

# 目的予測型カーナビゲーションのためのマップマッチングアルゴリズム

宮下 浩一<sup>†</sup> 寺田 努<sup>††</sup> 西尾章治郎<sup>†</sup>

<sup>†</sup> 大阪大学大学院情報科学研究科

<sup>††</sup> 神戸大学大学院工学研究科

あらまし 本稿では目的予測型カーナビゲーションシステムのためのマップマッチング手法について述べる。目的予測型カーナビゲーションシステムとは、ユーザの入力なしに目的地や目的を予測し、予測結果にあわせて様々な情報を提供するシステムである。このシステムでは、精度良く目的地を予測するために、スタート地点から現在地に至るまでに経由した道路を正確に取得することが求められる。そこで筆者らは以前、走行軌跡を一定間隔で分割し、小区間ごとに最短経路探索アルゴリズムを拡張したマップマッチング手法を用いる手法を提案した。しかしこの手法には、正しくマッチングできない走行軌跡が数パターン存在していたため、本稿ではその問題を解決する新たなマッチング手法を提案する。

キーワード センサデータ処理, ITS

## A Map Matching Algorithm for Car Navigation Systems that Predict User Destination

Koichi MIYASHITA<sup>†</sup>, Tsutomu TERADA<sup>††</sup>, and Shojiro NISHIO<sup>†</sup>

<sup>†</sup> Graduate School of Information Science and Technology, Osaka University

<sup>††</sup> Graduate School of Engineering, Kobe University

**Abstract** In this paper, we propose a map matching algorithm for car navigation systems that predict user destination. This car navigation system is a novel system that automatically predicts user purpose and destination to present various effective information based on the predicted purpose without user interaction. It requires not only the correct road links where the car drives in real time, but also the route from start position to current position correctly. Therefore, we have proposed the divided shortest path method that is enhanced shortest path algorithm. This method have some weak points that it cannot match several patterns of trajectory to correct road links. To solve this problem, we propose a new map matching method called dynamic division method.

**Key words** sensor data, ITS

### 1. はじめに

近年のカーエレクトロニクス技術の発展に伴い、カーナビゲーションシステムに対する注目が高まっている。カーナビゲーションシステムの主目的はユーザを指定の目的地まで正確にナビゲートすることであるが、次世代のカーナビゲーションシステムには単なる受動的な道案内だけでなく、車内における活動全般を支援するための情報サーバとしての役割や、ユーザの行動を予測して運転支援や安全管理を行うなどの高度な機能が要求される。そこで筆者らの研究チームでは、次世代のカーナビゲーションシステムに要求される機能の一つである能動的な情報提示を実現するために、ユーザの行動を予測し、その予測結果を元に情報を提示するシステムを構築している。具体的には、非誘導時（普段の生活乗車時）に、ユーザがわざわざ目的を入力をしなくても、システムが目的地や行動目的を自動的

に推測し、目的地の関連情報や効率的な誘導情報を自動的に提示する目的地予測手法を提案し、実際にナビゲーションシステムを開発している。

筆者らの提案する目的地予測手法は、他の従来研究で用いられている予測手法 [4] と同様に、過去にユーザが通過した道路の連なりから現走行の目的地を予測するため、システムが正確に目的地を予測するためには正しい道路通過情報を必要とする。しかし、走行履歴は全地球測位システム (Global Positioning System: GPS) の測位誤差や、各種センサを利用した自律走行法における累積誤差を原因とするノイズを含んでおり、正しい走行経路を得られないという問題があった。

この問題を解決するために、筆者らはこれまで、最短経路アルゴリズムを用いたマップマッチング手法を提案した。これは道路網をネットワークに見立て、各道路にスコアリングを行い、スコアの和が最小となる経路を選択するという手法である。し



図1 プロトタイプシステムの表示例

かし、スコアリングの方法についての議論が行われていないため、本稿では、はじめにこのスコアリングの方法についての検討を行う。また、最短経路アルゴリズムでは回り道や折り返しを含む走行を正しくマッチングできないため、走行区間を一定の間隔で分割し、区間内に回り道を作らないようにすることで対処した。しかし、分割した区間内に回り道などが存在している場合、上記の方法では正しくマッチングできない。そこで、経路の分割を効果的に行うために、回り道や折り返し走行を検出し、動的な経路分割を行うマップマッチング手法を提案する。

本稿の構成は以下の通りである。2章では想定環境である目的地予測型カーナビゲーションシステムについて述べ、3章で既存のマップマッチング手法を簡単に紹介する。4章で筆者らが提案するマップマッチング手法のベースとなる最短経路手法を説明し、そのスコアリング手法を検討する。また、5章では分割最短経路手法について述べ、動的に経路を分割する手法について説明し、それらを評価・考察する。最後に6章で本稿のまとめを行う。

## 2. 目的地予測型カーナビゲーションシステム

### 2.1 想定シナリオ

筆者らの提案するカーナビゲーションシステムは、ユーザの目的地を予測し、その予測結果を元にユーザに適した情報を自動的に提示する。以下に筆者らの想定するシナリオを挙げる。

- ユーザがショッピングモールへ向かっていることを予測したシステムが、普段使用するショッピングモールの駐車場へ向かう道路が混雑していることを認識し、別の駐車場をユーザに薦める。
- ユーザが友人を送迎しに駅へ向かっていることを予測したシステムが、その駅の電車の時刻表を提示する。
- 車両のガソリンの残量が少ないことを検知したシステムが、ユーザに警告を出すと共に、走行経路沿いのガソリンスタンドを知らせる。

このシステムのプロトタイプの表示例を図1に示す。図のように、プロトタイプでは目的地に応じた情報を複数のキャラクターの発言としてアニメーションを交えながら提示する。

### 2.2 目的地予測手法

筆者らの研究グループが提案している目的地予測手法[5]では、現走行における出発地からの走行の軌跡と過去の車両の走

表1 走行履歴

ルート	走行回数
$O \rightarrow b \rightarrow B$	6
$O \rightarrow a \rightarrow c \rightarrow C$	3
$O \rightarrow a \rightarrow e \rightarrow A$	2
$O \rightarrow d \rightarrow c \rightarrow C$	5
$O \rightarrow d \rightarrow e \rightarrow A$	4

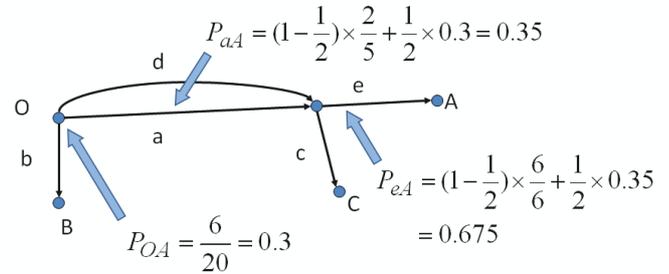


図2 目的地予測手法の例 ( $\alpha = 0.5$ )

行履歴を比較し、一致度が高い目的地を現在ユーザが向かっている目的地と推測する。

システムは道路リンクの遷移の履歴を保持しており、また、それぞれの道路リンクは目的地ごとの走行頻度を保持している。具体的には、現走行の道路リンクの遷移の履歴  $L = (l_0, l_1, l_2, \dots, l_i)$  が与えられており、ユーザは現在、道路リンク  $l_i$  上を走行しているものとする。この場合、ユーザが道路リンク  $l_i$  から目的地の候補地  $d$  へ行く確率  $P_{l_i d}$  は式1で表される。

$$P_{l_i d} = (1 - \alpha) \frac{N_{l_i d}}{N_{l_i}} + \alpha P_{l_{i-1} d} \quad (1)$$

式中の  $N_{l_i}$  は以前道路リンク  $l_i$  を走行した回数、 $N_{l_i d}$  は道路リンク  $l_i$  を通過し目的地  $d$  へ行った回数、 $\alpha$  は現在位置に達するまでの移動経路をどの程度重視するかを表す係数であり0から1の値をとる。ただし、 $P_{l_{0j}}$  は総走行回数に対する、目的地  $d$  へ行った回数の割合とする。筆者らのシステムでは、ユーザが走行する道路リンクが遷移すると上式を再計算する。例えば、図2において  $P_{OA}$  を算出する場合について述べる。 $O \rightarrow a \rightarrow e \rightarrow A$  という道順でユーザが走行した場合、過去の走行履歴(表1)を用いて図中の数式のように計算する。

## 3. マップマッチング

マップマッチングとは、GPS やその他のセンサから得られた暫定的な位置情報を道路上へマッチングさせる技術である。このマップマッチング機能は現在のカーナビゲーションシステムのみならず、ユーザの目的地を推測するシステムや、交通流を推測するシステムなどにおいても必要とされている。

特に、2章で述べた筆者らのシステムにおいては欠かせない機能である。また、吉岡らの提案する目的地予測手法[4]など従来研究においても経路通過履歴が用いられていることから、正確な走行経路は、目的地到達確率を正しく算出し、正しい目的地を予測するために不可欠である。

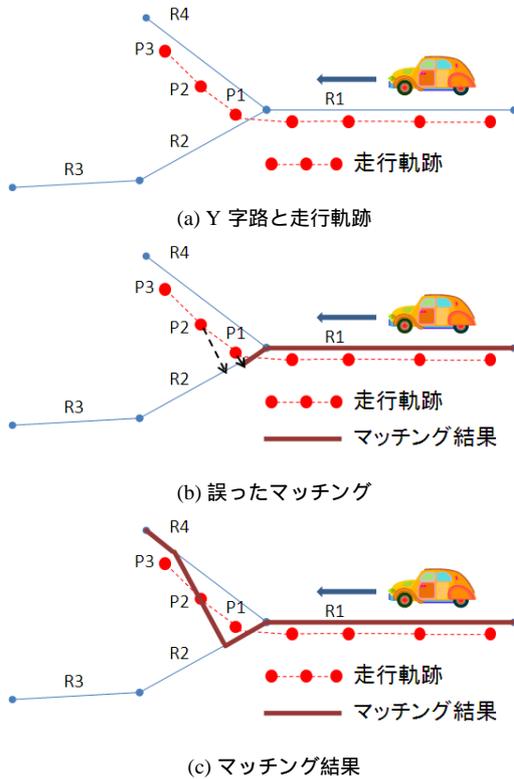


図3 マッチングミスの一例

### 3.1 現行のカーナビゲーションシステムにおけるマップマッチング

現在のカーナビゲーションシステムは、GPS と自律航法を組み合わせて現在地を特定する。自律航法とは、加速度センサや角速度センサから速度ベクトルを算出し、それらを積分することで車両の走行の軌跡を求め、得られた軌跡に類似する道路へマッチングする手法である。この自律航法だけでは時間経過と共に誤差が累積するため、GPS から取得した位置情報を用いて車両位置を修正し、誤差の累積を防いでいる。このカーナビゲーションに搭載されているマップマッチング手法は、現在の車両位置を尤もらしい道路へマッチングするという特徴があり、そのため即時性の求められるカーナビゲーションには非常に適している。

しかしながら、この手法はマッチングミスが発生した際に経路の軌跡に不自然なジャンプが発生するという問題を有する。例えば図 3(a) において、車両が Y 字路に差し掛かり、現在位置  $P_1$  と周囲の道路リンクとの類似度を算出した結果、 $P_1$  が誤った道路  $R_2$  へマッチングされた場合、以降はその誤ってマッチングした道路  $R_2$  と接続関係にある道路へマッチングされる。しかし、車両位置が  $P_3$  に差し掛かり、類似する道路が候補にない場合（図 3(b)）、接続関係にないが類似性の高い道路  $R_3$  へマッチングされてしまい、マッチング結果において、 $R_2$  から  $R_3$  への道路間のジャンプ（図 3(c)）が発生する。このような問題が発生すると、筆者らの提案するシステムでは正しく目的地を予測できないため、道路の接続関係を保ったまま、正確にマップマッチングを行なう手法が求められる。

## 3.2 関連研究

マップマッチングに関する研究は、スタティックな手法とインクリメンタルな手法とに分類される。

### 3.2.1 スタティックな手法

静的なマップマッチングとは、運転後の出発地点から到着地点までの走行軌跡に対しマッチングを行なう手法である。静的なマップマッチング手法の一つに、Brakatsoulas ら [1] が提案する、*Fréchet Distance* を用いたマップマッチング手法がある。*Fréchet Distance* とは、一般に次の例えを用いて説明される。飼い主と犬が散歩に出掛ける際に、飼い主と犬はそれぞれ異なる道（軌跡）を歩く。各々は自身の歩く速度を調整できるが、来た道に戻ることはできないものとする。このような環境において、飼い主と犬が散歩をするために必要な最低の鎖の長さを *Fréchet Distance* という。彼らは、出発地点から到着地点へ到る様々な経路のうち、GPS が描く走行軌跡との *Fréchet Distance* が最小となるような経路を見つける手法を提案している。この手法は静的な環境においては有効であるが、リアルタイムに位置情報が更新される環境で用いた場合、位置情報の更新が起こる度に出発地点から現在地点までの経路を再計算しなければならず、リアルタイムな処理には適さない。

### 3.2.2 インクリメンタルな手法

リアルタイムなマップマッチングとは、出発地点から現在地点へ到る走行軌跡をリアルタイムに得ることのできる手法である。Yang [3] は GPS から 2 分間隔で位置情報が得られる環境において、リアルタイムにマップマッチングする手法を提案している。しかし、この環境は筆者らの想定するカーナビでの位置取得という環境とは異なっており、また、位置取得が低頻度であるため、正確な走行軌跡を得る手法ではなく、GPS のデータを補完し、尤もらしいルートを作成する手法となっている。

Brakatsoulas ら [1] や Greenfield ら [2] は、高頻度で GPS による位置取得が可能な環境において、リアルタイムなマップマッチングを行なう手法を提案している。彼らの手法の概要は次の通りである。始めに GPS の測位開始地点の周辺の道路リンクを探し、近い道路リンクへマッチングする。位置情報が更新されると、GPS から得られた現在地とその一点前の位置を結ぶ線分と、先ほどマッチングした道路リンクやその道路リンクと接続している道路リンクとを距離や角度などの項目で比較し、最も類似性の高い道路にマッチングする。ここでは、Brakatsoulas らの用いた類似度の算出方法を紹介する。現在の位置を  $p_i$ 、マッチング候補の道路リンクを  $c_j$ 、距離のパラメータ評価値を  $S_d$ （式 2）、角度のパラメータの評価値を  $S_\alpha$ （式 3）とし、この二式の和  $S_d + S_\alpha$  が最大となる道路リンク  $c_j$  へ  $p_i$  をマッチングする。式中の関数  $d(p_i, c_j)$  は  $p_i$  と  $c_j$  との間の距離を、 $\alpha_{i,j}$  は  $p_i$  と  $c_j$  の成す角の大きさを意味する。また、 $\mu_d, a, n_d, \mu_\alpha, n_\alpha$  はそれぞれスケーリング用のパラメータであり、経験的に文中ではそれぞれ  $\mu_d = 10, a = 0.17, n_d = 1.4, \mu_\alpha = 10, n_\alpha = 4$  と設定している。

$$S_d(p_i, c_j) = \mu_d - a \cdot d(p_i, c_j)^{n_d} \quad (2)$$

$$S_\alpha(p_i, c_j) = \mu_\alpha \cdot \cos(\alpha_{i,j})^{n_\alpha} \quad (3)$$

この手法は、全ての GPS の点において上記の処理を施すため、GPS に含まれるノイズの影響を強く受けてしまうという欠点がある。加えて、一度誤った道路へマッチングされた場合、以降もその誤った道路と繋がる道路へマッチングしようとするため、マッチングミスを引きずってしまうという問題も抱えている。特に運転開始地点のマッチングミスが発生すると、長期間誤った道路へマッチングされ続ける恐れがある。

#### 4. 最短経路手法

本章では、我々が提案しているマップマッチング手法のベースとなる最短経路手法 [6] を説明する。本手法は他のマップマッチング手法と同様に、道路網をネットワークと見立て、交差点をネットワークの節、交差点間の道路（道路リンク）を枝として考える。ただし、道路リンクの端点である交差点ノード以外では、他の道路リンクと接続しないものとする。また、GPS から得られた離散的な点列をマッチング対象とする。

最短経路アルゴリズムとは、ネットワークを構成するリンクに任意のスコアを付加し、始点と終点を結ぶ経路において経由したリンクのスコアの総和を算出し、この総和が最小となる経路を導出するアルゴリズムである。我々はこの最短経路アルゴリズムを用いたマップマッチング手法を提案している。走行軌跡を元に各道路リンクにスコア付けを行い、始点から終点までのスコアの和が最小となる経路を正解ルートとする。本手法のマッチングの精度はスコアリング方法に依存しており、本当に走行した道路に小さなスコアを、走行していない道路には大きなスコアを付与すれば正しい経路を導出することができる。

以降では、複数のスコアリング手法を比較し、マップマッチングに適した方法を求める。

##### 4.1 スコアリング手法

スコアリングに用いる指標には、走行軌跡と道路リンクとの距離、進行方向と道路リンクとの偏角、道路リンク長などが挙げられ、これらを組み合わせてスコアリングを行う。比較した手法は次の通り。

**距離** 走行軌跡と道路リンクとの距離をスコアとする。走行軌跡に近い道路が選ばれる。

**偏角** 進行方向と道路リンクとの偏角の正弦をとった値をスコアとする。走行軌跡と並行して走る道路が選ばれる。

**リンク長** 道路リンクの長さをスコアとする。始点と終点を結ぶ経路のうち、実環境での最短経路を選択する。

**距離・偏角** 走行軌跡と道路リンクとの距離と進行方向と道路リンクとの偏角の正弦の積をスコアとする。距離が近く、走行軌跡と並行して走る道路が選ばれる。

**距離・リンク長** 走行軌跡と道路リンクとの距離と道路リンク長の積をスコアとする。走行軌跡と経路が描く面の面積が小さくなる経路が選ばれる。

**距離・リンク長・偏角** 走行軌跡と道路リンクとの距離と道路リンク長、進行方向と道路リンクとの偏角の正弦の積をスコアとする。

##### 4.2 スコアリング手法の評価

上記のスコアリング手法について評価実験を行った。実験に

表 2 スコアリングの評価結果

スコアリング手法	一致率	過剰率	不足率
距離	74.35	6.24	19.42
偏角	47.48	18.04	34.48
リンク長	57.94	16.00	26.06
距離・偏角	67.93	10.41	21.67
距離・リンク長	89.28	3.13	7.59
距離・リンク長・偏角	73.42	10.90	15.68

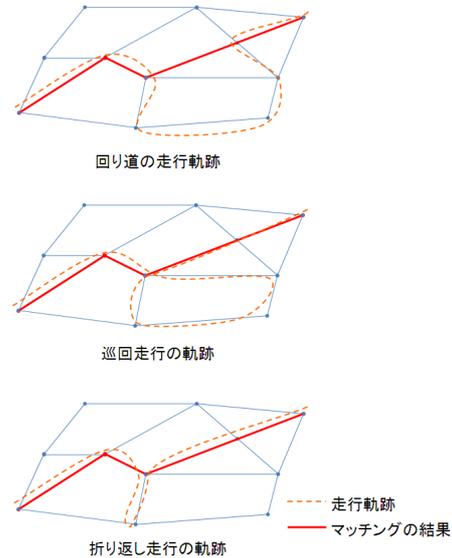


図 4 最短経路手法におけるマッチングミスの例

は国土地理院の発行する 1/2500 の日本の数値地図と GARMIN 社の GPS モジュール (Garmin GPS eTrex V1sta-C) を使用した。あるユーザの日常生活における車両での移動の記録 12ヶ月分 (総走行回数: 203 回) に対しマップマッチングを行なった。

##### 4.2.1 スコアリングの評価結果

走行軌跡全体を最短経路手法を用いてマッチングし、その結果得られた道路リンク群  $M$  と、正解ルートの道路リンク群  $T$  を比較した。なお正解ルートは、生データである GPS の点列から尤もらしい経路を手入力で作成したものを使用した。また道路リンク群  $A$  に含まれる道路リンクの個数を  $N(A)$  として、式 4, 5, 6 にて、一致率、過剰率、不足率を求めた。一致率は、正解ルートおよびマッチングで得られた経路に含まれる道路の総数に対する、2つの経路が一致している部分の道路の割合、過剰率は、マッチング結果の経路には含まれるが正解ルートではない、誤った道路にマッチングされた割合、不足率は、マッチング結果の経路には含まれていないが正解ルートである、すなわち誤ってマッチングされなかった道路の割合を意味する。

$$\text{一致率} = \frac{N(M \cap T)}{N(M \cup T)} \quad (4)$$

$$\text{過剰率} = \frac{N(M \cap \bar{T})}{N(M \cup T)} \quad (5)$$

$$\text{不足率} = \frac{N(\bar{M} \cap T)}{N(M \cup T)} \quad (6)$$

評価結果を表 2 に示す。この結果から、提案する最短経路手法では、スコアリングには距離とリンク長の積を用いるときに

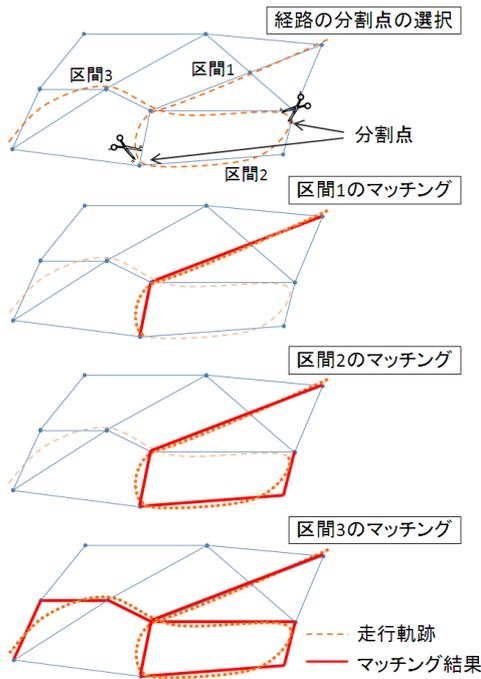


図5 分割最短経路手法の概要

最も良い結果となることがわかる。

距離・偏角・リンク長の3つの要素それぞれを個別に利用した場合、距離のみを考慮した手法が最も一致率が高いことから、マップマッチングにおいては、走行軌跡と道路との距離が最も重要な要素であると言える。また、リンク長のみを考慮した手法の一致率が5割程度であることから、実環境での走行において、ユーザが最短ルートを選んで走行する機会は全体の半分程度であり、道路幅や車線数などから走行しやすい経路を選択する・寄り道をするなどの理由により、最短ルートではない経路を選択することも多々あることがわかる。偏角を考慮した手法は、いずれもマッチングの精度が良くないことから、偏角を考慮してスコアリングを行うことは、マップマッチングには適していないと言える。

## 5. 分割最短経路手法

先に述べた最短経路手法を用いることで、大抵の走行履歴を正しくマッチングできるが、ある特定の状況においてマッチングエラーが発生する。例えば図4に示すように、回り道をした場合や、巡回走行した場合、走行中に折り返しが発生した場合などの状況では、最短経路アルゴリズムの特性が強く表れ正しくマッチングできない。上記のマッチングミスが発生する原因は、1走行を丸ごとマッチングするためである。また、このような走行全体をマッチングする手法は走行が終了した状態には適しているが、リアルタイムなマッチングには不向きである。そこで我々は、走行軌跡を一定間隔で分割し、分割された小区間ごとに前節のアルゴリズムを適用する手法[6]を提案した。例えば、図5のような巡回を含む走行軌跡でも、図のように経路を複数に分割し、各区間をマッチングしていくことで、最短経路手法において問題であった走行軌跡に対しても正しくマッチングできる。しかし、一定間隔で経路を区切る手法においても、分割した区間内に回り道などの最短経路手法が苦手とする走行

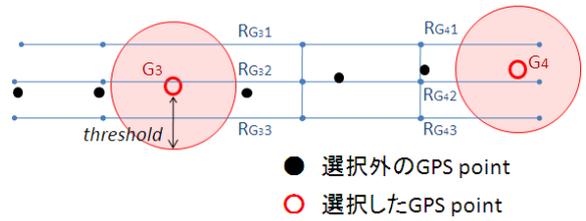


図6 分割最短経路手法の計算方法の例

	$R_{G_{41}}$	$R_{G_{42}}$	$R_{G_{43}}$
$R_{G_{31}}$	30	20	90
$R_{G_{32}}$	30	10	50
$R_{G_{33}}$	70	30	20

	Total Score
$TS_{G_{31}}$	100
$TS_{G_{32}}$	50
$TS_{G_{33}}$	70

があった場合は、同様の問題が発生してしまう。そこで本稿では、経路内の回り道や折り返しの有無を逐次確認し、それらが検出された位置で走行軌跡を分割する、動的分割手法を新たに提案する。

### 5.1 分割最短経路手法の詳細

手法の詳細は次の通りである。はじめに、次節で述べる分割点の選択手法にて、経路を分割する位置  $G_i$  を選択し、 $G_i$  から半径  $threshold$  以内にある  $J$  個の道路リンク  $R_{G_{ij}}$  ( $j = 1, 2, \dots, J$ ) を抽出する。

例えば図6において、現在の車両の位置が  $G_4$  である場合、 $G_4$  の周囲の道路リンク ( $R_{G_{41}}, R_{G_{42}}, R_{G_{43}}$ ) を抽出する。

この道路リンク  $R_{G_{ij}}$  と、一点前の  $G_{i-1}$  の周囲の  $J$  個の道路リンク  $R_{G_{i-1k}}$  ( $k = 1, 2, \dots, J$ ) を結ぶ経路のスコア  $S(R_{G_{i-1k}}, R_{G_{ij}})$  を最短経路手法を用いて算出する。このようなスコア付けは、分割された各区間 ( $G_0 \rightarrow G_1, G_1 \rightarrow G_2, \dots, G_{i-1} \rightarrow G_i$ ) で行なわれる。

次に、総合スコア  $TS_{G_{ij}}$  を下記のように定義する。

$$TS_{G_{ij}} = \min(TS_{G_{i-1k}} + S(R_{G_{i-1k}}, R_{G_{ij}})) (k = 1, 2, \dots, J)$$

総合スコアとは、走行開始地点  $G_0$  から  $G_i$  までの各区間におけるそれぞれのスコアを、和が最も小さくなるように足し合わせた値である。 $TS_{G_{i-1k}}$  は、一点前の  $G_{i-1}$  の周囲の道路リンク  $R_{G_{i-1k}}$  における総合スコアであり、 $G_{i-1}$  の時点で既に算出済みである。

以上の手法により、 $G_i$  の周囲の道路リンク群  $R_{G_{ij}}$  の総合スコア  $TS_{G_{ij}}$  を算出し、総合スコアの最も小さい  $R_{G_{ij}}$  を現在地  $G_i$  の暫定的な車両位置とする(式7)。

$$G_i = \min(TS_{G_{ij}}) (j = 1, 2, \dots, J) \quad (7)$$

例えば図6においては、最短経路手法を用いて、一点前にマッチングした  $G_3$  の周囲の道路リンク  $R_{G_{3k}}$  から  $R_{G_{4j}}$  へ到る経路とそのスコアを求める。この例では表3に示すような結果が得られたとする。また、走行開始地点  $G_0$  から  $R_{G_{3k}}$  へ到る経路の総合スコア  $TS_{G_{3k}}$  が表4である場合、 $TS_{G_{3k}} + S(R_{G_{3k}}, R_{G_{4j}})$  の計算結果は表5となる。

最後に  $R_{G_{4j}}$  の総合スコア  $TS_{G_{4j}}$  を決定し、 $G_4$  における処理が終了する。この例では、表5における各行の最小値が

表5  $TS_{G_{3k}} + S(R_{G_{3k}}, R_{G_{4j}})$  の計算結果

	$R_{G_{41}}$	$R_{G_{42}}$	$R_{G_{43}}$
$R_{G_{31}}$	130	120	190
$R_{G_{32}}$	80	60	100
$R_{G_{33}}$	140	100	90

表6  $TS_{G_{4j}}$

	Total Score
$TS_{G_{41}}$	80
$TS_{G_{42}}$	60
$TS_{G_{43}}$	90

表7 評価結果

マッチング手法	一致率	過剰率	不足率
インクリメンタルな手法	42.05	6.51	51.44
最短経路手法	88.53	2.36	9.11
分割最短経路手法			
一定間隔：7分	92.23	2.34	5.43
: 5分	94.43	2.07	3.50
: 3分	94.50	2.51	2.99
動的分割(上限値, 下限値)			
:(7分, 10秒)	94.92	2.91	2.17
:(5分, 10秒)	95.19	2.92	1.90
:(3分, 10秒)	94.95	3.03	2.01
:(7分, 3.5分)	95.17	2.27	2.57
:(5分, 2.5分)	95.51	2.42	2.07
:(3分, 1.5分)	95.52	2.55	1.93

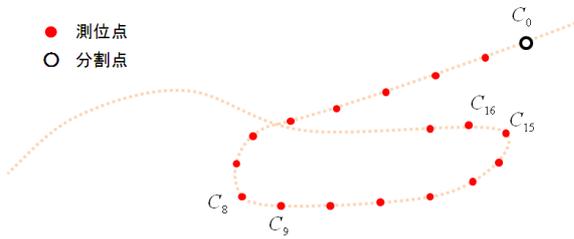


図7 動的分割の例

$TS_{G_{4j}}$  であり、その値を表6に示す。最も小さい総合スコア  $TS_{G_{4j}}$  をもつ  $R_{G_{42}}$  が、 $G_4$  をマッチングする道路リンクとなり、 $R_{G_{42}} \rightarrow R_{G_{32}} \rightarrow \dots \rightarrow R_{G_{0j}}$  というルートが作成される。

以上の手順で、分割した各区間において最適な経路を選択する。また、本手法では経路の分割は数分間隔で行うことを想定しており、分割点から次の分割点まで数分間は、既存のインクリメンタルな手法を用いて仮のマッチングを行う。これらの仮のマッチングを行った部分は、数分後の次の分割点にて提案手法よりもっともらしい経路が与えられると、その経路上へ再マッチングされる。

### 5.2 動的分割手法

本節では、最短経路手法が苦手とする回り道や折り返し走行などの走行パターンを検出し分割する手法について述べる。

経路分割には、区間の開始地点 ( $C_0$ ) から車両位置 ( $C_i$ ) までの直線距離 ( $L(C_0, C_i)$ ) と、その前に得られた車両位置 ( $C_{i-1}$ ) までの直線距離 ( $L(C_0, C_{i-1})$ ) を用い、 $L(C_0, C_i)$  が  $L(C_0, C_{i-1})$  より短い場合、 $C_i$  を新たな分割点とする。例えば図7の巡回を含む走行軌跡の場合、 $C_1$  から  $C_8$  は、 $C_0$  から離れていくが、次の  $C_9$  の直線距離  $L(C_0, C_9)$  は、 $L(C_0, C_8)$  よりも短いので、 $C_9$  が新たな分割点となる。同様に、その次の分割点は  $C_{16}$  となる。

ただし、分割が長い間発生しない場合でも、区間当たりの時間の上限値 ( $T_{max}$ ) を設け、区間の開始から  $T_{max}$  経過した地点を新たな分割点とする。また、分割最短経路手法において、短い時間で分割を繰り返すとマッチングの精度が低下するという結果が得られているため、分割の開始から、区間当たりの時間の下限値  $T_{min}$  が経過するまでは分割を行わないものとする。

### 5.3 評価・考察

この動的に分割を行う分割最短経路手法についての評価を行った。実験には4章と同様の環境で取得した実走行ログ8ヶ月分(総走行回数: 108回)を用いた。比較対象は、2章で紹介した Brakatsoulas らのインクリメンタルな手法と、4.1節で述べた最短経路手法、および走行データを一定間隔で区切って最短経路手法を適用する分割最短経路手法である。さらに分割最短経路手法は、一定間隔(7分, 5分, 3分)で分割したものと、5.2章で提案する動的分割手法(区間当たりの時間の上限

値:  $T_{max} = 7$ 分, 5分, 3分)を比較した。動的分割手法における区間当たりの時間の下限値 ( $T_{min}$ ) は、10秒とした場合と、 $T_{max}$  の2分の1とした場合を比較した。

最短経路手法および分割最短経路手法にて用いたスコアリング手法は4.2節の距離とリンク長である。また、分割する点  $G_i$  において抽出する道路リンクは、点  $G_i$  から半径100m以内にある道路リンクとし、 $G_i$  に近い道路リンクから5個を選択し、マッチング対象の道路リンクとした。すなわち5.1節における変数をそれぞれ  $threshold = 100$ ,  $J = 5$  と設定した。

#### 5.3.1 全体的な評価

4章の一致率・過剰率・不足率を用いてマッチングの精度を評価した。評価結果を表7に示す。

この結果より、動的に分割した方が、一定間隔で分割するよりも一致率が向上していることがわかる。また、動的分割手法において、分割した1区間当たりの時間の上限値を7分・5分・3分として比較したが、各々に明確な差は表れなかった。

分割間隔の下限値を10秒と設定した場合と、上限値の半分とした場合で比較すると、後者の方がマッチングの精度が高い。下限値を小さな値に設定した方が、回り道等に細かく対応できるため精度が向上すると予想していたが、その逆の結果となった。これについては、個々のマッチング結果を比較し考察する。

#### 5.3.2 個々のマッチング結果の評価

動的分割手法における下限値の違いにより、マッチング結果が異なる例を図8に示す。赤い点線がGPSから得られた軌跡であり、緑の実線がマッチング結果である。この走行軌跡は、図中の丸の部分において走行軌跡が一部下方へずれている。分割間隔の下限値が小さい上図では、矢印の位置で分割が行われ、その結果、平行する誤った道路へ一部マッチングされている。分割間隔の下限値が小さい場合、このような局所的な走行軌跡のずれを拾ってしまうため、マッチング精度が低下していた。

図9に、動的分割手法を用いてマッチングした結果の走行軌跡を示す。この走行軌跡は、図中の丸Aの部分の先で巡回する走行であり、最短経路手法が苦手とする軌跡の一つであるが、図のように正しい経路へマッチングできている。また、丸Bの部分のように、平行している道路が何本もあり、かつ、走行軌

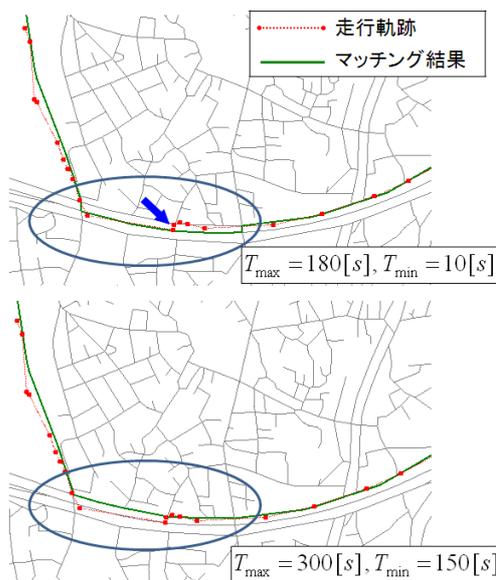


図8 マッチング結果 (1)

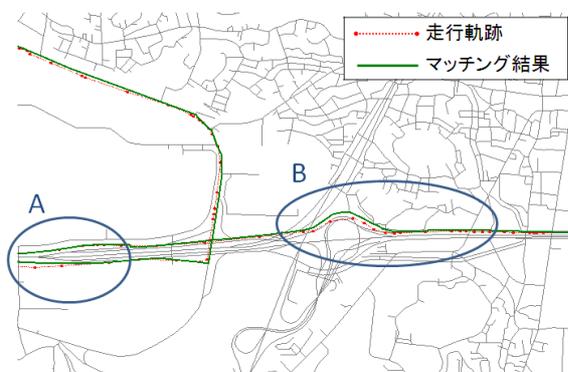


図9 マッチング結果 (2)

跡が平行する他の道路と重なっている状況であるにも関わらず正しい経路上へのマッチングができています。このように、提案手法はインクリメンタルな手法とスタティックな手法の欠点をともに解決する手法となっている。

### 5.3.3 マッチング精度の時間遷移

ある走行における動的分割手法（上限値：5分，下限値：2.5分）のマッチング精度の時間遷移を図10, 11に示す。横軸は運転開始からの時間，縦軸はマッチングされたGPSの点のうち、正しい経路上にマッチングされた点の割合を示す。

図10, 11より、数分間隔で大きく上昇し、その後緩やかに減少している様子がわかる。このようなグラフは他の多くの走行ログにおいてみられた。これは、分割点ごとに道路のつながりを考慮した最短経路手法によるマップマッチングが行われ、過去のデータにさかのぼって再マッチングするため、マッチング精度が大きく上昇するからである。また、その後徐々に精度が低下する区間もあるが、これは、インクリメンタルな手法が適応されている区間に誤った道路にマッチングされ、そのまま誤りが続いたためである。図10の走行での精度が低下している区間は、高速道路と環状線および側道が平行している部分を走行していたため、インクリメンタルな手法で誤って平行する他の道路へマッチングされていた。

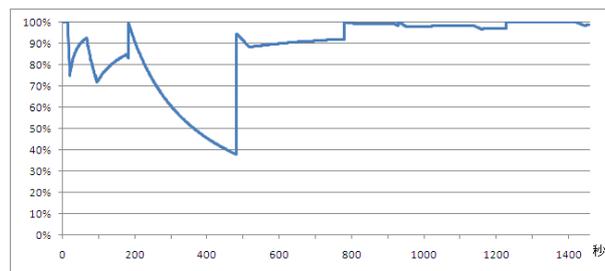


図10 マッチング精度の時間遷移 (1)

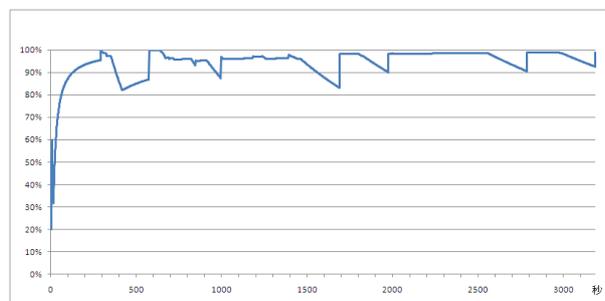


図11 マッチング精度の時間遷移 (2)

## 6. まとめ

本研究では、目的地予測型カーナビゲーションシステムが適切な目的地予測を行なうための、正しい経路履歴をリアルタイムで作成する手法について提案した。走行軌跡を動的に分割することで、回り道や折り返しの軌跡に柔軟に対応し、従来手法よりも精度よくマッチングできる。また、評価の結果、適度な間隔で分割することで、インクリメンタルな手法を適応時に生じたマッチングミスから正しい経路上へ復帰し、また、過去の走行軌跡も併せて再マッチングしていることが確認できた。

## 謝辞

本研究の一部は、文部科学省科学研究費補助金基盤研究(A)(17200006)の研究助成によるものである。ここに記して謝意を表す。

## 文献

- [1] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk: "On Map-Matching Vehicle Tracking Data," in *Proc. of the 31st Very Large Data Base Conference (VLDB)*, pp. 853–864 (2005).
- [2] J. Greenfeld: "Matching GPS Observations to Locations on a Digital Map," in *Proc. of the 81th Annual Meeting of the Transportation Research Board* (Jan. 2002).
- [3] J. Yang, S. Kang, and K. Chon: "The map matching algorithm of GPS data with relatively long polling time intervals," in *Jour. of Eastern Asia Society for Transportation Studies(EASTS)*, pp. 2561–2573 (2005).
- [4] V. Kostov, J. Ozawa, M. Yoshioka, and T. Kudoh: "Travel destination prediction using frequent crossing pattern from driving history," in *Proc. of IEEE Intelligent Transportation Systems*, Vol. 13, pp. 343–350 (Sept. 2005).
- [5] T. Terada, M. Miyamae, Y. Kishino, K. Tanaka, S. Nishio, T. Nakagawa, and Y. Yamaguchi: "Design of a Car Navigation System that Predicts User Destination," in *Proc. of the 1st workshop on tools and applications on mobile contents (TAMC)*, pp. 54–59 (May 2006).
- [6] 宮下浩一, 寺田 努, 西尾章治郎 "目的地予測型カーナビゲーションシステムのためのマップマッチング手法," 情報処理学会研究報告(ユビキタスコンピューティングシステム研究会 2007-UBI-16), Vol. 2007, No. 118, pp. 121–127 (Nov. 2007).