言える。

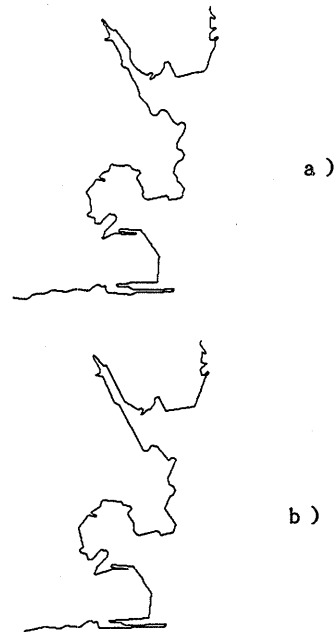


Fig.4-8 埋立地等の人工的な地形を含む地図データへの適用例。

a)原データ, b)近似結果

5. Conclusion and Future Works

本方式は、複雑さに応じた近似を行なうため、大きな部分と細かな部分とが共存する地図などの自然図形を効率良く近似できるだけでなく、格子点表現からの局所線分の抽出処理により、折れ線表現を含む人工図形にも正しく対応できるものである。

さらに、本手法はスパイクやコーナーを保存するだけでなく、滑らかな接続点の位置ずれについても、補正を行っているので、線図形の視覚的特徴点を保存する近似手法と言えよう。

しかし本手法では、近似図形が不自然に見える要因の内、幾何学的関係に起因するゆがみの一部を解決したにすぎないので、今後は対象関係などの大局的關係に起因する歪みの補正に対しても対処できるようにする予定である。

[謝辞]

最後に、本研究にあたり、有益な御助言・御討論を頂いた棟上ソフトウェア部長、藤村室長をはじめとする情報システム研究室の各位に感謝の意を表する。

[References]

1. J.Roberge:"A Data Reduction Algorithm for Planar Curves", CVGIP.,vol.29,pp.168-195 (1985).
2. Y.Kurozumi and W.A.Davis:"Polygonal Approximation by the Minimax Method", CGIP., vol.19,pp.248-264 (1972).
3. C.M.Williams:"An Efficient Algorithm for Piecewise Linear Approximation of Planar Curves",CGIP.,vol.8,pp.286-293 (1978).
4. K.Wall and P.E.Danielsson:"A Fast Sequential Method for Polygonal Approximation of Digitized Curves", CVGIP.,vol.28,pp.220-227 (1984).
5. M.A.Fischler and R.C.Bolles:"Perceptual Organization and Curve Partitioning",IEEE Trans.,vol.PAMI-8,no.1,pp.100-105 (1986).
6. 青山宏,河越正弘,佐藤孝紀:"曲率に大きな幅を持つ線図形の直線近似", 情報処理学会第32回全国大会 7N-8 (1986).
7. 青山宏,河越正弘:"滑らかな接合点を考慮した平面図形の近似法", 情報処理学会第34回全国大会 2D-3 (1987).
8. T.Pavlidis and S.Horowitz:"Segmentation of Planar Curves", IEEE Trans.,vol.C-23,pp.860-870 (1974).
9. A.Rosenfeld:"Digital Straight Line Segments" IEEE Trans.,vol.c-23,pp.1264-1269 (1974).
10. J.G.Dunham:"Optimum Uniform Piecewise Linear Approximation of Planar Curves",IEEE Trans., vol.PAMI-8,pp.67-75 (1986).