

複数台のロボットによる通行者情報地図における空間情報の更新手法

An Update Method of Spatial Information about Human Frequency Map by Multiple 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する範囲に限界が存在する。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り、ロボットの稼働環境に依存させなくてはならない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・更新を行う枠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枠組みとしてブロックチェーンを提案し、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) を用いた開発も行なっている [3][4]。複数台の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において HFM をロボットが取得する際、人が観測地点まで手動でロボットを移動させている。

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 HFM 更新手法

2.1 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、グリットごとの観測数とロボットの現時点からの移動距離を利用する。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移動距離 x_d を求める。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。

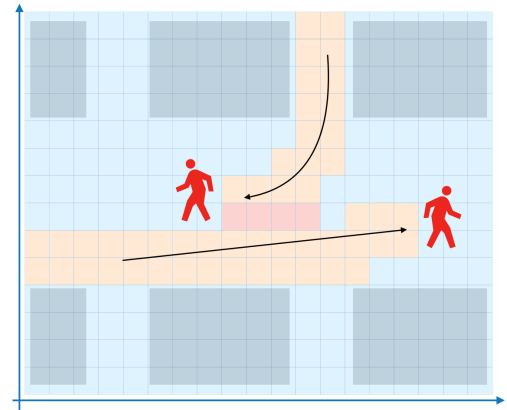


Fig.1 Human Frequency Map

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

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 x_o, x_d 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。

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

2.1.1 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トの数に考慮する必要がある。観測不可能なグリット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トの合計をとり、観測不可能グリットの影響を除去する。式 (4) は矩形範囲内の平均観測数を求める式である。

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

2.1.2 移動距離の計算

移動距離の計算においてはグリット単位での計算を行なっている。図 2 ではグリットの一边が 0.5m の際の例を示したものである。また本研究でも一辺 0.5m のグリットを利用している。式 (4)

は移動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。

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

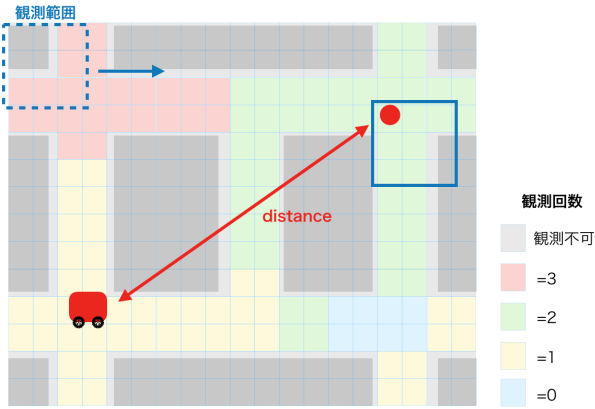


Fig.2 Decision of Observation Point

2.2 グリッドごとの観測回数の共有

2.1 項の観測地点の決定方法を複数台のロボットで行うためにはグリッドごとの観測回数をロボット間で共有する必要がある。そこで HFM がブロックチェーンで共有されるたびにグリッドごとの観測数を逐次更新する手法をとる。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して接続しこれを実現させる。(図 3)

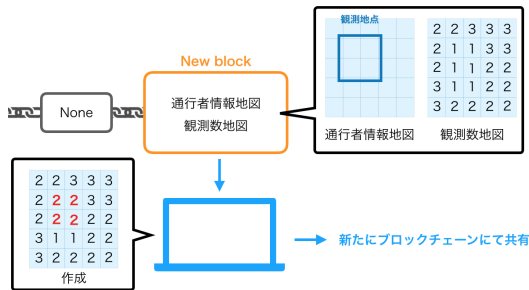


Fig.3 Share a Number of Observation

3 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は python にて開発を行なった。

3.1 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
3. 2.1 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繊維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 4 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、下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。

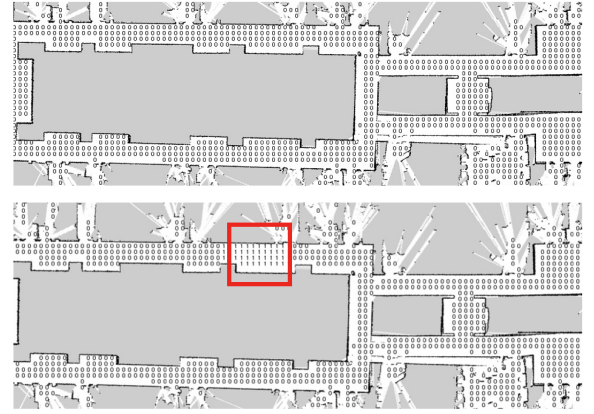


Fig.4 Example of Simulation

3.2 シミュレーション実験

3.1 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。これは移動距離を無視することにより、観測回数のみを考慮した理論的な数値を得ることができる考えたためである。図 5 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。

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

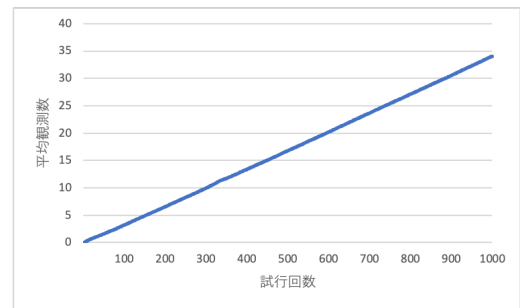


Fig.5 Result of Experiment

この結果より移動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 6 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 なのがわかる。

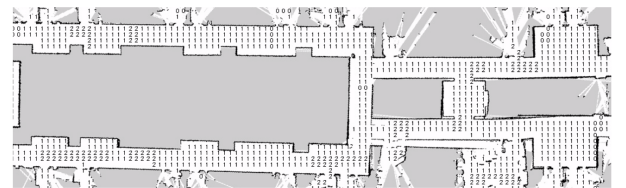


Fig.6 試行回数 $n = 30$

4 重みの評価

3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
- 総移動距離の比較

3.2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。

4.1 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7は w_d ごとの観測したグリッド数のグラフである。

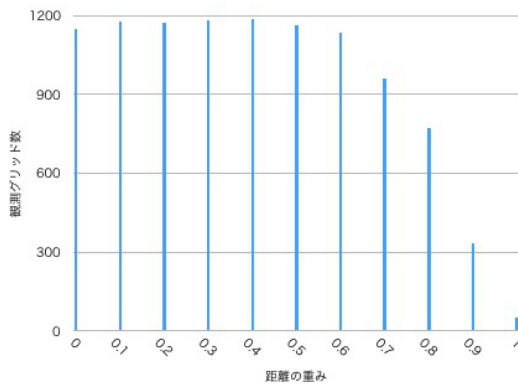


Fig.7 観測したグリッド数による評価

4.2 総移動距離の比較

地図全体の観測を効率的に行うためには、移動距離を最小限に行なっていく必要がある。図8は w_d ごとの総移動距離のグラフである。

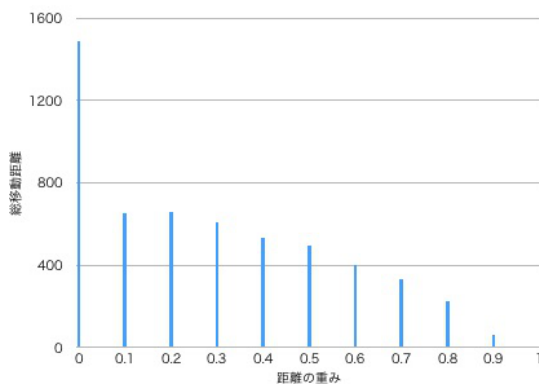


Fig.8 総移動距離による評価

4.3 重みの評価に対する考察

図7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8から w_d が増えるに当たり、総移動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0.5前後が最適だと考えられる。

5 複数台のロボットを想定したシミュレーション

5.1 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台であるとして実証を行なっている。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。観測したグリッド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9のグラフにて示す。横軸は試行回数 n 、縦軸は観測したグリッド数となっている。

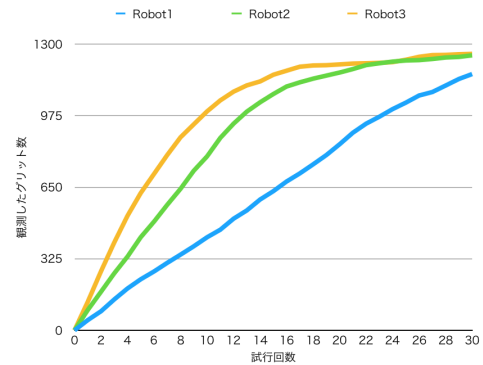


Fig.9 複数台ロボット

5.2 シミュレーションに対する考察

図9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ド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。

6 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10)

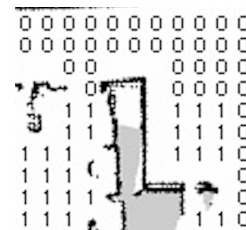


Fig.10 同時に観測不可な地点

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.