

# 道路地図の高速スプライン近似表示アルゴリズム

山路アンゼリカ, 鶴岡信治, 木村文隆, 三宅康二  
三重大学工学部電子工学科

本稿では、道路地図を表示するための再帰スプライン近似表示について述べる。道路、河川、海岸線などの不規則な形状を表現するためのベクトルデータは一部の地域を拡大して表示する時には曲線補間し、広い地域を表示する時には、再サンプリングして折線近似して表示すると必要な表示精度を保ちながら表示速度を向上させることができる。そのために、再帰的な折線近似法と二次スプライン曲線の幾何学的生成手法を単一のアルゴリズムに統合し、ベクトルデータを表示する時の表示尺度に応じて適応的に表示精度（ディスプレイ画面に対する相対誤差）をほぼ一定に制御できるようにする。アルゴリズムの有効性を東海地方の道路地図に対する表示実験の結果によって示す。

## HIGH-SPEED SPLINE APPROXIMATION ALGORITHM FOR DISPLAYING A ROAD MAP

Angelica YAMAJI, Shinji TSURUOKA, Fumitaka KIMURA, Yasuji MIYAKE  
Faculty of Engineering, Mie University  
Kamihama 1515, Tsu, 514 JAPAN

This paper describes an recursive spline approximation algorithm for displaying a road map. To satisfy the both requirements for representation accuracy and speed, vector data representing the shapes of roads, rivers and coast lines should be interpolated by curves in the case of magnified representation, while they should be resampled and approximated by polygonal lines in the case of reduced representation.

We integrate a recursive polygonal approximation and a geometric quadratic spline generation into an algorithm, which controls the representation accuracy to be nearly constant adaptively to the representation scale of the vector data. The effectiveness of the algorithm is shown by the result of experiments for road map data of Tokai district.

## 1. まえがき

道路、河川、海岸線などの不規則な形状を表現するためのベクトルデータは一部の地域を拡大して表示する時には曲線補間し、広い地域を表示する時には、再サンプリングして折線近似表示すると必要な表示精度を保ちながら表示速度を向上させることができる。筆者らは前報<sup>(1)</sup>、<sup>(2)</sup>で、再帰折線近似法を提案しその効果を東海地方の道路地図を対象とする実験により検討した。本稿では更に、拡大時にベクトルデータをスプライン近似し曲線状の道路を滑らかに表示する機能を持った再帰スプライン近似法を提案し有効性を実験により評価する。

再帰折線近似は、分割統治法(divide and conquer)に基づく近似手法で、画像や図形を折線近似してベクトルデータを得たり<sup>(3)</sup> コンピュータグラフィクスの分野で海岸線や岩肌を表示する曲線や曲面を再帰的に定義し、生成するために<sup>(4)</sup> 従来から使われている。また、本稿で用いる二次スプライン曲線の幾何学的生成手法は、Chaikin<sup>(5)</sup>の手法に基づくものである。本研究の目的は、これらの手法を単一のアルゴリズムに統合し、ベクトルデータを表示する時の表示尺度に応じて適応的に表示精度(ディスプレイ画面に対する相対誤差)をほぼ一定に保つことである。

## 2. システムの基本機能

地図の必要な部分を任意の縮尺で表示するために必要なシステムの基本機能として拡大、縮小、平行移動等が考えられる。ここでは、道路地図の高速表示に関係する拡大と縮小について述べる。

### (1) 拡大

マウスを使ってウィンドウパラメータを入力し、その部分をビューポートに拡大表示する(図1)。

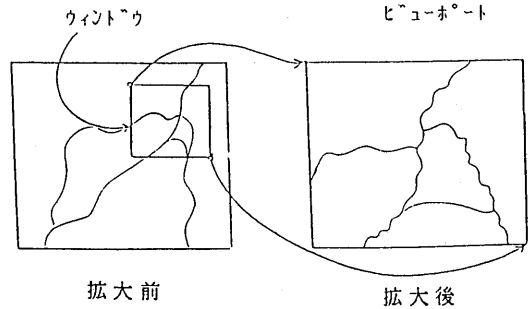


図1 拡大

### (2) 縮小

マウスを使って仮想ビューポートのパラメータを入力し、現在表示されている部分を仮想ビューポート内に縮小表示する。この時、仮想ビューポート外にも対応する地図を表示する(図2)。

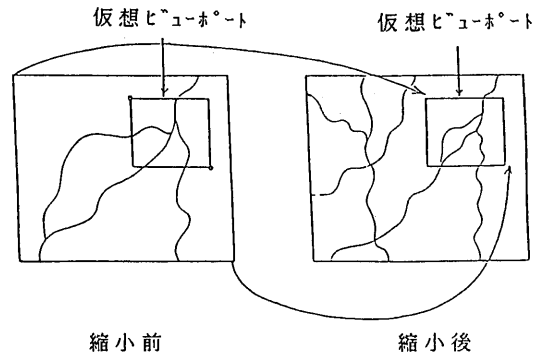


図2 縮小

## 3. 高速表示アルゴリズム

道路地図を表示するためには、膨大なベクトルデータが必要になる<sup>(6)</sup>。例えば本研究で用いた地図の主要地方道以上の道路を表示するために必要なデータは約46,000ベクトルであり、実用的な表示のためには更に10倍以上のデータが必要であろう。このように多量のデータを高速表示する場合次ぎの問題が生じる。

### (1) ウィンドウが大きい場合

ウィンドウが大きく東海地方全体を含むような場合描画するベクトル数が大きく時間がかか

る。

(2) ウィンドウが小さい場合

ウィンドウ内に含まれるベクトル数は小さく、描画に必要な時間は短い、それらのベクトルを全ベクトルデータの中から選択するのに時間がかかる。

(3) 折線近似の影響

滑らかな曲線状の道路セグメントを折線で近似するため、拡大に伴ってディスプレイ上での相対誤差が大きくなり、実際の道路形状から著しく異なる表示結果となる(図3)。

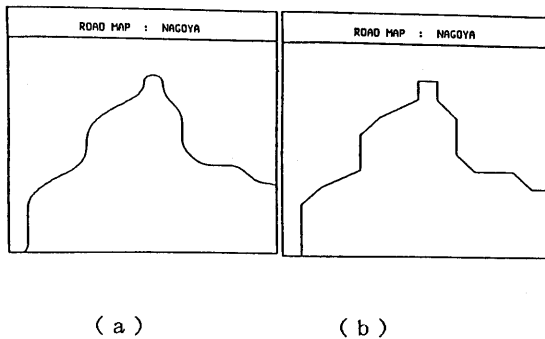


図3 実際の道路形状 (a), 折線近似表示 (b)

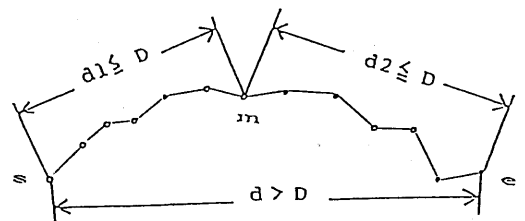
以下に、問題(1)に対処するための再帰折線近似法と問題(2)に対処するためのボクシングについて述べ、さらに(1)-(3)の問題に対処するために、再帰折線近似と再帰ボクシングおよびスプライン近似を併用する再帰スプライン近似法について述べる。

(1) 再帰折線近似法

表示用のベクトルデータは、ある特定のウィンドウサイズで表示した時に道路形状を必要かつ十分な精度で近似できるようにサンプリングされている。したがって、ウィンドウがより大きく、広い地域を表示する場合には、相対的にサンプリング間隔が狭くなり、極端な場合はディスプレイ装置の解像度以下になるなど、表示に必要以上の時間がかかる。これを避けるため

には、ウィンドウの大きさに対する、相対的な近似精度をディスプレイ装置の解像度等から決まる一定値に近い値に保つように近似表示すればよい。

再帰折線近似は道路セグメント単位で行う。道路セグメントは、1つの交差点から他の交差点に至る道路形状を表すサンプリング点の座標の系列のことである。道路セグメントの両端の距離がある定数  $D$  より小さければ、その道路セグメントの両端を結ぶ線分をディスプレイに表示する。両端の距離が  $D$  より大きければ、道路セグメントを2つに分割して、その各々を再帰折線近似する(図4)。分割して得られた座標系列も道路セグメントとして扱う。



- s : 始点番号
- e : 終点番号
- m : 中点番号  $(m = \frac{s+e}{2})$

図4 再帰折線近似

定数  $D$  の値は次式で決める。

$$D = \text{ウィンドウ幅} \times a \quad (1)$$

実験では、 $a = 0.01 \sim 0.05$ とした。また、道路セグメントの両端(s,e)の距離としては、シティブロック距離

$$d(s,e) = |x_e - x_s| + |y_e - y_s| \quad (2)$$

を用いた。

(2) ボクシング

問題(2)に対処するために、道路セグメント単位のボクシングを、再帰折線近似ルーチンの先頭で実行する。道路セグメントの両端を結ぶ線分を対角線とする座標軸に平行な4辺を持つ長方形とウィンドウが共通部分を持たなければ即座に再帰折線近似ルーチンから復帰 (return) し、その道路セグメントは再帰折線近似表示しない、共通部分があれば再帰折線近似のための処理を続ける (図5)。

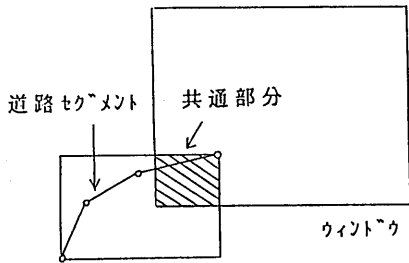


図5 再帰ボクシング

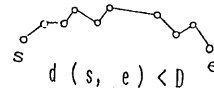
(3) 再帰スプライン近似アルゴリズム

(a) 始点  $s$  と終点  $e$  の距離がある定数  $D$  以下ならば、 $\widehat{s}e$  を直線で近似する (図6 (a))。

(b) 始点  $s$  と終点  $e$  の距離がある定数  $D$  以上で、 $\widehat{s}e$  の構成線分数が 2 なら  $\widehat{s}e$  を後述するアルゴリズムによりスプライン近似する (図6 (b))。

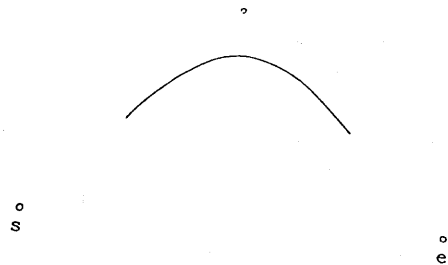
(c) 始点  $s$  と終点  $e$  の距離がある定数  $D$  以上で、 $\widehat{s}e$  の構成線分数が 3 以上ならば、 $m_1 = m_2 = (s + e) / 2$  とし、 $\widehat{s}m_1$  と  $m_2e$  を再帰スプライン近似し (図6 (c))、それらを次に述べる方法で接続する。

再帰スプライン近似ルーチンの先頭においても再帰折線近似と同様にボクシングを行う。

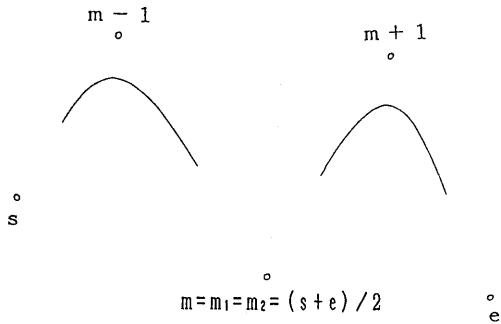


$$D = 0.01 \times \text{ウィンドウ幅}$$

(a)



(b)



(c)

図6 再帰スプライン近似

(4) 接続処理

(a)  $m_1, m_2$  がスプラインの端点ならば、 $m-1$   $m+1$  をスプライン近似する (図7 (a))。

(b)  $m_1$  が直線の端点で、 $m_2$  がスプラインの端点ならば、 $m$  と  $(m$  と  $m+1$  の中点) を直線で近似する (図7 (b))。

(c)  $m_1$  がスプラインの端点で、 $m_2$  が直

線の端点ならば、(m-1 と m の中点)と m を直線で近似する(図7(c))。

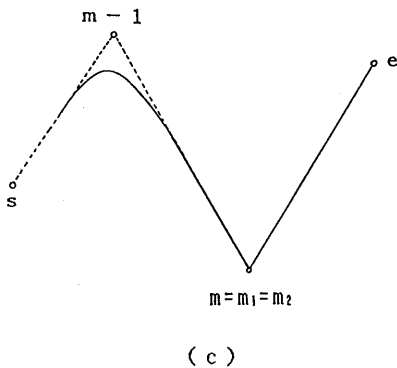
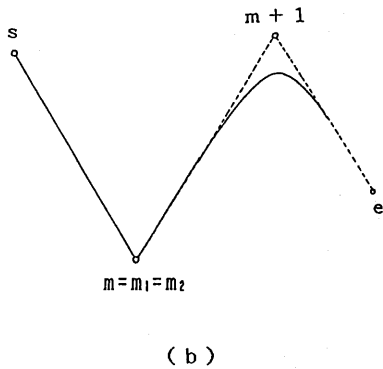
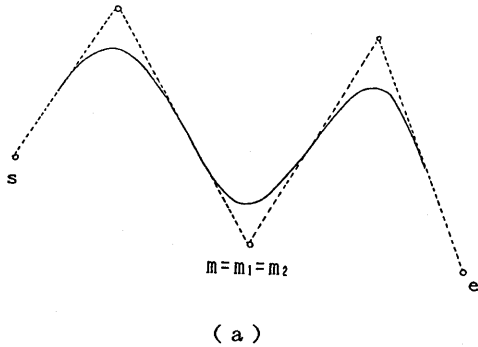


図7 接続処理

### (5) スプライン近似

構成線分数が2(頂点数が3)のセグメントを高速に二次スプライン近似する, Chaikin<sup>(5)</sup>の手法に基づくアルゴリズムを以下に示す。

連続する3個の頂点の位置ベクトルをX1, X2, X3とする。

(a)  $|X3-X1| < D$  ならX1(始点)とX3(終点)を直線近似する。さもなければ, X1', X2', X3'およびX2', X3', X4'を連続する3個の頂点の位置ベクトルとするセグメントの各々にこのアルゴリズムを再帰的に適用する。但し,

$$X1' = (3X1 + X2) / 4$$

$$X2' = (X1 + 3X2) / 4$$

$$X3' = (3X2 + X3) / 4$$

$$X4' = (X2 + 3X3) / 4$$

(3)

である(図8)。

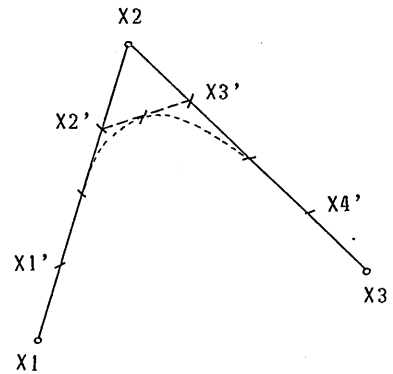


図8 Chaikinのアルゴリズムに基づく二次スプライン近似

### 4. データ構造

実験に用いたデータは名古屋, 豊橋, 静岡を含む東海地方の1/20万道路地図をデジタイザを使ってトレースし, 計算機に入力したものである。各道路セグメントは, 道路種別, 始点座標, 中間点座標列, 終点座標の各項目からなる可変長レコードで表されている。

1道路セグメント分のデータの記憶の様子を図9に示す。

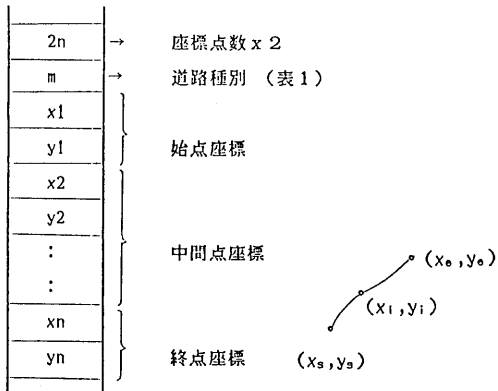


図9 主記憶上の道路セグメントデータ

### 5. 処理手順

処理手順を図10に示す。

ハードディスクに格納された地図データを主記憶上の配列 VECT に入れる。パラメータをマウスで入力し、ウィンドウパラメータを計算し、ウィンドウの中にあるセグメントを表示アルゴリズムに従って表示する。

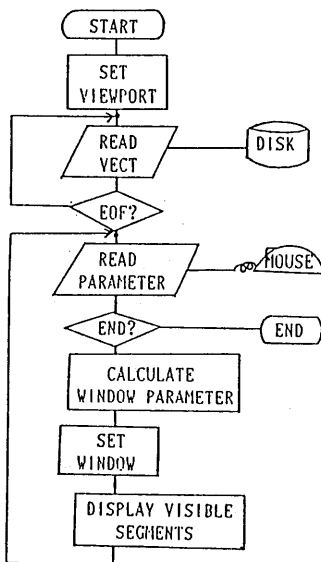


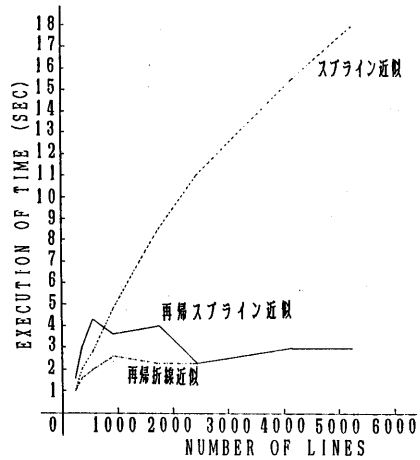
図10 処理手順

### 6. 実験結果と表示例

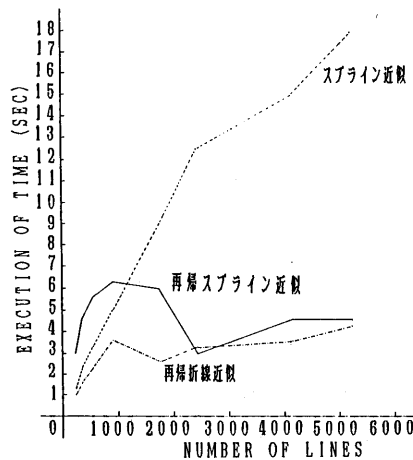
再帰スプライン近似アルゴリズムによる表示の効果を調べるために、再帰折線近似、Chaikinのスプライン近似および再帰スプライン近似を用いる場合の各々について表示時間とウィンドウ内ベクトル数の関係を調べる実験を行った。

使用計算機は、News NWS830で、4Mバイトの主記憶と156Mバイトのハードディスクを持ち、ビットマップディスプレイ及びマウスが接続されている。使用言語は、Cである。

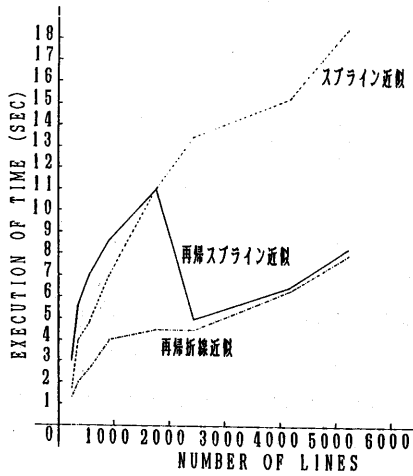
結果を図11(a) - (c)に示す。Dはそれぞれウィンドウ幅の0.05, 0.02, 0.01倍である。



(a) D = ウィンドウ幅 × 0.05



(b) D = ウィンドウ幅 × 0.02



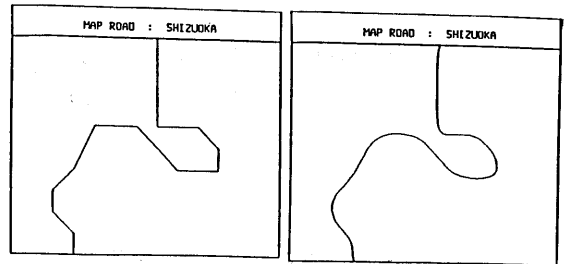
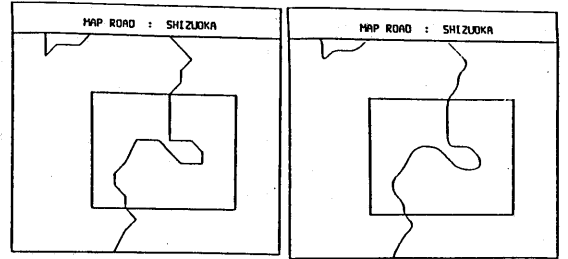
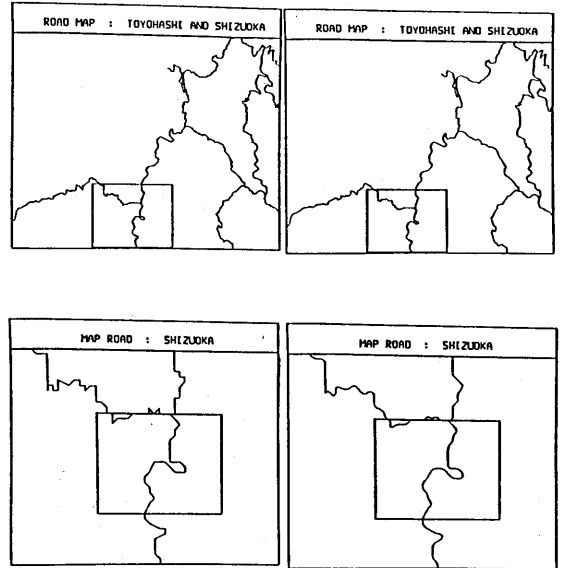
(c)  $D = \text{ウィンドウ幅} \times 0.01$

図 1.1 表示時間とウィンドウ内ベクトル数の関係

この結果から再帰スプライン近似により拡大の時には、スプライン近似し、縮小の時には、折線近似されていることがわかる。さらに、処理時間はウィンドウ内ベクトル数に無関係に一定に抑えられることがわかる。

図 1.2 は、静岡を拡大して行ったところである。拡大に伴って道路形状がより詳しく表示されていくことが分かる。図 1.2 (a) は、再帰折線近似を適用した表示例で図 1.2 (b) は、再帰スプライン近似を適用した表示例である。(a) では折線近似表示しているので拡大すると道路の形が詳しく表示され、直線で近似されていることが目だつようになる。(b) ではスプライン近似を併用しているのので拡大後も道路が滑らかに表示されている。図 1.3 は、同様に豊橋を拡大して行ったところである。

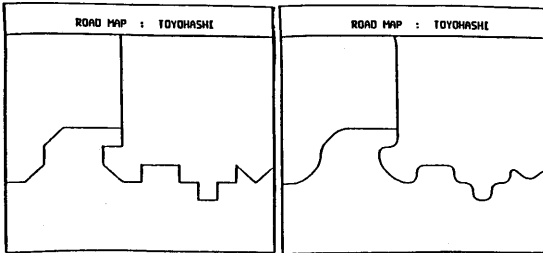
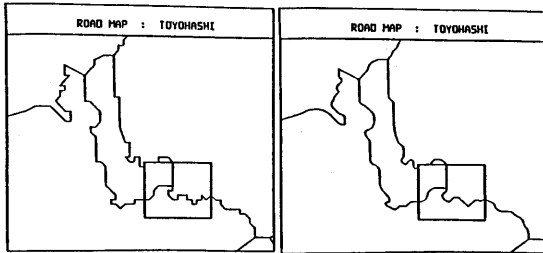
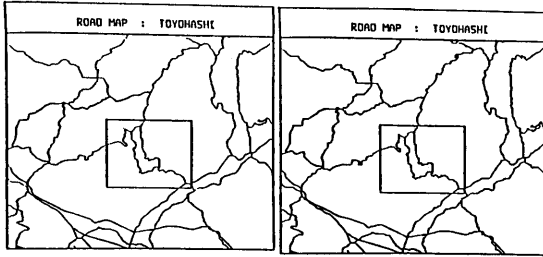
河川、海岸線などは表示されていないが同様に表示することができる。



(a)

(b)

図 1.2 再帰折線近似の適用例(a)と再帰スプライン近似アルゴリズムの適用例(b)



(a)

(b)

図13 再帰折線近似の適用例(a)と再帰スプライン近似アルゴリズムの適用例(b)

## 7・むすび

地図の必要な部分を任意の縮尺で表示できる道路地図表示システムを試作し、表示の高速化の方法について検討した。

その結果、折線近似とスプライン近似を併用する再帰スプライン近似法が道路地図の表示に適していることがわかった。この方法の特徴を以下に述べる。

(1) 縮小時には高速化、拡大時には、スプライン近似により道路の滑らかさを保つことができる。

(2) アルゴリズムが単純なため、ソフトウェアによる実現の他、道路、河川、海岸線などの不規則な形状を表示するための専用ハードウェア化に適している。

(3) 特殊なデータ構造を必要とせずデータ量も少ない。

(4) 地図データベース中の一道路セグメントのベクトル数(サンプリング点数)を増加させてもある程度以上大きなウィンドウでは表示速度には影響しない。

課題としては、

- (1) 連続性、表示速度について更に検討する、
- (2) スプライン近似時にサンプリング点を通るアルゴリズムの開発、などが残されており今後検討する予定である。

## 参考文献

- (1)山路, 木村, 鶴岡, 三宅: "道路地図の高速表示アルゴリズムについて", 信学技報 PRL85-80,1986.
- (2)山路, 木村, 鶴岡, 三宅: "道路地図の高速表示アルゴリズムについて(2)", 昭和62年信学情報システム部門全大107.
- (3) R.O.Duda and P.E.Hart: "Pattern classification and Scene Analysis", Willey,pp.338,(1973).
- (4)A.Fournier,D.Fussel,L.Carpenter: "Computer Rendering of Stochastic Models",comm.ACM,25,6,pp.371-384(June 1982).
- (5)Chaikin George Merrill: "An Algorithm for High-Speed Curve Generation",Computer Graphics and Image Processing(1974) 3, pp. 346-349.
- (6)安居院, 鈴木, 中嶋: "道路網データに関する一考察", 昭和60年度信学総全大 1678.