

ALVのための道路の3次元形状復元

小山千鶴子, 渡辺一成, 金谷健一

群馬大学工学部情報工学科

自動走行車 (ALV = Autonomous Land Vehicle) のための、道路画像から道路の3次元形状を復元するアルゴリズムを示す。まず、理想化した道路が満たすべき道路モデルを設定する。次に、画像上に観察された道路がその道路モデルの条件を満たしていると仮定して、道路の空間中での接線ベクトルを表す連立微分方程式を導く。これらの微分方程式を数値的に解けば道路の3次元形状が求まるが、誤差の蓄積によって計算が不安定となる可能性がある。そこで、それを防ぐために数値積分の各ステップで解を補正して計算を安定化する手法を提案する。また、数値積分を行うための初期値を求めるための緩和法のアルゴリズムを示す。最後に実際の画像を用いて復元を行った結果を示す。

RECONSTRUCTION OF 3D ROAD GEOMETRY FOR ALV

Chizuko Koyama, Kazunari Watanabe and Ken-ichi Kanatani

Department of Computer Science

Gunma University, Kiryu, Gunma 376, Japan

A new algorithm for reconstructing 3D road geometry from images is presented for the purpose of autonomously navigating land vehicles. The reconstruction is based on a road model idealizing real roads: A road is assumed to be generated by a horizontal line segment of a fixed length sweeping in the scene. Then, the "constraints" that ideal road images must satisfy are expressed as a set of "differential equations". The 3D road geometry is reconstructed by numerically integrating these differential equations. In order to prevent "numerical instability", a "correction scheme" is proposed to stabilize computation: At each numerical integration step, the computed solution is modified in such a way that the required constraint is always satisfied. We also present a "relaxation scheme" for computing the initial data for the numerical integration. Examples based on real road images are also given.

1. 序論

近年, ALV (Autonomous Land Vehicle: 自動走行車) に関する研究が米国において盛んである. 特に DARPA (Defense Advanced Research Projects Agency) が支援し, Maryland 大学, Carnegie-Mellon 大学などの大学と Martin-Marietta 社などの企業とが提携した Strategic Computing プロジェクトはわが国でも注目を集めている [4]. このプロジェクトの目標は, 車が前方の景色をカメラで捕らえ, 道路を認識し, その3次元形状を計算し, その進路を決定することである.

これを自動化するためにはいくつかの段階が必要である. 第一に入力画像の領域分割, および道路と障害物の認識, 道路の境界線の検出のための画像解析, 第二に3次元道路形状の計算, 第三に最短距離, および最小コスト, 障害物回避等を考慮した車の進路の計算, 最後に, 計算通りに走行させるための車両制御が必要である. [1, 8~10]

本論文では, 3次元道路形状の計算に焦点をおく. 直接的な手段としてステレオや距離センサー等があるが [3], 本研究では道路画像の解析と道路形状の拘束条件を組み合わせることで, 単一の画像から道路の3次元形状を復元することを試みる.

従来から提案されている方法は, 道路の両側の境界線上に, 互に対応する平行な線分の対を発見し, その「消失点」を計算することに基づいている [8~10]. そのような方法は, 道路が水平面上にあるか, または(必ずしも水平でない)直線状であれば計算は簡単なものになる. しかし, 一般に, そのような局所的に平行な対を見つけることは容易ではない. Sakurai ら [6] や 潮田ら [7] は道路形状のプロトタイプをあらかじめ数種類用意しておいて, パラメトリックなあてはめを試みている.

Dementhon [2] は純粋に2次元の情報からの3次元道路形状の復元を目指し, Ozawa and Rosenfeld [5] の道路モデルに基づいた離散的な方法を示した. そのアルゴリズムは, 出発点の3次元的位置を既知と仮定し, 道路の3次元的位置を反復的に計算するもので, 道路形状の離散的近似に基づく方法である. しかし, 各ステップで三次方程式を解いて得られる三つの解の中から正しいものを選ばなければならないので, 解が極めて不安定である(いったん誤った解を選ぶと後のステップはすべて無意味になってしまう恐れがある).

本論文では DeMenthon [2] の「離散的」アプローチに対して「連続的」アプローチを示す. まず, 3次元道路形状とその投影画像とを関係づける「連立微分方

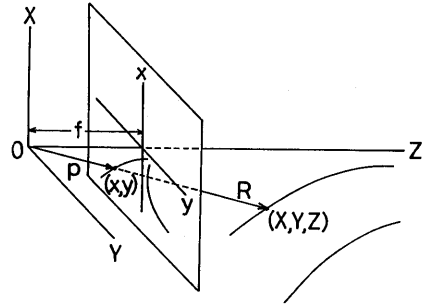


図1 道路の透視変換

程式」を導き, 与えられた初期値のもとでそれを「数値積分」することにより3次元道路形状を復元する. こうすると解は一意的に定まる. しかし, 数値積分の過程で蓄積する誤差のため, 解がしだいに不安定になっていくことは避けられない. そこで本論文では, 数値積分の各ステップごとに解を修正して, 計算を安定化する手法を提案する. 最後に, 実際の道路画像に基いた例を示す.

2. 道路モデル

理想的な道路は次のような条件を満たすものとする. 道路のセンターラインは任意の形状をとる滑らかな空間曲線とする. 長さ一定で, しかもセンターラインと中点で垂直に交わり, かつ水平に保たれている線分を考える. 以後, このような線分を「生成セグメント」と呼ぶ. 道路は生成セグメントがセンターライン上を滑るとき, その両端の描く軌跡により得られるものとする [2, 5].

3. カメラモデル

車に取り付けられたカメラを基準とするXYZ座標系を考える. レンズの中心を座標原点とし, カメラの光軸をZ軸とする. レンズの中心と撮像面との距離をfとおくと, 投影の関係は, 座標原点を視点とし, 平面Z=fを画像面とする透視変換として表せる(図1). したがって, 空間中の点(X,Y,Z)が画像面上の点(x,y)に投影されるとすれば, 次の関係が成り立つ.

$$x = f \frac{X}{Z}, \quad y = f \frac{Y}{Z} \quad (3.1)$$

4. 道路エッジの透視変換

画像上で検出した2本の道路エッジはスプライン補間により, 2つのなめらかな平面曲線として得られているとする(付録A参照). それらを $c_1(x_1(l), y_1(l))$, $c_2(x_2(r), y_2(r))$ とする. ここで, l と r は, それぞれの曲線ごとに適当に定めたパラメータである. また, これらの道路像に対応する2本の道路エッジを空間曲線 $C_1(X_1(l),$

$Y_1(l), Z_1(l), C_r(X_r(r), Y_r(r), Z_r(r))$ で表す。すると、式 (3.1) より次の関係が得られる。(l は左側, r は右側の意味である。両方を同時に示すときは $s = (l, r)$ を用いる。)

$$X_s = \frac{x_s Z_s}{f}, \quad Y_s = \frac{y_s Z_s}{f}, \quad s=l, r \quad (4.1)$$

x_s, y_s は画像上で得られるデータであるから、残りの $Z_s, s=l, r$, さえ定まれば道路の 3 次元形状は復元できる。

微分をドットにより表し、その変数に含まれるパラメータによる微分を表すことにする。たとえば、 \dot{x}_l, \dot{Z}_r は $dx_l/dl, dZ_r/dr$ を意味する。式 (4.1) を微分して、

$$\dot{X}_s = \frac{1}{f}(Z_s \dot{x}_s + x_s \dot{Z}_s), \quad \dot{Y}_s = \frac{1}{f}(Z_s \dot{y}_s + y_s \dot{Z}_s), \quad s=l, r \quad (4.2)$$

が得られる。シーン中の道路エッジ上の 1 点を $R_s = (X_s, Y_s, Z_s)$, $s=l, r$, で表すと、その点での道路の接線ベクトルは、 $\hat{R}_s = (\dot{X}_s, \dot{Y}_s, \dot{Z}_s)$, $s=l, r$, で表される。式 (4.2) で $x_s, y_s, \dot{x}_s, \dot{y}_s$ はスプライン補間によって既知であるから、 \dot{Z}_s さえ定めれば接線ベクトル \hat{R}_s が定まる。

5. 道路の接線ベクトルの方向

2本の空間曲線 C_l, C_r が、道路モデルの条件を満たしているとする、これらは常に一定の距離を保つ。言い換えると、2つの曲線 C_l, C_r の間にはある 1対1対応 (すなわちパラメータ l と r の間に 1対1対応) が存在し、対応する 2点は一定の距離を保ち、かつそれらをつなぐ道路セグメントは 2曲線 C_l, C_r に直交する (図 2)。

今、点 $R_s, s=l, r$, まで 3 次元形状がすでに復元されているものとして、接線ベクトル $\hat{R}_s, s=l, r$, を決定することを考える。前節で示したように、 \dot{Z}_s さえ決定できればよい。そのために、次のような 2つの制約条件について考える。

- (1) 接線ベクトル $\hat{R}_s, s=l, r$, は道路セグメントベクトル $R_l - R_r$ に直交しなければならない。
- (2) 接線ベクトル $\hat{R}_s, s=l, r$, は道路面の法線に直交しなければならない。

「単位道路セグメントベクトル」 T を次のように定義する。

$$T \equiv \frac{R_l - R_r}{\|R_l - R_r\|} = \frac{(X_l - X_r, Y_l - Y_r, Z_l - Z_r)}{\sqrt{(X_l - X_r)^2 + (Y_l - Y_r)^2 + (Z_l - Z_r)^2}} \quad (5.1)$$

条件 (1) は $(\hat{R}_s, T) = 0, s=l, r$, と表せるので、 $T = (T_1, T_2, T_3)$ とおくと、次のように表せる。

$$T_1 \dot{X}_s + T_2 \dot{Y}_s + T_3 \dot{Z}_s = 0, \quad s=l, r \quad (5.2)$$

また、条件 (2) は $S = (S_1, S_2, S_3)$ を道路面の単位法線ベクトルとすると、 $(\hat{R}_s, S) = 0, s=l, r$, と表せる。ここで、現在の位置までの道路の 3 次元形状はすでに復元されていると仮定しているので、現在の単位法線ベクトル S はわかっている。したがって、次のように表せる。

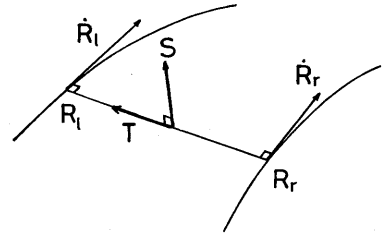


図 2 道路の接線ベクトルの方向

$$S_1 \dot{X}_s + S_2 \dot{Y}_s + S_3 \dot{Z}_s = 0, \quad s=l, r \quad (5.3)$$

このように、接線ベクトル $\hat{R}_s, s=l, r$, は単位ベクトル T と S の両方に直交しなければならない。しかし、 \dot{Z}_s を決定するためにはただ 1つ条件があれば十分なので、これら 2つの条件を同時に満足させることは一般にはできない。これらのうち任意の 1つを選んで採用してもよいが、データに含まれる誤差によって計算が不安定になる恐れがある。式 (4.2) を式 (5.2), 式 (5.3) に代入すると、それぞれ次のようになる。

$$(T_1 x_s + T_2 y_s + T_3 f) \dot{Z}_s = -(T_1 \dot{x}_s + T_2 \dot{y}_s) Z_s, \quad s=l, r \quad (5.4)$$

$$(S_1 x_s + S_2 y_s + S_3 f) \dot{Z}_s = -(S_1 \dot{x}_s + S_2 \dot{y}_s) Z_s, \quad s=l, r \quad (5.5)$$

ベクトル $p_s = (x_s, y_s, f)$ を考えると、これは原点 O と画像面 $Z=f$ 上の点 (x_s, y_s) を結ぶベクトルであり、シーン中の道路エッジ上の点 R_s の 3 次元的方向を表している (図 1)。式 (5.4) の左辺は $(T, p_s) \dot{Z}_s$ となるので、 T と p が直交する点で \dot{Z}_s を決定できないし、 $(T, p_s) \approx 0$ でも計算が不安定となる。同様に、式 (5.5) の左辺は $(S, p_s) \dot{Z}_s$ となるので、 S と p が直交する点で \dot{Z}_s を決定できないし、 $(S, p_s) \approx 0$ でも計算が不安定となる。

そこで、式 (5.4), 式 (5.5) を最小二乗の意味で平均的に満足するように $\dot{Z}_s, s=l, r$, を決定する。すなわち、次式を最小にするように \dot{Z}_s を求める。

$$[(T_1 x_s + T_2 y_s + T_3 f) \dot{Z}_s + (T_1 \dot{x}_s + T_2 \dot{y}_s) Z_s]^2 + [(S_1 x_s + S_2 y_s + S_3 f) \dot{Z}_s + (S_1 \dot{x}_s + S_2 \dot{y}_s) Z_s]^2 \quad (5.6)$$

この結果、 \dot{Z}_s は次のように与えられる。

$$\dot{Z}_s = Z_s \frac{(T_1 x_s + T_2 y_s + T_3 f)(T_1 \dot{x}_s + T_2 \dot{y}_s) + (S_1 x_s + S_2 y_s + S_3 f)(S_1 \dot{x}_s + S_2 \dot{y}_s)}{(T_1 x_s + T_2 y_s + T_3 f)^2 + (S_1 x_s + S_2 y_s + S_3 f)^2}, \quad s=l, r \quad (5.7)$$

したがって、式 (4.2), 式 (5.7) により接線ベクトル $\hat{R}_s, s=l, r$, を決定することができる。

6. 道路の接線ベクトルの大きさ

前節の方法により、2本の接線ベクトルの方向は左右独立して定まる。しかし、これでは左右の接線ベクトルに沿ってどれだけ進んでよいかわからない。それらが進んだ先が新しい対応点となるので、パラメータ l と

rの間の1対1対応を与える何らかの条件を与えれば、その条件を満たす点まで道路は接線ベクトルに沿って進むことになる。lとrとの間の制約条件を与える方法としてはいろいろ考えられる。たとえば、対応する2点を結ぶ道路セグメントが常に水平であるという条件で、2つのパラメータを関係づけることができる。しかし、その結果得られる微分関係は、道路の一部が完全に水平であると不安定になる。

さまざまな条件を試した結果、次のような近似条件が最も適当と思われる。それは、パラメータlとrがそれぞれ微小量dl, drだけ増加したとき、「道路エッジ上の点R_l, R_rは同じ距離だけ進む」というようにlとrを関係づけることである。式で表すと次のようになる。

$$\|\hat{R}_l\| dl = \|\hat{R}_r\| dr \quad (6.1)$$

これは、次のように表される。

$$\frac{dr}{dl} = \frac{\|\hat{R}_l\|}{\|\hat{R}_r\|} \quad (6.2)$$

この関係式を用いれば、パラメータlとrの間の対応をたどっていくことが可能となる。

7. 道路の連立微分方程式

X_l, X_r, Y_l, Y_r, Z_l, Z_r はパラメータlとrが与えられれば、道路エッジの像からスプライン補間によって計算できる。X_l, Y_l, X_r, Y_r は式(4.1)よりZ_l, Z_rによって表される。したがって、式(6.2)を用いれば、独立変数として例えばパラメータlを用いることにより、次のような形の連立微分方程式を得る。

$$\frac{dr}{dl} = F_0(l, r, Z_l, Z_r), \quad \frac{dZ_l}{dl} = F_1(l, r, Z_l, Z_r), \quad \frac{dZ_r}{dl} = F_2(l, r, Z_l, Z_r) \quad (7.1)$$

3番目の式は、(dZ_r/dr)・(dr/dl)より得られる。これらはパラメータlを独立変数とした未知数rとZ_lとZ_rに関する連立微分方程式である。したがって、道路の3次元形状はある初期位置l=l₀での既知の初期値Z_l=Z_l(l₀), Z_r=Z_r(r₀), r=r₀, からパラメータlに沿ってこれらを数値積分することによって得られる。

8. 解の補正による計算の安定化

式(6.1)で与えた条件は、近似的にのみ成立する。これは道路が左右にカーブしていないときは正確であるが、道路が左右にカーブしていればカーブの外側ではカーブの内側より多く進まなければならない。また、微分方程式を数値的に解かなければならないので、誤差の蓄積は避けられない。以上のような理由から、数値積分の各ステップで、要求される条件を満たすように解を補正することを考える。

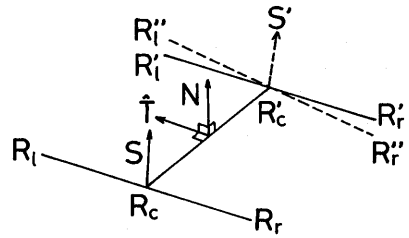


図3 計算解の補正

前のステップですでにその3次元的位置が決定している(または、初期値として与えられている)道路エッジ上の対応する2点をR_l, R_r(出発点)とする。また、道路セグメントベクトルR_l-R_rは水平であるとする。R_l', R_r'を数値積分の1ステップを適用して計算した位置とする(予測点)。まず、これらの中点を計算する。

$$R_c = \frac{1}{2}(R_l + R_r), \quad R_c' = \frac{1}{2}(R_l' + R_r') \quad (8.1)$$

次に、計算した位置R_l', R_r'を次の4つの条件を満たすように新しい位置R_l'', R_r''(補正点)に補正する(図3)。

(1) R_l''とR_r''の中点はR_c':

$$\frac{1}{2}(R_l'' + R_r'') = R_c' \quad (8.2)$$

(2) 道路セグメントベクトルR_l''-R_r''は水平:

$$(R_l'' - R_r'') \cdot N = 0 \quad (8.3)$$

ただし、Nは鉛直方向の単位ベクトルである。

(3) 道路セグメントベクトルR_l''-R_r''は、道路セグメントベクトルR_l-R_rと同じ長さをもつ。

$$\|R_l - R_r\| = \|R_l'' - R_r''\| \quad (8.4)$$

(4) 道路セグメントベクトルの平均

$$((R_l - R_r) + (R_l'' - R_r'')) / 2$$

はセンターラインベクトルR_c'-R_cと直交する。

$$\frac{((R_l - R_r) + (R_l'' - R_r'')) \cdot (R_c' - R_c)}{2} = 0 \quad (8.5)$$

ここで、次のようなベクトルを定義する。

$$\hat{T} = \frac{N \times (R_c' - R_c)}{\|N \times (R_c' - R_c)\|} \quad (8.6)$$

これは、NとR_c'-R_cの両方に直交する単位ベクトルである。したがって、単位ベクトルTは常に水平で、かつセンターラインに直交する。この単位ベクトルTを用いて補正点R_l''とR_r''を次のように定義する。このとき、上の(1)~(4)の条件がすべて満たされることは容易に確認できる。

$$\begin{aligned} R_l'' &= R_c' - (R_l - R_r) + 2(R_l - R_c) \hat{T} \\ R_r'' &= R_c' + (R_l - R_r) - 2(R_l - R_c) \hat{T} \end{aligned} \quad (8.7)$$

9. 道路面の垂線の計算

先に示したように、数値積分を行なうには道路面の垂線Sが必要である。厳格に言えば、垂線Sは位置により変わる。例えば、道路セグメントの一方の端点R_l

での垂線と他方の端点 R_T での垂線とは異なる。しかし、ここでは近似的に考えて、中点 R_C の位置での垂線 S を用いることにする。予測点 R'_1, R'_T はこの垂線 S を用いて決定するが、その次のステップで用いる新しい垂線 S' は次式のものを用いる (図 3)。

$$S' = \frac{(R_C' - R_C) \times \hat{T}}{\| (R_C' - R_C) \times \hat{T} \|} \quad (9.1)$$

10. パラメータの補正

予測点 R'_s を新しい位置 R''_s に補正したら、対応するパラメータもまた補正しなくてはならない。出発点 R_s 、予測点 R'_s に対応する像をそれぞれ (x_s, y_s) 、 (x'_s, y'_s) とする。また、補正点 R''_s に対応する像を (x''_s, y''_s) とする。点 (x''_s, y''_s) は道路エッジの像 c_s 上にとるには限らない。そこで、曲線 c_s 上で点 (x''_s, y''_s) から最も近い点に割り当てることで、パラメータの補正を行なう (図 4)。まず、次のような画像面上のベクトルを考える。 $\Delta r_s \equiv (x'_s - x_s, y'_s - y_s)$ 、 $\Delta r'_s \equiv (x''_s - x'_s, y''_s - y'_s)$ (10.1) ベクトル $\Delta r'_s$ をベクトル Δr_s 上に投影したときの長さは次式で与えられる。

$$\Delta s = \|\Delta r'_s\| \cos \alpha = \frac{(\Delta r'_s, \Delta r_s)}{\|\Delta r_s\|} \quad (10.2)$$

ただし、 α は Δr_s と $\Delta r'_s$ のなす角である。また、

$$\Delta s' = \|\Delta r_s\| - \Delta s \quad (10.3)$$

とおく。点 (x_s, y_s) 、 (x'_s, y'_s) でのパラメータ値をそれぞれ s と s' とすると、点 (x''_s, y''_s) でのパラメータ値 s'' を次のように割り当てればよい。

$$s'' = \frac{s' \Delta s + s \Delta s'}{\Delta s + \Delta s'}, \quad s=1, r \quad (10.4)$$

Δs と $\Delta s'$ のどちらか一方または共に 0 であっても負であってもよい。

以上で、道路形状の 3 次元復元のためのアルゴリズムは構成できた。このアルゴリズムは、与えられた初期値から出発して数値積分の各ステップで補正を行いながら次のステップへ進むことを繰り返す。

11. 実験結果

群馬大学付近の実際の道路を写真に撮り、それを復元した結果を示す。この実験では、画像処理による道路エッジの検出のステップは省き、道路エッジ上のサンプル点を適当に選んで、付録 A に示すスプライン補間を施した。復元アルゴリズムのための初期値は、付録 B に示す緩和法アルゴリズムで推定した。そして、Runge-Kutta 法により独立変数 l に沿って式 (7.1) を数値積分することで、道路の 3 次元形状を復元した。

Runge-Kutta 法の積分ステップは、任意に定めることができるが、ここではセンターライン R_C に沿って Z 軸方向にほぼ一定値 ΔZ_C だけ進むように積分ステップ

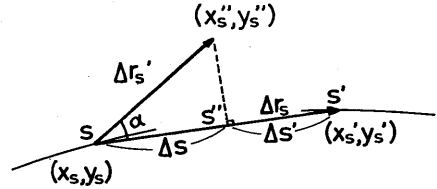


図 4 パラメータの補正

Δl を定める。そのために、関係

$$\frac{dZ_C}{dl} = \frac{d((Z_1 + Z_2)/2)}{dl} = \frac{1}{2} [F_1(l, r, Z_1, Z_2) + F_2(l, r, Z_1, Z_2)]$$

より、積分ステップ Δl を次のようにとった。

$$\Delta l \approx \frac{dl}{dZ_C} \Delta Z_C = \frac{2 \Delta Z_C}{F_1(l, r, Z_1, Z_2) + F_2(l, r, Z_1, Z_2)}$$

例えば、 $\Delta Z_C = 0.5$ [m] とすれば、Runge-Kutta 法の各ステップで、センターライン R_C に沿って Z 軸方向に毎回約 0.5 [m] だけ復元されることになる。

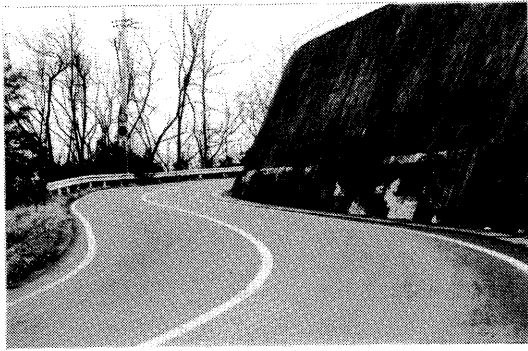
また、復元の出発点では道路面の垂線 S は仮に $S = 0$ として出発する。このことは式 (5.2) の条件だけを用いて出発することを表す。次のステップからは、式 (9.1) によって S を定める。

道路画像 (a)、そのスプライン補間図 (b)、復元図でそれぞれ復元結果を上からみた図、横からみた図 (c) を示す (図 5, 6)。

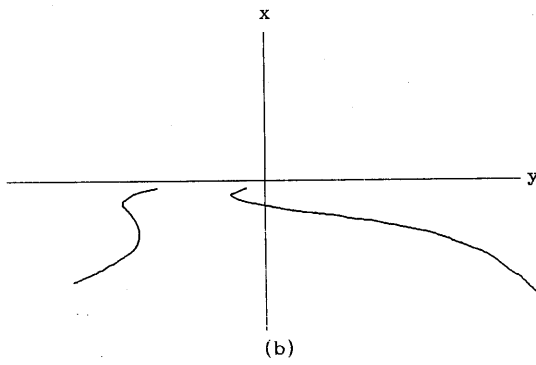
12. 結論

単一画像から、道路形状の 3 次元復元を行うアルゴリズムが構成できた。このアルゴリズムを用いれば、道路の 3 次元形状が一意に定まる。また、道路モデルの条件を必ずしも満足しているとは限らない実際の道路についても、このアルゴリズムが適用できた。得られた結果については、復元が遠くへ行けば行くほど復元の精度は落ちてくる。これはカメラから遠くにある道路像になるほど、道路エッジの読み取り誤差などの問題が急速に増大し、また、数値積分の誤差が蓄積することなどを考えると当然のことであろう。特に、実際の道路写真ではカメラの角度、高さ、焦点距離等の測定誤差がかなり含まれており、あまりよいデータとは言えない。しかしながら、その復元結果については実際の値がわからないので正確なことは言えないが、おおよその復元はできたものと思われる。なお、このアルゴリズムは道路面の変化がなるべく滑らかになるように条件を与えているので、道路面の小さな変化は平均化して復元する傾向がある。

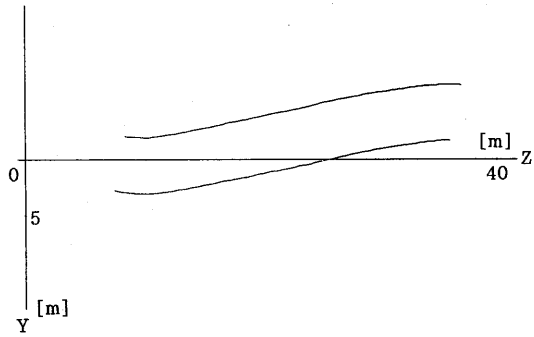
謝辞。本研究に関する有益な討論を頂いた米国メリランド大学の Azriel Rosenfeld 教授、Larry Davis 教授、Daniel DeMenthon 博士、慶応義塾大学の小沢慎治



(a)

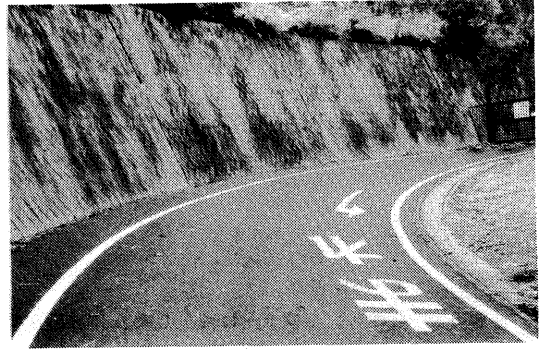


(b)

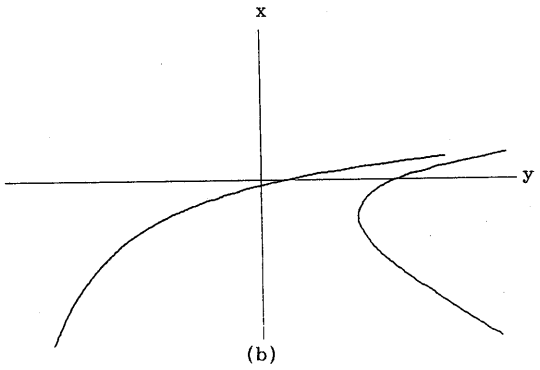


(c)

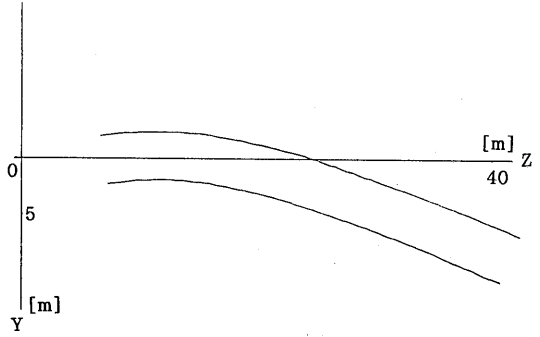
図5 3次元道路形状復元の例1



(a)

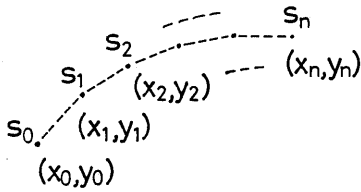


(b)

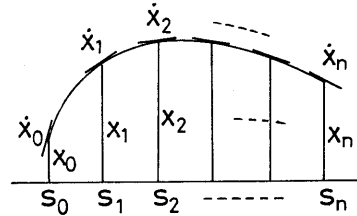


(c)

図6 3次元道路形状復元の例2



(a)



(b)

図A スプライン補間

教授に感謝する。本研究の一部はカシオ科学振興財団，矢崎科学技術振興記念財団，稲盛財団，文部省科学研究費一般研究(C)(No.63550268)の助成による。

参考文献

- [1] L. S. Davis, T. R. Kushner, J. LeMoigne and A. M. Waxman, Road boundary detection for autonomous vehicle navigation, Optical Engineering, 25(3) (1986), pp. 409 - 414.
- [2] D. DeMenthon, A zero-bank algorithm for inverse perspective of a road from a single image, Proc. IEEE Int. Conf. Robotics and Automation, April 1987, Raleigh, NC, pp. 1444 - 1449.
- [3] M. Hebert and T. Kanade, 3-D vision for an autonomous vehicle, Proc. Int. Workshop Indust. Appl. Machine Vision Machine Intell., Tokyo, February 1987, pp. 375 - 380.
- [4] 金出武雄, 米国における自律移動ロボット研究の動向, 日本ロボット学会誌, 5-5 (1987), pp. 44 - 51.
- [5] S. Ozawa and A. Rosenfeld, Synthesis of a road image as seen from a vehicle, Pattern Recognition, 19 (1986), pp. 123 - 145.
- [6] K. Sakurai, et al., Analysis of a road image as seen from a vehicle, Proc. IEEE 1st Int. Conf. Comput. Vision, London, June, 1987, pp. 651 - 656.
- [7] 瀬田・太田・小沢, 車両から見た道路像の解析, 電子通信学会技術研究報告 PRU86-4 (1986).
- [8] C. Thorpe, et al., Vision and navigation for the Carnegie-Mellon Navlab, IEEE Trans. Pattern Anal. Machine Intell., PAMI-10(3) (1988), pp. 362 - 373.
- [9] M. A. Turk, et al., VITS - A vision system for autonomous land vehicle navigation, IEEE Trans. Pattern Anal. Machine Intell., PAMI-10(3) (1988), pp. 342 - 361.

- [10] A. M. Waxman, et al., A visual navigation system for autonomous land vehicles, IEEE J. Robotics Automation, RA-3(2) (1987), pp. 124 - 141.

付録A 道路エッジのスプライン補間

本論文の数値積分に基づく方法では，道路エッジを滑らかな曲線として扱わなければならない。しかし，画像処理によって得られる道路エッジは不完全である。そこで，次のようなスプライン補間を行う。(コンピュータグラフィックスの分野でよく用いられる「スプライン」とは少々異なっている。)

画像面上の道路エッジ上の点列 (x_k, y_k) , $k=1, \dots, N$, に対してパラメータ s を

$$s_k = \sum_{i=1}^k \sqrt{((x_i - x_{i-1})^2 + (y_i - y_{i-1})^2)}, \quad k=1, \dots, N \quad (A.1)$$

によって定義する。ただし，点 (x_0, y_0) は $s_0=0$ とする。このパラメータ s_k は点 (x_0, y_0) から点 (x_k, y_k) までの点列を結ぶ折れ線に沿う長さを表している(図A(a))。

ここでは， $x(s_k)=x_k$, $y(s_k)=y_k$, $k=0, \dots, N$ を満たす3次曲線 $(x(s), y(s))$ を求める。 y_k の補間は x_k と同様であるので， x_k だけを考える。各点， $s=s_k$, $k=0, \dots, N$, における導関数 dx/ds の値 \dot{x}_k が推定されたとすれば，各区間 (s_{k-1}, s_k) , $k=1, \dots, N$ で3次式 $x(s)$ を，次の条件を満たすように定める(図A(b))。

$$\begin{aligned} x(s_{k-1}) &= x_{k-1}, \quad x(s_k) = x_k, \\ \frac{dx}{ds}(x_{k-1}) &= \dot{x}_{k-1}, \quad \frac{dx}{ds}(x_k) = \dot{x}_k \end{aligned} \quad (A.2)$$

$\Delta s_k = s_k - s_{k-1}$ とおけば，3次式 $x(s)$ は「Hermiteの補間公式」により次のように与えられる。

$$\begin{aligned} x(s) &= \left(\frac{s-s_{k-1}}{\Delta s_k}\right)^2 \left[3x_{k-1} + \dot{x}_{k-1} \Delta s_k - \frac{2x_k + \dot{x}_k \Delta s_k}{\Delta s_k} (x-x_{k-1})\right] \\ &+ \left(\frac{s-s_k}{\Delta s_k}\right)^2 \left[3x_k + \dot{x}_k \Delta s_k - \frac{2x_{k-1} + \dot{x}_{k-1} \Delta s_k}{\Delta s_k} (x-x_k)\right] \end{aligned} \quad (A.3)$$

これらの多項式を結べば， $x(s)$ は連続であり，次に示すような連続な一次導関数 dx/ds を持つ。

$$\begin{aligned} \frac{dx(s)}{ds} &= \frac{2(s-s_k)}{\Delta s_k^2} [3x_{k-1} + \dot{x}_{k-1} \Delta s_k + \frac{2x_k + \dot{x}_k \Delta s_k}{\Delta s_k} (x-x_k)] \\ &- \frac{2(s-s_{k-1})}{\Delta s_k^2} [3x_k + \dot{x}_k \Delta s_k - \frac{2x_{k-1} + \dot{x}_{k-1} \Delta s_k}{\Delta s_k} (x-x_{k-1})] \end{aligned}$$

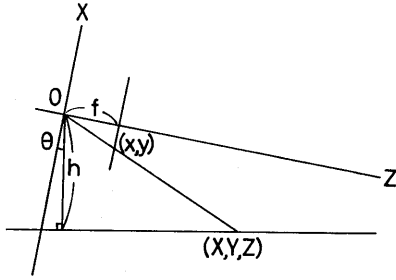


図 B1 道路が水平な場合の投視関係

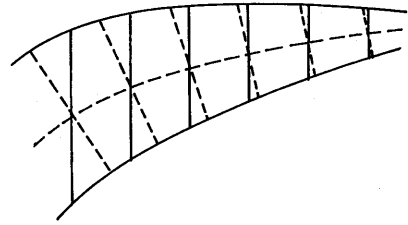


図 B2 生成セグメントを求める緩和法

$$\begin{aligned}
 & + \frac{2(s-s_{k-1})}{\Delta s_k^2} [3x_k + \dot{x}_k \Delta s_k + \frac{2x_k - \dot{x}_k \Delta s_k}{\Delta s_k} (x - x_{k-1})] \\
 & + \frac{2x_{k-1} + \dot{x}_{k-1} \Delta s_k}{\Delta s_k} \left(\frac{s-s_k}{\Delta s_k} \right)^2 - \frac{2x_k - \dot{x}_k \Delta s_k}{\Delta s_k} \left(\frac{s-s_{k-1}}{\Delta s_k} \right)^2 \quad (A.4)
 \end{aligned}$$

残る問題は、各点 $s=s_k, k=0, \dots, N$ における微分係数 dx/ds の値 \dot{x}_k を推定することである。ここでは、 \dot{x}_k は、以下の条件を満たす 2 次式 $x(s)$ の $s=s_k$ における dx/ds によって近似する。

$$x(s_{k-1})=x_{k-1}, \quad x(s_k)=x_k, \quad x(s_{k+1})=x_{k+1} \quad (A.5)$$

そのような 2 次式は、「Lagrange の補間公式」により

$$\begin{aligned}
 x(s) = & \frac{(s-s_k)(s-s_{k+1})}{(s_{k-1}-s_k)(s_{k-1}-s_{k+1})} x_{k-1} + \frac{(s-s_{k-1})(s-s_{k+1})}{(s_k-s_{k-1})(s_k-s_{k+1})} x_k \\
 & + \frac{(s-s_{k-1})(s-s_k)}{(s_{k+1}-s_{k-1})(s_{k+1}-s_k)} x_{k+1} \quad (A.6)
 \end{aligned}$$

と与えられる。したがって、 \dot{x}_k は

$$\dot{x}_k = \frac{1}{\Delta s_k + \Delta s_{k+1}} \left[\frac{\Delta s_k}{\Delta s_{k+1}} (x_{k+1} - x_k) + \frac{\Delta s_{k+1}}{\Delta s_k} (x_k - x_{k-1}) \right] \quad (A.7)$$

となる。ただし、 \dot{x}_0, \dot{x}_N に関しては、以下の条件を満たす 2 次式を用いる。

$$\begin{aligned}
 x(s_0) = x_0, \quad x(s_1) = x_1, \quad x(s_2) = x_2, \\
 x(s_{N-2}) = x_{N-2}, \quad x(s_{N-1}) = x_{N-1}, \quad x(s_N) = x_N \quad (A.8)
 \end{aligned}$$

「Lagrange の補間公式」を用いて同様にすれば

$$\begin{aligned}
 \dot{x}_0 = & \frac{1}{\Delta s_2} \left[\frac{\Delta s_1 + \Delta s_2}{\Delta s_1} (x_1 - x_0) - \frac{\Delta s_1}{\Delta s_1 + \Delta s_2} (x_2 - x_0) \right] \\
 \dot{x}_N = & \frac{1}{\Delta s_{N-1}} \left[\frac{\Delta s_N}{\Delta s_{N-1} + \Delta s_N} (x_N - x_{N-2}) \right. \\
 & \left. - \frac{\Delta s_{N-1} + \Delta s_N}{\Delta s_N} (x_N - x_{N-1}) \right] \quad (A.9)
 \end{aligned}$$

を得る。

以上を二本の道路エッジに施すことにより、平面曲線 C_D, C_r が画像面上に定義され、必要ときに $x_s(s), y_s(s), \dot{x}_s(s), \dot{y}_s(s), s=1, r$ を「関数副プログラム」として使用できる。

付録 B. 初期値を求める緩和法

数値積分をはじめするには、 $l=r=0$ における R_D, R_r

の初期値が必要になる。これらの点は観測点近辺の道路上から選べばよいから、ステレオや、距離センサーなどを用いれば簡単に解決する。しかし、道路画像のみから求めればそれにこしたことはない。このために次のような緩和法によるアルゴリズムを考える。これは、道路は観測点付近ではほとんど水平であるという仮定に基づく。

道路は水平と仮定し、カメラのレンズの中心の高さを h 、カメラの光軸を水平方向から下方へ測った角度を θ 、とする。図 B1 より次の関係がわかる。

$$\begin{aligned}
 X = & \frac{hx}{f \sin \theta - x \cos \theta}, \quad Y = \frac{hy}{f \sin \theta - x \cos \theta}, \\
 Z = & \frac{hf}{f \sin \theta - x \cos \theta} \quad (B.1)
 \end{aligned}$$

定数 h, θ, f は既知であるので、道路が水平であれば観測点付近の 3 次元道路形状は式 (B.1) により完全に決定できる。これにより得られた道路は一定の幅を持つはずである。逆に、幅が一定でなければ、道路は水平ではないと結論できる。しかし、幅は「生成セグメント」によって定義され、生成セグメントを定義するには「道路の左右の境界線上の対応関係」を定めなくてはならない。これを定めるために、次のような緩和法アルゴリズムを用いる (図 B2)。

はじめに左右の曲線間に勝手な対応を与えて仮の生成セグメントとする。つぎに、各生成セグメントの中心を結び、仮のセンターラインとする。さらに、センターラインの各点で垂直に交わる直線を引き、両曲線との交点を新しい対応点対とする。以下、同様の操作を収束するまで反復し、最終的な生成セグメントとセンターラインを定める。(両曲線の端点付近は適当に延長しておくとする。)

このように定めた生成セグメントの長さがすべて仮定した一定値 e であれば道路は仮定どおり水平であると見なす。もし一定でなければ、計算による幅 e' に対応する両端点 R_D, R_r を $(e/e')R_D, (e/e')R_r$ でおきかえる。