

# 車両データ選択的収集アルゴリズム

道路上にある障害物の状態変化を追跡調査する「障害物監視ユースケース」では、障害物の最新状況をとらえた車載カメラ画像を継続的に収集する必要があります。画像認識技術を活用すれば正確に画像選定できますが、車載の計算資源はごく限られており、クラウドに画像を転送するにも携帯網の通信コストがかかります。そこで、車両の位置や進行方位、カメラ画角といったメタ情報に基づいて各車載カメラの撮影範囲を推定し、必要な画像のみを選択的に収集する技術を考案しました。

たかぎ  
高木

まさる  
雅

まつお  
松尾

かずや  
和哉

なかだ  
中田

りょうた  
亮太

もり  
森

こうや  
航哉

NTT人間情報研究所

## コネクティッドカーの可能性

道路交通分野におけるICT化の要として期待されているコネクティッドカーですが、実はセンシング基盤としても大きな可能性を秘めています。各種の高性能センサと通信モジュールを搭載し、電池切れの心配が事実上なく、街中を縦横無尽に走り回るコネクティッドカーは、移動式センサノードとして理想的な特徴を兼ね備えています。もし、数1000万台のコネクティッドカーから車載カメラ画像やLiDAR (Light Detection and Ranging)\* 点群データを収集できれば、街全体をくまなく、絶え間なくスキャンして街のデジタルツインを構築できるようになるでしょう。

## 想定ユースケース

しかし、数1000万台のコネクティッドカーで生成されるセンサデータの総

\* LiDAR：光を用いたリモートセンシング技術の一種で、各方向に照射したレーザー光の反射時間から対象物までの距離を計測する技術。

量は10~100 Tbit/sと膨大であり、通信網、計算機、ストレージのいずれの観点でも全量収集は困難です。一方で、これらのセンサデータの中には、即座に収集すべきデータだけでなく、多少の収集遅延が許されるデータ、定期的な収集で十分なデータ、利用価値のないデータも含まれます。したがって、必要なセンサデータを取捨選択し、重要なセンサデータを優先的に収集することが重要となります。また、通信網や計算機の負荷状況に応じてセンサデータの収集ペースや収集タイミングを調整することで、負荷の時間変動を平準化し設備利用効率を高めることも可能です。

トヨタ自動車との協業においても、センサデータを選択的収集を必要とするユースケースはいくつかありましたが、本稿では「障害物監視ユースケース」に沿って技術課題と実装詳細、性能改善について紹介します。このユースケースでは、道路上の障害物が風に吹かれて移動したり、道路管理者によって撤去されたりすることを想定し、

障害物の近傍を通過した車両から車載カメラ画像を継続的に収集することで最新状況を把握します(図1)。

## 画像認識によるアプローチ

コネクティッドカーで撮影された大量の画像の中から、監視対象の障害物をとらえた画像を選出したいという要件に対しては、画像認識によるアプローチが一見すると順当に思えます。このアプローチでは大まかに2通りの構成が考えられますが、いずれも大きな課題を抱えています。

・構成①：コネクティッドカー内で処理

もっともシンプルな構成としては、コネクティッドカーに計算機を搭載して画像認識処理を行う方式が考えられます。しかし、この方式には自動車に特有の3つの課題があります。

第一に、真夏の炎天下における車内温度は80℃弱に達する場合<sup>(1)</sup>があり、計算機を安定稼働させるのは容易ではありません。第二に、乗用車の稼働率は平均5%といわれており、せっかく

搭載した計算機も休眠時間ばかりではコストに見合いません。第三に、自動車と計算機では製品寿命が大きく異なるため、計算機が先に寿命を迎えるか性能面での陳腐化が進みます。これらの事情を勘案すると、コネクティッドカーに計算機を搭載することは現実的な選択肢とはいえません。

・構成②：データセンタ内で処理

コネクティッドカー基盤の存在を前提とした場合、すべての画像をいったんデータセンタに転送して、データセンタ内で画像認識処理を行う構成も考えられます。この構成では先に指摘した3つの課題を回避できますが、画像転送に用いる携帯電話網の通信コストが課題となります。また、数1000万台規模で同時通信を行うケースを想定すると、携帯電話網に掛かる負荷も無視できなくなります。たとえコネクティッドカー1台当たり1 Mbit/sの通信トラフィックでも、1000万台では合計10 Tbit/sにも達します。これは我が国における現在の移動通信トラフィックの15倍という膨大な分量です<sup>(2)</sup>。

位置情報によるアプローチ

前述の課題に直面した私たちは、「障害物監視ユースケース」の要件を改めて吟味しました。収集された画像は後段の処理で細かく分析されるため、画像選定の時点で障害物の有無を高精度に判定する必要はありません。一方で、この画像分析処理は非常に高負荷であるため、分析する画像の個数はなるべく抑えたいという事情があります。また、監視対象とする障害物の位置情報は既知であり、これを手掛かりとして活用しない手はありません。

そこで私たちは、画像認識技術に頼ることなく収集すべき画像を選定する技術を考案しました。具体的には、周辺車両の位置情報や進行方位、車体サ

イズ、カメラ画角、カメラ解像度といったメタ情報に基づいて各車載カメラの撮影範囲を推定したうえで、障害物と各車両の位置関係を踏まえて、障害物をとらえたと推定される画像を選定します。この選定処理は以下の5つの工程から構成されています(図2)。

- ・工程①：車両情報データベースの整備

画像選定の際に重要な手掛かりとなる車両のメタ情報は、コネクティッドカーから定常的に収集しておき、高速時空間データベースAxispot<sup>®</sup>に格納しておきます。

- ・工程②：車両情報データベースの検索

障害物監視のリクエストが届いたら、障害物地点と現在時刻に対応する

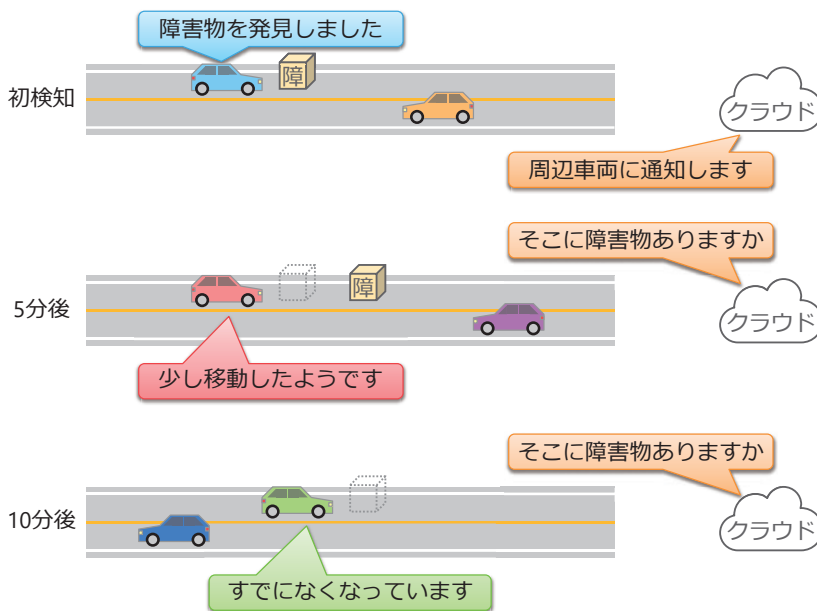


図1 障害物監視ユースケースの概要

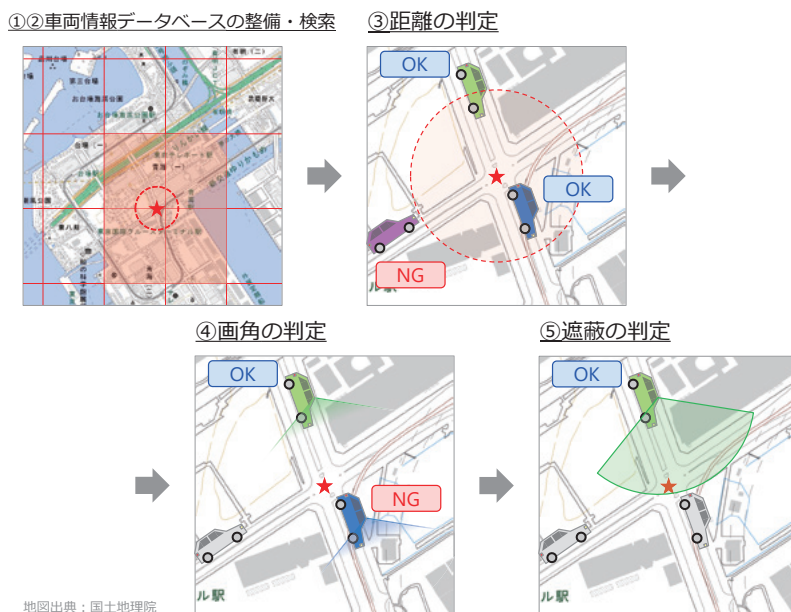


図2 メタ情報に基づく画像選定の流れ

時空間インデックスで車両情報データベースを検索し、ヒットした車両を候補車両とします。

・工程③：距離の判定

次に、各候補車両と障害物の間の距離を計算し、車載カメラの解像度から算定した有効射程距離より遠く離れた車両を候補から除外します。

・工程④：画角の判定

さらに、各候補車両から見た障害物の方向を計算し、車載カメラの画角内に障害物をとらえていないと推定される車両を候補から除外します。

・工程⑤：遮蔽の判定

最後に、周辺車両どうしや地物の位置関係を調べて、他の車両や地物に遮蔽されて障害物が見えないと推定される車両を候補から除外します。

遮蔽判定の計算量削減

これら5つの工程の中でもっとも複雑な計算処理を要するのが遮蔽判定です。後に紹介するように、遮蔽判定は処理時間の支配項となるため、実証実験期間中にアルゴリズムの大幅改良を実施しました。本稿では改良前と改良後のアルゴリズムを順に紹介します。

遮蔽判定のもっとも直感的な実装方法としては、各候補車両の視界範囲を計算して障害物地点と当たり判定を行う方法が考えられます。例えば、図3のように障害物の近傍に6台の車両が存在する場合、6台の車両それぞれについてカメラ画角と有効射程距離から扇形の視界範囲を描きます。ここから、他の5台の車両により遮蔽されている部分がないか、1つひとつ調べて視界範囲から差し引きます。この方式では、車両どうしが互いに影響を及ぼし合う視界範囲の遮蔽を総当たりで調べる必要があるため、車両数nに対して計算量は $O(n^2)$ となります。障害物近傍の車両数が増加するに従って処

理時間が急速に増大するため、監視対象の障害物周辺で渋滞により多数の車両が密集した状況では、処理時間が非常に長くなるおそれがあります。

そこで私たちは、計算量を抑えるためのアルゴリズム改良に挑戦しました。改良のポイントは「車載カメラの視界範囲を計算する」という固定観念を捨て去ったことです。ここでは「車両から障害物が見える」ならば「障害物から車両が見える」という逆転の発想に基づき、車両の視界範囲ではなく障害物の視界範囲を計算し、各車両と当たり判定を行います(図4)。この方式では各車両について1回ずつ遮蔽の有無を調べて視界範囲から差し引け

ばよく、総当たりでの計算処理が不要となるため、車両数nに対して計算量を $O(n)$ に抑えられます。また、視界範囲を扇形の集合で近似することで、各車両との当たり判定も非常にシンプルに実装できます。

性能評価

ここでは、アルゴリズム改良前と改良後の各実装について、さまざまな車両配置パターンにおける処理時間を評価しました(図5)。車載カメラの有効射程距離を100mと仮定し、障害物地点の半径100m以内に位置する車両数を仮想的に変化させます。最悪ケースの評価を行ううえでは、車両数の上

- ①各車両の視界範囲を1台ずつ計算
- ②障害物と当たり判定

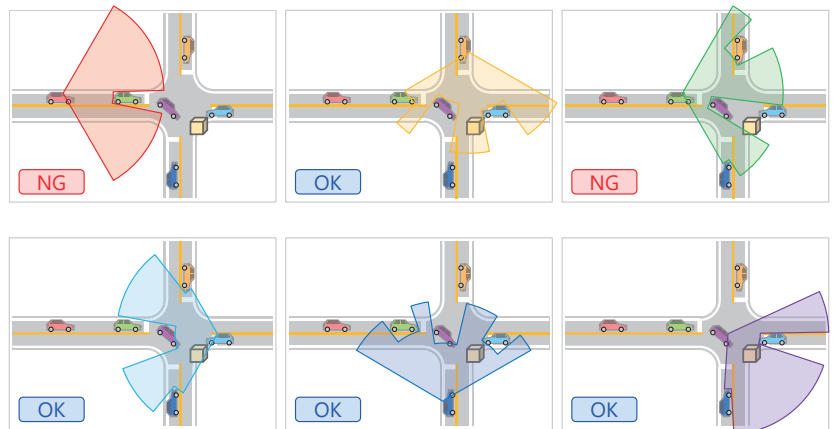


図3 遮蔽判定の実施例(アルゴリズム改良前)

- ①障害物の視界範囲を計算
- ②各車両と当たり判定

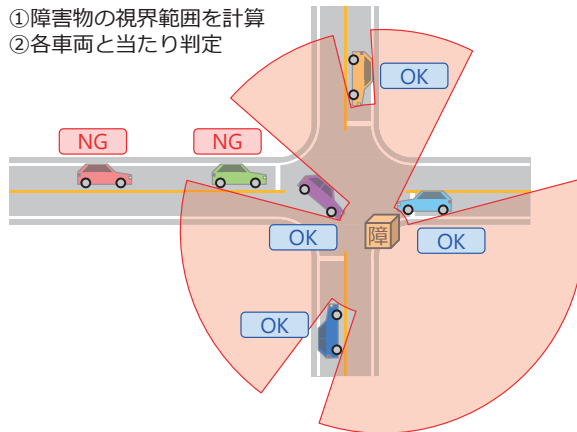


図4 遮蔽判定の実施例(アルゴリズム改良後)

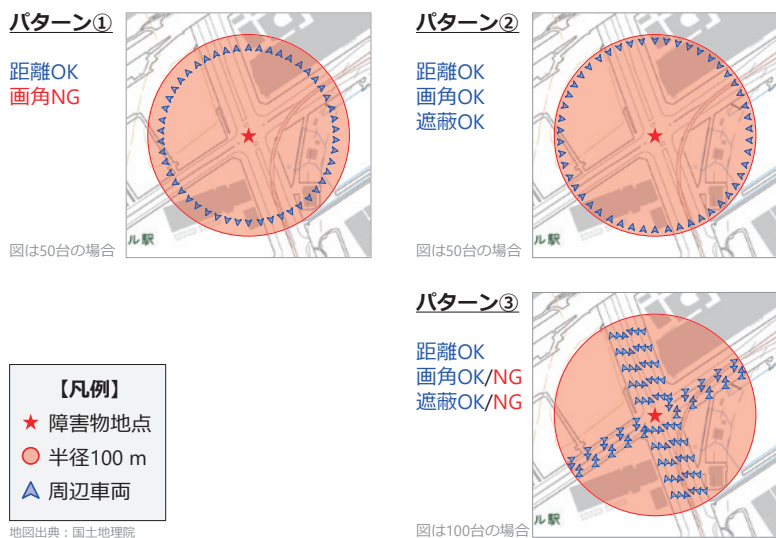


図5 性能評価における車両配置パターン

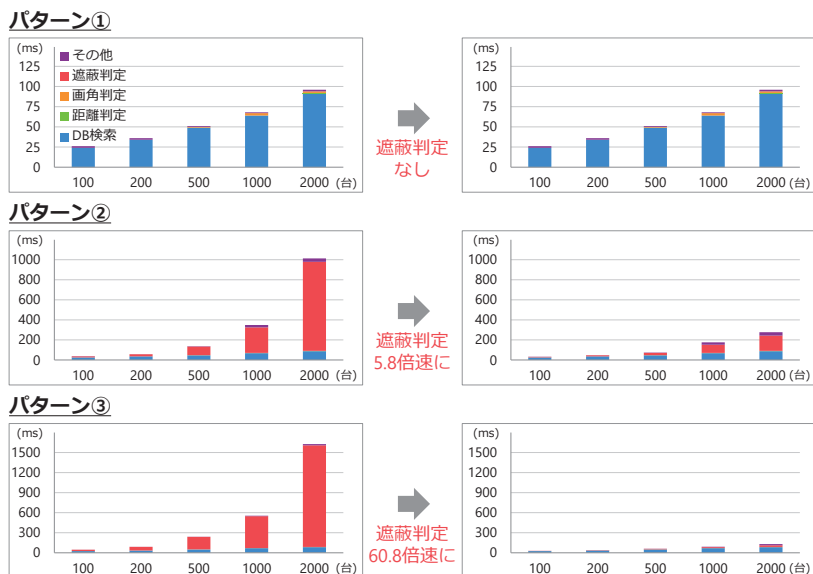


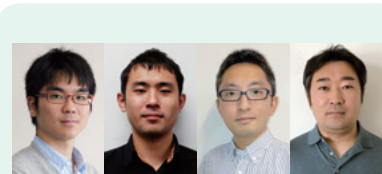
図6 車両配置パターンと処理時間の関係

以上高速な処理を実現できました。今後は、この成果をコネクティッドカー以外の用途にも応用できないか検討を進めていきます。

例えば、街のデジタルツイン構築では、一度センシングした領域についても定期的に再センシングして最新の状況を反映する必要があります。街の景色には変化の頻度が高い部分と低い部分があるため、街全体を均一に再センシングするより、要所要所をピンポイントで計画的に再センシングするほうが効率的です。このような場面では、本稿で紹介したアルゴリズムを活用することで、収集すべきデータの選定を効率化できます。さらに応用的な課題として、監視対象が面的な広がりを持つケースや、位置情報に誤差が含まれるケースに対応する手法についても検討を進めていきます。

参考文献

- (1) <https://jaf.or.jp/common/safety-drive/car-learning/user-test/temperature/summer>
- (2) [https://www.soumu.go.jp/main\\_content/000791761.pdf](https://www.soumu.go.jp/main_content/000791761.pdf)
- (3) <https://locationtech.github.io/jts/>



(左から) 高木 雅 / 松尾 和哉 / 中田 亮太 / 森 航哉

デジタルツインの社会実装においては、コネクティッドカーのような「つながるモノ」を駆使して、路上落下物のような「つながらないモノ」の情報をいかに収集するかが、今後ますます重要となると考えています。

◆問い合わせ先

NTT 人間情報研究所  
 デジタルツインコンピューティング  
 研究センタ  
 E-mail dtc-office-ml@hco.ntt.co.jp

限設定が重要となりますが、ここでは駐車場並みの密度で車両が密集した状態を想定して上限を2000台としました。選定アルゴリズムの実装言語としてはJavaを採用し、改良前のみJTS Topology Suite<sup>(3)</sup>という幾何計算ライブラリを併用しました。

各車両配置パターンで車両数を変化させたときの処理時間を図6に示します。アルゴリズム改良前は、車両数の2乗に比例するかたちで遮蔽判定の処理時間が急速に増大していましたが、

アルゴリズム改良後は、車両台数に比例する水準に抑えられています。車両2000台が密集する最悪ケースにおける遮蔽判定の処理時間は25msと、アルゴリズム改良前より60倍以上高速であることを確認できました。

今後の展望

今回の実証実験では、「障害物監視ユースケース」の要件に対して実装を最適化することで、幾何計算ライブラリを用いた汎用的な実装と比べて60倍