

Vehicular Edge Computingにおける 処理フレームレートを向上するためのEdge支援型SLAM

武藤 晟^{1,a)} 豊田 睦¹ 花輪 麻衣奈² 重野 寛^{1,b)} 森 康祐^{1,c)}

受付日 2022年5月2日, 採録日 2022年11月8日

概要: 自律走行の実現を目指して, 自己位置を正確に把握することや, 自動車周辺をセンシングする自動車アプリケーションが開発されている. 特に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. 既存手法では SLAM を Vehicular Edge Computing (VEC) で処理する研究がされている. しかし, 既存のエッジサーバでの SLAM 処理では限られた自動車の計算資源, V2I 通信のネットワーク帯域幅を考慮できていないという問題点があり, ネットワーク帯域が乏しい環境では, 処理フレームレートが低下してしまう. そこで, 本論文では処理フレームレートの向上を目的とした, 自動車の計算資源と V2I 通信のネットワーク帯域幅を考慮した Edge 支援型 SLAM を提案する. 提案手法のプロトタイプを実装し, 性能評価を行う. 動作確認の結果より, 提案手法が自動車の計算資源と V2I 通信のネットワーク帯域幅を効率的に利用し, 精度に与える影響を最小化する非処理フレームの決定を行っていることを確認した.

キーワード: Vehicular Edge Computing (VEC), SLAM, 車載計算資源, ネットワーク帯域幅

Edge-assisted SLAM to Improve Processing Frame Rate in Vehicular Edge Computing

AKIRA MUTO^{1,a)} MUTSUMI TOYODA¹ MAINA HANAWA² HIROSHI SHIGENO^{1,b)} KOSUKE MORI^{1,c)}

Received: May 2, 2022, Accepted: November 8, 2022

Abstract: A number of automotive applications have been developed to accurately determine self-position and to sense the surroundings of a vehicle in order to realize autonomous driving. In particular, SLAM (Simultaneous Localization and Mapping), which performs both self-positioning and environmental mapping simultaneously, has been attracting attention. In the related studies, SLAM is processed by Vehicular Edge Computing (VEC). However, the existing SLAM processing at the edge does not take into account the limited computational resources of the car and the network bandwidth for V2I communication, so the processing frame rate decreases. In this paper, we propose an edge-assisted SLAM that takes into account the computational resources of the car and the network bandwidth. We implement a prototype of the proposed method that shows the proposed method efficiently utilizes the computational resources of the car and the V2I network bandwidth, and determines the non-processed frames to minimize the impact on the accuracy.

Keywords: vehicular edge computing (VEC), SLAM, onboard computing resources, network bandwidth

¹ 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

² 慶應義塾大学理工学部情報工学科
Faculty of Science and Technology, Keio University,
Yokohama, Kanagawa 223–8522, Japan

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

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

c) kosukemori@keio.jp

1. はじめに

近年, IoV (Internet of Vehicle) の発展にともない遅延に敏感な自動車アプリケーションが開発されている [3]. そこで, 自動車アプリケーションをリアルタイムに実行するために, 自動車ネットワークにクラウドの機能をネットワークエッジに提供するモバイルエッジコンピューティング (MEC) [1]

を導入した VEC (Vehicular Edge Computing) [2] が注目されている。VEC とはネットワークのエッジ部分に計算資源を配置して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。

自動車アプリケーションの 1 つとして、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) [4] が注目を浴びている。SLAM では、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、自動車の限られた計算資源で QoS 要求を満たすことは困難である [13]。そのため、車載計算機の計算資源、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。

VEC による SLAM の実行では、SLAM の一部をエッジサーバにオフロードして実行する。また、SLAM のモジュールの 1 つであるトラッキングでは高精度な自己位置推定を行うためにフレーム間で共有する特徴点である Match Point (MP) を多くとる必要がある。しかし、自動車の高い移動性によって共有する特徴点を維持することは課題である。既存の VEC による SLAM の実行手法では精度に関する議論が多く存在するが自動車の計算資源、V2I (Vehicle to Infrastructure) 通信のネットワーク帯域に関して議論していない。限られた自動車の計算資源では処理フレームレートが低下し、それによって MP 数が減少し精度が低下する。同様に、ネットワーク帯域が乏しい環境ではネットワークの遅延によって処理フレームレートが低下し、精度が低下することが課題である。

本論文では、自動車の計算資源と V2I ネットワーク帯域幅を考慮した Edge 支援型 SLAM を提案する。本論文の目的は、SLAM の精度を維持しながら自動車の計算負荷を軽減することである。提案手法は、3 つのパートに分割できる。1 つ目は、自動車とエッジサーバの共同で SLAM のモジュールを処理することである。エッジサーバと共同で処理することにより、自動車の計算負荷を軽減する。2 つ目は、ネットワーク帯域と自動車の計算資源の制限を考慮して、SLAM の精度を維持する非処理フレームの決定を行う。非処理フレームとは、カメラから取得したフレームのうち、SLAM 処理を行わないフレームである。精度に与える影響度を共有する MP 数をもとに定義し、影響度の小さい非処理フレームを決定する。3 つ目は、MP 数の予測とフレーム配置決定アルゴリズムである。物理環境に依存し予測困難な MP 数を移動平均法によって算出する。フレーム配置決定アルゴリズムでは自動車の計算資源と V2I 通信のネットワーク帯域幅から処理フレームレートを計算し、影響度の大きいフレームからエッジサーバで処理する。エッジサーバで処理することにより、自動車の計算負荷を軽減する。自動車の計算資源と V2I 通信のネットワーク帯域の制限により、処理不可能なフレームを影響度の小さい非処

理フレームとして決定する。以上の 3 つのパートは SLAM の精度を維持しながら自動車の計算負荷を軽減することを實現する。本論文では提案手法のプロトタイプを實現し、計算資源の乏しい自動車と計算資源の潤沢な自動車を想定して実験を行う。実験では SLAM の精度や処理フレームレートおよび自動車の計算負荷の軽減について評価を行う。これにより、本論文で提案する手法が自動車の計算資源とネットワーク帯域を考慮し、SLAM の精度を維持しながら自動車の計算負荷の軽減を實現しているか確認する。

本論文の構成は以下のとおりである。2 章では背景である SLAM や VEC について述べ、VEC 環境における SLAM の実行に関する関連研究について述べる。3 章では提案手法について述べる。4 章では提案手法の實現と実験評価について述べる。5 章では本論文のまとめについて述べる。

2. 関連研究

2.1 VEC と SLAM

本研究の背景として、VEC と SLAM について説明する。自動車の計算負荷を軽減する目的で、自動車ネットワークにクラウドコンピューティング [5] やエッジコンピューティング [6] を導入する研究が行われている。自動車ネットワークにおけるエッジコンピューティングでは、地理的に分散した RSU (Road Side Units) にエッジサーバを搭載し、自動車と RSU の間で V2I 通信が可能である。自動車はアプリケーションの実行に必要なタスクを豊富な計算資源を持つエッジサーバに転送して実行することで、車載計算機の負荷を軽減する。コアネットワークのエッジサーバに計算資源を配置することで、通信距離の短縮によるレイテンシの低減が期待できる。

SLAM は自動運転に必要な自己位置特定と環境マッピングを同時に行うものである。本論文では、カメラを用いる Visual-SLAM に分類される ORB-SLAM2 について言及する。ORB-SLAM2 は、すべてのモジュールが ORB 特徴量と呼ばれる同一の特徴点を使用するため、効率的でシンプル、かつ信頼性の高いシステムを提供することができる。ORB-SLAM2 は、共通の地図情報を参照しながら、トラッキング、マッピング、ループクロージングの 3 つのスレッドで実行される。トラッキングモジュールは、ORB 特徴点を計算し、前フレームの点との対応関係からカメラ姿勢を計算する。また、すべてのフレームをマッピングに使用するのは冗長なので、すべてのフレームからマッピングのための KF (Key Frame) を抽出する。その後、マッピングモジュールで抽出した KF 用いてマッピングを行う。ループクロージングは、環境地図全体の歪みを補正し、累積誤差を最小にするものである。

2.2 エッジサーバにおける SLAM 処理

自動車側の計算負荷を軽減するために、SLAM 実行に必

要なタスクをエッジサーバと自動車側で分担する方式がいくつか提案されている。Ali ら [7] は、エッジコンピューティングに基づく 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。Edge-SLAM では、計算負荷が比較的軽いトラッキングは自動車側で処理され、計算負荷の大きいマッピングとループクロージングはエッジサーバで処理される。グローバルマップはエッジサーバに置かれ、エッジサーバがカバーする範囲の地図情報が格納される。自動車にはローカル地図を配置し、トラッキングに必要な自動車周辺の地図情報はグローバル地図から更新することで、ストレージの問題を解決した。Wright ら [8] は、予測不可能な KF を抽出するために、車両が必要とする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た。連続する KF 間の最小時間窓を指定することで、地図は近似的に一定周期で更新される。Xu ら [9] は、Edge-SLAM にセマンティックセグメンテーションを導入した Semantic-SLAM を提案した。セマンティックセグメンテーションによって動物体を検出することで、動的な情報による精度低下を防ぐことが可能である。

以上の研究において、SLAM の精度について多くの議論がなされているが、自動車の計算資源や V2I 通信のネットワーク帯域幅に関する議論は少ない。自動車の計算資源を考慮しない場合、処理フレームレートが低下し、MP 数が減少することが想定される。また、V2I 通信のネットワーク帯域幅を考慮しない場合、フレーム転送の遅延が発生し、処理フレームレートが低下する可能性がある。

3. 提案

本章では、自動車の計算資源と V2I 通信のネットワーク帯域幅を考慮した Edge 支援型 SLAM を提案する。

3.1 提案概要

提案手法は、SLAM の精度を維持したまま、自動車の計算負荷を軽減することを目的とする。自動車とエッジサーバがつねに通信可能であり、各自動車個別にエッジサーバと通信を行い、SLAM を実行することを想定している。各自動車はステレオカメラを搭載しており、計算資源には限りがある。自動車の計算資源と V2I 通信のネットワーク帯域幅の制限による処理フレームレートの低下は、非処理フレームを決定することで対応する。非処理フレームは、精度に与える影響を考慮し、MP の減少をできるだけ抑えるように決定する。提案手法の主な特徴は以下の 3 点である。

- SLAM のトラッキングモジュールの処理を自動車だけでなくエッジサーバにも分担させて対応する、自動車とエッジサーバで協調処理する Edge 支援型 SLAM により、自動車の計算負荷を軽減する。
- ネットワーク帯域と SLAM の QoS 要求を考慮して、

精度への影響を最小化する非処理フレームを決定する。

- 非処理フレームを決定しつつ、フレームの割当て先を決定するヒューリスティックなアルゴリズムを提案する。自動車の計算機資源と V2I 通信のネットワーク帯域幅から処理フレームレートを算出し、最も影響の大きいフレームをエッジサーバで処理する。自動車の計算資源や V2I 通信のネットワーク帯域幅の制限により処理できないフレームは、影響度の低い非処理フレームとして判断される。

3.2 想定環境

図 1 に提案手法の想定環境を示す。想定環境は、RSU、エッジサーバ、自動車から構成される。RSU は自動車との通信を可能にする。エッジサーバは、SLAM を含む自動車アプリケーションを処理する環境を提供する。自動車は、SLAM モジュールとローカルマップを保持している。提案手法の検討にあたっては、Visual-SLAM の一般的なタイプである ORB-SLAM2 [10] を対象とし、主にトラッキングモジュールの処理に焦点を当てる。

RSU

RSU は通信範囲内の自動車との通信が可能である。隣接する RSU の通信範囲は連続的であり、タスクはつねにサーバにオフロードすることが可能であるとする。RSU はエッジサーバに接続されており、RSU どちらかはコアネットワークを介して通信することができる。

エッジサーバ

エッジサーバは、SLAM を含む自動車アプリケーションを処理する環境を提供する。自動車よりも高い計算能力を持ち、複数の自動車からアプリケーションのリクエストを受け付けることができる。

自動車

自動車には通信機能を持つ車載計算機が搭載されており、RSU の通信範囲に入るとエッジサーバと通信することができる。各自動車は、トラッキングモジュールを独立に実行するために、独自のローカルマップを持っている。提案手法のフレーム割当てアルゴリズムは、フレームごとに実行される。

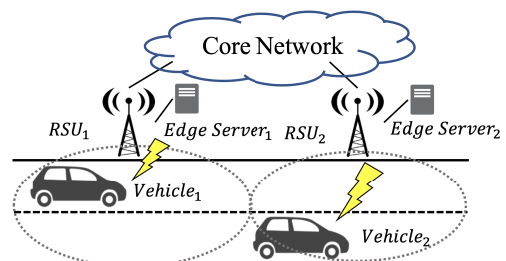


図 1 想定環境

Fig. 1 Assumption environment.

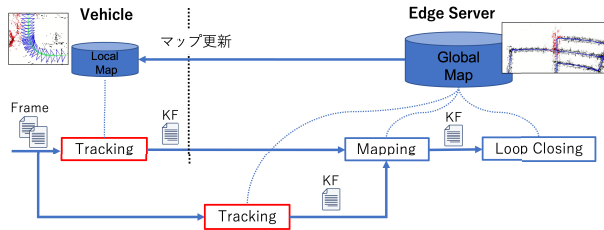


図 2 Edge 支援型 SLAM のシステム図

Fig. 2 System diagram of Edge-assisted SLAM.

3.3 自動車とエッジサーバの協調トラッキング処理

図 2 に提案する Edge 支援型 SLAM のシステム図を示す。自動車の計算負荷を軽減するために、トラッキングモジュールを車両とエッジサーバの両方に配置し、協調してトラッキング処理を行う。エッジサーバにはマッピングモジュールとループクローズングモジュールも配置されている。トラッキングモジュールを自動車で実行する場合とエッジサーバで実行する場合の 2 つの場合がある。自動車ですらトラッキングモジュールを実行した場合、抽出された KF はエッジサーバに転送される。エッジサーバでトラッキングモジュールを実行した場合、自動車は取得したフレームを直接エッジサーバに転送する。エッジサーバは、受信したフレームに対してトラッキングモジュールを実行し、KF を抽出する。

3.4 非処理フレームの決定

V2I 通信のネットワーク帯域幅や自動車の計算機資源が不足すると、処理できないフレームが発生する。処理できないフレームが SLAM にとって重要なフレームである場合、SLAM 精度が大きく低下する可能性がある。そこで、SLAM の精度に大きな影響を与えるフレームを処理から除外しないように、非処理フレームが SLAM のトラッキング精度に与える影響を計算し、影響度を最小化するように非処理フレームを決定する。非処理フレーム n の影響度 I_n を式 (1) に示す。

$$I_n = P_{n,n-k} - P_{n+1,n-k} \quad (1)$$

ここで、 $P_{i,j}$ はフレーム i とフレーム j の間に取得した MP 数、 $n-k$ はトラッキングが処理された最後のフレームのインデックスを示している。すなわち I_n は、トラッキングモジュールがフレーム n を処理する場合としない場合の MP 数の差を示す。

影響度の計算に使用する MP の数は、自動車の速度や周囲の物理環境に影響されるパラメータであり、予測し決定する必要がある。MP 数の予測については移動平均法を用いる。式 (2) に、移動平均法で算出された MP 数 $P_{n,n-k}$ を示す。

$$P_{n,n-k} = \frac{1}{m} \sum_{i=n-1}^{n-m} P_{i,i-k} \quad (2)$$

ここで、 m は移動平均法の次元数である。

ある単位時間の中で非処理フレームを決定する問題を以下の整数計画問題として定式化する。

$$\min \sum_{n=1}^N a_n I_n \quad (3)$$

制約は以下の式で表される。

$$S_n \left(N - \sum_{n=1}^N a_n \right) < B, \quad (4)$$

$$P_{n,n-k} > P_{min}, \quad (5)$$

$$\sum_{n=1}^N t_n < T_{max}, \quad (6)$$

$$a_n = \{0, 1\} \quad (7)$$

ここで、 N は単位時間あたりに自動車取得するフレーム数、 n は単位時間内のフレームのインデックスである。 a_n は決定変数であり、フレーム n を非処理フレームとして判定するか否かを示す変数であり、 $a_n = 1$ のとき、フレーム n は非処理フレーム、 $a_n = 0$ のとき、処理フレームとする。式 (4) は、単位時間あたりの自動車からエッジサーバへの転送データ量が V2I 通信のネットワーク帯域より小さいという制約を示す。 S_n はフレーム n のデータサイズ、 B は V2I 通信のネットワーク帯域幅である。式 (5) は、 P_{min} を保証することでトラッキング精度を維持する制約を示す。 P_{min} は ORB-SLAM2 で経験的に算出された追従精度を維持するために必要な MP 数の最小値である。式 (6) は、単位時間あたりのすべてのフレームを許容時間内に処理するという時間制約を示す。 t_n はトラッキングモジュールにおけるフレーム n の処理時間、 T_{max} は単位時間あたりにすべてのフレームを処理する許容時間である。自動車内でフレーム n を処理する場合、 t_n は車両内のトラッキングモジュールのみの処理時間である。エッジサーバで処理する場合、 t_n は自動車がフレームをエッジサーバに転送してから、エッジサーバがトラッキングモジュールを処理し、ローカルマップを更新するまでの時間である。そして、式 (7) は、非処理フレームの決定変数がバイナリであり、フレームが部分的に処理されないという制約を示す。また、フレーム数に対して NP 困難である。

3.5 処理フレーム決定アルゴリズム

単位時間あたりの全フレームを自動車、エッジサーバのどちらに配置するかを決定するとともに、非処理フレームの決定を行う。ヒューリスティックなアルゴリズムを提案する。アルゴリズムを Algorithm 1 に示す。トラッキングモジュールが車両で処理される際の処理時間 t_v と、エッジサーバで処理される際の時間 t_s から、自動車が処理可能なフレームレートを算出する (1 行目)。自動車は、式 (2) を用いて $frame_n$ からトラッキングモジュールで処理された最後の $frame_{n-k}$ までの MP 数を予測し、式 (1) を用い

Algorithm 1 処理フレーム決定アルゴリズム

Input: frame 1, 2, ..., N, B, t_s , t_v , Numbers of MP, S_n
Output: Determine allocation of all frames per unit time

- 1: Calculate the frame rates f_v and f_s that can be processed by the vehicle and the edge server
- 2: Prediction of $P_{n,n-k}$ of $frame_n$ using Eq. (7)
- 3: Calculate I_n using Eq. (1)
- 4: Sort in descending order of impact I_n
- 5: **for** $i = 1$ to N **do**
- 6: **if** ($S_i * f_s < B$) & ($t_s < T_{max}$) **then**
- 7: Processing frames on the edge server
- 8: **else if** ($f_v < N - i$) **then**
- 9: Processing frames on vehicle
- 10: **else**
- 11: Determined as non-processed frame
- 12: **end if**
- 13: **end for**

て $frame_n$ の影響 I_n を算出する (2, 3 行目). t_v と t_s , $P_{n,n-k}$ は ORB-SLAM2 初期化における次元と同じ時間に算出される. 影響度の小さいフレームを非処理フレームとして判定できるように, 制約 (5) を満たすフレームを影響度の大きい順にソートする (4 行目). 各フレームのデータサイズが V2I 通信のネットワーク帯域幅以下であり, t_s が T_{max} 以下であれば, ソートされたフレームの中で最も影響度が大きいフレームがエッジサーバに配置される (5, 6 行目). V2I 通信のネットワーク帯域幅が不足し, 自動車の計算資源で処理可能な場合は, 自動車に配置する (7, 8 行目). エッジサーバと自動車に配置できないフレームは非処理フレームとして判定する (9, 10 行目). エッジサーバにフレームを優先的に配置することで, 自動車の計算負荷を軽減することが可能である. また, エッジサーバと自動車の 2 つの計算資源で処理できない, 最も影響度の小さいフレームを決定することで, SLAM のトラッキング精度を維持することができる.

4. プロトタイプシステムの実装と評価

4.1 プロトタイプシステムの実装

提案手法の動作確認および性能評価を行うために, 研究室内にテスト環境を構築し, プロトタイプシステムを実装した. 図 3 にプロトタイプシステムの構成を示す. プロトタイプシステムは, 自動車とエッジサーバを模擬する PC を用いて実装した. 自動車コンポーネントは, フレーム取得 (Image Acquisition), フレーム配置アルゴリズムモジュール (Frame allocation algorithm), トラッキングモジュール (Tracking), 地図更新モジュール (Map Update), ローカル地図 (Local Map) で構成されている. エッジサーバコンポーネントは, トラッキングモジュール, マッピングモジュール (Mapping), ループクローズングモジュール (Loop Closing), マップアップデートモジュール, グローバルマップ (Global Map) から構成されている. 両コン

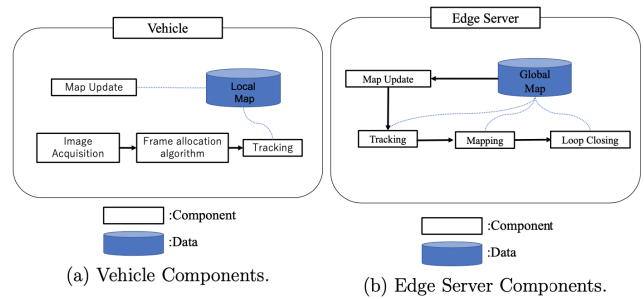


図 3 プロトタイプシステムの構成要素
Fig. 3 Prototype system components.

表 1 評価環境

Table 1 Evaluation environment.

| | | |
|-------------|-----|--|
| Edge Server | 製品 | Dell XPS |
| | CPU | 11th Gen Intel(R)core(TM) i7-11700 CPU@2.5 GHz × 16 |
| | メモリ | 16 GB |
| Vehicle1 | 製品 | Raspberry Pi |
| | CPU | BCM2711, Quad-core Cortex-A72 (ARM v8) 64 bit@1.5 GHz |
| | メモリ | 4 GB |
| Vehicle2 | 製品 | Intel Pentium(R) |
| | CPU | CPUG4560@3.5 GHz × 4 |
| | メモリ | 8 GB |

表 2 実験パラメータ

Table 2 Experimental parameters.

| | |
|-----------------|------------------|
| カメラモード | ステレオ |
| カメラフレームレート | 10 [fps] |
| フレームサイズ | 1,226 × 370 [px] |
| フレームデータサイズ | 260 [KB] |
| ネットワーク帯域幅 (V2I) | 10~50 [Mbps] |
| 通信遅延 | 5 [ms] |
| P_{min} | 30 [point] |
| m | 5 |
| T_{max} | 1 [sec] |

ポーネントに, 処理とデータ転送のために複数のスレッドが存在する. 実装にあたり, 自動車に対して, 計算資源が限られた環境と, 計算資源が豊富な環境の 2 つを想定した. 表 1 に評価環境を示す. 自動車を模擬した PC とアクセスポイントの接続には IEEE 802.11ac を用いた. アクセスポイントとエッジサーバを模擬した PC は Ethernet で接続した. 自動車とエッジサーバ間は TCP/IP ソケット通信を使用した. 各計算資源間のルーティングテーブルはあらかじめ設定されているものとする.

4.2 評価方法

実装したプロトタイプシステムを用いて評価実験を行った. 表 2 に実験パラメータを示す. カメラ画像には KITTI データセット [11] を使用した. カメラモードはステレオ

であり、カメラのフレームレートは 10 fps、フレームサイズは $1,226 \times 370$ px である。V2I 通信のネットワーク帯域幅は LTEV2X や NRv2X を想定し、10 から 50 Mbps とした。実装では tc コマンド [12] を用いて意図的に 5 ms の遅延を付与し、通信距離による遅延を再現した。実験環境は直線道路と右折を含む道路を想定した。一般道のデータを使用しているため、トラッキングモジュールは各フレームにおいて、周辺環境の木や建物など、約 1,000 点の ORB 特徴点を検出することが可能である。すなわち、毎秒 10 フレームでカメラの姿勢と地図情報の結果を得ることができる。また、MP 数の予測に用いる移動平均法の次元数 m は 5 である。

4.3 評価項目

実験における評価項目は以下のとおりである。

平均処理フレームレート

1 秒間に処理されるフレーム数を示す。トラッキングモジュールの処理時間から、提案手法が自動車とエッジサーバの限られた計算資源を効率的に利用していることを確認する。

トラッキング精度

1 m あたりの軌跡誤差を示す。軌跡誤差を E_t として、式 (8) に示す。

$$E_t = \sqrt{dx^2 + dy^2 + dz^2} \quad (8)$$

ここで、 dx , dy , dz は実際の位置と測定した位置との相対位置誤差を示す。相対位置誤差は、実際の位置と測定した位置それぞれにおいて、自動車が 1 m 進んだ際の最初のフレームと最後のフレームの位置から算出される。軌跡誤差の単位% は単位メートルあたりの誤差であり、1% の誤差とは自動車が 1 m 進むたびに 1 m の誤差が生じるという意味である。提案手法が、自動車の計算負荷を軽減することで SLAM 精度を向上させるとともに、影響の少ないフレームを非処理フレームとして判断することで、フレームレートの低下による精度低下を抑えることを確認する。

自動車の CPU 使用率

実行時間に対する自動車の CPU 使用率を示す。CPU 使用率は C++ の関数である `brief` を用いて計測した。提案手法では、自動車とエッジサーバでトラッキングモジュールを共同処理することで、自動車の計算負荷を軽減させていることを確認する。

4.4 平均処理フレームレート

図 4 に Vehicle1 における平均処理フレームレートを示す。図 4 に示すように、提案手法は既存手法に比べて高い処理フレームレートが得られていることが分かる。特に、ネットワーク帯域が 40~50 Mbps の場合、提案手法の処理フレームレートは 8 fps を示しており、ORB-SLAM2 より



図 4 Vehicle1 の平均処理フレームレート

Fig. 4 Average processing frame rate of Vehicle1.

も 4.7 fps、Edge-SLAM よりも 5 fps 高い処理フレームレートが得られた。Vehicle1 の場合、カメラのフレームレートである 10 fps の処理フレームレートを維持することは困難である。これは、Vehicle1 の計算資源が乏しく、非処理フレームが発生するためである。ORB-SLAM2 と比較して処理フレームレートが向上したことは、提案手法がネットワークリソースを効率的に利用し、トラッキング処理をエッジサーバで行っていることを示している。Edge-SLAM と比較して処理フレームレートが向上していることから、提案手法は自動車の計算負荷を考慮している。また、提案手法のアルゴリズムは、エッジサーバでの処理を優先するため、自動車の計算負荷が軽減され、ネットワーク帯域幅が 10 Mbps の場合でも、他の手法と比較して処理フレーム比率 60% 以上低くなることが確認された。以上のことから、提案手法では、自動車とネットワークのリソースを効率的に利用し、処理フレームレートを向上させることが確認された。

4.5 トラッキング精度

図 5、図 6 に、Vehicle1 の直線、右折路それぞれにおけるトラッキング精度を示す。図 5 の直線路の場合、V2I 通信のネットワーク帯域幅によらず、他の方式と比較して Vehicle1 の精度が向上していることが分かる。特に、ネットワーク帯域が広い場合、ORB-SLAM2 と比較して最大 0.33%、Edge-SLAM と比較して最大 0.19% 軌道の誤差が改善された。

図 6 に示すように、右折路でもトラッキング精度が改善されたことが分かる。特に、ネットワーク帯域が広い場合、ORB-SLAM2 に比べ最大 0.43%、Edge-SLAM に比べ最大 0.26% 軌跡誤差が改善された。これらの結果から、提案手法はどのような状況でも精度を向上させることが分かった。

4.6 自動車の CPU 使用率

図 7、図 8 に、Vehicle1、Vehicle2 それぞれにおける CPU

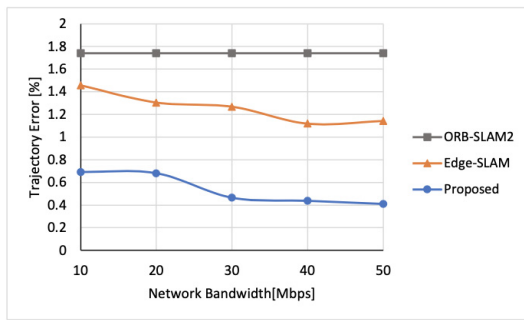


図 5 直線路における Vehicle1 の軌跡誤差

Fig. 5 Trajectory error of Vehicle1 on a straight road.

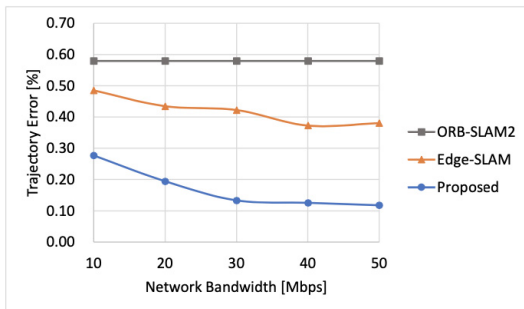


図 6 右折路における Vehicle1 の軌跡誤差

Fig. 6 Trajectory error of Vehicle1 on a right-turn road.

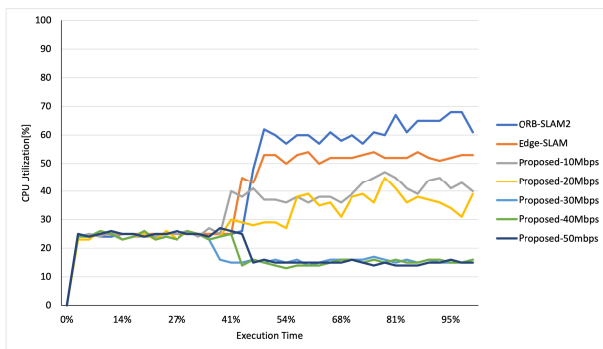


図 7 Vehicle1 の CPU 使用率

Fig. 7 CPU usage for Vehicle1.

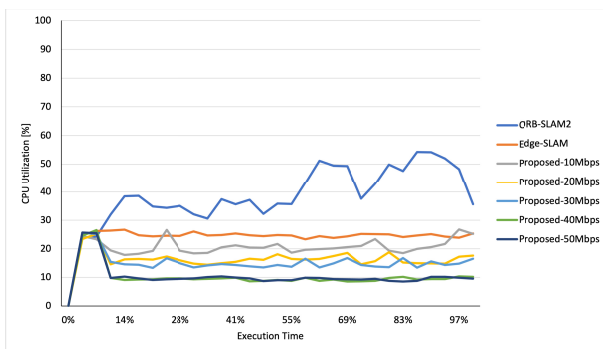


図 8 Vehicle2 の CPU 使用率

Fig. 8 CPU usage for Vehicle2.

使用率の推移を示す。図 7 に示すように、30~50 Mbps の広帯域ネットワークにおいて、提案方式は ORB-SLAM2 と比較して最大 49%、Edge-SLAM と比較して最大 33% の

CPU 使用率の削減が可能である。提案方式は、狭帯域ネットワーク環境において、他の手法と比較して CPU 使用率を削減することができる。

図 8 に示すように、30~50 Mbps の広いネットワーク帯域において、提案方式は ORB-SLAM2 と比較して最大 47%、Edge-SLAM と比較して最大 19% の CPU 使用率の削減が可能であることが分かる。これは、提案手法が Vehicle2 のような広いネットワーク帯域幅の環境において、ネットワーク資源を効率的に使用できるためである。提案方式ではエッジサーバの処理を優先させるため、ネットワーク遅延により処理フレームレートが低下する。しかし、他の方式と比較して CPU 使用率が大きく減少しており、提案方式が有効性が示された。

5. おわりに

本論文では、自動車の計算量と V2I 通信のネットワーク帯域幅を考慮して、SLAM の精度を維持しつつ自動車の計算負荷を軽減するために、取得したフレームを自動車とエッジサーバのどちらに割り当てるのかを決定する、Edge 支援型 SLAM を提案した。提案手法では、自動車とエッジサーバで SLAM モジュールを協調処理することで、自動車の計算負荷を軽減する。また、ネットワーク帯域と自動車の計算機資源の制限を考慮し、SLAM の精度を維持する非処理フレームを決定する。さらに、フレームアロケーションアルゴリズムで自動車の計算資源と V2I 通信のネットワーク帯域幅から処理フレームレートを算出し、最も影響の大きいフレームをエッジサーバで処理する。

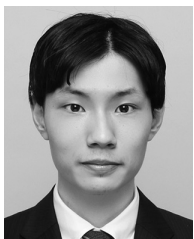
性能評価のために、提案手法のシステムを実装し、計算資源が限られた自動車と資源が豊富な自動車の 2 種類を対象に実験を行った。実験では、道路状況やネットワーク帯域を変化させながら、処理フレームレート、SLAM 精度、自動車の計算負荷の軽減を評価した。評価の結果、提案手法は自動車の計算資源や V2I 通信のネットワーク帯域幅を考慮し、SLAM の精度を維持したまま自動車の計算負荷を軽減できると結論づけられる。

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] ETSI-European Telecommunications Standards Institute (2015), ETSI GS MEC-IEG 004 V1.1.1 (2015-11)-Mobile-Edge Computing (MEC); Service Scenarios (online), available from https://www.etsi.org/deliver/etsi_gs/MEC-IEG/001_099/004/01.01.01_60/gs_MEC-IEG004v010101p.pdf(accessed 2021-01-08).
- [2] Raza, S., Wang, S., Ahmed, M. and Anwar, M.R.: A survey on vehicular edge computing: Architecture, applications, technical issues, and future directions, *Wireless Communications and Mobile Computing* (2019).
- [3] Eiter, T., Fureder, H., Kasslatter, F., Parreira, J.X. and

- Schneider, P.: Towards a semantically enriched local dynamic map, *International Journal of Intelligent Transportation Systems Research*, Vol.17, pp.32–48 (2019).
- [4] Wang, X., Zhao, G. and Pan, S.: Overview on simultaneous localization and mapping for uninhabited aerial vehicle vision, *Proc. 30th Chinese Control Conference*, pp.1–14 (2011).
- [5] Ataallah, S.M.A., Nassar, S.M. and Hemayed, E.E.: Fault tolerance in cloud computing – Survey, *2015 11th International Computer Engineering Conference (ICENCO)*, pp.241–245 (2015).
- [6] Toyoda, M., Satake, H. and Shigeno, H.: A dynamic task offloading method with centralized controller to improve task success rate, *International Workshop on Pervasive Information Flow (PerFlow '2021)*, pp.483–488 (2021).
- [7] Ali, A.J.B., Hashemifar, Z.S. and Dantu, K.: Edge-SLAM: Edge-assisted visual simultaneous localization and mapping, *Proc. 18th International Conference on Mobile Systems, Applications, and Services*, pp.325–337 (2020).
- [8] Wright, K.L., Sivakumar, A., Steenkiste, P., Yu, B. and Bai, F.: CloudSLAM: Edge offloading of stateful vehicular applications, *2020 IEEE/ACM Symposium on Edge Computing (SEC)*, pp.139–151 (2020).
- [9] Xu, J., Cao, H., Li, D., Huang, K., Qian, C., Shangguan, L. and Yang, Z.: Edge assisted mobile semantic visual SLAM, *IEEE INFOCOM 2020 – IEEE Conference on Computer Communications*, pp.1828–1837 (2020).
- [10] Mur-Artal, R. and Tard, J.D.: ORB-SLAM2: An open source SLAM system for monocular stereo and RGB-D cameras, *IEEE Trans. Robotics*, Vol.33, No.5, pp.1255–1262 (2017).
- [11] Geiger, A., Lenz, P. and Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite, *IEEE Conference on Computer Vision and Pattern Recognition*, pp.3354–3361 (2012).
- [12] Manpage maintained by bert hubert (ahu@ds9a.nl), tc-show / manipulate traffic control settings (online), available from <http://man7.org/linux/man-pages/man8/tc.8.html> (accessed 2021-01-08).
- [13] Li, Z., Wang, L., Jiang, L. and Xu, C.: FC-SLAM: Federated learning enhanced distributed visual-LiDAR SLAM in cloud robotic system, *2019 IEEE International Conference on Robotics and Biomimetics (RO-BIO)*, pp.1995–2000 (2019).



武藤 辰 (学生会員)

2021年慶應義塾大学理工学部情報工学科卒業。現在、同大学大学院理工学研究科修士課程在学中。



豊田 睦 (学生会員)

2020年慶應義塾大学理工学部情報工学科卒業。同大学大学院理工学研究科修士課程修了。



花輪 麻衣奈

2022年慶應義塾大学理工学部情報工学科卒業。



重野 寛 (正会員)

1990年慶應義塾大学理工学部計測工学科卒業。1997年同大学大学院理工学研究科博士課程修了。現在、同大学理工学部教授。博士(工学)。情報処理学会論文誌編集委員、同DPS研究会主査、同理事、同ITS研究会主査、電子情報通信学会英文論文誌B編集委員、Secretary of IEEE ComSoc APB等を歴任。現在、情報処理学会ITS研究会主査、Co-Chair of IEEE ComSoc APB CCC。ネットワーク・プロトコル、ITS等の研究に従事。著書『ユビキタスコンピューティング』(オーム社)、『情報学基礎第2版』(共立出版)等。電子情報通信学会、IEEE、ACM各会員。



森 康祐 (正会員)

2014年慶應義塾大学理工学部情報工学科卒業。2016年同大学大学院理工学研究科前期博士課程修了。現在、同大学理工学部助教(有期)。電子情報通信学会、IEEE各会員。