



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
寺田, 努
田中, 宏平
西尾, 章治郎

(Citation)

情報処理学会論文誌, 50(1):75-86

(Issue Date)

2009-01-15

(Resource Type)

journal article

(Version)

Version of Record

(URL)

<https://hdl.handle.net/20.500.14094/90002112>



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}
田中 宏平^{†1} 西尾 章治郎^{†1}

本論文では目的予測型カーナビゲーションシステムに適したマップマッチング手法を提案する。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。このシステムが精度良く目的地を予測するためには、スタート地点から現在地に至る経路の正確な取得が必要となるが、既存の位置測定手法では測位誤差のため正しい目的地予測ができなかった。そこで本研究では、経路の接続関係を考慮しつつリアルタイムに車両位置を道路上へ割り当てるマップマッチング手法を提案する。提案手法では道路網をネットワークに見立て、各道路をスコアリングすることで適切な経路を選択する。また、従来のマップマッチング手法では回り道や折り返しを正しくマッチングできなかったが、提案手法では走行区間の動的分割により精度良くマッチングを行う。実際の走行履歴を用いて提案手法を評価し、従来手法に比べて精度の向上を確認した。

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

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

In this paper, we propose a new map matching method for car navigation systems that predict user destination. This car navigation system automatically predicts user purpose and destination to present various information based on the predicted purpose without any user interaction. For accurate prediction, the system requires the route from start position to current position correctly. Therefore, we propose a dynamic division method that is an enhanced shortest path algorithm. Our method regards a road map as a network and intersections as network nodes, and road links between two intersections as network edges. This method divides trajectory with dynamic interval to avoid the incorrect matching derived from several patterns of running. In this paper, we

discuss a link scoring method and road dividing method to achieve accurate map matching. We confirmed the effectiveness of our method by comparing with conventional map matching 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ともない、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誘導することであるが、次世代のカーナビゲーションシステムは単なる道案内だけでなく、車内における活動全般を支援する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求められる。筆者らの研究チームでは、次世代カーナビゲーションシステムに要求される機能の1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、その予測結果をもと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、ユーザが目的を入力しなくても、システムが目的地や行動目的を自動的に推測し、目的地に関連する情報や効率的な誘導情報を自動的に提示するナビゲーションシステムを開発・運用している⁷⁾。

このシステムで用いられている目的地予測手法は、他の従来研究で用いられている手法⁵⁾と同様に、過去にユーザが通過した道路の連なりから目的地を予測するため、正確な予測には正しい道路通過情報が必要となる。しかし、走行履歴はGPSの測位誤差や、自律航法におけるセンサの累積誤差を原因とするノイズを含むため、正しい走行経路を得られなかった。さらに、既存のカーナビゲーションシステムにおいて道路上へと車両位置をマッチングするマップマッチング機能では、経路の遷移情報を考慮していないため、本来遷移できない道路間での遷移が発生し、目的地予測を正しく行うことができなかった。

そこで本研究では、経路の接続関係を考慮しつつリアルタイムにマップマッチングを行う手法を提案する。提案手法では、道路網をネットワークに見立て、各道路に走行軌跡との類似度に基づくスコアを付与することで適切なマップマッチングを行う。また、従来手法では回り道や折り返しを含む走行軌跡を正しくマッチングできないという問題があるため、提案

^{†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} 神戸大学工学研究科

Graduate School of Engineering, Kobe University

手法では回り道や折り返しが発生しないように動的に走行区間を分割する．提案手法により，リアルタイムに高精度な道路通過履歴を求められるようになる．

本論文の構成は以下のとおりである．2章では想定環境である目的予測型カーナビゲーションシステムについて述べ，3章で関連研究を紹介する．4章で提案手法を詳細に説明し，5章で評価実験および結果の考察を行う．最後に6章で本論文のまとめを行う．

2. 目的予測型カーナビゲーションシステム

筆者らの提案する目的予測型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，その予測結果をもとにユーザに適した情報を以下の例のように提示する⁷⁾．

- 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，その駅の時刻表を提示する．
- ガソリン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ともに，予測した目的地への走行経路沿いのガソリンスタンドを教える．

システムのプロトタイプを表示例を図1に示す．プロトタイプでは複数のキャラクタが目的地に応じた情報をアニメーション表示している．

2.1 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⁷⁾では，現走行における出発地か



図1 プロトタイプシステムの表示例

Fig.1 A screenshot of prototype system.

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， 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， $P_{l_0 j}$ 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．たとえば，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)を用いて図中の式のように計算する．

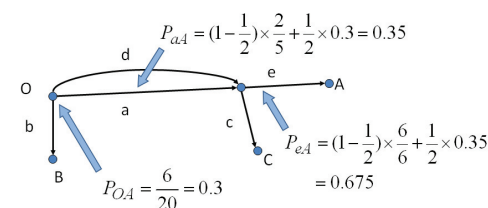


図2 目的地予測の例 ($\alpha = 0.5$)

Fig.2 An example of destination estimation.

表1 走行履歴の例

Table 1 An example of driving trajectory.

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

3. 関連研究

本研究で取り扱う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位置情報を道路上へマッチングする技術である。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だけでなく、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^{3),4)}などにおいても用いられる重要な技術である。特に、2章で述べた筆者らのシステムにおいてマップマッチングは欠かせない機能であり、また吉岡らの提案する目的地予測手法⁵⁾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、正しい目的地を予測するために必要不可欠である。

3.1 市販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

現在市販されているカーナビゲーションシステムのほとんどは、GPS と自律航法を用いて取得した現在地をもとにマップマッチングを行っている。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る手法である。自律航法だけでは時間経過とともに誤差が累積するため、GPS を用いて車両位置を修正し、誤差の累積を防いでいる。このようなマップマッチング手法は、現在地のマッチングに適しているため、カーナビゲーションシステムで広く採用されている。

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にジャンプする。たとえば図3(a)において、車両がY字路にさしかかり、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(図3(b))。しかし、車両位置が P_3 にさしかかり、類似する道路が候補にない場合、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が発生する(図3(c))。このような問題が発生すると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う手法が求め

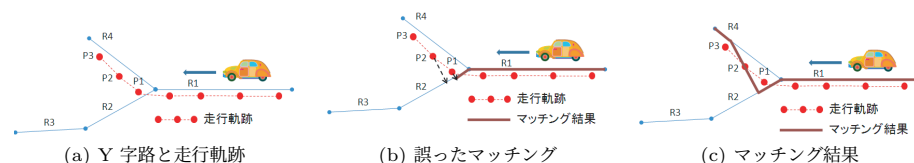


図3 マッチングミスの一例
Fig. 3 Examples of matching error.

られる。

3.2 静的な手法

静的な手法とは、運転後に得られる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う手法である⁸⁾。静的な手法の1つに、Brakatsoulas らが提案する、*Fréchet Distance* を用いたマップマッチング手法¹⁾がある。*Fréchet Distance* とは、一般に次の例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* と呼ぶ。この論文では、出発地点から到着地点へ至る様々な経路のうち、GPS を用いて取得した走行軌跡との *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。しかし、この手法を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、位置更新のたびに出発地点から現在地点までの経路を再計算する必要があり、処理コストが高くなる。

3.3 インクリメンタルな手法

インクリメンタルな手法とは、出発地点から現在地点へ至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⁶⁾。Brakatsoulas ら¹⁾やGreenfeld²⁾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う手法を提案している。彼らの手法の概要は次のとおりである。初めに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1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。この手法は、すべての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るという欠点がある。加えて、現在のカーナビゲーションのマップマッチング手法と同様に、1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接続する道路へマッチングしようとするため、マッチングミスを継続してしまうという問題もある。特に運転開始地点にて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

前章で述べたように、インクリメンタルな手法はリアルタイム処理に適する反面、マッチングミスが生じやすく、静的な手法は計算量が多くリアルタイム処理に向かないという問題

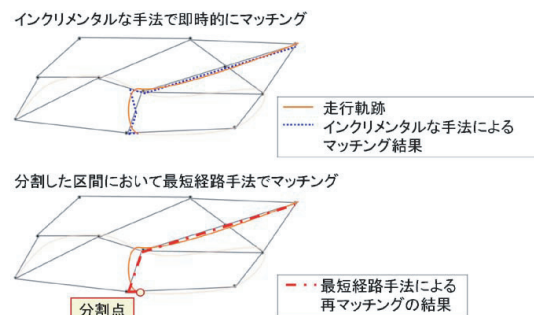


図 4 分割最短経路手法の処理の流れ
Fig. 4 The process of proposed method.

がある。そこで本研究では、両手法の欠点を解消する手法として、経路分割を行うことでインクリメンタルな手法と静的な手法の一種である最短経路手法を組み合わせ、さらに分割点を動的に決定することでマッチング精度を向上させる手法を提案する。

提案手法の動作は下記の手順を繰り返すことで行われる。

- (1) インクリメンタルな手法を用いてリアルタイムマッチングを行う。
- (2) ある区間走行すると、その区間に対して最短経路手法を適用する。
- (3) 区間の間の接続関係を修正する。

たとえば、図 4 上の実線のような走行軌跡が得られると、インクリメンタルな手法では接続関係を考慮していないため、図 4 上の点線のようなマッチング結果が得られる。そこで、走行中に走行軌跡を分割する点を設け、分割点と分割点の間の走行軌跡を最短経路手法を用いて再マッチングする。図 4 下では、始点から分割点までの走行軌跡を再マッチングすることで、破線のように接続関係を維持した経路が得られる。分割点での再マッチング後は、再びインクリメンタルな手法で次の分割が行われるまでは現在地を逐次更新し、新たな分割があると最短経路手法を用いて再マッチング処理を行う。一定区間ごとにマッチング結果を修正するため、その後のインクリメンタルな手法の精度は従来手法と比べて高くなる。また、静的な手法の適用は区間内に限られるため、静的な手法の計算量の大きさも問題とならない。

上に述べたように、提案手法はインクリメンタルな手法による精度の悪さを最短経路アルゴリズムによって補完する手法であるため、最短経路手法自体の精度および適切な経路分割方法が結果の精度に大きく影響する。そこで、以降では最短経路手法で用いるパラメータの

決定および経路分割を動的に行う方法について詳しく説明する。

4.1 最短経路手法の設計

ネットワークを構成するリンクにスコアを付加し、始点と終点を結ぶ経路上のリンクのスコアの総和が最小となる経路を導出するアルゴリズムを最短経路アルゴリズムと呼ぶ。本研究ではこの最短経路アルゴリズムにおけるリンクを道路としてマップマッチングを行う。各道路リンクのスコアリングに用いる指標としては下記のものが考えられる。

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リンクが選ばれる。関連研究 1) はこれに近い。

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リンクが選ばれる。

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リンクが選ばれる。関連研究 2) ではこのパラメータをもとに計算が行われている。

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。

ただし、距離には、走行軌跡を構成する測位点列のうち道路リンクに最も近い点との直線距離を、偏角には、道路リンクに最も近い点とその 1 つ前の点を結ぶ直線と道路リンクの始点と終点を結ぶ直線がなす角の大きさを用いる。

4.2 スコアリング手法決定のための実験

上記の各スコアリング手法を評価した。実験には国土地理院の発行する 1/2500 の数値地図を用い、GARMIN 社の GPS モジュール (Garmin GPS eTrex VISTA-C) を使用して取得した、2 ユーザの日常生活における車両での移動の記録 12 カ月分 (総走行回数: 約 200 回, 総走行時間: 約 60 h) の走行軌跡に対しマップマッチング処理を行った。1 走行あたりの時間を図 5 に示す。走行時間が 10 分から 15 分の走行が最も多いが、30 分超の長時間の走行も約 1 割ある。評価は、走行軌跡を最短経路手法を用いてマッチングした結果の道路リンク群 M と、正解ルートの道路リンク群 T を比較することで行った。なお正解ルートは手入力で作成した。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、

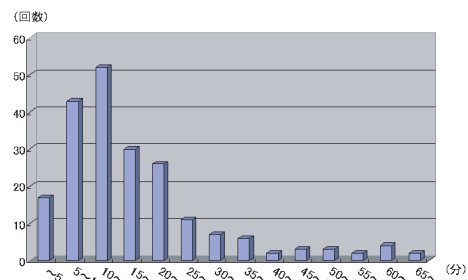


図 5 走行時間の分布

Fig. 5 Distribution of driving time.

表 2 スコアリングの評価結果

Table 2 Evaluation result.

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

下式で、一致率、過剰率、不足率を定義した。一致率は、正解ルートと、提案手法で得られた経路に含まれる道路リンクが一致している割合を示し、過剰率は、提案手法で得られた経路だけに含まれている余分な誤マッチングの割合、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道路リンクの割合を意味する。

$$\text{一致率} = \frac{N(M \cap T)}{N(M \cup T)}, \quad \text{過剰率} = \frac{N(M \cap \bar{T})}{N(M \cup T)}, \quad \text{不足率} = \frac{N(\bar{M} \cap T)}{N(M \cup T)}$$

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た場合が最も良い結果となることが分かる。

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で比較すると、距離のみを考慮した手法が最も一致率が高い。また、リンク長のみを考慮した手法の一致率が 6 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しかないことが分かる。さらに、偏角はマップマッチングのスコアリングには適していないことが分かる。

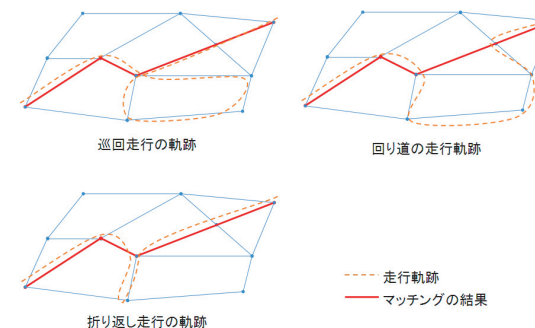


図 6 最短経路手法におけるマッチングミスの例

Fig. 6 Examples of matching error in simple method.

最短経路手法を用いることで、たいてい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ミスが多発した。詳細な調査により、マッチングミスは図 6 に示すような回り道や巡回、折り返しを含む走行パターンが原因となることが多いことが分かった。

4.3 分割方法の決定

最短経路手法を適用した場合、回り道や巡回、折り返しを含む走行軌跡においてマッチングミスが発生する。したがって、提案手法において経路分割を行っても、その分割経路内に上記の走行が含まれていた場合には精度が低下する。そこで、各分割経路においてこれらの状況が起こらないように経路の分割点を選択する手法を提案する。具体的には、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。たとえば図 7 の走行軌跡の場合、 C_1 から C_9 は、 C_0 から離れていくが、次の C_{10} の直線距離 $L(C_0, C_{10})$ は、 $L(C_0, C_9)$ よりも短いため、 C_{10} が新たな分割点となる。

作成した新たな分割点を新しい区間の開始地点 C_0 とし、以降はこの点からの直線距離の比較を行う。ただし、分割が長い間発生しない場合でも、区間の開始から T_{max} 秒以上経過した地点を新たな分割点とする。また、予備実験により、短い時間で分割を繰り返すとインクリメンタルな手法に近くなるためマッチング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、区間あ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。

提案する走行軌跡の分割手法を 4.2 節で示したマッチングミスが発生する走行パターンに

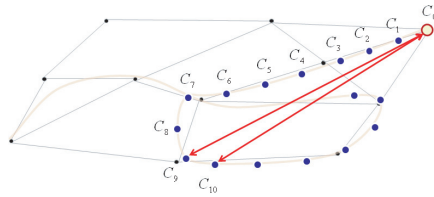


図 7 動的分割の例
Fig. 7 Dynamic division.

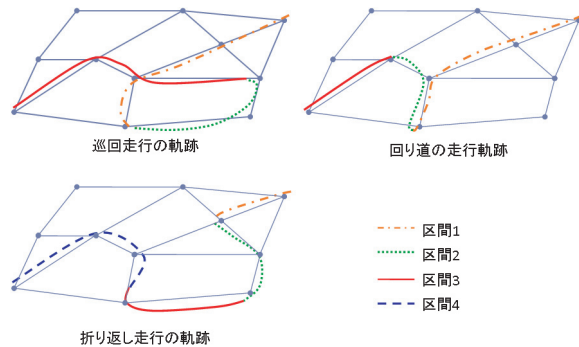


図 8 マッチングミスが発生する走行パターンの分割例
Fig. 8 Division patterns against matching errors.

適用した例を図 8 に示す。いずれの走行パターンも、3 つないし 4 つの区間に分割され、分割された各区間は、4.2 節で示したマッチングミスが発生する走行パターンに該当しない。このように分割処理を行うことで、静的なマップマッチング手法が苦手とする走行パターンを、正しくマッチングできる走行パターンの集合へ変換できる。

4.4 分割点での再マッチング処理

最短経路手法は、始点の道路リンクと終点の道路リンクを結ぶ経路の中で、走行軌跡に近い経路を探す手法である。そのためマッチング開始時には始点・終点の道路リンクを指定する必要があるが、単純に始点・終点に最も近い道路リンクを採用すると、インクリメンタルな手法においてマッチングミスが発生していた場合にそのリンクをそのまま採用してしまう可能性がある。分割最短経路手法では、分割した区間ごとに始点・終点の道路リンクを与える必要があるため、単純に分割点に最も近い道路リンクを選択するだけではマッチング精度

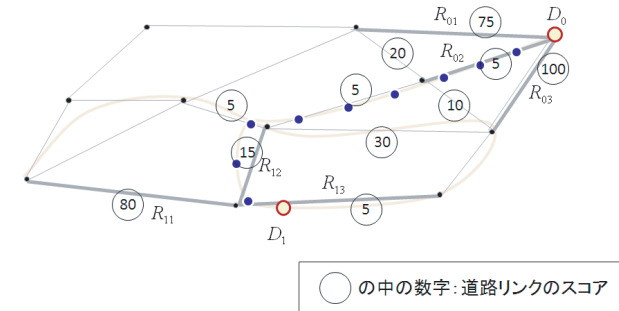


図 9 区間 1 の周辺リンクへのスコアリング
Fig. 9 Scores around division 1.

が大幅に低下する可能性がある。そこで提案手法では、区間の始点・終点となる分割点において、マッチングする道路リンクの候補を複数個保持し、新たな分割経路が評価される際に再マッチング処理を行うことで上記の問題を回避する。

手法の詳細は次のとおりである。はじめに、前節で述べた分割点の選択手法を用いて、新たに走行軌跡を分割する点 D_i が選択された場合、 D_i から半径 r 以内にある N 個の道路リンク $R_{i1} \sim R_{iN}$ を選択する。これらの道路リンクは分割点におけるマッチング対象リンクとなる。分割点により新たに作成された区間の始点は $D_{(i-1)}$ であり、この区間の始点リンクの候補となる道路リンクは、 $D_{(i-1)}$ の周囲の道路リンクである $R_{(i-1)1} \sim R_{(i-1)N}$ となる。 $N = 3$ の場合の例を図 9 に示す。図では、区間 1 の始点である D_0 の周囲にある道路リンクを 3 本 (R_{01} , R_{02} , R_{03}) 選択し、同様に、終点である D_1 の周囲にある道路リンク R_{11} , R_{12} , R_{13} を選択している。

次に提案手法では、最短経路手法を用いて始点リンクを $R_{(i-1)k}$ 、終点リンクを R_{il} とした場合の、最短経路手法により得られる経路のスコア $S(R_{(i-1)k}, R_{il})$ を、すべての k と l ($k = 1, 2, \dots, N$, $l = 1, 2, \dots, N$) について求める。

たとえば、図 9 のようにスコアリングされている区間において、始点リンクを R_{01} 、終点リンクを R_{11} のときの最短経路手法により得られる経路のスコアは 180 となる ($S(R_{01}, R_{11}) = 180$)。同様に、すべての始点リンク・終点リンクの組合せのスコアを求め、表 3 に示すような、 $D_0 \cdot D_1$ 間のスコア表を作成する。

次に、総スコア $TS(R_{il})$ を下記のように定義する。

$$TS(R_{il}) = \min(TS(R_{(i-1)k}) + S(R_{(i-1)k}, R_{il})) \quad (2)$$

表 3 $D_0 \cdot D_1$ 間のスコア表
Table 3 Scores between D_0
and D_1 .

	R_{11}	R_{12}	R_{13}
R_{01}	180	100	105
R_{02}	105	25	30
R_{03}	205	125	130

表 6 D_2 の総合スコア
Table 6 Scores for D_2 .

	スコア	リンク
$TS(R_{21})$	80	R_{12}
$TS(R_{22})$	75	R_{12}
$TS(R_{23})$	160	R_{12}

表 4 D_1 の総合スコア
Table 4 Scores for D_1 .

	スコア	リンク
$TS(R_{11})$	105	R_{02}
$TS(R_{12})$	25	R_{02}
$TS(R_{13})$	30	R_{02}

表 7 $D_2 \cdot D_3$ 間のスコア表
Table 7 Scores between D_2
and D_3 .

	R_{31}	R_{32}	R_{33}
R_{21}	50	60	150
R_{22}	60	70	160
R_{23}	110	120	190

表 5 $D_1 \cdot D_2$ 間のスコア表
Table 5 Scores between D_1
and D_2 .

	R_{21}	R_{22}	R_{23}
R_{11}	100	95	180
R_{12}	55	50	135
R_{13}	50	45	130

表 8 D_3 の総合スコア
Table 8 Scores for D_3 .

	スコア	リンク
$TS(R_{31})$	130	R_{21}
$TS(R_{32})$	140	R_{21}
$TS(R_{33})$	310	R_{21}

$$(k = 1, 2..N, l = 1, 2..N)$$

総合スコアとは、走行開始地点 D_0 から D_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。 $TS(R_{(i-1)k})$ は、1 点前の分割点 $D_{(i-1)}$ の周囲の道路リンク $R_{(i-1)k}$ における総合スコアであり、分割点 $D_{(i-1)}$ が選択されたときに、すでに算出されている。ただし、走行開始地点 D_0 における総合スコア $TS(R_{0k})$ はすべて 0 とする。

以上の手法により、 D_i の周囲の道路リンク群 R_{il} の総合スコア $TS(R_{il})$ を算出し、総合スコアの最も小さい R_{il} を現在地 D_i を表す道路リンクであるとす。

$$D_i = \min(TS(R_{il})) \quad (l = 1, 2..N) \quad (3)$$

具体的に説明すると、図 9 において、 D_0 は運転開始の点であるため、 $TS(R_{01})$, $TS(R_{02})$, $TS(R_{03})$ はいずれも 0 である。次に D_1 の周辺の道路リンクの総合スコアを求める。たとえば $TS(R_{11})$ は、式 (2) より

$$TS(R_{01}) + S(R_{01}, R_{11}) = 0 + 180 = 180$$

$$TS(R_{02}) + S(R_{02}, R_{11}) = 0 + 105 = 105$$

$$TS(R_{03}) + S(R_{03}, R_{11}) = 0 + 205 = 205$$

の中で、値が最小である R_{02} が R_{11} に接続する道路リンクとなり、また $TS(R_{11})$ は 105 となる。同様の処理を行った結果、表 4 に示すような結果が得られ、この中で最もスコアが小さくなる R_{12} が D_1 をマッチングする道路リンクとなる。

さらに車両が走行し新たな分割点 D_2 が現れると上記の処理を再び行い、 $D_1 \cdot D_2$ 間のス

コア表を作成する。このスコア表と、 D_1 における総合スコア (表 4) を用いて、式 (2) から、 D_2 における総合スコアと接続する道路リンクを決定する。この結果、誤った道路リンクである R_{22} がマッチングされる。

しかし、その後運転終了地点 D_3 に到達し、 D_2 から D_3 までの区間の再マッチングが行われる際にこのマッチングミスは訂正される。具体的には、表 5 に示すスコア表が得られる場合、式 (2) を求めると表 8 となる。その結果、 D_3 は総合スコアの最も小さい R_{31} 上にマッチングされ、さらに、接続する道路リンクを確認することで、 D_2 は R_{21} 上にマッチングされる。同様の処理により、 $R_{31} \rightarrow R_{21} \rightarrow R_{12} \rightarrow R_{02}$ と運転開始地点までさかのぼって再マッチングが行われる。さかのぼる処理は分割点のみで行われ、さかのぼる段数もただか分割点の数だけであるため、この処理による計算負荷は小さい。

5. 評価と考察

3 章の実験で用いたものと同じ走行ログを用いて提案手法を評価した。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 章で述べた最短経路手法、および走行データを区切って最短経路手法を適用する提案手法である。ただし、提案手法における動的分割の効果を測るため、一定間隔 (9 分, 7 分, 5 分, 3 分, 1 分) で分割するものや、区間あたりの時間の下限値 T_{min} と上限値 T_{max} を変化させたものも評価した。また、提案手法における即時的なマッチング処理には Brakatsoulas らの手法を用いた。

最短経路手法および分割最短経路手法で用いたスコアリング手法は、距離とリンク長の積を採用した。また、分割点 D_i において抽出する道路リンクは、予備実験の結果から、点 D_i から半径 100m 以内にある D_i に近い 5 個の道路リンクとした。すなわち 4 章における変数をそれぞれ $r = 100$, $N = 5$ と設定した。

5.1 運転終了時のマッチング結果の評価

目的予測型カーナビゲーションシステムは、過去の走行履歴を学習し、学習した走行履歴をもとに、現走行におけるユーザの目的地を推測する。この学習には運転終了時に得られる走行経路を用いるため、高い精度でのマッチングを要求される。そこで本節では、運転終了時に得られる走行経路と正解経路を比較した。評価指標には 4 章で定義した一致率・過剰率・不足率を用いた。

全走行の平均の一致率・過剰率・不足率を計算した結果を表 9 に示す。この結果より、提案手法はインクリメンタルな手法よりも高い精度でマッチングできていることが分かる。インクリメンタルな手法は、最も走行している可能性の高い道路へ逐次マッチングする手法

Table 9 Evaluation results.

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	80.23	15.25	4.53
最短経路手法	89.28	3.13	7.59
分割最短経路手法 一定間隔：9分	92.03	3.07	4.90
：7分	91.96	3.16	4.88
：5分	93.58	3.00	3.42
：3分	93.86	3.22	2.92
：1分	93.30	4.00	2.70
動的分割：（上限値，下限値）			
下限値固定：（9分，1分）	94.06	3.58	2.36
：（7分，1分）	94.18	3.51	2.31
：（5分，1分）	94.29	3.49	2.22
：（3分，1分）	93.85	3.60	2.54
上限値固定：（9分，7分）	92.62	3.18	4.19
：（9分，5分）	93.64	3.29	3.08
：（9分，3分）	93.96	3.38	2.66
：（9分，1分）	94.06	3.58	2.36
：（9分，なし）	93.72	3.90	2.38

であるため、正しい道路を選択することが多く不足率は低い値を示している。一方で、過剰率が高いことから、正解経路以外の道路上へも多数マッチングされていることが分かる。また、最短経路手法のみを用いた際の不足率が7.6%と、インクリメンタルな手法よりも悪いことから、この手法が回り道などを考慮できていないことが分かる。

提案手法において一定間隔で分割点を決定した場合は、分割間隔が狭いほど過剰率が上昇していることから、分割間隔が狭い場合は正解経路から外れた道路へマッチングされやすいという性質を持つことが分かる。逆に、分割間隔が広いほど不足率が上昇していることから、分割間隔を広く設定すると、最短経路手法において問題となっていた回り道や巡回・折り返し走行を含む走行軌跡に対して正しくマッチングできていないことが分かる。また、一定間隔で経路を分割する場合は、3~5 分ごとに分割点を設けることが適切であることが分かる。

提案手法において分割間隔の下限値を 1 分とした場合、上限値にかかわらず高い一致率を示している。これは、動的に分割を行う提案アルゴリズムが有効に働き、回り道や巡回・折り返し走行などの走行軌跡を検出し、柔軟に分割しているためであると考えられる。一方、分割の上限値を 9 分とし、下限値を変化させた場合、下限値が大きくなるほど一致率が低下していることから、分割を行うべき走行軌跡が短時間のうちに複数回発生している

Table 10 Relationship between concordance and max/min threshold.

		上 限 值				
		9 分	7 分	5 分	3 分	1 分
下 限 值	9 分	92.03				
	7 分	92.62	91.96			
	5 分	93.64	93.93	93.58		
	3 分	93.96	94.18	94.48	93.86	
	1 分	94.06	94.18	94.29	93.85	93.30

と予測できる。ただし、下限値を設定しない、すなわち、回り道検出アルゴリズムが反応したすべての点を分割する場合には一致率が低下していることから、下限値をなしにすると、小刻みに分割点が生成され、その結果、インクリメンタルな手法と同様に、個々の測位点が有するノイズに強く影響されてしまうことが分かる。

動的分割を行う手法において、適切な上限値と下限値を求めるために、上限値を9分・7分・5分・3分・1分、下限値を9分・7分・5分・3分・1分と変化させた場合のすべての組合せについて一致率を評価した。結果を表10に示す。ただし、一定間隔で分割した結果を、上限値と下限値が等しい欄に記す。表より、上限値が5分の場合の一致率が他の条件の場合よりも一致率が高いことから、動的に分割する点を選択する場合、5分以上分割が発生しない区間でも、一度走行軌跡を分割した方が、より正解経路に近い経路へマッチングできることが分かる。

これらの評価結果より、提案手法はインクリメンタルな手法と比べ、マッチング精度が大幅に向上していることが分かる。提案手法の動的な分割における、下限値・上限値を比べると、一致率の結果と同様、上限値を 5 分と設定した場合が他の条件よりも良い結果が得られることが分かる。また、上限値にかかわらず、下限値を 1 分や 3 分とした場合が他の条件よりも精度良くマッチングできている。

5.2 個々の走行経路の評価

実際の走行軌跡とマッチング結果の例をあげる．図 10 に，最短経路手法では正しくマッチングできなかった巡回と回り道を有する走行軌跡を，提案手法を用いてマッチングした結果を示す．なお，正解経路はマッチング結果と重なっているため，図中に表示していない．図より，八の字に巡回している部分まで正確にマッチングできており，また，回り道走行の部分も，最短経路の道路にマッチングされることなく，正しい道路にマッチングされていることが分かる．また，図 11 は，最短経路手法では正しくマッチングできなかった折り返

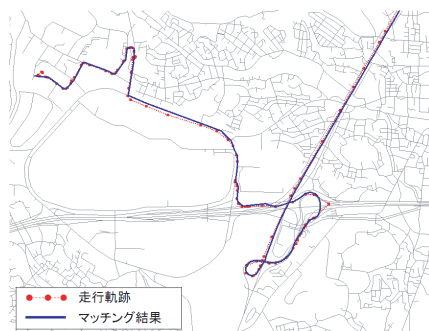


図 10 巡回と回り道がある場合のマッチング
Fig. 10 Matching result (1).

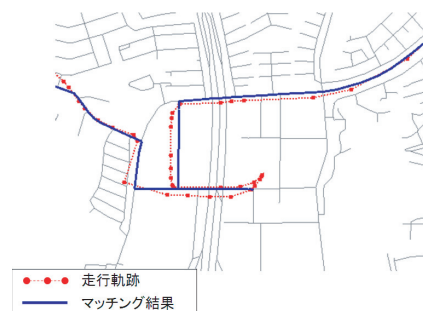


図 11 折り返しがある場合のマッチング
Fig. 11 Matching result (2).



図 12 分割点の選択の様子
Fig. 12 Division points.

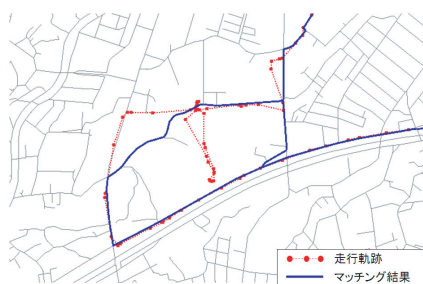


図 13 正しいマッチングができない例
Fig. 13 An example of matching error.

しを有する走行軌跡を、提案手法を用いて正確にマッチングした例である。

図 12 は、動的分割手法において分割点に選ばれた点を地図上に太丸で描画したものである。図からも分かるとおり、動的分割手法が対象としている、回り道を検出し分割点として選択できていることが分かる。

最後にマッチングミスの例として図 13 をあげる。この例は、地図に登録されていない道路を走行した際のマッチング結果である。このような場合、道路上へマッチングすることを前提とした提案手法では対処できず、図の例では右側を迂回したような経路にマッチングされている。

表 11 平均マッチング精度
Table 11 Accuracy.

		上限値				
		9 分	7 分	5 分	3 分	1 分
下限値	9 分	78.53				
	7 分	79.40	80.36			
	5 分	80.93	81.25	83.12		
	3 分	84.77	84.87	85.83	87.25	
	1 分	88.80	89.01	89.19	90.15	93.05

5.3 走行中に得られるマッチング結果の評価

目的予測型カーナビゲーションシステムにおける目的予測は現在地の影響が大きいため、現在地が正しくマッチングされることが重要となる。そこで、毎秒ごとに、正しい経路上に各点がどの程度マッチングされているのかを下記の式に基づき評価した。

$$(\text{ある時刻の}) \text{ マッチング精度} = \frac{\text{正解経路上の測位点数}}{(\text{ある時刻までの}) \text{ すべての測位点数}}$$

各走行における上式の平均値を求め、さらに全走行に対してそれらの平均を算出した結果を表 11 に示す。表より、動的分割における下限値および上限値が小さいほど平均マッチング精度が高いことが分かる。また、1 分間ごとに一定間隔で分割した場合が最も平均マッチング精度が良い。これは他の手法よりも分割間隔が狭く、分割回数が多いためである。特に、運転開始直後で正解経路上にマッチングされていない場合、分割間隔が狭い方が早く再マッチング処理を行うことができ、さらに、運転開始直後はすべての測位点の数が少なく各点が正解経路上か否かがマッチング精度に与える影響が大きいため、このような結果となったと考えられる。しかしこれまでの評価から明らかなように、経路全体で見ると短すぎる分割間隔は悪影響が大きいため、経路全体での精度を下げない範囲で分割の下限値を下げるのが有効である。

5.4 実験データに関する考察

本実験の走行履歴は 2 ユーザ分と少ないものであるが、一般にマップマッチングミスが発生しやすい都心部や住宅地といった道路が密な地域の走行が多数であることや、図 5 に示すように、走行時間のばらつきが大きく多様な走行の記録を用いたこと、約 200 走行あることより、走行データの数や種類は十分であるといえる。

また、たとえば交差点ごとに右左折する経路はあま選択しないなど、運転者の経路選択の特性を用いたマッチングを行っていないため、他のユーザの走行履歴に提案手法を適用した

場合も本評価結果と同様の結果になると予想できる。

5.5 プローブシステムにおけるマップマッチング手法との比較

本手法と類似するマップマッチング手法に、移動体の位置情報をプローブサーバに収集し、交通流予測や渋滞情報を推測するプローブシステムにおける、サーバ上でのマッチング手法があげられる。プローブシステムは、通信の制約により短くても 3~5 分間隔で情報をアップロードするため、マップマッチングもそのタイミングで行われる。しかし、本研究の前提である目的予測カーナビにおいては、カーナビにて目的地予測を行うため、逐次マッチングすることが求められ、プローブシステムにおけるマップマッチング手法をそのままでは適用できない。一方、本研究における「分割」をプローブサーバへのアップロードのタイミングと見立てると、本研究の成果をプローブシステムへ応用できる。プローブサーバでは一般に、本手法の一定間隔で分割する手法に類似する経路探索アルゴリズムを用いて、定期的にアップロードされた位置情報間の経路を探索するため、4.2 節で議論した走行パターンにおいては、本研究と同様のマッチングミスが起こると考えられる。そこで、本手法における、動的な分割が発生するタイミングで移動体からプローブ情報をアップロードすることで、プローブサーバ上でのマップマッチングの精度が向上すると考えられる。

5.6 計算量についての考察

提案手法の処理はインクリメンタルな手法でリアルタイムに処理をするフェーズと、分割点が発生した場合に行う、最短経路手法を用いた再マッチングのフェーズに分けられる。インクリメンタルな手法は逐次処理に適した手法で、計算量は小さい。分割点での処理において、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 N 、分割した区間内に含まれる GPS の計測点数を g とすると、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 $O(\log m)$ 。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。また、最短経路手法は N^2 回実行されるため $O(N^2 m \log m)$ を要する。ただし、評価環境において $N = 5$ としたように、 N^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用するため、 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 ($r = 100$) において、ある走行記録を 5 分ごとに一定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分かる。

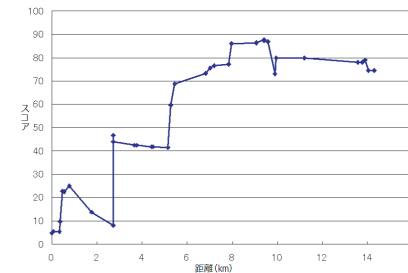


図 14 マッチング前のデータを用いた予測結果
Fig. 14 Destination prediction in conventional method.

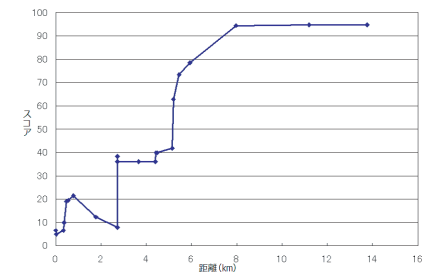


図 15 マッチング後のデータを用いた予測結果
Fig. 15 Destination prediction in proposed method.

5.7 目的予測型カーナビゲーションへの適応結果

提案手法でマップマッチングした走行軌跡を用いて目的地予測を行った。1 人のユーザの走行履歴 1 年分を利用し、ランダムに選択した半分を学習用データ、残りを評価用データとして実験を行った。

まず、ある走行のマッチング前の走行軌跡とマッチング後の走行軌跡を用いて目的地予測した結果を図 14、図 15 に示す。縦軸は、ある候補地へ向かうとシステムが予測する度合いを表し、高いほどその目的地へ向かっているとシステムが認識していることを意味する。スコアの算出法には式 (1) を用いた。また、横軸は走行距離を意味する。グラフは、多数の目的地候補のうち、この走行における正解目的地に対する予測結果を示しており、走行開始時は手がかりとなる情報が少なくスコアは低い値を示すが、走行を続けるにつれスコアが高くなっていることが分かる。この例では、両者とも最終的には正しい目的地を予測しているが、図 14 では、走行距離 10 km 付近や 14 km 付近でスコアが低下している。これらは、目的予測手法に起因する予測ミスではなく、走行した道路とは異なる道路へマッチングされたことを原因とするスコアの低下である。一方、提案手法を用いた目的地予測では、図 15 に示すとおり上述のスコアの低下が減少している。

次に、評価実験に用いた全走行において、先ほど問題にした正しい目的地のスコアが低下する回数をカウントした。マップマッチング前のデータを用いた場合、スコアの低下は 393 回（うちスコアが 10 以上低下する回数は 95 回）、マッチング後のデータを用いた場合、スコアの低下は 366 回（うちスコアが 10 以上低下する回数は 85 回）であった。

さらに、正しい目的地へのスコアが上位にあるか否かを確認する。著者らが提案する目的

予型測カーナビゲーションシステムでは、つねに正しい目的地を予測することは困難であるという認識から、ある候補地のスコアが上位4に含まれる場合、その候補地に関する情報を提示するように実装されている。そこで、総走行距離のうち正しい目的地が上位4に含まれている区間の割合を確認したところ、マッチング前のデータでは90.1%、マッチング後のデータでは90.3%であった。

この評価実験より、提案するマップマッチング手法を用いて目的地を予測することで、正しい目的地に対するスコアが上位以内に入る割合はほとんど変化しなかったものの、正しい目的地に対するスコアが途中で低下する回数が減少していることが確認できた。このことは、提案するマップマッチング手法が目的地予測システムの情報提示において有効であることを示している。一方、提案するマップマッチング手法を用いることで正しい目的地に対するスコアが低下する回数を減らすことはできたが、依然、スコアの低下が366回発生していることから、マップマッチング以外の部分での予測精度の向上が求められる。現在、著者らの研究グループでは、上記の問題に対し、運転者の状況（日時・乗員数・天気など）を用いることを検討している。

6. ま と め

本研究では、目的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。提案手法は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できる。評価結果より、提案手法は従来手法よりも精度良くマッチングでき、適切なパラメータを設定することで、インクリメンタルな手法を適用時に生じたマッチングミス正しい経路上へ修正し、また、過去の走行軌跡もあわせて低コストで再マッチングできていることが確認できた。今後の課題としては、インクリメンタルな手法の特性を利用した分割点の決定があげられる。

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)および特定領域研究(19024046)の研究助成によるものである。ここに記して謝意を表す。

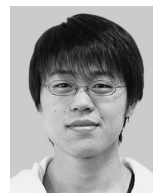
参 考 文 献

- 1) Brakatsoulas, S., Pfoer, D., Salas, R. and Wenk, C.: On Map-Matching Vehicle Tracking Data, *Proc. 31st Very Large Data Base Conference (VLDB)*, pp.853–864 (2005).
- 2) Greenfield, J.: Matching GPS Observations to Locations on a Digital Map, *Proc. 81st Annual Meeting of the Transportation Research Board* (Jan. 2002).

- 3) 飯田祐三, 岩辺路由, 菊池 輝, 北村隆一, 佐々木邦明, 白水靖郎, 中川 大, 波床正敏, 藤井 聡, 森川高行, 山本俊行: マイクロシミュレーションアプローチによる都市交通計画のための交通需要予測システムの提案, *土木計画学研究・論文集*, No.22, pp.853–856 (1999).
- 4) 石川 亮, 風間 洋, 本多中二, 板倉直明, 猪飼國夫: 交差点走行車両の運転動作のモデル化と交通解析, *情報処理学会論文誌*, Vol.44, No.14, pp.71–80 (2003).
- 5) Kostov, V., Ozawa, J., Yoshioka, M. and Kudoh, T.: Travel destination prediction using frequent crossing pattern from driving history, *Proc. IEEE Intelligent Transportation Systems*, Vol.13, pp.343–350 (2005).
- 6) Quddus, M.A. and Ochieng, W.Y.: A general map matching algorithm for transport telematics applications, *GPS Solutions*, Vol.7, No.3, pp.157–167 (2004).
- 7) Terada, T., Miyamae, M., Kishino, Y., Tanaka, K., Nishio, S., Nakagawa, T. and Yamaguchi, Y.: Design of a Car Navigation System that Predicts User Destination, *Proc. 1st Workshop on Tools and Applications on Mobile Contents (TAMC)*, pp.54–59 (May 2006).
- 8) Yin, H. and Wolfson, O.: A weight-based map matching method in moving objects databases, *Proc. IEEE Conference on Scientific and Statistical Database Management*, pp.437–438 (2004).

(平成 20 年 4 月 3 日受付)

(平成 20 年 10 月 7 日採録)



宮下 浩一

2006 年大阪大学工学部電子情報エネルギー工学科卒業。2008 年同大学院情報科学研究科マルチメディア工学専攻博士前期課程修了。現在、株式会社三菱総合研究所に在籍。ユビキタスコンピューティング、ITS に関する研究に従事。



寺田 努（正会員）

1997 年大阪大学工学部情報システム工学科卒業。2000 年同大学院工学研究科博士後期課程退学。同年より大阪大学サイバーメディアセンター助手。2005 年より同講師。2007 年神戸大学大学院工学研究科准教授。現在に至る。2004 年より特定非営利活動法人ウェアラブルコンピュータ研究開発機構理事。2004 年には英国ランカスター大学客員研究員を兼務。博士（工学）。ウェアラブル・ユビキタスコンピューティングの研究に従事。IEEE，電子情報通信学会，日本データベース学会，ヒューマンインタフェース学会の各会員。



田中 宏平

2005 年大阪大学工学部電子情報エネルギー工学科情報システム工学科目卒業。2007 年同大学院情報科学研究科マルチメディア工学専攻博士前期課程修了。現在，同専攻博士後期課程に在籍。ウェアラブルコンピューティング，ユビキタスコンピューティングに興味を持つ。



西尾章治郎（正会員）

1975 年京都大学工学部数理工学科卒業。1980 年同大学院工学研究科博士後期課程修了。工学博士。京都大学工学部助手，大阪大学基礎工学部および情報処理教育センター助教授，同大学大学院工学研究科教授を経て，2002 年より大阪大学大学院情報科学研究科教授となり，現在に至る。2000 年より大阪大学サイバーメディアセンター長，2003 年より大阪大学大学院情報科学研究科長，その後 2007 年より大阪大学理事・副学長に就任。この間，カナダ・ウォータールー大学，ピクトリア大学客員。データベース，マルチメディアシステムの研究に従事。本会理事を歴任。ACM，IEEE 等 8 学会の各会員。