

移動軌跡データモデルと領域に基づく問合せ処理 Query Processing for Topological Relation between Trajectory and Regions

柳沢 豊[†]
Yutaka Yanagisawa

赤埴 淳一[†]
Junichi Akahani

小暮 潔[‡]
Kiyoshi Kogure

1. はじめに

近年, GPS などの位置取得デバイスにより継続的に取得されたユーザの移動軌跡データを分析することで, ユーザの行動の特徴を抽出しようとする研究が進んでいる. たとえば, オフィス内を移動する多くのユーザの移動軌跡を分析すれば, オフィスの構造を最適化するための情報を得られると考えられる. 同様に, 車の移動軌跡データから交通渋滞の予測や, 道路の新設の必要性などを検討する手がかりが得られる.

こうした分析を行うには, 移動軌跡データの特徴を記述するための方法が必要になる. 従来より, 時空間上の図形データを扱うデータベースである時空間データベース [1] では, 図形の特徴を記述するための方法が提案されている [2]. しかし, 移動軌跡データのような複雑な形状をもつ, 向きのある線分の特徴を記述し検索する方法についての議論は十分ではない.

そこで本論文では, 文献 [2] で述べられている線分と領域との位相幾何学的関係を記述する方法を拡張し, 有向線分と領域との間にできる位相幾何学的な関係を利用して, 移動軌跡データの特徴を記述する方法を提案する. さらに, この記述を用いて指定された移動軌跡データを検索するためのインデックス手法を提案し, 手法の評価について述べる.

2. 移動軌跡データ

移動軌跡 λ とは, 質点が時間軸に沿って移動していった軌跡であり, 空間 R^2 上にプロットすると, 向きと有限の長さをもつ曲線 (あるいは直線) になる (図 1 左). しかし, 実際にデータが取得される際には位置取得デバイスの性質上, 移動軌跡データ λ は時系列に沿って離散的に取得された座標情報の列になる (図 1 右). 空間上に描画された領域との位相幾何学的な関係を考える上では, 移動軌跡データが離散的な数列のままでは扱いにくい. そのため, 本論文では移動軌跡データを, 離散的な座標データを連続化する方法として一般的に用いられる方法である折れ線近似 (Piecewise Linear Approximate) を用いて連続な有向線分にしておく. その上で, 空間 R^2 上での向きと有限の長さをもつ線分 λ と境界のある連続な領域 R との位相幾何学的な関係について論じる.

3. 領域と移動軌跡との関係記述

空間 R^2 上にある, 端点のあるひとつの連続な線分 λ と境界のある連続なひとつの領域 R との関係を記述する方法として, 線分の両端 $\partial\lambda$, 線分の内部 $\lambda^\circ = \lambda - \partial\lambda$ と, 領域 R の境界 ∂R , 領域の内部 $R^\circ = R - \partial R$, および R の外部 $R^- = R^2 - R$ の共通部分の有無を 3×2 行列として表す方法が提案されている [2].

[†]日本電信電話株式会社 NTT コミュニケーション科学基礎研究所
[‡]ATR メディア情報科学研究所

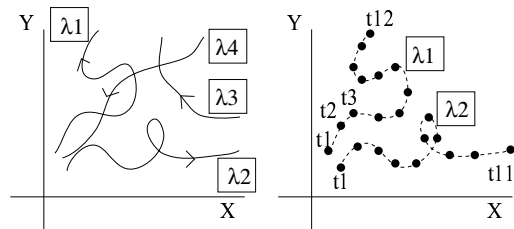


図 1: 移動軌跡データ

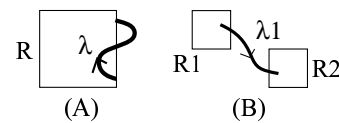


図 2: 有向線分と領域との関係の例

移動軌跡は向きがある線分なので, 従来の記法では関係の記述が不十分である. そこで本論文では, 両端 (始点と終点) が区別できるように, 始点 $\partial_0\lambda$ と終点 $\partial_1\lambda$ を用いて関係記述を拡張する. このようにすると, 移動軌跡 λ と領域 R との位相幾何学的な関係は, $\partial_0\lambda$, $\partial_1\lambda$, λ° と ∂R , R° , R^- の共通部分の有無を表す 3×3 行列で記述できる. 例えば図 2 のような関係 $T(R, \lambda)$ は, 次の行列として記述できる.

$$T(R, \lambda) = \begin{pmatrix} \partial R \cap \partial_0\lambda & \partial R \cap \partial_1\lambda & \partial R \cap \lambda^\circ \\ R^\circ \cap \partial_0\lambda & R^\circ \cap \partial_1\lambda & R^\circ \cap \lambda^\circ \\ R^- \cap \partial_0\lambda & R^- \cap \partial_1\lambda & R^- \cap \lambda^\circ \end{pmatrix} = \begin{pmatrix} -\emptyset & -\emptyset & -\emptyset \\ \emptyset & \emptyset & -\emptyset \\ \emptyset & \emptyset & -\emptyset \end{pmatrix}$$

可能な行列の種類は 26 通り存在する. ここで, 関係行列 $T_A(R, \lambda_A)$ と $T_B(R, \lambda_B)$ (ただし $\partial_1\lambda_A = \partial_0\lambda_B$) の間の演算である CONNECT 演算 $T_A \bullet T_B$ を次のように定義する.

$$T_A \bullet T_B = \begin{pmatrix} A_{11} & A_{21} & A_{31} \\ A_{12} & A_{22} & A_{32} \\ A_{13} & A_{23} & A_{33} \end{pmatrix} \bullet \begin{pmatrix} B_{11} & B_{21} & B_{31} \\ B_{12} & B_{22} & B_{32} \\ B_{13} & B_{23} & B_{33} \end{pmatrix} = \begin{pmatrix} A_{11} & B_{21} & A_{21} + A_{31} + B_{11} + B_{31} \\ A_{12} & B_{22} & A_{22} + A_{32} + B_{12} + B_{32} \\ A_{13} & B_{23} & A_{23} + A_{33} + B_{13} + B_{33} \end{pmatrix}$$

この演算の導入により, 26 通りの関係は図 4 に示す基本的な 7 つの関係と CONNECT 演算の組み合わせで

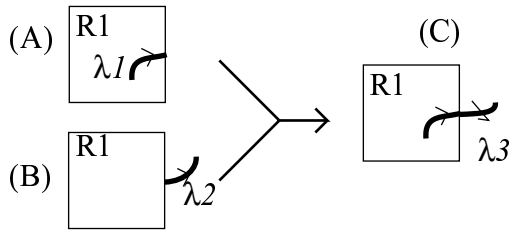


図 3: CONNECT 演算の例

すべて表現することができる。例えば、図 3 の (c) のような関係 $T(\lambda_c, R_1)$ は、図 4 の各関係を表す行列式をそれぞれ E_A, E_B, \dots, E_G とすると、CONNECT 演算を使って $T(\lambda_c, R_1) = E_B \cdot E_D$ と記述できる。

領域が複数ある場合の代表的な例として、図 2(B) のように、移動軌跡が片方の領域から出発してもう一方の領域へ入って終わる関係を考える。このような複数個の領域と移動軌跡 λ の関係は、移動軌跡 λ を適当に複数の移動軌跡の列 $\langle \lambda_1, \lambda_2, \dots \rangle$ に分割して、それぞれの λ_i が領域 R_j と関係の列 T として表すことができる。例えば図 2(B) のケースでは、 λ_1 を λ_2, λ_3 に分割し (ただし $\partial_1 \lambda_2 = \partial_0 \lambda_3$)、 $\langle T(\lambda_2, R_1), T(\lambda_3, R_2) \rangle = \langle E_B \cdot E_D, E_E \cdot E_C \rangle$ と記述できる。これは直感的には「 λ_1 を λ_2, λ_3 に分割すると、 λ_2 と R_1 の間の関係が $E_B \cdot E_D$ であり、 λ_3 と R_2 の間の関係が $E_E \cdot E_C$ になる」という意味である。

図 2(B) のように複数の領域を順番に通過する軌跡を探すクエリは、複数の領域を通過する移動軌跡を探すもとも基本的なクエリであり、本論文ではこれを PATH Query と呼ぶ。

4. グリッドリンクを用いたインデックス

以下では PATH Query を効率的に計算するためのインデックスの作成手法について述べる。線分を扱うデータベースでは、グリッドを使って線分をグリッドごとに分割し、インデックスをつける方法が一般的である [1]。グリッドによるインデックスを用いると、特定の領域を通過する線分を探すクエリに対して、まずその領域を含むグリッドを探し、そのグリッド内を通る線についてだけ指定された領域を通るかどうかを判別すればよいから、検索効率が大幅に向上する。

一方、PATH Query のような複数の領域を指定した順序で通過する移動軌跡を探すときには、領域を通過するかどうかのチェックに加え、通過する順序についてのチェックが必要になる。このチェックのコストを下げるために、本論文では図 5 のように、各移動軌跡ごとにそれが通過するグリッドの順序を予め記録しておく、グリッドリンクと呼ぶ方法を導入する。これを用いると、複数の領域を指定された順序で通過する PATH Query を計算するときには、最初の領域を含むグリッドをまず探し、次にグリッドを通過する各移動軌跡のグリッドリンクをたどり、2 番目以降に通過する領域を含むグリッドを順番に通過しているかどうかを調べればよい。これにより通過順序のチェックが単純化され、PATH Query の検索効率の向上を図ることができる。

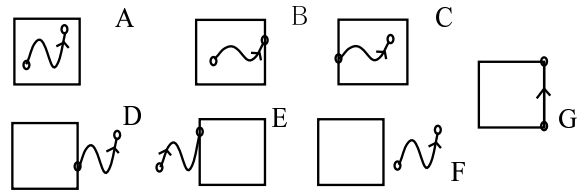


図 4: 7つの基本的な関係

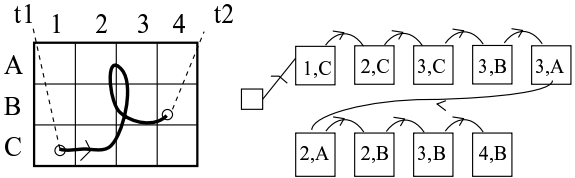


図 5: グリッドを用いたインデックス

5. 評価

実験のために、グラウンドのような道路のない平面上を自由に歩く人間の移動軌跡に近い移動軌跡データを、シミュレーションによって作成した。このデータに対してインデックス付けおよび検索の実験を行った。データとして 1000 個の座標データを含む、平均長が 50 km 前後の移動軌跡データ 1000 本を用意した。グリッドの一边の長さを 5km に固定してインデックスを作成した。そして、ランダムに選んだ 2 つのグリッドを領域 R_1, R_2 とする PATH Query について、グリッドリンクを使う場合と使わない場合での検索速度の比較を行った。検索速度は、選ばれた 2 つのグリッド間の直線距離に反比例する。これは、グリッドが近いほどそれら双方を通過する移動軌跡の数 (= 計算対象) が増えるためである。比較の結果、グリッドリンクを用いると用いない場合に比べて、平均 6.45 倍、グリッドが隣接する場合で 10.38 倍高速に検索することができた。

6. まとめ

本論文では、移動軌跡データを空間上の有向線分としてモデリングし、空間上の複数の領域と線分との間にできる関係を行列式を用いて記述することで、移動軌跡の特徴を一意に記述する方法について述べた。また、指定された条件を満たす移動軌跡データを探すためのインデックスの作成方法についても述べた。今後は実際に GPS など取得したデータに対して提案方式を適用し、評価を行う予定である。

参考文献

- [1] H.D. Chon, D. Agrawal, and Amr El Abbadi: Query Processing for Moving Objects with Space-Time Grid Storage Model, In Proc. of MDM2002, pp.121-129 (2002).
- [2] M. Egenhofer, E. Clementini, and P. di Felice: Topological Relations between Regions with Holes, Intl. Journal of Geographical Information Systems, Vol. 8, No. 2, pp.129-142(1994).