大規模プローブカー情報を処理するための高速リンクマッチング手法

今井 照之† 藤山 健一郎† 喜田 弘司† 中村 暢達†
†NEC サービスプラットフォーム研究所

1 はじめに

近年，プローブカーから情報を収集，処理して交通管理などに活用するプローブカー情報システムが注目されている．プローブカーとは，速度センサや GPS などのセンサを搭載した車で，走行情報，位置情報を収集し，サーバに送信する．プローブカーの位置情報を活用するには，まずその車が走行中の道路(リンク)を判定するリンクマッチング処理が重要である．

現在の多くのプローブカー情報システムでは，1 千台から 1 万台のプローブカーを使用し，サーバ上で数分単位で処理している [1]．今後プローブカーが増加し，より広い用途に情報を活用するためには，数万台から数十万台規模のプローブカー情報のリンクマッチング処理を秒単位で処理する必要がある．

そこで，本稿では，プローブカー情報システムの大規模化に対応するために，リンクマッチング処理を精度を保ちながら高速化する方法を提案する．

2 従来方式とその課題

2.1 リンクマッチング

リンクマッチングの最も単純な方法は，車と道路の距離を計算し，距離最小の道路上を走っていると判定する方法である [2]．しかし，全ての道路について車との距離比較を行うと，距離計算の回数が増えるため大量の車を高速に処理できない．そこで，距離計算回数を削減するモザイクマッチング方式を提案した [3]．モザイクマッチング方式では，事前に地図領域を格子分割し，格子毎に，その領域内に含まれる道路への対応を示すリンクマッチング表を生成する．格子領域内に含まれる道路を道路候補と呼ぶ．図 1 の例では，左の道路 $r1,\ldots,r6$ が含まれる地図領域を 3×3 に格子分割して右のリンクマッチング表を得る．各車のリンクマッチング処理では，まず，車の位置から，その車が存在する格子を判定する．次に，リンクマッチング表から，その格子に対応する道路候補を求める．最後に，その道路候補中の各道路と車の距離を計算し，距離が最小の道路を走行中の道路とする．事前にリンクマッチング表を用意することで，車毎の距離計算の回数を減らし，処理速度を向上できる．

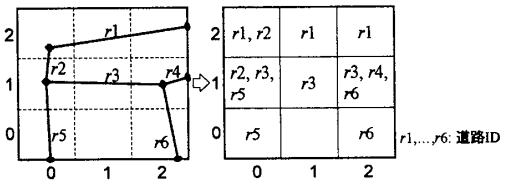

図 1: 地図領域の格子分割とリンクマッチング表

2.2 課題

モザイクマッチング方式のリンクマッチング処理速度は，格子内の道路数が少ない程高速である．格子サイズが小さい程，格子当たりの道路数は少なくなるため，モザイクマッチング方式では格子サイズを道幅程度に小さくすることで高速化を実現している．

しかし，この手法は車の位置の誤差を考慮しておらず，格子サイズを小さくするとリンクマッチング処理の精度が下がるという問題がある．なぜなら，誤差があると，車が本来存在する格子の隣の格子に存在すると判定する場合があるからである．格子判定を誤ると，誤った道路候補の中から距離比較を行うため，道路の判定を誤ってしまう．格子サイズが小さくする程，誤差により格子判定を誤る可能性が高くなる．よって，誤差がある場合でも，高速かつ精度を落とさずにリンクマッチングできる方式が必要である．

3 提案手法

3.1 隣接格子補完

誤差により格子判定を誤ると，道路候補から正解の道路が漏れるため，リンクマッチングを誤る．そこで，格子判定を誤った場合でも正解道路が道路候補に含まれるようにするため，隣接する 8 つの格子領域に含まれている道路をリンクマッチング表の格子に対応する道路候補に加える．この手法を隣接格子補完と呼ぶ．図 2 の例において，格子 (2,1) に対応する隣接格子補完前後の道路候補 $R_{2,1}, R'_{2,1}$ はそれぞれ，$R_{2,1}=\{r6\}$，$R'_{2,1}=\{r2,r3,r5,r6,r9,r10\}$ である．

隣接格子補完を行ったリンクマッチング表を使用することで，車が存在する格子の判定を誤った場合にでも，格子に対応する道路候補に正解の道路が含まれる．これにより，正解の道路の漏れを減らすことができる．

Link Matching Method for High Speed Processing of Large Scale Probe-car Information

† Teruyuki Imai, Ken-ichiro Fujiyama, Koji Kida, and Nobutatsu Nakamura
Service Platforms Research Laboratories, NEC Corporation (†)

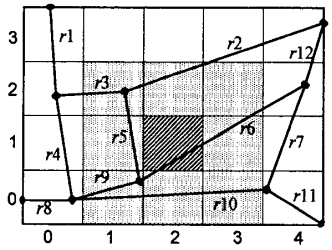

図 2: 隣接格子補完の例

3.2 最適な格子サイズ

隣接格子補完を用いた場合でも，格子サイズが小さ過ぎると誤差に対応できない．これは，格子サイズが小さいと，車の位置が本来存在する格子に隣接しない格子に対応し，道路候補から正解の道路が漏れるからである．そこで，図 3 を用いて，隣接格子補完を行う場合における精度と速度を両立するために適切な格子サイズを考察する．

位置の誤差を ε，格子サイズを l とおく．格子 G 内部の領域にある車が出力する位置の範囲は，斜線で示す領域である．一方，隣接格子補完により，G に対応する道路候補は，図中の 9 つの格子領域内に含まれる道路である．隣接格子補完を行ったリンクマッチング表から道路候補の漏れをなくすには，9 つの格子内に斜線領域が含まれていれば良い．従って，$l \geq \varepsilon$ であれば良いといえる．

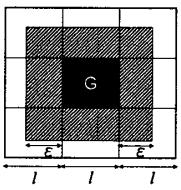

図 3: 格子サイズと誤差の関係

4 実験

3.1 の手法により精度が向上することと 3.2 による格子サイズ設定が適切であることを示すため，各手法のリンクマッチング処理の速度と精度を測定，比較する．

4.1 方法

3.2 の有効性を評価するため，誤差を考慮して格子サイズを設定したリンクマッチング表と考慮せずに道幅程度に格子サイズを設定したリンクマッチング表を使用して精度を比較する．また，3.1 の有効性を評価するため，前述の 2 種類の格子サイズのリンクマッチング表について，3.1 の補完を行ったものと行わないものを使用して精度を比較する．

実験では，南北 8km，東西 10km の地図を格子分割して用いた．格子サイズ 30m のリンクマッチング表が 3.2 を考慮したもので，格子サイズ 10m のリンクマッチング表が 3.2 を考慮せず道幅程度に格子分割したものである．さらに，前述 2 種類のリンクマッチング表について隣接格子補完を行ったものと行わなかったもの，および基準として従来手法どおり格子分割を行わなかったもの，合計 5 種類のリンクマッチング表を使用した．プローブカー情報はシミュレーションで生成し，位置に分散 $\sigma = 30$m の正規乱数による誤差を加えた．

4.2 結果

実験結果を表 1 に示す．

表 1: 実験結果

#	サイズ	補完	速度 [万件/秒]	正解率 [%]
0	N/A	なし	0.510	73.3
1	30m	なし	117	65.0
2	30m	あり	85.3	73.3
3	10m	なし	158	28.6
4	10m	あり	107	72.7

結果 1 ～ 4 の全ての場合において毎秒数十万台以上の速度であり，従来手法による結果 0 と比べて格子サイズの縮小により十分な速度が得られることがわかる．

結果 1 と結果 3 を比較すると，格子サイズを単純に縮小すると精度が著しく低下するが，誤差を考慮することで，精度低下が軽減される．

結果 1 と結果 2，または結果 3 と結果 4 を比較すると，隣接格子補完により精度が向上したことがわかる．特に，結果 2 の格子サイズを誤差を考慮して決定し隣接格子補完を適用することで，従来手法と同程度の精度が得られながら，毎秒数十万台のリンクマッチング処理が可能となるといえる．

5 おわりに

本研究では，大規模プローブカー情報を処理するために，リンクマッチング処理の速度と精度の両立する手法を提案した．実験により，精度を従来手法と同様に保ちながら，高速化を達成し，数十万台の車を毎秒処理可能であることを示した．

参考文献

[1] P-DRGS プローブ情報を活用した動的経路案内システムの研究開発 http://www.p-drgs.com/.

[2] 国土交通省 国土交通政策研究所：次世代マルチモーダル ITS 研究会報告－プローブデータを活用した交通情報の把握に関する研究－ (2005).

[3] 喜田弘司 他：次世代プローブ情報システム (2) 大規模高速マップマッチングアルゴリズムの提案, in *DICOMO2007* (2007).