

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

宮下 浩一^{†1} 寺田 努^{†2} 西尾 章治郎^{†1}

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、正しい現在地をリアルタイムで取得することや、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、最短経路探索手法を拡張することで、リアルタイムで正確なルート情報を生成するマップマッチング手法を提案する。実際の走行データを用いた評価実験の結果、従来手法と比べて高い精度でマップマッチングが行なえていることが明らかとなった。

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

KOICHI MIYASHITA,^{†1} TSUTOMU TERADA^{†2} and SHOJIRO NISHIO^{†1}

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 correct road links where the car drives in real time, and it also need to know the route from start position to current position correctly. The proposal method achieves a real time map matching that can create a correct route by using the enhanced shortest path algorithm. The simulation study confirmed the effectiveness of our method.

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ひとつである、ユーザの行動を予測し、ユーザがわざわざ入力をしなくても適切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、自動的にユーザの目的地や行動目的を推測し、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。システムは正確に目的地を予測するため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、このノイズが目的地予測に大きな影響を与えるという問題があった。そこで本研究では、これら走行データに含まれるノイズを除去し、正確な道路通過情報を生成する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を説明し、5章にて提案手法の評価結果を示し考察する。最後に6章で本稿のまとめを行う。

^{†1} 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University

^{†2} 神戸大学工学部
Faculty of Engineering, Kobe University



図1 プロトタイプシステムの表示例

2. 目的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を表示例を図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³⁾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

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

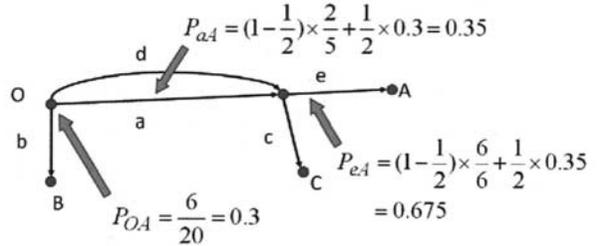


図2 目的地予測手法の例 ($\alpha = 0.5$)

地の候補地 d へ行く確率 $P_{i,d}$ は式 2.1 で表される。

$$P_{i,d} = (1 - \alpha) \frac{N_{i,d}}{N_{i,i}} + \alpha P_{i-1,d} \quad (2.1)$$

式中の $N_{i,i}$ は以前道路リンク l_i を走行した回数、 $N_{i,d}$ は道路リンク l_i を通過し目的地 d へ行った回数、 α 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、 $P_{i,oj}$ 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、車両は道路上を走行するという特性を用いて、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⁴⁾の提案する目的地予測手法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。

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ることで現在地を特定する。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、得られた軌跡と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、静的なマップマッチングとリアルタイムなマップマッチングとに大きく分類される。

3.2.1 静的なマップマッチング

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、

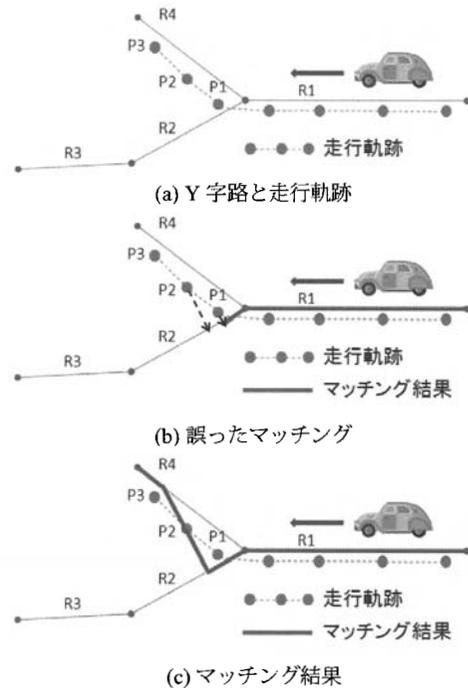


図 3 マッチングミスの一例

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 リアルタイムなマップマッチング

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。Yang²⁾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 3.1)、角度のパラメータの評価値を S_α (式 3.2) 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、 $\mu_d, a, n_d, \mu_\alpha, n_\alpha$ はそれぞれスケール用パラメータであり、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。

$$S_d(p_i, c_j) = \mu_d - a \cdot d(p_i, c_j)^{n_d} \quad (3.1)$$

$$S_\alpha(p_i, c_j) = \mu_\alpha \cdot \cos(\alpha_{i,j})^{n_\alpha} \quad (3.2)$$

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. 提案手法

本研究で提案するマップマッチング手法は最短経路アルゴリズムを拡張したものである。はじめにベースとなる最短経路手法について説明し、次に最短経路手法の拡張版である分割最短経路手法について述べる。

4.1 最短経路手法

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。また、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に観測された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。提案手法におけるスコア付けは、道路リンクの長さ ($length$) と、道路リンクと車両の走行軌跡との間の最短距離 ($distance$) の積と定義した。図 4 の例では、 $length$ と $distance$ は図中の二本の矢印で表現している。

$$Score = length \times distance \quad (4.1)$$

ただし、始点・終点となる道路に対して式 4.1 を適用すると、道路のリンク長の長短の影響が大きく正しくスコアリングできないため、始点・終点では観測地点から候補の道路の端点までの距離を $length$ とする。例えば図 4 において、始点となる道路リンクが $R0$ で

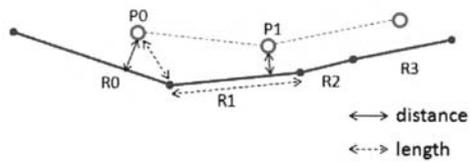


図 4 スコアリングの例

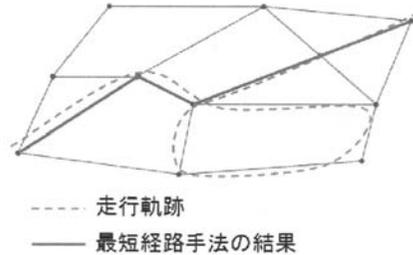


図 5 最短経路手法におけるマッチングミスの例 (1)

あった場合に、もし $length$ に道路リンク $R0$ の長さを用いると $R0$ のスコアが非常に大きくなり、 $P0$ は $R0$ ではなく $R1$ にマッチングされてしまう。このようなマッチングミスを防ぐために、上記の例外処理を行う。

最後に始点候補と終点候補の道路を結ぶ経路を、最短経路アルゴリズムを用いて導出し、始点から終点までのスコアの和が最小の経路を正解ルートとする。

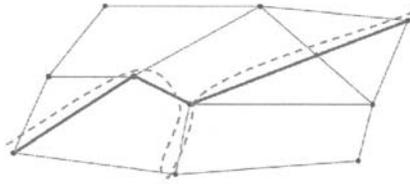
4.2 分割最短経路手法

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、巡回走行した場合 (図 5) や、走行中に折り返し (図 6) が発生した場合、回り道をした場合 (図 7) などの状況では、最短経路アルゴリズムの特性が強くと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。

そこで走行軌跡を一定区間で分割し、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を提案する。具体的には次の通りである。まず一定時間ごとに、暫定的な車両位置 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 $R_{G_i,j}$ ($j = 1, 2, \dots, J$) を抽出する。

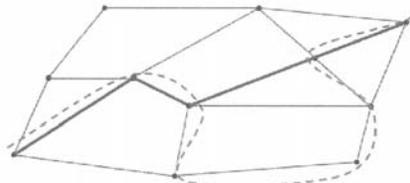
例えば図 8 において、現在の車両の位置が G_4 である場合、 G_4 の周囲の道路リンク ($R_{G_4,1}, R_{G_4,2}, R_{G_4,3}$) を抽出する。

この道路リンク $R_{G_i,j}$ と、一点前の G_{i-1} の周囲の J 個の道路リンク $R_{G_{i-1},k}$ ($k = 1, 2, \dots, J$) を結ぶ経路のスコア $S(R_{G_{i-1},k}, R_{G_i,j})$ を最短経路手法を用いて



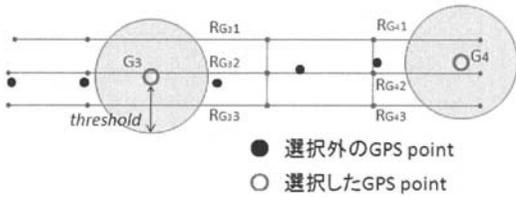
----- 走行軌跡
 —— 最短経路手法の結果

図6 最短経路手法におけるマッチングミス例(2)



----- 走行軌跡
 —— 最短経路手法の結果

図7 最短経路手法におけるマッチングミス例(3)



● 選択外のGPS point
 ○ 選択したGPS point

算出する。このようなスコア付けは、分割された各区分 ($G_0 \rightarrow G_1, G_1 \rightarrow G_2, \dots, G_{i-1} \rightarrow G_i$) で行なわれる。

次に、総合スコア $TS_{G_{i,j}}$ を下記のように定義する。

$$TS_{G_{i,j}} = \min(TS_{G_{i-1,k}} + S(R_{G_{i-1,k}}, R_{G_{i,j}})) \quad (k = 1, 2, \dots, J) \quad (4.2)$$

総合スコアとは、走行開始地点 G_0 から G_i までの各区分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。 $TS_{G_{i-1,k}}$ は、一点前の G_{i-1} の周囲の道路リンク $R_{G_{i-1,k}}$ 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 $R_{G_{i,j}}$ の総合スコア $TS_{G_{i,j}}$ を算出し、総合スコアの最も小さい $R_{G_{i,j}}$ を現在地 G_i の暫定的な車両位置とする(式 4.3)。

$$G_i = \min(TS_{G_{i,j}}) \quad (j = 1, 2, \dots, J) \quad (4.3)$$

例えば図8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 の周辺の道路リンク R_{G_3k}

表2 $S(R_{G_3j}, R_{G_4j})$

	R_{G_41}	R_{G_42}	R_{G_43}
R_{G_31}	300	200	900
R_{G_32}	300	100	500
R_{G_33}	700	300	200

表3 TS_{G_3j}

	Total Score
TS_{G_31}	1000
TS_{G_32}	500
TS_{G_33}	700

表4 $TS_{G_3k} + S(R_{G_3k}, R_{G_4j})$ の計算結果

	R_{G_41}	R_{G_42}	R_{G_43}
R_{G_31}	1300	1200	1900
R_{G_32}	800	600	1000
R_{G_33}	1400	1000	900

表5 TS_{G_4j}

	Total Score
TS_{G_41}	800
TS_{G_42}	600
TS_{G_43}	900

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2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3である場合、 $TS_{G_3k} + S(R_{G_3k}, R_{G_4j})$ の計算結果は表4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4における各行の最小値が TS_{G_4j} であり、その値を表5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。

5. 評価

5.1 評価環境

これまで述べてきた手法についての評価を行なった。実験に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 Vista-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 3ヶ月分 (総走行時間: 約 20h, 総走行回数: 57回, 平均乗車時間: 21分/回) に対しマップマッチングを行なった。比較対象は、2章で紹介した Brakatsoulas らのリアルタイムなマッチング手法 (以降、単純手法) と、4.1節で述べた最短経路手法、および走行データを区間ごとに区切って最短経路手法を適応する分割最短経路手法である。分割最短経路手法では、分割する間隔を 10分, 7分, 5分, 3分, 1分, 1秒間隔としてそれぞれマッチングを行なった。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 3.2 節における変数をそれぞれ $threshold = 100, J = 5$ と設定した。

5.2 評価結果

5.2.1 全般的な評価

マッチング結果の道路リンク群 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

表 6 評価結果

	一致率	過剰率	不足率
最短経路手法	88.74	1.90	9.36
分割最短経路手法 (10分)	90.18	2.52	7.30
分割最短経路手法 (7分)	91.56	2.54	5.90
分割最短経路手法 (5分)	94.92	2.21	2.87
分割最短経路手法 (3分)	94.00	3.09	2.91
分割最短経路手法 (1分)	93.58	3.54	2.87
分割最短経路手法 (1秒)	84.91	9.91	9.91
単純手法	41.30	9.15	49.55

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 5.1, 5.2, 5.3 にて、一致率、過剰率、不足率を求めた。

$$\text{一致率} = \frac{N(M \cap T)}{N(M \cup T)} \quad (5.1)$$

$$\text{過剰率} = \frac{N(M \cap \bar{T})}{N(M \cup T)} \quad (5.2)$$

$$\text{不足率} = \frac{N(\bar{M} \cap T)}{N(M \cup T)} \quad (5.3)$$

評価結果を表 6 に示す。最短経路手法を用いた手法のみでも 9 割近い一致率を示していることがわかる。

一致率は、分割の間隔を 5 分間隔とした際は最も高い値を示し、1 分間隔や 1 秒間隔で分割した際は低下した。分割の間隔が狭くなるほど不足率が低下したことから、間隔を狭くした方が、正解道路リンクをより多く抽出できていることがわかり、また、分割間隔が狭いほど過剰率が上昇することから、不要な道路リンクにまでマッチングしていることがわかる。

また、10 分間隔や 7 分間隔で分割した際も、一致率は低下する。分割間隔が広くなるにつれ過剰率と不足率が共に上昇することから、最短経路手法で問題となっていた、順回路、折り返し、回り道などが発生し、マッチング精度が悪くなっていると考えられる。

5.2.2 個々のマッチング結果の評価

全体評価の結果をより正しく理解するために、各々の走行についての評価も行なった。全走行データのうち新しいもの 19 走行におけるマッチング結果を正解ルートと比較した。過剰率を図 9、不足率を図 10 に示す。

No.3, No.4 の走行は折り返し運転を含んでいた。特に No.4 の走行に関しては、自宅を出発し、駅まで同乗者を送り帰宅するという走行であった。そのため、最短経路アルゴリズムを用いた結果、自宅周辺の道路リンクにしかマッチングされず、不足率が著しく上昇していた。これの問題に対しては、細かな間隔で分割をすることで不足率が低減できていることがわかる。

次に、No.8 の走行を 1 秒毎に分割してマッチング結果を図 11 に示す。1 秒毎に細かくマッチングすると、正しい道路から大きく逸れてしまった GPS のログに対し、適切なマッチングができていないことがわ



図 11 マッチング結果 (No.8)



図 12 出発地点におけるマッチングミス

かる。これは GPS 1 秒毎に周辺のマッチング対象リンクを探し、それらのいずれかにマッチングするという手法となってしまうため、GPS に含まれるノイズが大きい場合、そのノイズの影響を強く受けてしまうからだと考えられる。一方、ある程度の間隔を開けて分割した場合、正しくマッチングできていることがわかる。

また No.19 の走行に関しては、いずれのマッチング手法でも比較的良好な結果が得られているが、正解ルートと完全に一致してはならず、一定のマッチングミスが常に発生している。マッチング結果をプロットしたところ、これらは始点・終点付近で起きており、このような結果は他のログにも見られた (図 12)。運転開始時・運転終了時は住宅地などの GPS の電波の受信の妨げとなる要素が多い地点であることに加え、特に運転開始時においては、GPS 衛星の捕捉に時間を要するため、その間の位置情報が不安定であることも原因であると考えられる。ただし実車環境においては、運転開始地点は前回の運転終了地点と同じ位置であるため、上記の運転開始時の測位精度はあまり問題にならないと考えられる。

5.2.3 計算量

提案手法がリアルタイムに使用できることを示すために、計算量についての考察を行う。マッチング対象の地図内に含まれる道路リンクの数を m 、選択した GPS の周辺の道路リンクを抽出する数を J 、分割した区間内に含まれる GPS の計測点数を g とする。

最初に、選択された GPS の周囲の道路リンクを抽出する処理に要する計算量が $O(\log m)$ 。次に、マッチング対象の地図内に含まれる道路リンクに対しスコア付けを行う処理が $O(g \log m)$ である。最後に、最短経路

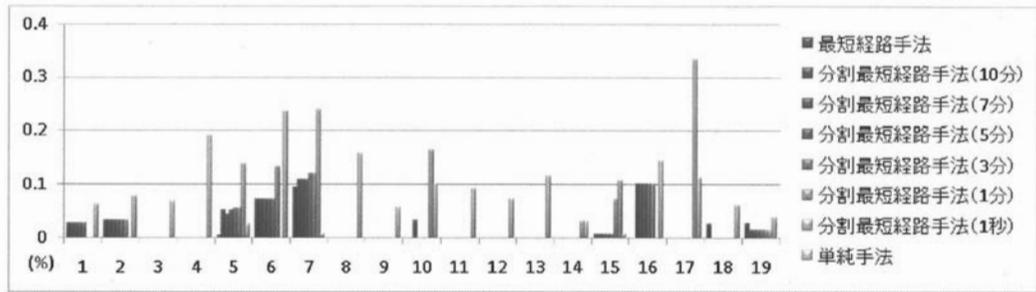


図9 過剰率

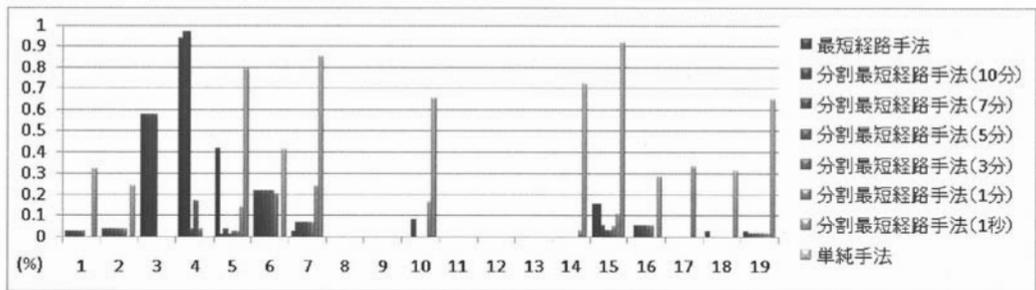


図10 不足率

手法を J^2 回実行するために $O(J^2 m \log m)$ を要する。ただし、評価環境において $J = 5$ とした様に、 J^2 は一定値であり計算量に及ぼす影響は小さい。以上をまとめると、全体の計算量は $O((g + m) \log m)$ となる。提案手法においては、数分間隔で GPS の軌跡を区切り、各々に対して最短経路を適応するため、 g や m が与える影響も限られるためリアルタイムな処理に適しているといえる。実際の評価環境 ($threshold = 100$) において、ある走行記録を 5 分間隔で分割した場合、 $g = 300$, $m = 50 \sim 148$ であったことから、 m の値がある程度の範囲で収まっていることがわかる。

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。評価の結果、従来手法よりも精度が向上していることがわかった。また、5 分間隔で走行履歴を分割してマッチングすると、最も良い結果が得られた。

今後の課題としては、マッチング精度のなお一層の向上へ向け、マッチングアルゴリズムの改善を行なう予定である。また、最適な分割間隔や、スコアリング手法についても議論していく必要がある。

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。

参考文献

- 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 (Sep. 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).