

# 複数台の移動ロボットによる通行者情報地図の更新手法

## An Update Method of Human Frequency Map by Multiple Mobile Robots

○学 横山光希 (明治大) 森岡一幸 (明治大)

Koki YOKOYAMA, Meiji University, y.koki0601@gmail.com

Kazuyuki MORIOKA, Meiji University

By acquiring, sharing and updating sensor data in space on multiple robots and utilizing it with other robots, it is possible to efficiently recognize surrounding information and lead to better decision making. In this research, we have developed a sensor data sharing system for mobile robot using block chain sharing Human Frequency Map as sensor data. In this paper, we describe a method of updating multiple robot Human Frequency Map, and created a simulator and evaluated the method.

**Key Words:** Multiple robots, Human Frequency Map, Update Spatial Information

### 1 緒言

自律移動ロボットは測域センサやカメラなどの情報に基づき周囲の環境を認識し行動をしている。しかしロボット自身に搭載されているセンサデータだけでは取得できる情報に限界が存在する。そこで空間に分散させたセンサノードから情報を取得する方法が提案されている。これを空間の知能化という。カメラを空間上に分散させロボットの人の追従を行なった研究事例 [1] も報告されており、これによりロボットが外部のセンサ情報を利用する有用性が確認できる。しかしこの手法ではセンサノードを空間内に設置する際に、設置位置などの詳細な設定を行う必要があるという問題点も存在する。

そこで本研究では環境にセンサノードを固定させるのではなく、同一空間上に存在する複数のロボットがセンサデータを取得・共有・更新を行う仕組みの提案および開発を目的としている。著者らはセンサデータ共有の仕組みとしてブロックチェーンを用いたシステム提案し、開発に着手してきた [2]。またロボットが共有するセンサデータとして通行者情報地図 (HFM: Human Frequency Map) を用いたシステムを提案し、開発を行なっている [3][4]。複数台の移動ロボットを同一空間上に存在させ、HFM を取得し共有するロボットと共有された HFM を活用するロボットの開発を行なった。しかしこのシステムでは HFM を共有し、他のロボットが活用する流れの動作確認を目的としており、HFM を取得しにいく際には人間が手動でロボットを移動させていた。

本稿ではロボットが自動かつ継続的に HFM を更新しデータを蓄積する手法を提案する。また本稿ではシミュレーターを作成し、提案手法の実験を行なった。

### 2 提案システム

#### 2.1 Human Frequency Map

Human Frequency Map とは占有格子地図をグリッドに分けた時にグリッドごと保持している人の経路情報のことである。人の経路情報とは通行頻度や人の進行方向などのことである。本研究では通行頻度を利用しており、図 1 はそのイメージを示している。この図では通行頻度を色で示しており、実際はグリッドごとに人が通行した回数が数値として格納されている。

#### 2.2 システム概要

本研究では HFM を共有するセンサデータとしてシステムを提案し、その開発に着手してきた。複数台の移動ロボットが地図空

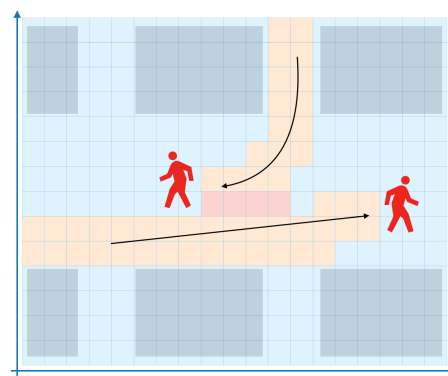


Fig.1 Human Frequency Map

間の様々な場所に移動し HFM を観測する。その観測した HFM をブロックチェーンを介し共有を行うことで、空間上にいる他のロボットもそのデータを活用するという一連の流れの開発を目的としている。図 2 は提案システムの全体像を示している。この提案システムを実現するために次の 4 つの段階があると考ええる。

- Acquire(HFM の取得)
- Share(HFM の共有)
- Utilize(HFM の活用)
- Update(HFM の更新)

著者らは取得・共有・活用までは [4] で開発しているので、本稿では HFM の更新を主な目的としている。

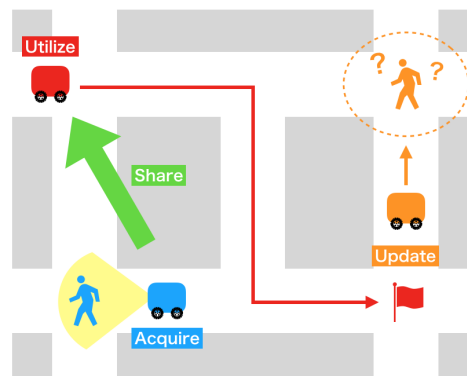


Fig.2 Proposed System

### 2.2.1 HFM の取得

HFM の取得では北陽電機社製のレーザー測域センサ UTM-30LX を利用して行なっている。測域センサをロボットに搭載、取得されたセンサ情報を OpenCV で描画し、テンプレートマッチングを用いて HFM の取得をしている。また取得した HFM は csv ファイルにて保存している。

### 2.2.2 HFM の共有

ブロックチェーンを用いて複数台の移動ロボットでのネットワークを構築し、ブロックチェーンのシステム上で HFM を共有している。本研究ではブロックチェーンとして Ethereum を採用している。IPFS という分散型ファイルシステムと Ethereum を利用することで csv ファイルの共有を実現している。

### 2.2.3 HFM の活用

HFM をもとにコストマップを作成し、A\*アルゴリズムを用いて大域経路計画を行う。これは現在地から目的地まで与えられたコストが最小になるような経路計画を立てるものである。

## 3 HFM 更新手法

### 3.1 グリッドごとの観測回数の共有

HFM を更新していくためにはロボットが共有された情報を基に次の観測地点を決定していく必要がある。そこでグリッドごとに観測回数を持たせ、それを共有する手法をとる。これを基に HFM を更新するロボットが次この地点を観測すればいいのかを判断する。またこの手法では HFM がブロックチェーン上で共有されるたびに観測されたグリッドの観測回数を更新する必要がある。今回は HFM が共有されるのを監視する PC をブロックチェーンのノードとして接続しこれを実現させる (図 3)。

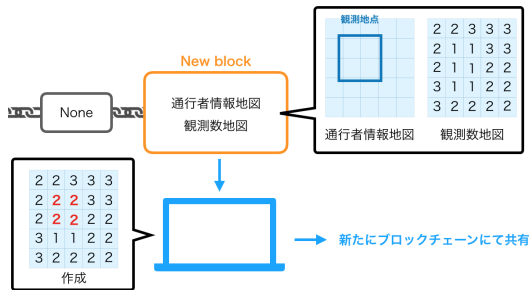


Fig.3 Share a Number of Observation

### 3.2 観測地点の決定方法

次の観測地点を決定する手段として、グリッドごとの観測数とロボットの現時点からの距離を利用する。本研究ではロボットを静止させ、前方 6mx6m の範囲で HFM を観測を行わせているため、一度の観測における観測範囲は固定である。その観測範囲に対応した矩形範囲内で地図の左上から右下まで探索を行い、矩形範囲内ごとの平均観測数  $x_o$  およびロボットの現時点からの距離  $x_d$  を求める。図 4 の青い矩形が矩形範囲内に相当し、矩形範囲内の左上のグリッドが距離である。これらの数値にそれぞれ重み  $w_o, w_d$  をかけ足し合わせたものを矩形範囲における  $value$  とする。この  $value$  が最小の地点を次の観測地点と決定する。式 (1) は  $value$  の計算式である。

$$value = w_o x_o + w_d x_d \quad (1)$$

$x_o, x_d$  は絶対値の違いによる作用を考慮し、全ての矩形範囲での平均観測数と距離を計算したのち、それぞれ標準化したものを

利用している。また  $w_o, w_d$  の範囲は 0~1 であり、式 (2) の条件を満たしている。

$$w_o = 1 - w_d \quad (2)$$

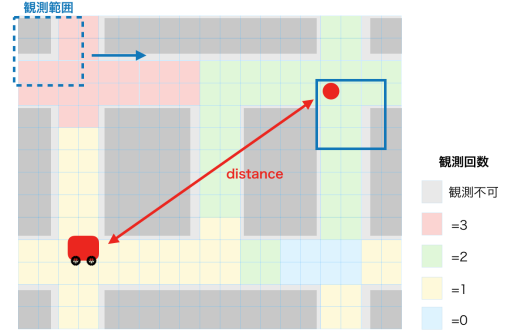


Fig.4 Decision of Observation Point

### 3.2.1 平均観測数の計算

矩形範囲における平均観測数を変数として利用するためには矩形範囲内に存在する観測不可能なグリッドの数を考慮する必要がある。観測不可能なグリッドとは壁や地図外の未確認の地点である。そこで今回求める矩形範囲内の平均観測数は分母に矩形範囲内の観測可能なグリッドの合計を取り、観測不可能グリッドの影響を除去する。式 (4) は矩形範囲内の平均観測数を求める式である。

$$\text{平均観測数} = \frac{\text{観測回数の合計}}{\text{観測可能なグリッド数}} \quad (3)$$

### 3.2.2 距離の計算

距離の計算においてはグリッド単位での計算を行なっている。本研究でも一辺 0.5m のグリッドを利用している。式 (4) は距離の計算式である。ここではロボットの現時点での座標を  $(x_0, y_0)$  、次の地点を  $(x_1, y_1)$  とし表している。

$$distance = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2} \quad (4)$$

## 4 シミュレーターの開発

本稿では実際にロボットに走行させるのではなく、シミュレーターを開発し観測回数の遷移および最適な重みの探索を行う。今回のシミュレーターでは事前に SLAM で作成した占有格子地図を 0.5m x 0.5m のグリッドに分割したものを利用している。

### 4.1 シミュレーション手順

1. グリッドごとの観測数の初期化 (全て 0 にする)
  2. ロボットの初期位置を決定 (地図上の地点からランダムに決定)
  3. 3.2 項の手法により観測地点の決定
  4. 決定した観測地点をもとにグリッドごとの観測数を +1
- 手順の 3,4 を任意の回数繰り返すことにより、地図全体の観測回数の遷移を確認することができる。この任意の回数のことをここでは試行回数  $n$  と呼ぶこととする。また  $w_o, w_d$  はプログラム上で変更することができる。図 5 は初期段階から試行回数を 1 度行なった時の図である。上の画像が観測数が初期の図であり、赤く囲まれた部分が初期地点として観測を行った地点であり、観測数が増えているのがわかる。

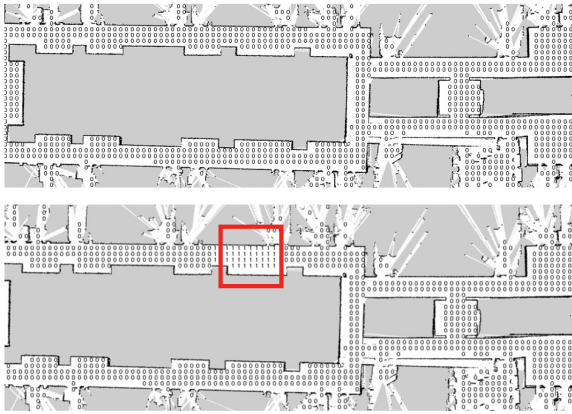


Fig.5 Example of Simulation

#### 4.2 シミュレーション実験

4.1 項で開発したものを明治大学中野キャンパス 11F の地図にてシミュレーションを行う。 $n = 1000, w_o = 1.0, w_d = 0.0$  とし、シミュレーションを行なった。 $w_o = 1.0, w_d = 0.0$  とすることで距離を無視し、観測回数だけを考慮して次の観測地点を決定する。これにより観測回数が少ない条件の地点を見に行くことができ、最短の試行回数で全てのグリッドを観測しに行くことができ、5 章の重みの評価で用いる試行回数の基準を決定することが可能になる。図 6 のグラフはその結果である。横軸が試行回数  $n$ 、縦軸が地図全体の平均観測数である。地図全体の平均観測数は以下の式 (5) で定義している。また地図全体で観測可能なグリッド数はこの地図では 1272 箇所である。

$$\text{地図全体の平均観測数} = \frac{\text{地図全体の観測回数の合計}}{\text{地図全体の観測可能なグリッド数}} \quad (5)$$

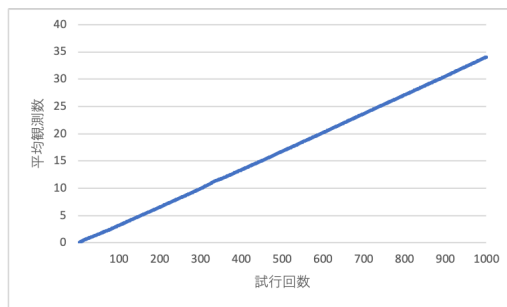


Fig.6 Result of Experiment

この結果より距離を考慮しない場合、試行回数  $n = 1000$  の時地図全体の平均観測数が約 35 回になることがわかる。つまりこの実験環境においておおよそ試行回数  $n = 30$  で全てのグリッドの観測を行うことができる。図 7 は試行回数  $n = 30$  の時のグリッドごとの観測数を地図に描画したものであり、ほとんどのグリッドにおいて観測数が 1 か 2 なのが見える。

### 5 重みの評価

4 章で開発したシミュレーターを用いて最適な  $w_o, w_d$  の探索を行う。本稿では以下の 2 点に着目し重みの評価を行った。

- 観測したグリッド数の比較
- 総距離の比較

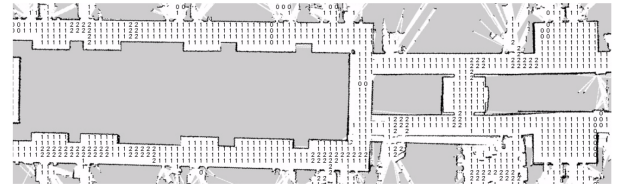


Fig.7 試行回数  $n = 30$

4.2 項により試行回数が約 30 で全てのグリッドを観測するのがわかった。そこでシミュレーション上にて距離の重みを 0~1 の範囲で 0.1 ずつ変化させ、それぞれの重みで 30 回の試行回数を 5 回行うことにする。5 回シミュレーションを行ったあと結果の平均を取り、それぞれの重みでの比較を行う。

#### 5.1 観測したグリッド数の比較

地図全体で観測可能なグリッドは 1272 箇所あり、試行回数 30 の際に 1272 箇所に近いグリッド数を観測するのが理想的である。そこで  $w_d$  で観測グリッド数を算出し比較を行う。図 8 は  $w_d$  ごとの観測したグリッド数のグラフである。

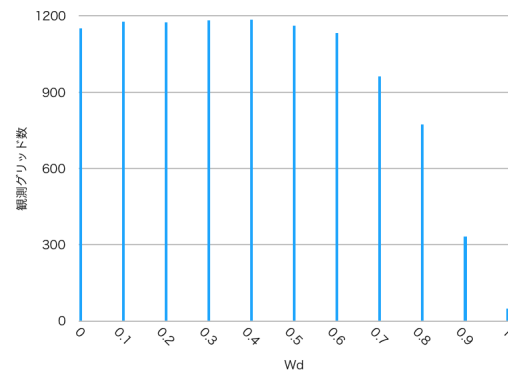


Fig.8 Evaluation with a Number of Observed Grid

#### 5.2 総距離の比較

地図全体の観測を効率的に行うためには、ロボットの移動時間も考慮し距離を最小限に行なっていく必要がある。図 9 は  $w_d$  ごとの総距離のグラフである。

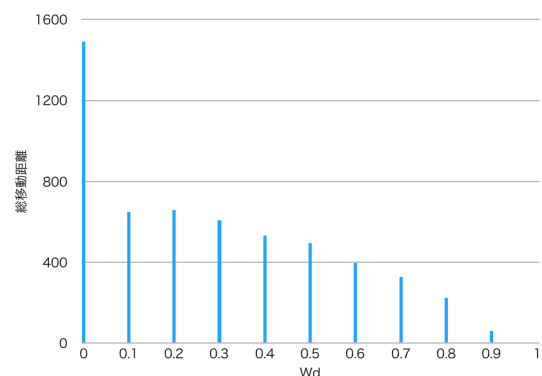


Fig.9 Evaluation with Total Distance

#### 5.3 重みの評価に対する考察

図 8 から  $w_d$  が 0.5 までは観測回数が 1200 弱で横並びであり、それ以降減少していることがわかる。また図 9 から  $w_d$  が増える

にあたり、総距離が少なくなっていることがわかる。この結果から  $w_d$  は 0.4~0.6 がバランスの取れているものと考えられる。

## 6 複数台のロボットを想定したシミュレーション

### 6.1 シミュレーション内容および結果

前章までに行なったシミュレーションはロボットが1台として実験を行なっている。そこでロボットの数を増やした時の場合を想定した時のシミュレーションを行う。観測したグリット数が試行を繰り返すごとにどのように遷移していくかをロボットの数ごとに比較を行う。今回ロボットの数は1~3台、試行回数  $n = 30, w_o = 0.5, w_d = 0.5$  とし、シミュレーションを行なった。ロボットの台数によりシミュレーションを5回ずつ行い、その結果をそれぞれ平均をとったものを図11のグラフにて示す。ロボットの台数によりシミュレーションを5回ずつ行い、その結果をそれぞれ平均したものを利用し台数ごとの比較を行う。図10が台数ごとの結果、図11が平均を取り比較した結果のグラフである。ともに横軸は試行回数  $n$ 、縦軸は観測したグリット数となっている。

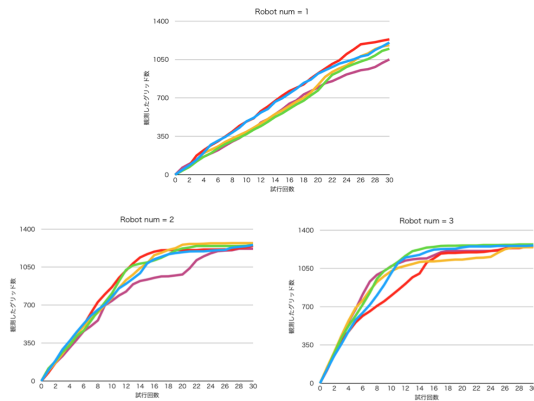


Fig.10 それぞれの台数での結果

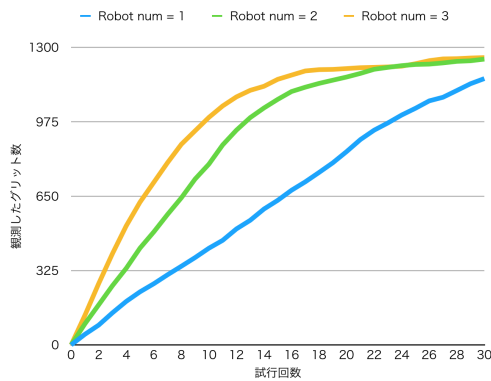


Fig.11 複数台ロボット

### 6.2 シミュレーションに対する考察

図11からロボットの数が増えるほど少ない試行回数で地図全体のグリットを観測できることがわかる。これより今回提案した手法を用い複数台のロボットで観測を行うことで、効率的に地図全体のHFMの観測を行うことができることを確認することができた。

## 7 結言

本稿では通行者情報地図を共有情報とし、それを更新していくための手法について提案した。またシミュレーションを開発し、そ

の手法に対する検証も行なった。しかし今回の手法およびシミュレーション以下のことが考慮されていない。

- 今回の距離は単なる点と点との距離でしかなく、実際のロボットの移動距離ではない。
- 観測地点でのロボットの位置、向きという情報は設定されていない。
- 矩形範囲内は全て+1しているのので、壁などの影響で同時に観測できない地点もカウントしてしまっている。(図12)

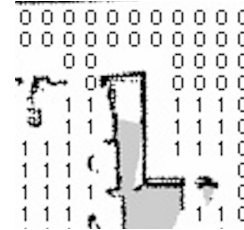


Fig.12 同時に観測不可な地点

今後は上記を考慮したシミュレーターを作成し提案手法の評価を行い、実環境での実験を目指す。

## 参考文献

- [1] Kazuyuki Morioka, Joo-Ho Lee, Hideki Hashimoto, "Human-following mobile robot in a distributed intelligent sensor network", IEEE Transactions on Industrial Electronics, Vol.51, No.1, pp.229-237, 2004.
- [2] 横山光希, 森岡一幸, "ブロックチェーンを用いた知能化空間のためのセンサデータ共有システム", ROBOMECH2018. 2A2-D08, 2018.
- [3] 横山光希, 森岡一幸, "ブロックチェーンを用いた移動ロボットのための環境地図共有システム", SI2018. 2D4-06, 2018.
- [4] 加藤勇気, 森岡一幸, "移動ロボットによる人間観測に基づく通行者情報の作成と柔軟な経路計画", ROBOMECH2017, 2A2-D08, 2017.