経路履歴抽出の為のオフラインマップマッチング方式

路 琳　藤野　和久　大沢　裕　曽根原登

Off-line Map Matching Method for Route History Acquisition

Lin LU，Kazuhisa FUJINO，　Yutaka OHSAWA，Noboru SONEHARA

Abstract: Moving object monitoring have been studied actively for two decades. When we deal with a trajectory of a car moving along road, the trajectory has to be related on road. This process is called map-matching. For the evaluation of the matching measure in previous studies, evaluation measure composed mixture distance and angle is apt to be used. However, these distance and angle are different nature, then it deteriorates versatility of map-matching. This paper proposes a versatile map-matching method by dividing the procedure into several steps which applies different matching measure in series.

Keywords: マップマッチング(Map matching), 移動体(moving object), リアルタイムモニタリング(real-time monitoring)

1. はじめに

GPSなどの測位装置を用いて取得された移動体位置を道路網上に対応付ける処理は，マップマッチングと呼ばれ，従来よりカーナビにおける主要技術として研究されてきた(Quddus et al. 2003)．筆者らは，移動体位置の実時間モニタリングのために，個々の移動体の良く通るルート情報を用いたモニタリング方式を提案している(藤野ほか 2010)．この際にもマップマッチングは重要な要素技術となっている．

マップマッチング方式は，それが利用される環境に応じて２つのカテゴリに分類することができる．まず，ある旅行の経路履歴があり，そのデータを道路上に対応付けるマップマッチング(map matching: MM)があり，これをオフラインMMと呼ぶ(Wenk et al. 2006)．一方，カーナビではまさに現在移動している位置を道路網上に対応付ける必要がある．これをオンラインMMと呼ぶ(Civilis et al. 2005)．取得されるデータ点が密度高く(例えば1秒毎) 存在するか，疎(例えば数分毎)であるかによっても最適な方式は異なる．更に移動予定経路が与えられ，現在点がその経路上に存在するか，あるいは経路から外れたかを判定すれば十分な場合もある．例えば，カーナビで目的地を指定することにより予定経路が決定されている場合である．また，GPSによる測位データのみが与えられる場合と，車速，方向，回転角度などが与えられる場合にも最適な方式は異なる．本稿では，これらの組み合わせの内，1秒間隔で取得されるGPSの測位データのみを用い，予定経路は与えられないオフラインMM方式について述べる．

大沢：〒338-8570 埼玉県さいたまま市桜区
下大久保 225　埼玉大学大学院理工学研究科
TEL(FAX):048-858-3717
email:ohsawa@mail.saitama-u.ac.jp

２．提案方式

従来方式の多くは，マップマッチングする経路の尤もらしさを評価する基準として，マッチ経路とGPS軌跡の方向性の類似，マッチ位置とGPS取得位置との距離の二つを同時に用いていた．この二つの評価は全く異なる単位である為，これらを同時に利用し，尤もらしさを定めるには多くのパラメータによって調整する必要がある．また，このパラメータはGPS装置の精度や，地図データの種類など，実行環境によって大きく左右さ

れる．そのため，高い汎用性を得にくい．そこで，本節では，マップマッチングの手順を複数に分解し，距離と方向を別の手順において考慮することで，設定すべきパラメータの少なく，汎用性の高い提案方式について説明する．

Algorithm 1 mapmatchin

Require: $gpsPoints$,ϵ
Ensure: $matchedLine$
1: $cand \Leftarrow$getCandidateLines($gpsPoints$,ϵ)
2: $cand \Leftarrow$fillBlank($cand$)
3: $adjustedPoints \Leftarrow$adjustPoint($gpsPoints$)
4: $landmark \Leftarrow$makeLandmark($adjustedPoints$)
5: $pq \Leftarrow \phi$
6: **for each**node n near landmark[0] **do**
7: $pq.Add(\langle n.id, 0, \phi, null\rangle)$#(ID,cost,visited,path)
8: **end for**
9: **while** pq isn't empty **do**
10: $now \leftarrow pq.deque()$
11: **if** $now.path$ near from $landmark[now.lid]$ **then**
12: $now.lid \leftarrow now.lid + 1$
13: **if** now visited all landmark **then**
14: return All passed path
15: **end if**
16: **end if**
17: **if** $now.id$ was visited by same pattern **then**
18: continue
19: **end if**
20: **for each** segment s in $cand$ adjacent to $now.id$ **do**
21: $pq.Add(\langle s'$s next node id, $now.cost+s.cost, now.visited, s\rangle)$
22: **end for**
23: **end while**

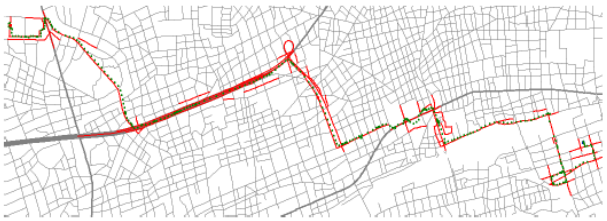
このアルゴリズムでは入力として移動を記録したGPS軌跡と，地図データとGPS装置の持つ精度から考えられる誤差(ε)を受け取る．また，出力は$gpsPoints$を地図データ上の経路データに置き換えた道路セグメントの群を返す．

提案方式で行う操作は大きく分けて4つである．まず始めに，マップマッチング対象となる道路セグメントの絞込みを行う(アルゴリズム中のgetCandidate-Lines)．絞込みでは，GPS点を取得時間順に二点ずつ利用し，現在注目しているGPS点からの距離がε以内である全ての道路セグメントの中で，二点の成す方向と類似した道路セグメントのみを候補として保存していく．この操作をGPS軌跡の始点から終点までの組で繰り返し行う．この操作によってマップマッチングすべき道路セグメントが絞られ，ミスマッチする確率が下がる．また，候補を絞り込んだ際に道路セグメントごとに，近いGPS点との距離の平均と道路セグメント長の積をコストとして割り振っておく．

次の段階の処理として，先ほど求められたマッチング候補の補間を行う(*fillBlank*)．補間は，GPS装置が閉鎖された空間(例えばトンネル）において，現在位置を取得できないことや，数秒間に一度現在位置を取得している場合，GPS点間隔が広がってしまうため，一つ目の操作で求まった道路セグメントが始点から終点まで繋がらなくなってしまうことから必要になる．候補の補間操作ではまず，交差点を一つの要素と見て，道路セグメントで繋がっている交差点を全て同じ集合とする．次に，この集合の数が一つになるまで，集合同士の間の経路を最短経路で補間し，１つの集合を形成する．最短経路で補間する理由としては，例えばトンネルにおいて，移動体がトンネルに入った位置と抜けた位置がわかっているため，この間の経路は一般的に一意に定まると考察できるからである．また，移動速度が速く，GPS間隔が広い為に繋がらなくなる場合は，移動速度が速いことから，右左折を繰り返していることは考えにくく，道路に沿って直進したと考えられる為である．また，ここでマッチング候補に追加された道路セグメントは，道路セグメント長をそのままマッチングした際のコストとしてみる．

3段階目の操作として，GPS軌跡における特徴点の抽出を行う(アルゴリズム中の*makeLandMark*)．本方式における特徴点とは，移動体が右左折を行ったであろうGPS点と，移動開始と停止点のことを指す．本方式では，移動経路がどのような順番で経路を通ったかを求めるために必要となる．特に，経路中に存在する閉路や，コンビニやレストランといったPOI（point of interest）に立ち寄った際の折り返し経路が存在する場合に必要となる．この特徴点の判断基準としては，GPS取得時間順に三つずつのGPS点の成す角度が，ある閾値を越えた際に，その中央の点を右左折した点として判断する．また，この特徴点に大きな誤差が乗っている場合マップマッチングに大きく影響するため，特徴点候補となった点のε/2の範囲にマッチング候補である道路セグメントがあるかを確認し，なければ周辺にマッチング候補のある特徴点を追加する．ただし，特徴点を抽出する前に，移動体の停止時に取得された

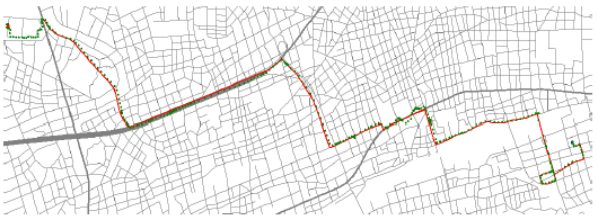
GPS点を取り除く必要がある(アルゴリズム中の *adjustPoint*)．停止時に取得された点はGPSの含む誤差による移動軌跡のふらつきによって様々な角度が存在しており，右左折と誤認識してしまう為である．そこで，密集したGPS点を取り除く為，GPS点の間隔が一定間隔(*error*)以上になるようにGPS 点を間引き，その後特徴点抽出を行う．



(a) 候補セグメントの選択

(b) 候補セグメントの補間

(c) 特徴点の抽出



(d) マッチング結果

図１　マップマッチング処理の流れ

最後の操作として，先ほど求めた特徴点を順番どおりにコスト最小となるように通過する経路を求める(アルゴリズム中の5～23行目)．マッチングでは，マッチングした経路のコストが最小のものを取り出すための優先度付きキュー(*pq*)を用いる．操作は，最初に通過する特徴点(移動開始点)付近の交差点を*pq*にコスト0で挿入する(6～8行目)．次に，全ての特徴点を通過するまで以下の操作を行う(10～23行目)．まず， *pq*から最もコストの低いノード(*now*)を取り出す(10行目)．以前通過した道路セグメントが次に通過すべき特徴点付近であった場合， *now*が特徴点を通過したと見なす(11, 12行目)．全ての特徴点にたどり着いている場合(終点付近にたどり着いた場合)，これまでに辿ってきた経路を返す(13～15行目)．また，現在注目しているノードが既に同じパターンの特徴点で通過している場合，9行目に戻る(17～19行目)．最後に*now*の繋がる全ての道路セグメントの先にあるノードを*pq*に追加する(20～22行目)．

以上の処理過程を図１に示す．

３．実験結果

提案方式を評価するために，GPSロガーを用いて筆者の一人の車での移動を約1年にわたり収集し，実データを取得した．今回用いたGPSデータは，主として自宅(図２中のA点）からの移動と勤務先（同B点）からの移動であり，主に自宅から勤務先への経路と勤務先から自宅への経路(約9km)が大半を占める．

表１は本実験で用いたパラメータを示している．

この表１に示したパラメータでの実験結果を表２示す．また，図２は移動軌跡を示しており，図中のA，Bがそれぞれ表２の移動起点を示している．成功の評価としては，マップマッチングによって求まった地図上の経路が全て正しい場合を成功．一箇所でも失敗があった場合には失敗と判断した． また，判断基準は実際に通過した経路とマッチングによって求まった経路を目視によって判断した．

表１　実験に用いたパラメータ

設定値	意味	値
誤差範囲	GPS, 地図の誤差	70m
類似角度	類似とになす誤差角度	25°
特徴点角度	右左折と判断する際の角度	135°

表２　実験結果

移動起点	データ数	平均処理時間	成功率
A	183	2373(秒)	85.2%
B	218	2486	97.2%
その他	15	2339	93.3%
全データ	416	2431	91.8%

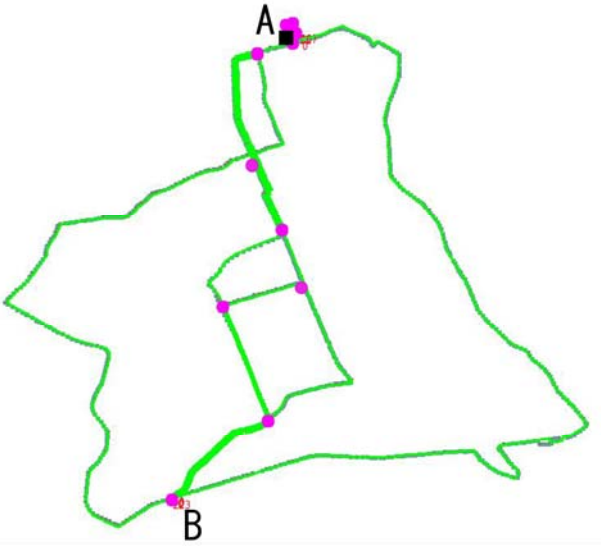


図２　実験に用いた経路履歴

マッチングミスは大きく三種類に分けられる．まず第１は，図３のようにビルや高架によるマルチパスや高速道路の高架下，トンネル内におけるGPS測位点の欠落などの影響である．この原因によるマッチングミスは全マッチングミスの内6割以上を占めた．

二つ目は，使用した道路データが進路方向を持たないものであったため発生した誤りであり，実際は左側通行であるのにも関わらず，右側通行道路とマッチングしたものが見られた．

最後に閉路におけるマッチングミスがある．閉路が存在する移動軌跡のうち9割以上でマッチング成功しているものの，一部の移動軌跡に おいてマッチングが失敗している．今後，道路セグメントに対するスコアリング方式の更なる改良が必要があるといえる．

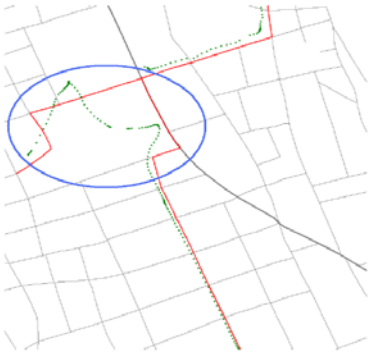


図３　マッチングミスの例

４．おわりに

本稿では，オフラインマップマッチングを対象として，多段階処理によりパラメータ設定が容易なマップマッチング方式を提案した．本方式は，車両の実時間モニタリングや時空間データマイニングなどへの応用を予定している．

マップマッチングは測位誤差の影響が避けられず，それによるミスマッチが発生する．今後，マッチングの確からしさを評価しやすい方式の研究が望まれる．

参考文献

Brakatsoulas, S. et al. (2005) On-Map-Matching Vehicle Tracking Data, VLDB 2005, pp.853-864

Civilis, A. and Jensen C.S. (2005) Techniques for Efficient Road-Network-Based Tracking of Moving Objects, IEEE Trans. KD, 17, pp.698-712

Quddus, M.A. et al.(2003) A General Map Matching Algorithm for Transport Telematics Applications, GPS Solutions, 7, pp.157-167

Wenk, G., Salas, R., and Pfoser D. (2006) Addressing the Need for Map-Matching Speed: Localizing Global Curve-Matching Algorithms, SSDBM 2006, pp.379-388

藤野，Htoo，大沢(2019) 経路履歴を用いて経路予測を行なう移動体の実時間モニタリング，電子通信学会技報，ITS2010-5