

# SLAMのQoS要求を考慮した 車両-エッジ間のトラフィック削減手法の提案

豊田睦<sup>1,a)</sup> 武藤晟<sup>1,b)</sup> 花輪麻衣奈<sup>1,c)</sup> 重野寛<sup>1,d)</sup>

**概要:** 概要: 現在, IoV (Internet of Vehicle) の発展に伴い遅延に敏感な車両アプリケーションが開発されている。その中でも自律走行に要求される自己位置推定と環境地図作成を同時に実行する SLAM (Simultaneous Localization and Mapping) が注目を浴びている。既存のエッジコンピューティングを用いた SLAM の実行手法では, 比較的計算量の大きいマッピングとループ閉じ込みをエッジサーバ側で実行し, トラッキングをクライアント側で実行する。しかし, 既存手法では物理環境によらず固定周波でのマップ更新を実行するため, SLAM のトラッキング精度が低下したり, ネットワーク帯域を不必要に圧迫するという問題点がある。そこで, SLAM の QoS 要求を考慮した車両-エッジ間のトラフィック削減手法を提案する。トラフィックを削減する問題を整数計画問題として定式化し, 近似アルゴリズムによって SLAM の精度を維持したままトラフィックの削減を目指す。SLAM の QoS 要求を制約に, 物理環境を考慮した動的なマップ更新の決定を行う。近似アルゴリズムでは, 車両の走行軌道を前 Key Frame との相対角度によって算出し, マップ更新の決定およびマップデータサイズの決定を前マップ更新からの経過時間, SLAM のトラッキングに要求される最低限のマップ更新から算出した。本研究では提案したトラフィック削減手法のプロトタイプを実装して実験を行い, 動作確認と評価を行う予定である。

## 1. はじめに

近年, IoV (Internet of Vehicle) の発展に伴い遅延に敏感な車両アプリケーションが開発されている。その中でも自律走行に要求される自己位置推定と環境地図作成を同時に実行する SLAM (Simultaneous Localization and Mapping) [1] が注目を浴びている。SLAM はトラッキング, マッピング, ループ閉じ込みの3つのスレッドに分かれて実行され, 共通のマップ情報を参照して動作する。また, マップ作成のための局所的な最適化や, マップの歪みを最小にする全体的な最適化を繰り返し実行するため, 車載コンピュータに対して計算負荷が膨大である [2]。そのため, 車載コンピュータの計算リソース, ストレージの面から車両で SLAM のアプリケーション要求を満たすことは現実的ではない。そこで, エッジコンピューティング [3] を用いて SLAM を実行する研究が行われている。エッジコンピューティングとはネットワークのエッジ部分に計算

資源を配置し, クライアントの計算を肩代わりすることでクライアントの計算負荷を軽減することが可能である。

エッジコンピューティングによる SLAM の実行では, 比較的計算量の大きいマッピングとループ閉じ込みをエッジサーバ側で実行し, トラッキングをクライアント側で実行する。また, トラッキングに必要なマップ情報はエッジサーバ側から逐一クライアント側に更新する必要がある。しかし, マップを生成するために選定される KF (Key Frame) は物理環境に依存し, 予測不可能である。既存手法では, 物理環境によらず固定周波でのマップ更新を実行するため, 車両の周辺環境や走行軌道は考慮されておらず, SLAM のトラッキング精度の低下やネットワーク帯域を不必要に圧迫するという問題点がある。そのため, SLAM の QoS 要求を考慮し, ネットワークのトラフィックを最小にするマップ更新頻度を決定することが必要であると考えられる。

本稿では, SLAM の QoS 要求を考慮した車両-エッジ間のトラフィック削減手法を提案する。提案手法では SLAM の QoS 要求を制約に車両-エッジ間のトラフィックの最適化問題を近似アルゴリズムで解く。近似アルゴリズムでは, 車両の周辺環境や走行軌道を考慮した動的なマップ更新とデータサイズを決定する。車両の走行軌道は前 KF との相対角度から右左折を検知し, 最適なマップ更新を決定する。更新するマップのデータサイズは前マップ更新から

<sup>1</sup> 慶應義塾大学大学院理工学研究科  
Graduate School of Science and Technology, Keio University,  
Yokohama, Kanagawa, 223-8522, Japan

a) toyoda@mos.ics.keio.ac.jp

b) muto@mos.ics.keio.ac.jp

c) hanawa@mos.ics.keio.ac.jp

d) shigeno@mos.ics.keio.ac.jp

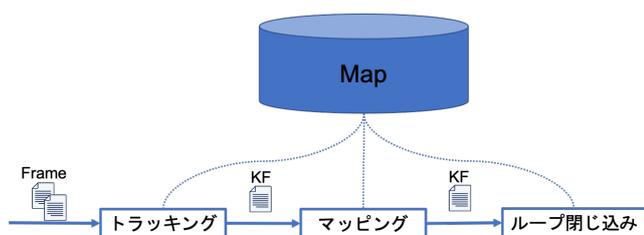


図 1 SLAM のシステム図  
Fig. 1 System of SLAM

の経過時間と SLAM の QoS 要求から決定する。予備実験では、単一のエッジサーバで SLAM の実装を行い、動作確認と各モジュールにおける処理時間を計測した。評価の予定として、提案手法のプロトタイプシステムを実装し、2 台の物理マシンを車両とエッジに想定することによって実験を行う。実験では、車両-エッジ間のトラフィック量や車両側のトラッキング精度について提案手法の有効性を評価しようと考えている。これにより、提案手法が SLAM の QoS 要求を満たしながらネットワークトラフィック量の削減を実現できているか確認する予定である。

## 2. 関連研究

本章では、SLAM, エッジコンピューティング, エッジコンピューティングによる SLAM 実行について述べ、これらから考えられる既存のエッジコンピューティングによる SLAM 実行手法の問題点を抽出する。

### 2.1 SLAM

近年、車両の自律走行を実現するために自己位置推定と環境地図作成を同時に実行する SLAM が注目を浴びている。SLAM は LiDAR-SLAM と Visual-SLAM の 2 つに分類されるが、本稿では Visual-SLAM について言及する。

図 1 に SLAM のシステム図を示す。SLAM はトラッキング, マッピング, ループ閉じ込みの 3 つのスレッドに分かれて実行され、共通のマップ情報を参照して動作する。トラッキングは MP (Map Point) と呼ばれる特徴量を計算し、前フレームの MP との対応からカメラの姿勢を計算し、冗長を防ぐため全てのフレームからマッピングに使用する KF を抽出する。KF の抽出は車両の周辺環境や軌跡に大きく依存するため、予測不可能である。マッピングはマップ情報にある隣接する KF 群との局所的な最適化を行い、環境地図を作成する。ループ閉じ込みは環境地図全体の歪みを修正し、蓄積された誤差を最小化するための全体的な最適化問題を解く。マップ作成のための局所的な最適化や、マップの歪みを最小にする全体的な最適化を繰り返し実行し、マップ情報は 1 マイルあたり平均 200MB 容量を必要とするため、車載コンピュータに対して必要計算資源およびストレージが大きいことがわかっている [4]。

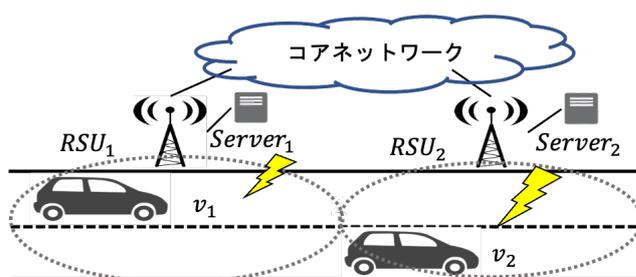


図 2 車両エッジコンピューティングの全体図  
Fig. 2 Overview of Vehicular Edge Computing

### 2.2 車両エッジコンピューティング

IoV の発展に伴い、車両は他の車両との車車間通信 (V2V) や路側機 (RSU) との路車間通信 (V2I) が実現し、リアルタイム性を要求する車両アプリケーションの需要が増加している。しかし、前節で述べた通り自律走行に要求される車両アプリケーションは計算負荷、ストレージの観点から車載コンピュータで実現することが厳しいとされている。そこで、車両の計算負荷を軽減することを目的にした、車両ネットワークへクラウドコンピューティングやエッジコンピューティングを導入する研究がされている [5]。

図 2 に車両ネットワークにおけるエッジコンピューティングの展望を示す。図 2 のように地理的に分散された路側機にエッジサーバが備わっており、車両と路側機は V2I 通信が可能である。車両はアプリケーション実行に必要なタスクを、計算資源が豊富なエッジサーバに転送し、実行することにより、車載コンピュータの負荷軽減を実現する。コアネットワークのエッジ部分に計算資源を配置することで、通信距離の短縮による低遅延化が見込まれている。

### 2.3 エッジコンピューティングによる SLAM 実行

車両の計算負荷を軽減するために、SLAM の実行に必要なタスクをエッジサーバと車両に分割する手法が提案されている [6]。

Ali J ら [7] はエッジコンピューティングによる SLAM の代表的なフレームワークである Edge-SLAM を提案している。図 3 に Edge-SLAM 実行のシステム図を示す。図 3 のように比較的計算負荷の小さいトラッキングは車両で実行し、計算負荷の大きいマッピングとループ閉じ込みをエッジサーバで実行する。マッピングをエッジサーバで実行するため、エッジサーバ側にグローバルマップが配置されエッジサーバが包括するエリアのマップ情報が格納される。車両にはローカルマップが配置され、トラッキングに必要な車両周辺のマップ情報をグローバルマップから更新することでストレージの問題を解決している。

K. L. Wright ら [8] は予測不可能な KF の抽出に対し、車両に必要な計算資源とメモリを削減する KF 送信フレームレートを提案している。連続した KF 間の最小時間ウィンドウを指定することで、近似的に固定周波でのマップ更

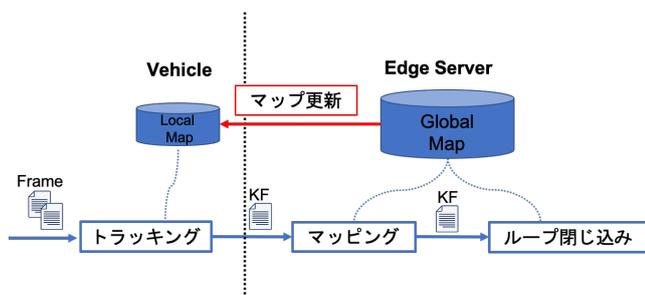


図 3 Edge-SLAM のシステム構成  
Fig. 3 System of Edge-SLAM

新を実現している。

しかし、既存のエッジコンピューティングによる SLAM の実行手法では、車両-エッジ間のトラフィック量は考慮されておらずローカルマップの更新が冗長する可能性がある。そのため、SLAM の精度を維持しながら車両-エッジ間のトラフィックを考慮する最適なマップ更新の決定が必要であると考えられる。

### 3. 提案手法

本章では、SLAM のマップ精度を維持しながら車両-エッジ間のトラフィックを削減するための最適なマップ更新の決定手法を提案する。

#### 3.1 提案概要

提案手法は SLAM のトラッキング精度を維持したまま、車両とエッジサーバ間のトラフィック量を削減することを目的とする。本稿では SLAM の QoS 要求およびネットワーク資源を制約に、車両-エッジ間のトラフィック量の最小化を整数計画問題として定式化する。しかし、整数計画問題は NP 困難である。そこで、車両-エッジ間での連携したトラフィック量を削減するマップ更新とマップデータサイズを近似アルゴリズムによって算出する。近似アルゴリズムでは SLAM のトラッキング精度とトラフィック量に焦点を当てる。トラッキング精度の維持は車両の走行軌道に合わせてマップの更新を決定することで実現する。トラフィック量の削減は前マップ更新からの経過時間を算出し、車両のトラッキングに必要なマップ情報を計算することで実現する。車両の走行軌道は前 KF との相対角度によって算出し、トラッキングモジュールを実行する車両によって計算される。マップ更新の決定およびマップデータサイズの決定は前マップ更新からの経過時間、SLAM のトラッキングに要求される最低限のマップ更新頻度からエッジサーバによって計算される。

#### 3.2 車両-エッジ間のトラフィック量

エッジサーバによって、車両とエッジサーバのトラフィック量が最小になるマップ更新とマップデータサイズを決定

する。トラフィック量とは単位時間当たりに車両-エッジ間の双方向で転送されるデータ量を指す。車両-エッジ間のトラフィック量を整数計画問題に基づいて目的関数 (1) を定式化する。

$$\min \sum_{n=1}^N a_n * D_n \quad (1)$$

制約は以下の式で表される。

$$KF_n > KF_{min} \quad (2)$$

$$\sum_{l \in L} MP_l > MP_{min} \quad (3)$$

$$T_n - T_{n-1} < T_{max} \quad (4)$$

$$a_n \in \{0, 1\} \quad (5)$$

$$D_n < D_W \quad (6)$$

目的関数 (1) において、 $N = \{1, 2, \dots, N\}$  は車両のトラッキングによって抽出された KF,  $A = \{a_1, a_2, \dots, a_N\}$  は  $KF_n$  のマップ更新の決定変数,  $D = \{D_1, D_2, \dots, D_N\}$  は  $KF_n$  は更新マップデータサイズを表す。  $a_n = 1$  の場合、マップデータサイズ  $D_n$  でのマップ更新を実行し、  $a_n = 0$  の場合、マップ更新を実行しない。

制約 (2) (3) (4) は SLAM のトラッキングの精度に関する QoS 制約である。制約 (2) において  $KF_n$  は車両における  $n$  番目の KF 抽出時点でのローカルマップに存在する KF 数であり、  $KF_{min}$  はトラッキングの実行に必要な最低 KF 数を表す。つまり、制約 (2) は車両のマップに更新される KF 数はトラッキングの実行に必要な最低 KF 数を下回らないことを示す。制約 (3) において  $L = \{1, 2, \dots, L\}$  は車両のある時点での車両に保持される KF を表し、  $MP_l$  は車両が保持する  $KF_l$  に含まれる MP 数を示す。つまり制約 (3) は車両のマップに更新される MP 数は車両のトラッキングの実行に必要な最低 MP 数を下回らないことを示す。制約 (4) において  $T_n$  は  $KF_n$  のマップ更新時刻を示し、  $T_{max}$  は SLAM の QoS 要求を満たす許容される最大マップ更新頻度を表す。つまり制約 (4) は前マップ更新から少なくとも許容される最大マップ更新時間は超えないことを示す。制約 (5) はマップを部分的に更新しないことを示す。制約 (6) は更新するマップデータサイズが使用するネットワーク帯域を超えないことを示す。

目的関数 (1) は KF 数に対して整数計画問題であり、NP 困難である。つまり、整数計画問題を複数の部分問題に分割し、それぞれを独立に解く必要がある。

#### 3.3 SLAM の KF 抽出特性

3.2 で述べた整数計画問題を解くために車両の走行軌道に合わせた SLAM の KF 抽出特性について述べる。図 4 に単一のコンピュータで動作した SLAM の実行画面を示

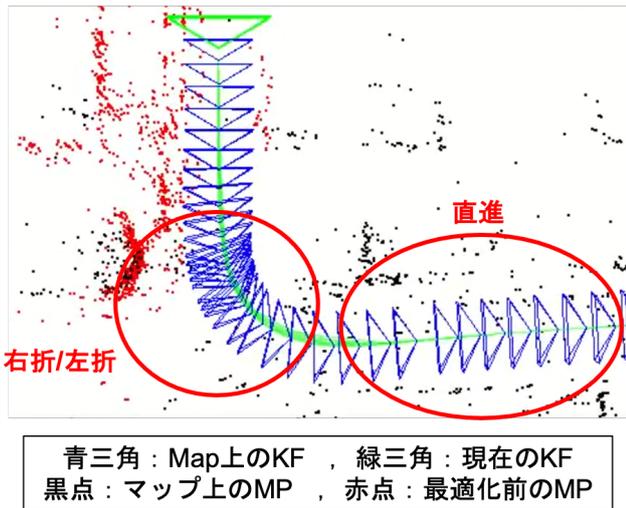


図 4 SLAM の KF 抽出特性  
Fig. 4 Selective Property of SLAM

す。SLAM のトラッキングにおいて直進の際には物理環境の変化量が小さく KF が疎になり、右折や左折には KF が密になることがわかる。直進の場合、新たな MP を検知しないことから車両の速度に合わせた均一な KF の抽出が行われることが考えられる。右左折の場合、車両の速度が低下すること、新たな MP が検知されトラッキングにおいて KF が頻繁に抽出されることが考えられる。そこで直進および右左折を検知するための前 KF との相対角度は以下のように算出する。

$$\theta = |\theta_n - \theta_{n-1}| \quad (7)$$

式 (7) において  $\theta_n$  は  $KF_n$  における世界座標系の角度を示す。つまり直進の場合、 $\theta_n = 0$  となり、左折の場合  $\theta_n > 0$ 、右折の場合  $\theta_n < 0$  となる。また、 $\theta_n$  は SLAM のトラッキングにおける姿勢計算によって算出される。

### 3.4 近似アルゴリズム

3.2 で定式化したトラフィック量の最小化問題を解くために、3.3 で述べた SLAM の KF 抽出特性を用いて、トラッキング精度を考慮した車両-エッジ間のトラフィック量が最小になるマップの更新とマップデータサイズを決定する近似アルゴリズムについて述べる。アルゴリズム 1 は本稿で目的関数 (1) の近似解を求めるアルゴリズムを示す。

アルゴリズム 1 において、車両はエッジサーバに対し相対角度、KF およびローカルマップに存在する KF 数、共有する MP 数を転送する。次に制約 (2) から (6) を満たすマップの更新とデータサイズの決定を行う。マップデータサイズの決定では SLAM の QoS 要求としてマップの更新制約 ITC (Initial Time Constraint) を用いることで必要マップデータサイズを近似的に算出する。また  $L$  は SLAM のトラッキングにおいて許容される物理的距離を示す。このアルゴリズムにより、SLAM の時間的および地理的な

### Algorithm 1 トラフィック量削減手法

**Require:** Relative angle between KFs  $\theta$ , Vehicle's selected KF, Number of KFs in local map  $KF_n$ , Number of MPs shared local map  $MP_l$ .  
**Ensure:** Determined Map update and map data size.

```

for n = 1... N do
     $a_n = 0$ 
    if  $T_n - T_{n-1} < T_{max}$  then
         $a_n = 1$ 
         $D_n = \alpha * |ITC - (T_n - T_{n-1})| < D_W$ 
    end if
    if  $KF_n > KF_{min} \& MP_l > MP_{min}$  then
        if  $k * \theta + V_f * (T_n - T_{n-1}) > L$  then
             $a_n = 1$ 
             $D_n = \alpha * |ITC - (T_n - T_{n-1})| < D_W$ 
        end if
    end if
end for
    
```

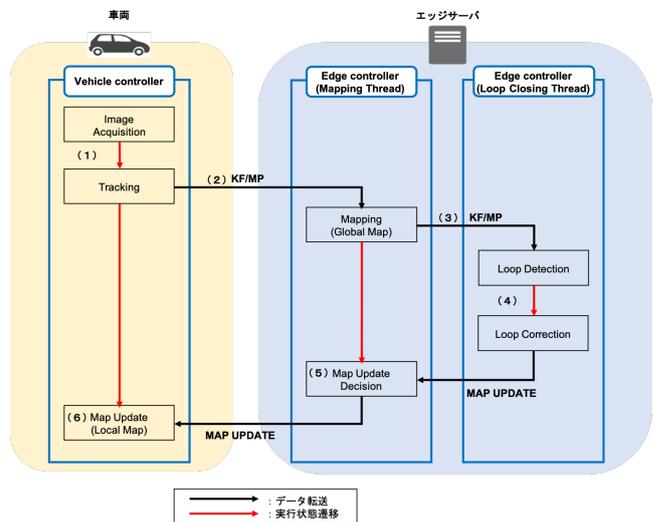


図 5 提案手法のシステム構成  
Fig. 5 System of Proposed Method

QoS 要求を考慮し、トラフィック量を削減するマップ更新とマップデータサイズの決定を実現する。

### 3.5 システム構成とマップ更新決定手順

提案手法のシステム構成と車両-エッジ間のマップ更新手順について述べる。図 5 に提案手法のシステム構成を示す。車両には画像取得、トラッキング、ローカルマップの更新モジュールが備わっている。エッジサーバは 2 つのスレッドで動作しており、それぞれグローバルマップのマッピング、ループ閉じこみを実行する。3.4 で述べた近似アルゴリズムはマッピングのマップ更新決定モジュールによって動作する。ループ閉じ込みは KF 数の増加に伴い数秒の処理遅延が発生するためである。以下に車両-エッジ間のマップ更新手順を示す。

- (1) フレーム取得モジュールによって取得したフレームとローカルマップからトラッキングを実行する。

表 1 実験パラメータ

Table 1 Experimental Parameters.

| パラメータ   | 値                 |
|---------|-------------------|
| SLAM    | ORB-SLAM [9]      |
| データセット  | KITTI [10]        |
| カメラ     | 単眼 Monocular      |
| フレームレート | 30fps             |
| 画像サイズ   | 1241*376px(840KB) |
| ITS     | 300ms             |

- (2) トラッキングによって算出された,  $\theta$ ,  $KF$ ,  $KF_n$ ,  $MP_l$  をエッジサーバに転送する.
- (3) マッピングスレッドでマッピングを実行し, さらに抽出された  $KF$  をループ閉じ込みスレッドに転送する.
- (4) ループ閉じ込みスレッドではループの検知およびループの閉じ込みを実行する.
- (5) マップ更新決定モジュールは, ループ閉じ込みによるグローバルマップの更新が実行されるかタイムアウトした時点で, 近似アルゴリズムによって決定する.
- (6) 車両のローカルマップを更新する.

#### 4. 実験・プロトタイプ実装計画

本章ではプロトタイプを用いた実験の計画について述べる.

##### 4.1 実験環境

提案手法の動作確認および性能評価を行うために, 提案手法のプロトタイプを実装する. 実験では2台のコンピュータをそれぞれ車両, エッジサーバと想定し, 有線接続する. エッジサーバはCPUにAMD Ryzen 7 3700x 8core 16 processorを使用し, メモリは16GB, ストレージは1TBである. 車両はCPUにIntel Pentium G4500 @3.5GHz\*4を使用し, メモリは12GB, ストレージは500MBである. また, 使用するOSはUbuntu 18.04 LTSである. 実装するマップ更新決定手法において, 各コンポーネントの実装にはC++を用い, 3.5で述べたような各計算資源上で必要となるソフトウェアおよびコンポーネントを作成し, 動作させる. トラッキングモジュールとマップ更新は同期しているため, データの転送はSocketによるTCPコネクションを用いる.

実装するマップ更新決定手法の実験パラメータを表1に示す. SLAMはVisual SLAMの中でも一般的なORB-SLAMをベースに実装を行う. 実験にはKITTIのデータセットの使用を予定している. KITTIはドイツのカーlsruエ工科大学による, SLAMによく用いられる車両のための画像データセットである. 画像入力フレームレートは30fpsを想定しており, 1秒間に30枚の画像を読み込む. 画像サイズは1241\*376pxでデータサイズは840KBである.

表 2 単一エッジサーバでのSLAM実行結果

Table 2 SLAM Execution Result on Edge Server

| モジュール        | サブモジュール             | 処理時間 [ms] |
|--------------|---------------------|-----------|
| Tracking     | ORB extraction      | 11.1      |
|              | Initial Pose Est.   | 3.4       |
|              | Track Local Map     | 15.2      |
| Mapping      | Key Frame Insertion | 10.7      |
|              | Map Point Culling   | 1.4       |
|              | Map Point Creation  | 45.5      |
|              | Local BA            | 124.3     |
|              | Kay Frame Culling   | 11.6      |
| Loop Closing | Loop Detection 1st  | 16.2      |
|              | 2nd                 | 18.1      |
|              | 3rd                 | 24.1      |
|              | Loop Correction 1st | 625.8     |
|              | 2nd                 | 930.2     |
|              | 3rd                 | 1329.5    |

##### 4.2 評価項目

実験における評価項目は以下を予定している.

- ネットワークトラフィック量  
車両-エッジ間のネットワークトラフィック量を評価する. トラフィック量の測定にはIftop [11]を使用する予定である. 車両からエッジサーバへのKFおよびエッジサーバから車両へのマップ情報の双方向を評価する.
- トラッキング精度  
車両におけるトラッキング精度を評価する. KITTIデータセットにはGround Truthがないため, 単一のマシンでSLAMを実行し, 画像の欠損がない場合と比較し精度を測定する予定である.

また, 上記の評価項目についてエッジサーバのグローバルマップ更新毎に車両のローカルマップの更新を行う手法と, 既存のEdge-SLAMにおける固定周波でのマップ更新を行う手法と比較する.

##### 4.3 予備実験

提案手法のプロトタイプ実装に関して, 予備実験を行った. 予備実験では単一エッジサーバでのORB-SLAMの動作確認および各モジュールの処理遅延を測定した. 表2に各モジュールの実行時間を示す. 表2からわかるようにトラッキングモジュールはSLAMの処理全体から見ても処理が小さく, マッピングではトラッキングに対し処理時間が約10倍かかることがわかる. また, 使用したKITTIのデータセットでは3回のループを検知しており, KFの増加に伴って処理時間がかかっていることがわかる. これはループ閉じ込みがマップ全体の誤差を最小化する問題を解くことが原因である.

## 5. おわりに

本稿では、SLAMのQoS要求を考慮した車両-エッジ間のトラフィック量を削減する手法の提案を行った。本稿で提案したトラフィック量を削減する手法では、近似アルゴリズムによって車両側のトラッキング精度を考慮したマップの更新とマップデータサイズを決定する。トラッキング精度の維持は車両の走行軌道に合わせてマップの更新を決定することで実現し、トラフィック量の削減は前マップ更新からの経過時間を算出し、車両のトラッキングに必要なマップ情報を計算することで実現する。近似アルゴリズムでは、車両の走行軌道は前KFとの相対角度によって算出し、マップ更新の決定およびマップデータサイズの決定は前マップ更新からの経過時間、SLAMのトラッキングに要求される最低限のマップ更新から算出する。

今後、本稿で提案した車両-エッジ間トラフィック量の削減手法を実装して、実験を行い、動作確認と評価を行う予定である。特にトラッキング精度を測定する方法の検討と実装を進め、実験により車両-エッジ間のトラフィック量が既存手法と比較して削減すること、車両のトラッキング精度を維持することで本提案の有効性を示す予定である。

**謝辞** 本研究はJSPS科研費JP20H04180の助成を受けたものです。

## 参考文献

- [1] X.Wang, G.Zhao, S.Pan, "Overview on simultaneous localization and mapping for uninhabited aerial vehicle vision," Proceedings of the 30th Chinese Control Conference, Jul. 2011.
- [2] S.Milz, G.Arbeiter, C.Witt, B.Abdallah, S.Yogamani, "Visual SLAM for Automated Driving: Exploring the Applications of Deep Learning," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), June 2018.
- [3] 佐竹颯太, 谷遼太郎, 豊田睦, 重野寛, "車載器非搭載車両を考慮した協調認識のためのモバイルエッジコンピューティング支援型車両情報共有システム," 情報処理学会論文誌, Vol.62, No.2, pp.475-483, 2021年2月.
- [4] Liu Haomin, Zhang Guofeng and Bao Hujun, "A survey of monocular simultaneous localization and mapping," Journal of Computer-Aided Design & Computer Graphics, vol. 28, no. 6, pp. 855-868, 2016.
- [5] Salman Raza, Shanguang Wang, Manzoor Ahmed, and Muhammad Rizwan Anwar, "Survey on Vehicular Edge Computing: Architecture, Applications, Technical Issues, and Future Directions," in Wireless Communications and Mobile Computing, 19pages, February 2019.
- [6] Z.ZHOU, X.CHEN, E.LI, L.ZENG, K.LUO, and J.ZHANG "Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing", Proceedings of the IEEE, pp.1738-1762, Jun.2019.
- [7] Ali J, Ben Ali, Zakieh Sadat Hashemifar, and Karthik Dantu "Edge-SLAM: Edge-Assisted Visual Simultaneous Localization and Mapping," In Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services (Toronto, Ontario, Canada)(MobiSys'20)
- [8] K.L.Wright, A.Sivakumar, P.Steenkiste, B.Yu and F.Bai, "CloudSLAM: Edge Offloading of Stateful Vehicular Applications," 2020 IEEE/ACM Symposium on Edge Computing (SEC), 2020, pp.139-151, doi: 10.1109/SEC50012, 2020, 00018.
- [9] R.Mur-Artal and J.D.Tard, "ORB-SLAM2: An Open-Source SLAM System for Monocular Stereo and RGB-D Cameras", IEEE Transactions on Robotics, vol.33, no.5, pp.1255-1262, 2017.
- [10] A.Geiger, P.Lenz and R.Urtasun, "Are we ready for autonomous driving the KITTI vision benchmark suite", 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp.3354-3361, 2012.
- [11] Paul Warren and Chris Lightfoot, "iftop: display bandwidth usage on an interface," [online]. 入手先 (<http://www.ex-parrot.com/pdw/iftop/>) (参照 2021-07-15).