

# 類似画像検索と SLAM の協調的位置推定における負荷分散法

山崎康平<sup>†</sup> 宋戸英彦<sup>‡</sup> 北原格<sup>†</sup> 亀田能成<sup>†</sup>

**概要**：歩行者ナビゲーションシステムには自己位置推定が必要であり、頑健性の高い手法が求められる。我々は利用場所を問わない自己位置推定手法としてカメラを用いた手法に取り組んでいる。これまでの取り組みでは類似画像検索と SLAM を組み合わせて自己位置推定を行う方法を考案し、頑健性の調査を行ってきた。予備調査の結果からこの 2 つの手法を 1 つの計算機資源上で実行するには負荷が大きく実現が難しいことがわかっている。本稿では、1 つの計算機資源上で実行するには負荷が大きく実現が難しい類似画像検索と SLAM の 2 つについて別々の計算機資源上で実行することで協調的位置推定を行う負荷分散法を提案する。

**キーワード**：自己位置推定，ナビゲーション，RealSense，ZeroMQ

## Load Distribution of Harmonic Position Estimation of Image Retrieval and SLAM

KOHEI YAMASAKI<sup>†</sup> SHISHIDO HIDEHIKO<sup>‡</sup>  
ITARU KITAHARA<sup>†</sup> YOSHINARI KAMEDA<sup>†</sup>

### 1. はじめに

歩行者を対象としたナビゲーションシステムには頑健性の高い自己位置推定手法が必須である。特に視覚障がい者を対象としたナビゲーションには歩行したい範囲で利用場所を問わず、頑健性の高い自己位置推定手法が必要となる。自己位置推定手法の一つとして GPS が挙げられるが、地下や屋内では利用できない。RFID(Radio Frequency Identifier) タグ、Wi-Fi の信号強度を用いた自己位置推定手法では電波の届く範囲では頑健性の高い位置推定が可能であるが、範囲外では利用できない問題がある。

我々は視覚障がい者の外出では経路を予め決めておくという前提 [1] を利用し、先に述べた問題を解決する手法としてカメラを用いた位置推定に取り組んでいる。利用場所を問わないカメラのみを用いた手法として類似画像検索と RealSense SLAM(以下 SLAM)による位置推定をそれぞれ検証し、頑健性の調査を行ってきた。予備調査の結果からこの 2 つの手法を組み合わせることで頑健性の高い自己位置推定が期待できると考え、本稿では類似画像検索と SLAM を同時に実行することとした。しかし、予備調査の評価での事実としてそれぞれの自己位置推定手法の処理によって計算機資源にかかる負荷が大きくなることがわかっている。そのため、歩行者が利用するウェアラブルなデバイス上で 2 つの自己位置推定手法を 1 つの計算機資源上で実行することは現実的に難しい。

本稿では、1 つの計算機資源上で実行するには負荷が大きく実現が難しい類似画像検索と SLAM の 2 つについて別々の計算機資源上で実行することで協調的位置推定を行う負荷

分散法を提案する。

### 2. 自己位置推定手法

#### 2.1 類似画像検索

類似画像検索による位置推定手法では釜坂らの手法[2]と同様の手法を用いる。類似画像検索では予めナビゲーションを行う経路の映像を事前に撮影しておく。事前撮影映像を数フレームごとに分割し類似画像検索の参照画像とする。参照画像から特徴点を検出し、キーとしてデータベースに登録する(図 1(a))。実際に位置推定を行う際には事前撮影映像と同じ経路を歩行する。実時間で撮影されている映像フレームのキーとデータベースに保存されている画像のキーの一致率を利用し、データベースから参照画像を選択することで位置推定を行う(図 1(b))。

#### 2.2 SLAM

カメラを用いる Visual SLAM[3]は撮影された映像から画像特徴量などの情報をもとにカメラの位置姿勢推定と撮影環境の 3 次元地図作成を同時に行う。Visual SLAM の中には RGB-D カメラを利用した Dense SLAM[4]や ORB 特徴量を利用した SLAM[5]などがある。本稿ではハードウェアとして Intel® RealSense™ Camera ZR300 (図 2) を用いる。ZR300 で利用できる RealSense SLAM ライブラリを導入する。ZR300 には可視光カメラ、魚眼カメラ、加速度計、ジャイロスコープ、およびデプスカメラが搭載されており RealSense SLAM ではこれらの情報を利用して位置推定を行う(図 3)。RealSense SLAM ライブラリでは図 4 に示す

<sup>†</sup> 筑波大学 大学院 システム情報工学研究科

<sup>‡</sup> 筑波大学 計算科学研究センター

データフローとなっている。

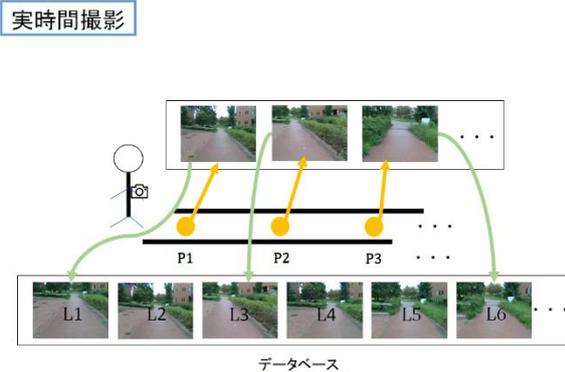
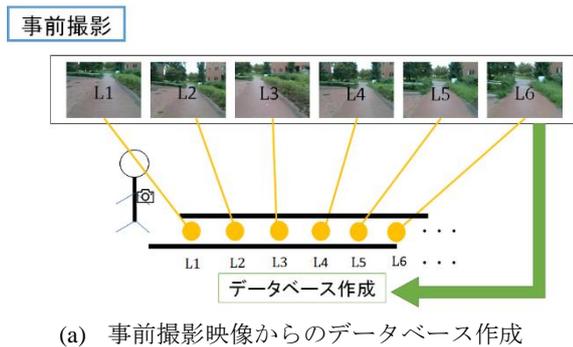


図 1 類似画像検索



図 2 Intel®RealSense™Camera ZR300[6]

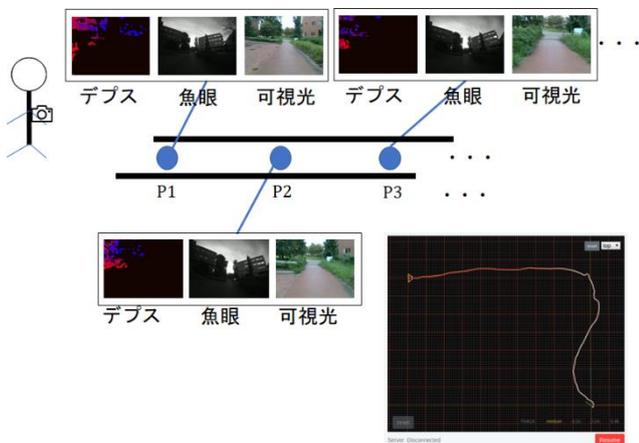


図 3 SLAM

### SLAM data flow

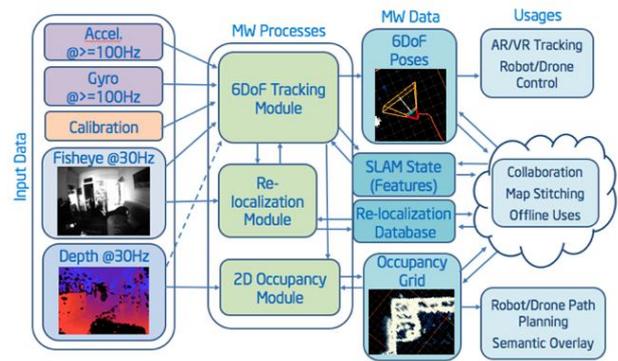


図 4 RealSense SLAM データフロー[7]

### 3. 提案手法

類似画像検索と SLAM の 2 つの自己位置推定手法を 1 つの計算資源上で実行するには負荷が大きく実現が難しい。

2 つの手法をもし 1 つの計算資源上で同時に実行すれば、類似画像検索では入力画像から類似度が第一位の参照画像を検索するためにかかる時間の増大を引き起こす可能性がある。SLAM では処理速度の低下によって単位時間あたりで処理できるフレーム数が減り、結果として位置推定に失敗する可能性が大きくなる。カメラの動きが大きいときにこの傾向が顕著になると考えられる。

そこで、2 つについて別々の計算資源上で実行することで協調的位置推定を行う負荷分散法を提案する。提案手法の概要を図 5 に示す。

まず、2 つに必要な情報をあげる。

類似画像検索：可視光カメラから取得されるグレースケール画像

SLAM：Intel® RealSense™ Camera ZR300 から得られるセンサ情報（デプスカメラ，魚眼カメラ，加速度計，ジャイロスコープ）

これらの情報から、ウェアラブル計算機資源上のプロセスからリモート計算機資源上のプロセスに ZR300 の可視光カメラから取得した画像を送信する機構を作成することができれば 2 つの処理を別々の計算資源上で実行できると考えた。

リモート計算機資源上で類似画像検索を実行しウェアラブル計算機資源に結果を送信するプロセス，ウェアラブル計算機資源上でユーザに結果を返す機能とリモート計算機資源上で実行された類似画像検索の実行結果を受信する機能を有する統括プロセスを作成することで 2 つの位置推定手法を組み合わせる方法を考案した。

本稿ではネットワークを介してデータのやり取りを行うことで類似画像検索，SLAM それぞれの処理を別々の計算資源上で実行し協調的位置推定を行う負荷分散法を提案

する。

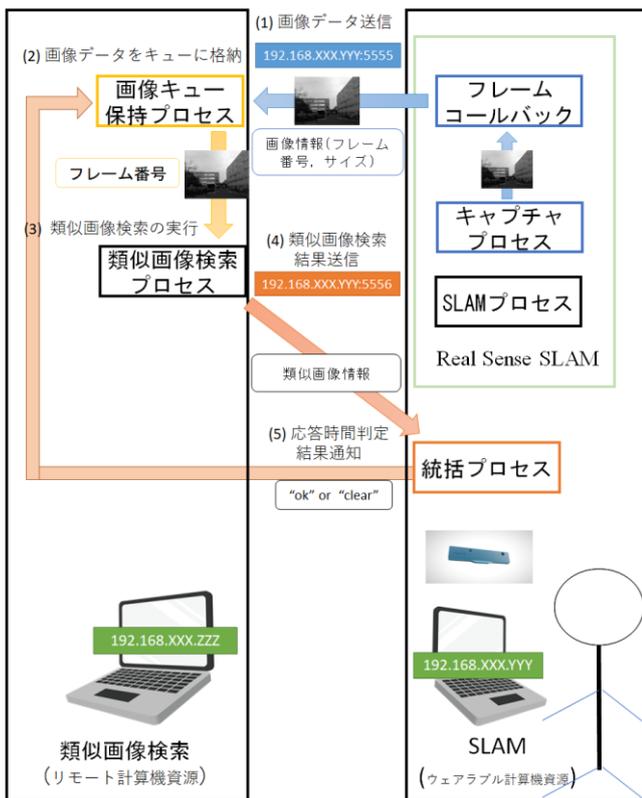


図5 提案手法 プロセスフロー

図5に示す手順について説明する。2つのプロセス間のデータ送受信にはZeroMQ[5]というライブラリを利用している。

(1) 画像データ送信

RealSense SLAM ライブラリで動作する SLAM プロセスでは処理実行の際に毎フレームコールバックが呼ばれる。フレームコールバック内では RealSense SLAM のキャプチャプロセスで取得された可視光カメラの画像を取得することができる。フレームコールバックから画像キュー保持プロセスに TCP/IP を利用し「画像データ」、「フレーム番号」、「画像タイプ (グレースケール)」、「画像サイズ (640)」、「画像サイズ (480)」を送信する。

(2) 画像データをキューに格納

フレームコールバックから送信された「画像データ」、「フレーム番号」、「画像タイプ (グレースケール)」、「画像サイズ (640)」、「画像サイズ (480)」を画像キュー保存プロセスが受信し「画像データ」、「フレーム番号」をキューに格納する。画像サイズは画像データが正常に受信されたか判断するために利用する。

(3) 類似画像検索の実行

キューに格納されている画像データとフレーム番号を取り出し類似画像検索を実行する。

(4) 類似画像検索 結果送信

類似画像検索を実行した結果第一位となった参照画像の情報を類似画像検索プロセスから統括プロセスに送信する。

(5) 応答時間判定 結果通知

統括プロセスにおいて計測した(1)から(4)までに要した時間が一定時間内であれば“ok” (そのまま続行)、一定時間を超えていれば“clear”を画像キュー保持プロセスに送信する。“clear”を受け取った場合はキューを空にする。

4. 実験

提案手法による協調的位置推定を行う負荷分散を実際に2つの計算資源上で実行した。実行時間の計測などから考察を行う。

4.1 実験環境

利用する PC を表1に示す。ウェアラブル計算機資源が iiyama STYLE, リモート計算機資源が Surface Pro である。今回は、図6に示すようにそれぞれの PC を 100BASE-TX の有線接続で通信を行った。

表1 検証に使用した PC

| PC名             | CPU              | RAM  | OS                  |
|-----------------|------------------|------|---------------------|
| iiyama<br>STYLE | Corei7-<br>8550U | 16GB | Ubuntu<br>16.04     |
| Surface Pro     | Corei7-<br>6550U | 16GB | Windows 10<br>64bit |



図6 実験環境 PC 接続

検証を行う前に経路の事前撮影を行い類似画像検索で利用するデータベースを作成する。本稿では類似画像検索に画像特徴量に SURF を利用しデータベース作成と位置推定を行う。事前撮影とデータベースの作成には iiyama STYLE を使用した。Intel® RealSense™ Camera ZR300 に搭載されている可視光カメラを用いてフレームレート 30fps, 画像サイズ 640x480 画素で撮影を行った。撮影経路は筑波大学敷地内の屋内外を含む経路を対象とした。撮影した映像の合計フレームは 11535 フレーム、経路の長さは約 400m である。撮影した経路の一部を図 7 に示す。



3 節で述べたそれぞれのプロセスにおいて 1 回のやりとりで送受信されるデータ量は以下の通りである。

・画像データ送信

画像データ(グレースケール) : 640x480x1 bytes = 307200 bytes = 300 kbytes

画像情報(サイズ, タイプ, フレーム番号) : 4×4 bytes = 16bytes

・結果送信

類似画像検索による位置推定結果 : 最高 80 bytes

・応答時間判定

結果送信に対する reply : 2 or 5 bytes

また、提案手法のプロセス(5)における応答時間の判定を 0.5 秒とした。

RealSense SLAM の位置推定にかかる時間を正確に計測することはできないが 30fps で位置推定情報の提示が可能になっていることから 1 フレーム分の処理に要する時間はそれ以下であることがわかっている。

## 4.2 実験結果

実験時のリモート計算機資源、ウェアラブル計算機資源の負荷の様子を図 8, 図 9 に示す。この結果から類似画像検索, SLAM の処理はどちらも CPU 使用率が 100%に近いことがわかる。計算機資源を 2 つに分割しても高い CPU 使用率であることから 2 つの処理はどちらも計算コストが高い手法であることがわかる。

3 節に示した(3)に要した時間を横軸に, (1)から(4)までに要した時間を縦軸にとった散布図を図 10(a)に示す。(1)から(4)までに要した時間を縦軸に, フレーム番号を横軸にとったグラフを図 10(b)に示す。(1)から(4)までに要した時間のヒストグラムを図 10(c)に示す。また, (3)に要した時間と(1)から(4)までに要した時間の最高値・最低値・平均値を表 2 に示す。

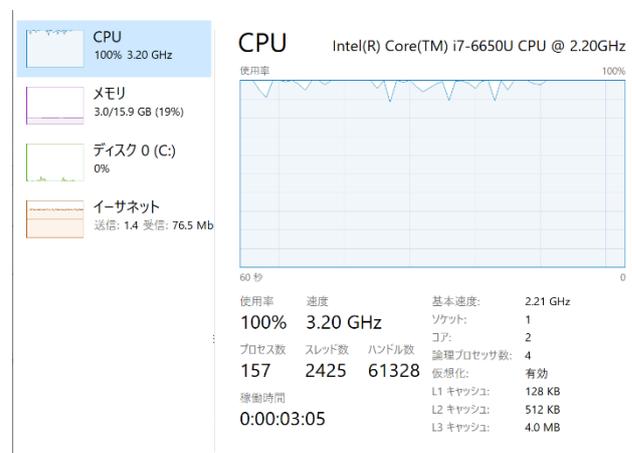


図 8 リモート計算機資源

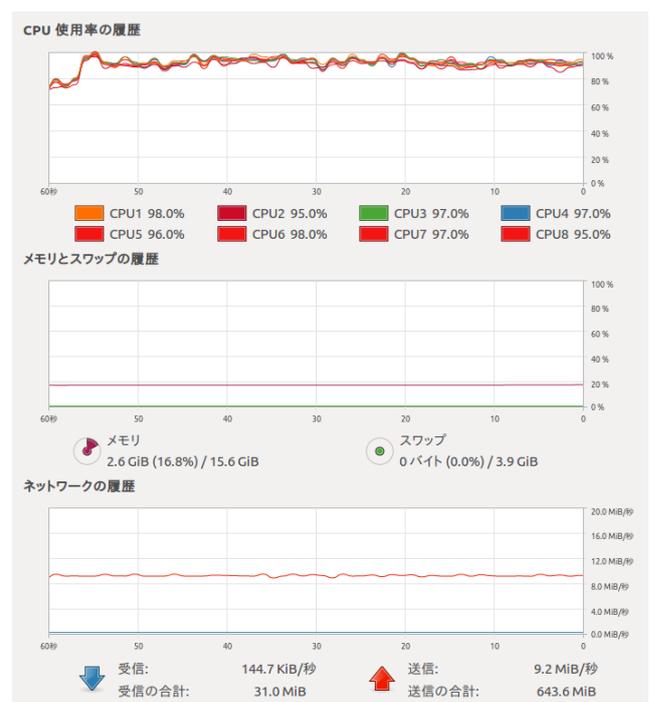
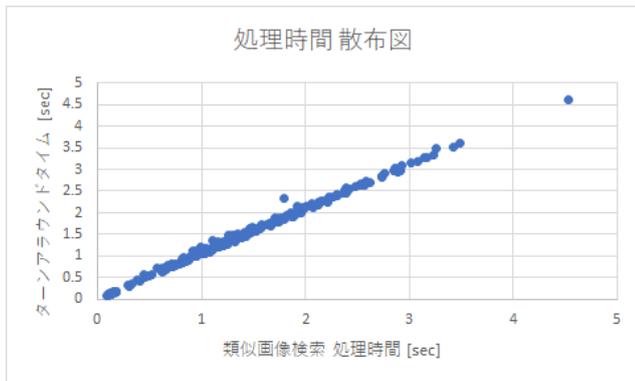


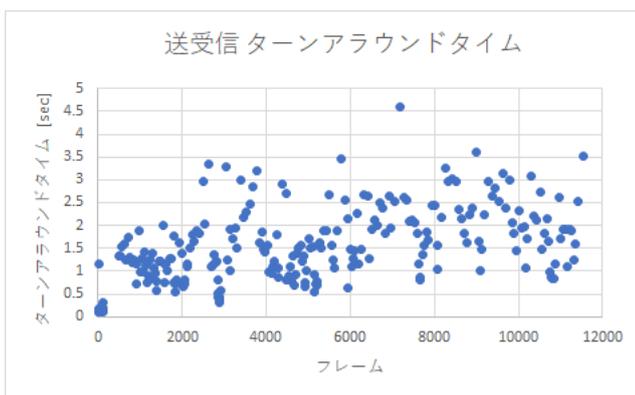
図 9 ウェアラブル計算機資源

表 2 処理時間計測結果

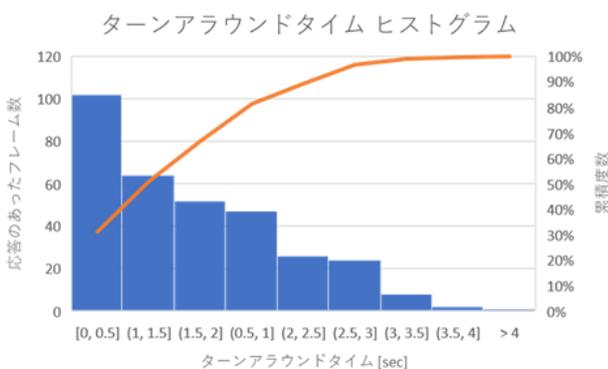
|                      | 最高値<br>[sec] | 最低値<br>[sec] | 平均値<br>[sec] |
|----------------------|--------------|--------------|--------------|
| (3)に要した時間            | 0.0859       | 4.535        | 1.122        |
| (1)から(4)までに<br>要した時間 | 0.0987       | 4.602        | 1.184        |



(a) (1)から(4)までに要した時間 — (3)に要した時間



(b) (1)から(4)までに要した時間 — フレーム番号



(c) (1)から(4)までに要した時間 ヒストグラム

図 10 処理時間計測結果 グラフ

図 10(a)から、(1)から(4)までに要した時間のうち(3)に要した時間がほとんどを占めている。

処理結果が設定した時間内に返ってきたフレームならびに設定した時間を超えた最初のフレームの合計は 11535 フレーム中 326 フレームであった。応答時間判定を 0.5 秒に設定していたところほとんどのキューはクリアされてしまう結果となった。

今回の実装では類似画像検索の処理時間 (3)に要した時間) の遅さが問題となり類似画像検索の処理結果が返ってきたフレーム数が少なくなった。類似画像検索プロセスのプログラム改良による高速化、リモート計算機資源のスペック向上などの可能性を考慮し類似画像検索と SLAM の協調的位置推定の改善に努めていきたい。

## 5. おわりに

本稿では、1 つの計算資源上で実行するには負荷が大きく実現が難しい類似画像検索と SLAM の2 つについて別々の計算資源上で実行することで協調的位置推定を行う負荷分散法を提案し検証を行った。検証結果から2つの負荷分散が可能であることを確認することができたが今回の実験においては類似画像検索の処理に時間を要していることが分かった。今後はこの問題を重点的に改善していき、類似画像検索と SLAM の協調的位置推定を目指していきたいと考えている。

本研究の一部は科研費 17H01773 と 18H03480 の助成を受けた。ここに謝意を表する。

## 参考文献

- [1] 本間昭信, “日常的な生活空間における視覚障がい者の空間認知”, 地理学評論 Ser. A, vol. 73, no. 11, pp.802-816, 2000.
- [2] 釜坂一步, 北原格, 亀田能成, 大田友一, “経路上の歩行者位置推定に適した事前撮影映像からのデータベース生成”, 電子情報通信学会技術研究報告, vol. 115, no. 494, pp.19-24, 2016.
- [3] Hugh D.W., Tim B., “Simultaneous localization and mapping: part I”, IEEE Robotics and Automation Society, pp.99-110, 2006
- [4] Christian K., Jurgen S., Daniel C., “Dense Visual SLAM for RGB-D Cameras”, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013
- [5] Raul M. A., Juan D. T., “Orb-slam: Tracking and mapping recognizable”, Conference: Workshop on Multi View Geometry in Robotics, 2014
- [6] “Intel®RealSense™Camera ZR300 | Intel®Software”, Intel®Software Developer Zone, <https://software.intel.com/en-us/realsense/zr300>
- [7] “Intel® RealSense™ for Linux - SLAM Library: Developers Guide”, Intel® RealSense™ for Linux - SLAM Library, [https://software.intel.com/sites/products/realsense/slam/developer\\_guide.html](https://software.intel.com/sites/products/realsense/slam/developer_guide.html)
- [8] “Distributed Messaging - zeromq”, <http://zeromq.org/>