

## Safe-Region を用いた道路網距離に基づく効率的な連続範囲検索 Efficient Continuous Range Query Using Safe-Region in Road Network Distance

鴨目 翔太\*  
Shota Kamome

Htoo Htoo\*  
Htoo Htoo

大沢 裕\*  
Yutaka Ohsawa

### 1. はじめに

移動体が道路網上を移動しつつ、常にある検索条件を満たすものを検索し続けるタイプの検索を連続検索と呼ぶ。この連続検索は、2つの分類で研究が行われてきた。1つは、検索対象が固定し、検索点のみが移動する場合である。他の1つは検索点と検索対象が共に移動するタイプの検索である。前者については多様な検索に対して効率化方式が提案されてきた。一方、後者の検索を道路網距離で行う研究は比較的新しく、2010年代に入ってから活発化している。本稿では、後者の検索点と検索対象が共に移動する場合を対象として、距離範囲検索の効率化方式を提案する。

### 2. 連続検索のモデル

本稿では、連続検索で通常用いられるサーバクライアントモデルを想定する。クライアントの位置をサーバが把握しており、各検索要求に対してサーバが検索結果を求め、クライアントに送出する。

連続検索では、検索対象と検索点が時々刻々位置を変える。そこで、単純には決められたタイムスタンプ毎に位置をサーバに送る必要がある。しかし、移動体が渋滞等でタイムスタンプ毎の位置変更が少ない場合にも定時的な位置の報告は無駄である。そこで、各移動体が safe-region という領域を持ち、その領域を出た時点でサーバに位置を報告し、新たな safe-region を設定するという方式が提案されている [1]。

safe-region の形態としては、道路網上の範囲をサーバが作成し、それを移動体に送る方式も考えられるが、本稿では、サーバ側はユークリッド距離での範囲として設定する。safe-region の範囲を  $r$  とすると、以前にサーバに報告した位置から道路網距離で  $r$  離れたとき、移動体はサーバに現在地を報告する。移動体が道路網距離を計算するためには移動体自身が地図データを所持している必要があるが、所持していない場合は代わりに移動距離を使用することで、効率は落ちるものの正しい結果は得られる。サーバ側では、移動体からの位置更新がなければ、その移動体の存在位置を前回の報告の位置から道路網距離で  $r$  内の範囲に限定できる。また、 $r$  をシステムパラメータとすれば、サーバ側では移動体の safe-region を算出することも不要となる。

道路網上での距離  $D$  以内に存在する他の移動体を、移動体  $q$  が連続検索するものとする。以降、移動体  $q$  は一定時間間隔で現在位置をサーバに報告する。サーバは必要が生じたとき、任意の時点で各移動体に対して位置の報告を求めることができる。サーバは  $q$  からの距離が  $D$  以内の他のオブジェクトを検索して、 $q$  に結果を伝えるが、その際に各移動体の safe-region を基に

対象の絞り込みを行い、それらが検索結果に含まれるか否かの判定時に、正確な位置が分らなければ判定できないオブジェクトに対して現在位置の報告を求める。

本研究の目的は、このモデルに基づき連続検索を行う際に、サーバの演算コストを削減する方式の提案である。

### 3. 方式の概要

連続検索は、フィルタステップと、検証ステップで構成される。フィルタステップは範囲検索の対象を限定するものであり、検証ステップは前者で得られた候補オブジェクトが真に検索条件を満たしているか検証を行い、検索点に報告する結果オブジェクトを得る。

#### 3.1. フィルタステップ

各移動体は中心からの距離が  $r$  の safe-region を保持しており、その中に存在する場合にはサーバに対して位置を報告しない。即ち、移動体位置には半径  $r$  だけの曖昧さが生じる。図1はその状況を示している。図中のオブジェクトの内、 $sr1$  は完全に範囲外のため検証対象外となる。一方、 $sr2, sr3, sr4$  は safe-region が検索範囲と重なっているため、検証を必要とする。

サーバは、全てのオブジェクトが前回 safe-region を設定した時の位置を記録している。移動体位置のインデックスとしては、他の研究でもよく用いられているグリッドインデックスを用いる。このフィルタステップは、ユークリッド距離での範囲検索を実行する。2点  $a, b$  間のユークリッド距離  $d_E(a, b)$ 、その道路網距離を  $d_N(a, b)$  と表す時、 $d_E(a, b) \leq d_N(a, b)$  が成り立つため、ユークリッド距離で範囲  $D + r$  を超えるオブジェクトは検索の対象外であることが分かる。重なりを持つオブジェクトは候補オブジェクト集合 ( $C$ ) として次の検証ステップに渡す。

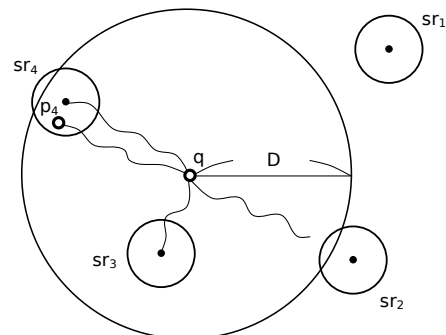


図 1: 検索の例

#### 3.2. 検証ステップ

検証ステップではまず、絞り込まれた各オブジェクトの前回の報告時の位置  $c_i$  と  $q$  間の距離、 $d_N(q, c_i)$  を

\*埼玉大学

求める。候補オブジェクトの内、 $d_N(q, c_i) - r > D$  のオブジェクトは検索範囲外となる (図 1 の  $sr2$ )。また、 $d_N(q, c_i) + r < D$  のオブジェクトは検索範囲内に含まれる (図 1 の  $sr3$ )。そして、それ以外のオブジェクト (図 1 の  $sr4$ ) に対して現在地  $p$  の報告を求め、正確な距離  $d_N(q, p_i)$  を計算し、それが  $D$  以下ならばそのオブジェクトは検索範囲内にあることがわかる。

タイムスタンプ間の時間経過は 10 秒程度の短い時間を想定しているため、各オブジェクトの移動距離は短いものと想定される。従って、各オブジェクトは前回の検索以降  $safe-region$  を新たに設定しておらず、 $d_{last} + M_q \leq D$  の条件を満たしていれば新たに  $q, c$  間の道路網距離を求める必要はない。但し、 $M_q$  は  $q$  がタイムスタンプ間で移動した距離、 $d_{last}$  は前回の検索時に求めた  $d_N(q, c) + r$ 、もし求めていけば  $d_N(q, p)$  を表す。

### 3.3. 高速化

この方式は、フィルタステップで絞り込まれた各移動体と検索点  $q$  からの距離の検証を高速化できる。候補集合  $C$  中の要素数が多い場合に、 $q$  から各候補点への道路網距離が必要になり、これを A\* アルゴリズムを繰り返して適用した場合に、同じノードが複数回調べられることになる。この効率化の為に、SSMTA\* アルゴリズム [2] を適用できる。

図 2 は  $q$  から  $p_1 \sim p_3$  の各点までの道路網距離を A\* アルゴリズムを繰り返し適用した時に連結リストが参照される部分を色付けで示している。この部分に対して隣接リストを 1 度のみ参照することにより効率化を図るのが SSMTA\* アルゴリズムである。また、 $p_i$  の各点への距離を求める場合には、ヒープから取り出されたレコードにおける  $q$  から現在点までの距離が  $D$  を超えた時点で探索を打ち切ることができる (図 1 の  $sr2$ )。

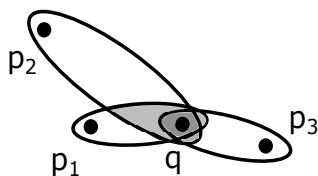


図 2: 探索領域の重複

## 4. 実験結果

提案方式の性能を評価するために実験を行った。道路地図は旧さいたま市で、ノード数は 16,284、エッジ数は 24,914 である。プログラムは Java で実装し、PC はのスペックは Intel Core i7 4790 CPU (3.60GHz) でメモリは 16GB である。各移動体はランダムに生成した 2 点間の最短経路を時速 20km で移動し、終点に到着した場合は同じ経路を戻るようにした。検索オブジェクト数は 10 とし、プログラムの開始から終了まで 10 秒置きにサーバに検索要求を出している。また、 $safe-region$  の半径は 300m、オブジェクト密度は 0.01、検索範囲は 2km をデフォルトとした。

図 3 は、オブジェクトのエッジ数に対する密度を変化させたときの 1 秒間あたりにクライアントがサーバ

に現在地を送信した回数である。“ $safe-region$  なし” は提案方式から  $safe-region$  を無くし、各クライアントは 10 秒置きに現在地をサーバに送るという方式である。通信回数は安定して提案方式の方が少なく半分程度に減らすことができた。

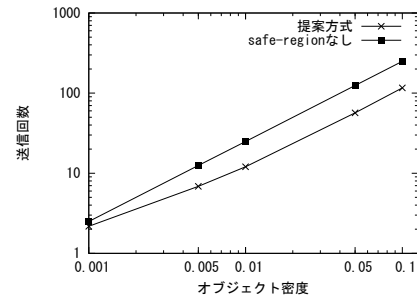


図 3: 密度に対する 1 秒あたりの送信回数

図 4 は  $safe-region$  の大きさを変化させた時の実行時間と送信回数である。 $safe-region$  が大きくなるに連れて検証を必要とするオブジェクトが増え、それに伴い実行時間も増加した。 $safe-region$  が大きくなるほど、移動体が  $safe-region$  から出る回数が減りその分位置を報告する回数も減るが、検索の際に検証ステップで現在地をサーバから要求されることが多くなる。そのため半径 300m までは送信回数が減少したが、それ以降は増加した。

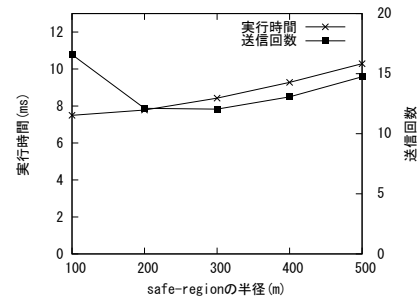


図 4:  $safe-region$  の半径に対する実行時間と送信回数

## 参考文献

- [1] M. A. Cheema, W. Zhang, X. Lin, Y. Zhang, X. Li, "Continuous reverse k nearest neighbors queries in Euclidean space and in spatial networks", The VLDB Journal, vol.21, pp.69-95, 2012.
- [2] H. Htoo, Y. Ohsawa, N. sonehara, and M. Sakauchi, "Incremental single-source multi target A\* algorithm for LBS Based on Road Network Distance", IEICE Transactions on Information and Systems, vol.E96-D, no.5, pp.1043-1052, 2013.