

# カメラ付きアームの能動的な姿勢変更による 3次元地図生成のための未知環境探索

魚躬 良太 萬 礼応 大矢 晃久 (筑波大学)

## 1. 緒言

近年、家庭用掃除ロボットは iRobot 社のルンバ [1] や Panasonic 社の RULO [2] などのように、カメラや LiDAR センサを搭載し、SLAM 技術を用いることで、部屋全体の領域を隈なく効率的に掃除できるようになった。しかし、掃除ロボットは低い位置からのセンシングを行うため、周囲の環境の認識が難しい場合がある。この課題の解決方法の1つとして、高い位置にカメラを取り付け、俯瞰的にセンシングを行い、周囲の環境からより多くの情報を観測できるようにすることが考えられる。しかし、実際にカメラをロボットの高い位置に搭載しようとすると、車高が高くなってしまい、掃除ロボットの特徴である机や椅子などの低い・狭い場所に潜り込んで清掃できる移動性能が犠牲になってしまう。そこで本研究では、掃除ロボット本来の移動性能を犠牲にせず、高い位置からの俯瞰的なセンシングを行えるように、図1に示すような RGB-D カメラ付きロボットアームを搭載させた掃除ロボットを提案する。

提案する掃除ロボットは一般的な掃除ロボットと同様の機能を備えなければならない。一般的な掃除ロボットにおいて必要な機能はだまかに分けて3つ存在し、1つ目は未知環境を探索して地図を生成する機能、2つ目は作成した地図を基に動作計画をする機能、3つ目は計画に沿って掃除を行う機能である。本研究ではまず、1つ目の機能に関して提案する掃除ロボットを使い、地図生成までにかかる探索時間を短縮することを目的とする。

カメラ付きアームを搭載した掃除ロボットではアームを動かすことができるので、移動中でもカメラの向きを能動的に変化させることが可能である。そのため、移動中にカメラを未知領域がある方向へ適切に向けながら移動することで、地図生成までにかかる時間を短縮することができると考えられる。そこで本研究で提案する手法では、既知領域と未知領域の境界部分を検出し、その境界部分を多く観測できるような位置とカメラ姿勢を算出する。この算出された位置の中から現在のロボットの位置と姿勢を考慮して目的地を決定する。そして、算出された位置とカメラ姿勢に合わせてカメラの向きを適切に変化させながら移動を行っていく。このような動作を環境中の未知領域が全て無くなるまで行うのが本研究の提案手法である。

既知領域と未知領域の境界を目的地として探索する手法としては Yamauchi [3] の手法が先行研究として挙げられる。また、既知領域と未知領域の境界を観測できるような位置とカメラ姿勢を算出し、その位置と姿勢を目的地として探索する手法としては 三明ら [4] の手法が先行研究として挙げられる。これらの先行研究と本研究には大きく異なる点が2つある。1つ目は、従来研究では移動中にカメラの姿勢は固定して移動を行っているのに対して、本研究では移動中にも既知領域と未知領域の境界部分を向くようにカメラの向きを適時変更している点で

ある。2つ目は、従来研究では候補となる目的地の中から、ロボットの現在地からの距離を考慮して次の目的地を決定しているのに対して、本研究では、ロボットの現在地からの距離に加えて、ロボットが現在どの方向を向いているかを考慮して目的地を決定している点である。

今回は提案手法において、カメラの向きを変更しながら移動する処理とロボットの現在地と姿勢を考慮して次の目的地を決定する処理が、従来手法と比べて地図生成にかかる探索時間をどの程度短縮できているかを検証し、その結果について述べる。

## 2. 関連研究

未知環境での地図生成のための探索アルゴリズムには様々な手法があるが、本研究では Yamauchi [3] が提案した Frontier Exploration アルゴリズムを参考にしている。この手法は既知領域と未知領域の境界をフロンティアと定義しており、フロンティアがある場所への移動を繰り返し、未知領域の情報を取得していくことで、部屋の地図を生成している。ただし、本研究ではカメラを利用しているため、ただフロンティアがある場所へ移動する Frontier Exploration アルゴリズムを本研究の探索アルゴリズムとしてそのまま適用するのは難しい。

それに対して、フロンティアを観測できるような位置とカメラ姿勢を計算し、探索を行う先行研究として、 三明ら [4] の手法がある。 三明らはフロンティアを観測できるような位置とカメラ姿勢を計算し、できるだけ多くのフロンティアを観測できるような位置とカメラ姿勢を目標位置として移動していく探索アルゴリズムを提案している。この手法を姿勢変更可能なカメラを搭載したロボットでの探索アルゴリズムに適用することで、移動中にカメラを能動的に動作させることが可能になり、地図生成効率の向上に繋がると考えられる。

## 3. 提案手法

本節では提案手法として、移動中にアーム姿勢をヨー角方向へ動かし、カメラの向きをフロンティアが存在する方へ向けながら探索を行うアルゴリズムについて説明する。

3.1 節ではこの手法を説明するにあたって前提となる



Fig. 1 使用するロボットのハードウェア構成

ロボットの仕様について、3.2 節で手法の概要について、3.3 節ではフロンティアの検出方法について、3.4 節ではフロンティア観測量マップの作成について、3.5 節では目的地の決定方法について、3.6 節では移動中におけるカメラの向き決定方法について述べる。

### 3.1 ロボットの仕様について

提案手法の説明をする前に、本研究で扱うロボットについて説明する。本研究では図 1 で示したようなロボットを用いている。水平方向への移動を行うことが可能であり、ロボットアームを搭載している。ロボットアームには、オレンジの枠で示されている位置に RGB-D カメラが取り付けられており、周辺環境の画像情報、深度情報、点群を取得することが可能である。また、ロボットアームの各関節はロール、ピッチ、ヨー角方向へ姿勢を変化させることが可能である。ただし、今回はロール、ピッチ方向にはアームの関節は動かさず、ヨー方向にだけアームを動かせるようにしている。カメラの位置はアームの先端に固定されており、垂直方向にアームは動かさないため、常に地表から 0.7m の高さとなっている。

### 3.2 手法の概要

提案手法の全体的な流れを図 2 に示す。まず初めに、Yamauchi [3] が提案した手法と同様に、既知領域と未知領域の境界をフロンティアとして検出している。そして、この検出したフロンティアを観測できるようなロボットの位置とカメラ姿勢を算出する。位置と姿勢の算出は三明ら [4] の手法と同様であり、平面上での位置座標とカメラが向く方向の 3 つの情報を持ったグリッドマップを生成し、各グリッドにはその位置と姿勢から観測できるフロンティアの量が格納されている。このようなグリッドマップを本研究ではフロンティア観測量マップと呼ぶ。そして、算出された位置と姿勢の中から、ロボットの正面方向かつ近くにある位置と姿勢が次に移動すべき目的地として決定される。図 3 の例では赤い矢印が目的地として決定されている。そして、移動中にフロンティア観測量マップを参照することで、最もフロンティアが観測できるカメラ姿勢となるようにカメラの向きを決定している。

また、Yamauchi や三明らの手法においては目的地に到着してから再度フロンティアの検出、位置とカメラ姿勢の計算を行っていたが、本研究では目的地への移動中にもフロンティア検出と位置とカメラ姿勢の計算を行っているため、目的地は随時更新されている。

### 3.3 フロンティアの検出方法

提案する手法では、まず、2.1 節で述べた Yamauchi [3] が提案した手法と同様に、既知領域と未知領域の境界をフロンティアと定義し、検出を行う。ただし、Yamauchi の手法では 2 次元平面上でフロンティアの検出を行っているが、本手法では RGB-D カメラを使用するため、3 次元空間上でフロンティアの検出を行っている。環境内の情報は Octomap [6] という 3D-SLAM の手法を用いてカメラから取得した空間の情報を 3 次元ボクセルに変換し、取得している。この 3 次元のボクセルは障害物が無い領域を Free、障害物がある領域を Occupied、センサから取得できていない領域を Unknown として 3 つの状態に区分されている。本研究では、この Free に隣り合っ

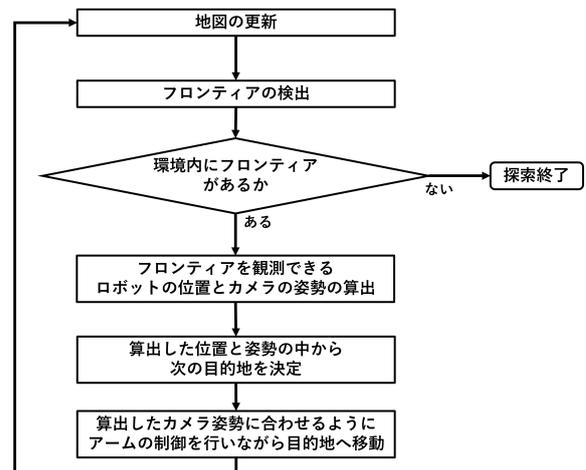


Fig. 2 提案手法のフロー図

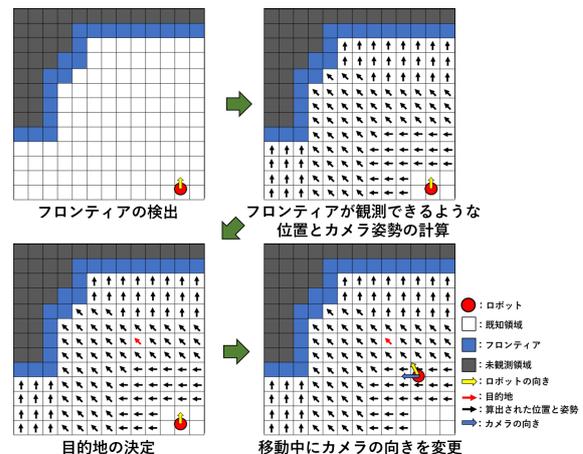


Fig. 3 目的地と移動中のカメラ姿勢を決定するまでの流れ

ている Unknown をフロンティアとして定義し、検出できるようにしている。また、フロンティアも 3 次元ボクセルとして表しているため、個数として量を測ることが可能になっている。

### 3.4 フロンティア観測量マップの作成

環境内で検出されたフロンティアの量はどのような位置とカメラ姿勢で観測するかによって異なる。そこで、本研究ではグリッドマップを作成し、そのグリッドごとに観測できるフロンティア量を格納したフロンティア観測量マップの作成を行っている。手法としては、図 3 のようなグリッドマップを用意する。そして、各グリッドで観測できるフロンティアの量を 0 から 360 度の範囲で 45 度ごとに算出し、その量を各グリッドに格納している。そのため、各グリッドには 8 方向分の観測可能なフロンティアの量が格納されていることになる。また、今回使用するカメラでは最大 5m までのセンシングを行い、水平視野角は 85.2 度となっている。そのため、算出されるフロンティアの量はこのセンシング範囲で観測可能なフロンティアの量を算出している。このように作成したフロンティア観測量マップは以降の目的地の決定方法と移動中のカメラの向き決定方法に使用される。

### 3.5 目的地の決定方法

作成したフロンティア観測量マップの各グリッド  $p$  には観測できるフロンティアの量を 0~360 度の範囲で 45

度おきに格納しているため、計 8 方向分のフロンティアの量が格納されている。その 8 方向の中から最大のフロンティアの量  $v(p)$  を算出した後、算出されたフロンティアの量  $v(p)$  に対して補正をかけ、補正された値  $f(p)$  を基に目的地を決定する。補正された値  $f(p)$  は次式を用いて計算される。

$$f(p) = \begin{cases} \frac{v(p)(\cos \theta + 2)}{d(p)} & \text{if } 1 < d(p) \\ v(p)(\cos \theta + 2) & \text{otherwise} \end{cases} \quad (1)$$

ここで  $\theta$  は、ロボット座標系において、対象のグリッド  $p$  がロボットに対してどの方向に位置しているかを表す角度である。この角度は対象のグリッド  $p$  の位置がロボットの正面に近ければ近いほど値は 0 に近づき、ロボットの後方に近ければ近いほど値は  $\pi$  に近づく。そのため、 $\cos \theta + 2$  の値域は  $1 \leq \cos \theta + 2 \leq 3$  となるので、 $f(p)$  の値はグリッド  $p$  がロボットの正面近くに位置しているほど大きくなる。また  $d(p)$  は、世界座標系におけるグリッド  $p$  とロボットとのユークリッド距離を表している。そのため、対象のグリッド  $p$  とロボットが離れていれば離れているほど  $f(p)$  の値は小さくなる。ただし、 $D$  の下限を 1 と設定しているため、グリッド  $p$  とロボットの距離が 1m 未満になっても  $d(p)$  の値は 1 のままであり、 $f(p)$  の値は大きくならない。これらの補正はロボットの正面方向かつ近くにあるグリッドを目的地として選択するために行われている。このようにして算出された  $f(p)$  の値が最も大きいグリッドの位置座標をロボットの目的地として決定している。

また、Ymauchi と三明らの手法では目的地を決定後、ロボットが目的地に到着するまで各グリッドで観測できるフロンティアの量の計算は行われず、目的地も更新されていなかったが、この提案手法においては目的地の決定までの処理は常に計算している。そのため、ロボットが目的地までの移動中にフロンティアを観測することで環境内のフロンティアの量と位置が変動した場合、算出される位置とカメラ姿勢も変化するので、随時目的地を更新するようにした。

### 3.6 移動中におけるカメラの向き決定方法

本節では移動中のカメラの向き決定方法について説明する。ロボットは移動中、ロボットに一番近いグリッドを参照する。3.4 節で算出された各グリッドには、0~360 度の範囲で 45 度ごとに観測可能なフロンティアの量、すなわち 8 方向分のフロンティアの量が格納されている。その 8 方向の中で、観測できるフロンティアの量が最も多い方向をカメラが向くべき方向として決定している。図 3 を例にとると、ロボットが目的地に向かうまでの途中、カメラの向きをロボットの一番近くにあるグリッドを参照し、観測できるフロンティアの量が最も多い方向にカメラの向きを合わせていることが分かる。このように移動中にフロンティアがある方へカメラを向けることで、カメラの向きを変更しないような三明らの手法と比べて、どの程度地図生成効率に影響があるのかについて本研究では検証を行っている。

## 4. 検証

提案手法と従来研究である三明らの手法と比較するために、カメラの向きを変更しない状態での探索時間を計

算し、結果の比較を行った。また、式 (1) による補正をかけてから目的地を決定する方法がどの程度地図生成効率に影響しているかを検証するために、式 (1) による補正をかけずに、最もフロンティアを観測できるグリッドを目的地として決定した場合の探索時間も計算し、比較を行った。

### 4.1 検証環境

検証は Gazebo [5] と呼ばれる 3 次元シミュレーター上で行った。地図生成を行う対象の部屋環境は図 4-(a) に示す。部屋の大きさは 10 m × 10 m であり、部屋内には等間隔で 0.5 m × 0.5 m の四角柱が 8 つ置かれている。また、3.4 節で説明したグリッドマップはグリッド幅が 0.2 となっており、そのサイズは部屋の大きさに合わせて 51 × 51 のサイズとなっている。スタート地点は図 4-(b) ~ 図 4-(f) のように計 5 回変更し、その 5 回分を平均したデータを結果として取得している。また、3.1 節でも説明したように図 1 のロボットを使用する。ロボットアームには X-Series Robotics Arms の WidowX 250 [7]、自動掃除ロボットには Kobuki [8] を使用している。また、RGB-D カメラは、シミュレーター上で水平視野角を 85.2 度、最大センシング距離を 5 m として定義したカメラを使用している。そして、今回はロボットの移動に move\_base [9] を使用するため、補助的に LiDAR センサを載せて使用した。LiDAR センサは北陽電機の URG UST-10LX [10] を使用している。シミュレーションに用いた PC のスペックは、CPU が Intel Core i5-8250U 1.6GHz、メモリが 8GB、GPU が Intel UHD Graphics 620 となっている。

### 4.2 検証結果・考察

提案手法の地図生成効率を検証するにあたって、今回は探索中にカメラの向きを変更しない場合、式 (1) の補正をかけずに最もフロンティアを観測できるグリッドを目的地とする場合、という 2 つのパターンの結果を取得し、提案手法の結果と比較した。表 1 に各検証方法に含まれる要素の違いを示し、表 2 にそれぞれの検証結果を示す。表 2 の検証結果において、ロボットが動き始めてから地図が作成されるまでの時間・距離をそれぞれ探索時間・移動距離として測定している。また、時間は全てシミュレーション内の時間に合わせて計測している。旋回角度は動き始めてから地図が作成されるまでにロボットが旋回した角度の総量を表している。また、図 5 に

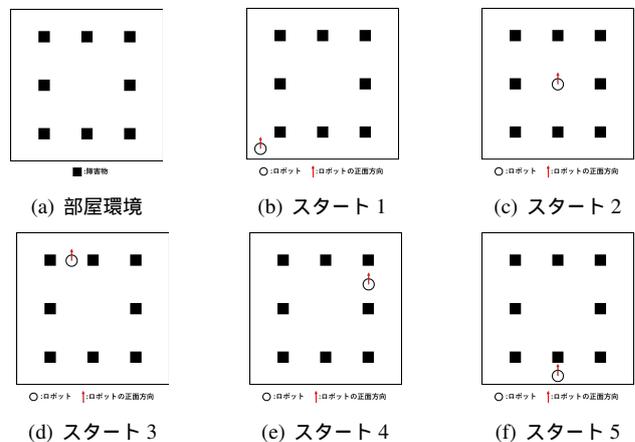


Fig. 4 部屋環境とスタート地点

実際の動作の様子を示す．図 5-(a) には Gazebo 上でのロボットの位置と部屋の様子を示している．赤い四角柱が環境内におかれている柱である．この環境に対してフロンティアの検出を行うと図 5-(b) のようになる．図中の青いボクセルがフロンティアを表している．この検出したフロンティアを観測できるような位置と姿勢を算出し，図 5-(c) の赤矢印で表された地点を目的地として決定している．この赤矢印の姿勢は目的地におけるカメラの姿勢を表している．また，移動中のカメラの姿勢は青い矢印で示されている方向を向くように決定されている．時間が進み環境内のフロンティアが変化すると，図 5-(d) のように目的地の位置も変化している．また，移動中のカメラ姿勢も目的地でのカメラ姿勢とは違う方向を向いており，現在のロボットの位置から最もフロンティアが見える方向にカメラの姿勢を変化させていることが分かる．このように，ロボットはカメラの向きを能動的に変化させながら地図の生成を行っている．

表 2 の結果から地図生成までにかかる探索時間は提案手法が最も短くなった．また，手法 2 が最も探索時間が長くなっており，その差は 179.2 秒であった．これは，提案手法のように探索中にカメラの向きをフロンティアが存在する方向へ向けることで，移動中に取得できる情報量が多くなることが要因であると考えられる．

また，手法 3 の式 (1) に従って目的地を決定しない場合，地図生成までにかかる探索時間は 2 番目に短かった．提案手法と比べると移動距離，旋回度が増加していることが分かる．目的地を式 (1) に従って決定せずに，一番フロンティアが多く観測できるグリッドを目的地としているため，ロボットの現在地と姿勢を考慮せずにロボットから離れた位置やロボットの後方の位置に目的地が決定される頻度が増加していた．そのため，ロボットが折り返して移動する頻度が多くなり，その分移動距離と旋

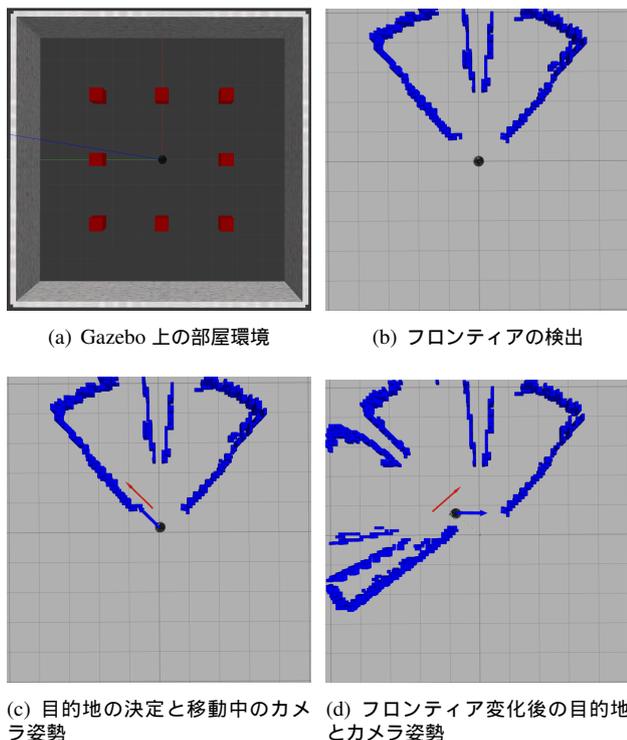


Fig. 5 ロボットの動作の様子

Table 1 各検証方法に含まれる要素

	カメラの向き変更	式 (1) に従って目的地を決定
手法 1(提案)		
手法 2	-	
手法 3		-

Table 2 検証結果の比較

	探索時間 [s] (5 回平均)	移動距離 [m] (5 回平均)	旋回角度 [rad] (5 回平均)
手法 1(提案)	349.7	30.2	65.6
手法 2	528.9	43.2	95.4
手法 3	451.5	35.8	92.7

回が増加し，全体の探索時間が長くなっていると考えられる．

## 5. 結言

本稿では，3 次元地図を生成するための探索アルゴリズムとして，探索中にフロンティアが存在する方へカメラを向けるながら探索するアルゴリズムを提案した．そして，従来手法のように探索中にカメラの向きを変更しない場合，ロボットの位置と向きを考慮せずに目的地を決定する場合と提案手法を比べて，地図生成効率にどのような違いが表れるかを検証した．

結果として，従来手法のようにカメラの向きを変更しない場合と比べて，地図生成にかかる時間を短縮することができた．また，ロボットの位置と姿勢を考慮して目的地を決定する場合と考慮せずに目的地を決定する場合を比べても，ロボットの位置と姿勢を考慮した場合の方が地図生成にかかる時間を短縮することができていた．このことから，提案手法のようにカメラの向きを変更しながら探索する処理とロボットの位置と姿勢を考慮して目的地を決定する処理は地図生成効率の向上に寄与していることが示された．

今後はカメラをヨー角方向だけでなくピッチ角方向にも変更できるようにした上で，より複雑な環境における 3 次元地図生成効率の検証を行いたい．

## 参考文献

- [1] iRobot 社: “<https://www.irobot-jp.com/roomba/>”, アクセス日:2021 年 6 月 27 日.
- [2] Panasonic 社: “<https://panasonic.jp/soji/products/rulo.html>”, アクセス日:2021 年 6 月 27 日.
- [3] Yamauchi, Brian.: “A frontier-based approach for autonomous exploration.”, *IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97*, 1997.
- [4] 三明 優介, 三浦 純.: “屋内物体探索における 3 次元未観測領域推定に基づく視点計画”, 第 38 回日本ロボット学会学術講演会, 2020.
- [5] Gazebo: “<http://gazebosim.org/>”, アクセス日:2021 年 7 月 11 日.
- [6] Kai M. Wurm et al.: “OctoMap: an efficient probabilistic 3D mapping framework based on octree”, *Autonomous Robots*, April 2013, Volume 34, 2013.
- [7] Interbotix 社: “<https://www.trossenrobotics.com/m/interbotix/>”, アクセス日:2021 年 6 月 27 日.
- [8] Yujin Robot 社: “<http://kobuki.yujinrobot.com/>”, アクセス日:2021 年 6 月 27 日.
- [9] move\_base: “[http://wiki.ros.org/move\\_base/](http://wiki.ros.org/move_base/)”, アクセス日:2021 年 7 月 11 日.
- [10] 北陽電機株式会社: “<https://www.hokuyo-aut.co.jp/search/single.php?serial=16>”, アクセス日:2021 年 6 月 27 日.