

Multi-resolution map を用いた未知環境の探査

Exploration using Multi-resolution map in Unknown Environment

○学 戸田 雄一郎 (首都大) 正 久保田 直行 (首都大)

Yuichiro TODA, Tokyo Metropolitan University, toda-yuuchirou@sd.tmu.ac.jp

Naoyuki KUBOTA, Tokyo Metropolitan University

Simultaneous localization and mapping (SLAM) is one of important topics in robotics. However, we must consider various intelligent behaviors in SLAM, e.g., the exploration of unknown areas and effective path planning of mobile robots. To realize these intelligent behaviors, we use a multi-resolution map. The multi-resolution map can be updated by the operators suitable to a specific aim. The first aim is to represent the occupied or empty cells in the built map. In this paper, the next aim is to represent the unknown areas in the built map. Therefore, we propose the exploration method using multi-resolution map in unknown area. The exploration method is based on k -certainty exploration. The experimental results show the effectiveness of the proposed method.

Key Words: Simultaneous localization and Mapping, Multi-resolution map, Exploration

1. 緒言

近年、ロボットテクノロジーの発展により、ロボットは徐々に我々の生活圏内へと進出している。今日、エンターテイメントロボットやペットロボット、施設の案内ロボットなど、人を楽しませるロボットの誕生により、知的なロボットが注目を集めている。これはロボットの存在がより身近なものとして一般的な地位を確立してきており、ロボットの活躍が期待されるシーンの拡大傾向を示しているといえる。

今後、ロボットの活躍が期待される分野として、レスキューロボット、セキュリティ目的の見守りロボット、床下など人間が作業することが難しい現場における点検のためのロボットなどがある。これらのロボットが活動する環境は、動的で且つ未知環境であることが多いと考えられる。こういった環境においては、ロボット自身が内部モデリングとして地図を持ち、自己位置推定をしながら行動していかなければならない。このようにロボットの自己位置推定と地図生成を同時に行う問題は、SLAM (Simultaneous Localization and Mapping) と呼ばれ盛んに研究が行われている。SLAM の問題は、近代のロボットの考案より前から数世紀にわたって存在する。動くセンサプラットフォームから物理的な構造をモデリングする問題は、地学、写真測量法など、多くの分野において核となる技術である。SLAM に関する研究は国内外問わず、世界中で盛んに行われている。

Online SLAM の分野では、Iterative Closest Point(ICP)法や Extended Kalman Filter(EKF)を用いたものが提案されている。[1,2] Full SLAM の分野では、様々な Graph SLAM やパーティクルフィルタを用いた Fast SLAM が提案され、多くの研究者によって、拡張した手法が提案されている [3,4]。国内においては、友納らがエッジ点追跡に基づく三次元 SLAM を提案している [5]。また、SLAM を行うための未知環境における探査計画の手法も数多く提案されている。これらの多くは、エントロピーを用いたグリーディー探査アルゴリズムに基づいている [6]。

一般的に地図を構築する際に、地図の粒度は細かいほうが正確な地図を得ることができる。しかしながら、これらの地図を用いて探査計画や経路計画などの行動計画を行うには、地図の粒度が細かく、一つ一つのセルの値を参照しては計算コストが高くなってしまいう問題点が存在する。よって本稿では、multi-resolution map を提案し、ロボットが効率よく行動計画を立てるためにセルの粒度を扱っていく。また、構築された multi-resolution map を探査計画に利用することにより、効率のよい未知環境の探査手法について提案する。

本稿は以下の流れで構成される。2章では、我々が用いた占有格子空間に基づく SLAM について説明する。3章では SLAM によって構築された地図からの multi-resolution map の生成方法についての説明を行う。4章では multi-resolution map を用いた未知環境における探査手法についての提案を行う。5章では提案手法に基づく実験結果を示す。

2. SLAM

本稿における SLAM の地図として、占有格子空間地図を用いる。図 1 に我々が提案する占有格子空間地図を示す。それぞれの地図の値を以下の式に示す。

$$map_0(x,y) = \begin{cases} 1 & \text{(occupied)} \\ 0.5 & \text{(partially occupied)} \\ 0 & \text{(unknown)} \\ -1 & \text{(empty)} \end{cases} \quad (6)$$

ここで、地図の初期値の値は全て 0 とする。LRF の計測距離計測データは (d_i, θ_i) , $i=1,2, \dots, M, j=1,2, \dots, L$ で表され、 d_i は計測距離、 θ_i は計測方向の角度、 M は全ての計測本数を表す。また、 $L_i (= [\alpha^{Res} \cdot d_i])$ は占有格子空間における解像度を表す。よって、地図の更新は以下のアルゴリズムによって示される。

Algorithm map-update:

for $i=1$ to M do

for $j=1$ to L_i do

$$u_{i,j} = \frac{j}{L_i} (d_i \cos(\theta_i + r_p)) + x_p \quad (7)$$

$$v_{i,j} = \frac{j}{L_i} (d_i \sin(\theta_i + r_p)) + y_p$$

$$x_{i,j} = [\alpha^{Map} \cdot u_{i,j}] \quad (8)$$

$$y_{i,j} = [\alpha^{Map} \cdot v_{i,j}]$$

$$map_0(x_{i,j}, y_{i,j}) = f(map_0(x_{i,j}, y_{i,j}), j) \quad (9)$$

endfor

endfor

ここで、 (x_p, y_p) はロボットの位置、 r_p は姿勢を表す。 d_i は LRF の i 番目の方向の計測距離、 q_i は計測方向の角度、 a^{MAP} は実環境から、占有格子空間へのスケール係数となっている。また、式(9)の $f(\cdot)$ は表1の IF-THEN rules に従って求められる。もし、ロボットの位置と姿勢が正しいならば、上述の地図は効果的に更新されるが、ロボットの自己位置推定の結果が正しくないならば、地図の構築も失敗してしまう。よって、自己位置推定の結果は精確でなければならない。表1に地図更新における地図の値の状態の遷移を示す。

我々は Steady-state genetic algorithm (SSGA) を自己位置推定の最適化計算に用いる。解候補はロボットの平行移動量 $(g_{k,x}, g_{k,y})$ 、回転移動量 $(g_{k,r})$ の数値パラメータによって構成される。SSGA では、僅かな解のみが交叉と突然変異によって生成された解候補に置き換えられる。我々は、エリート交叉と適応型突然変異を用いる。エリート交叉では、迅速に適切な解を見つけるために、ランダムに選択された1個体にベスト個体の遺伝子情報を組み込む。また、以下に適応度型突然変異の式を示す。

$$g_{k,h} \rightarrow g_{k,h} + \left(\alpha^{SSGA} \cdot \frac{f_{max} - f_k}{f_{max} - f_{min}} + \beta^{SSGA} \right) \cdot N(0,1) \quad (10)$$

ここで、 f_k は k 番目の個体の適応度、 f_{max} 、 f_{min} は適応度の最大値と最小値を表す。また、 $N(0,1)$ は正規乱数、 α^{SSGA} と β^{SSGA} は係数と補正値を表す。 k 番目の解候補は以下の式によって計算される。

$$fit_k^{Loc} = \sum_{i=1}^M map_0(x_{i,L}, y_{i,L}) \quad (11)$$

式(11)では、占有状態のセルが多いほど、適応度が高くなっていく。つまりこの問題は、最大化問題に帰着される。

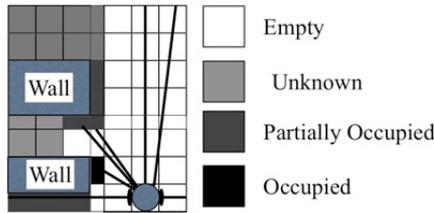


Fig.1 Concept image of the occupancy grid map

Table.1 State transition of map-update

Condition		Output
j	$map_0(x,y)$	
$j < L$	0	-1
$j < L$	1	0.5
$j = L$	0	1
$j = L$	-1	0.5

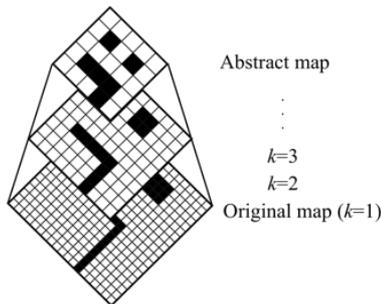


Fig.2 Concept image of multi resolution map

3. MULTI-RESOLUTION MAP

この章では、構築された地図から multi-resolution map を生成する手法について説明する。multi-resolution map の概念図を図2に示す。SLAM によって構築された地図の値は $\{-1, 0, 0.5, 1\}$ である。まず、1 レベルの地図の値として、partially occupied の値を以下の式のように occupied の値に置き換える。

$$map_1(x,y) = \begin{cases} 1 & \text{if } map_0(x,y) = 0.5 \\ map_0(x,y) & \text{otherwise} \end{cases} \quad (12)$$

次に、地図の粒度下げるために地図を再構築していき、我々は解像度のより低い地図を手に入れる。その地図の値は以下の式によって計算される。

$$map_{k+1}(x,y) = \sum_{i=0}^1 \sum_{j=0}^1 map_k(x+i,y+j) \quad (13)$$

$$x = 0, k^2, 2k^2, \dots, X, \quad y = k^2, 2k^2, \dots, Y, \quad k = 1, 2, \dots, K$$

ここで、 k は multi-resolution map における k 番目のレベルの地図を表す。 k の値が増加するにつれて、地図の情報はまばらになっていく。その地図の情報は以下の式によって示される。

$$map_k(x,y) \in \{-2^{2(k-1)}, -2^{2(k-1)} - 1, \dots, -1, 0, 1, \dots, 2^{2(k-1)} - 1, 2^{2(k-1)}\} \quad (14)$$

次に、地図の値を扱いやすくするために、地図の値の正規化を行う。また、以下のメンバシップ関数を定義することにより、Occupied, Uncertainty, Empty の度合いを扱っていく。また、各メンバシップ関数を図3に示す。

$$n(x,y) = \frac{1}{2^{2(k-1)}} map_k(x,y) \quad (15)$$

$$\mu_k^{Occ}(x,y) = \begin{cases} n(x,y) & \text{if } n(x,y) \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

$$\mu_k^{Unc}(x,y) = 1 - |n(x,y)| \quad (17)$$

$$\mu_k^{Emp}(x,y) = \begin{cases} -n(x,y) & \text{if } n(x,y) < 0 \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

ここで、 μ_k^{Occ} 、 μ_k^{Unc} 、 μ_k^{Emp} はそれぞれ、Occupied, Uncertainty, Empty の度合いを示す。我々はこれらの値を用い、効率のよい未探索領域の探索やそこまで、移動するための経路計画を行っていく。基本的に未探索のセルの値は $map_k(x,y)=0$ となる。しかしながら、 $map_k(x,y)$ は単純なセルの総和である。そのため、未探索でないにもかかわらずセルの総和が0になってしまう場合が存在する。よって、我々は以下の $map_k^{Unk}(x,y)$ を定義する。

$$map_{k+1}^{Unk}(x,y) = \sum_{i=0}^1 \sum_{j=0}^1 |map_k(x+i,y+j)| \quad \text{if } k = 1$$

$$map_{k+1}^{Unk}(x,y) = \sum_{i=0}^1 \sum_{j=0}^1 map_k^{Unk}(x,y) \quad \text{if } k > 1 \quad (19)$$

$$x = 0, k^2, 2k^2, \dots, X, \quad y = k^2, 2k^2, \dots, Y, \quad k = 1, 2, \dots, K$$

さらに未探索の度合いを以下の定義式により表す。

$$\mu_k^{Unk}(x,y) = 1 - \frac{1}{2^{2(k-1)}} map_k^{Unk}(x,y) \quad (20)$$

我々はこれらのメンバシップ関数の値を使用することにより、探索空間を下げることができ、より小さいサイズの抽象化された地図を手に入れることができる。

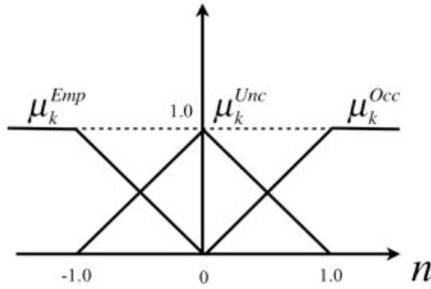


Fig.3 Member ship function of each state

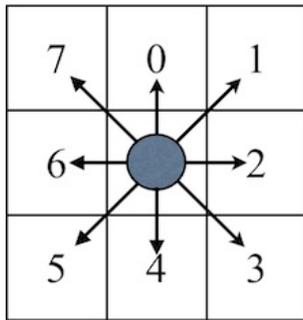


Fig.4 The moveable direction of the robot

4. 未知環境の探索

本稿における未知環境の探索方法として、強化学習における環境同定の方法として用いられる k -確実探索法の考え方を拡張したものを用いる [7]. 本稿では、 k -確実なルールの代わりにして、行動地図を定義し、行動地図を k 回探索を行うことを考えることにより効率の良い探索を実現する。

4.1. 行動地図

本稿では、ロボットが探索行動を行った履歴を地図として保存しておくことにより効率のよい未知環境の探索を行う。基本的なセルベースの地図においては、ロボットが行える移動方向は8方向に限定することができる (図4). よって行動地図 $map_k^{act}(x,y)$ の更新則を以下のアルゴリズムに、更新される地図の値の関数 $g(\cdot)$ を表2に示す。ここで、direction は図4の番号に基づいている。また、 N は一回の移動によってロボットが精度良く探索することが可能な範囲を示し、用いるセンサレンジによって変更される、 k は multi-resolution map での解像度のレベルを表し、行動地図においても、現在の multi-resolution map と同じ解像度のレベルを用いることにより、地図の参照回数の削減を行う。

Algorithm action map update(direction, x,y)
for $m=0$ to $N-1$ do
$map_k^{act}(x,y) = g(map_k^{act}(x,y), direction)$
endfor

Table 2 Update value of action map

direction	
0	$map_k^{act}(x-m,y) += N-m$ $map_k^{act}(x+m,y) += N-m$
1	$map_k^{act}(x-m,y) += N-m$ $map_k^{act}(x,y+m) += N-m$
2	$map_k^{act}(x,y-m) += N-m$ $map_k^{act}(x,y+m) += N-m$
3	$map_k^{act}(x,y-m) += N-m$ $map_k^{act}(x-m,y) += N-m$
4	$map_k^{act}(x+m,y) += N-m$ $map_k^{act}(x-m,y) += N-m$
5	$map_k^{act}(x+m,y) += N-m$ $map_k^{act}(x,y-m) += N-m$
6	$map_k^{act}(x,y-m) += N-m$ $map_k^{act}(x,y+m) += N-m$
7	$map_k^{act}(x,y+m) += N-m$ $map_k^{act}(x+m,y) += N-m$

4.2. Multi-resolution map を用いた未知環境の探索

ロボットが探索活動を行う初期値の $map_k^{act}(x,y)$ は全て 0 とする。ロボットが行動を行い、 $map_k^{act}(x,y) \geq 1$ となる地図上のセルでは $\mu_k^{unk}(x,y) = 0$ となると考えられる。よって本稿では、1-確実探索を行い、未知環境の探索を行うことを考える。以下に、探索のための解候補決定のアルゴリズムを示す。

Algorithm Exploration of unknown environment
$n^{can} = 0$
for $i=-K$ to K do
for $j=-K$ to K do
if $\mu_k^{safe}(x+i,y+j) = 1$ and $map_k^{act}(x+i,y+j) = 0$
add candidate $< x+i,y+j >$
$n^{can} += 1$
end for
end for
for $i=0$ to n^{can}
fitness calculation(candidate)
end for
return maximum candidate

ここで、 K は解候補の範囲、 n^{can} は解候補の数、candidate は解候補を示す。また、各解候補の適応度の値はブレゼンハムのアルゴリズムを用い現在ロボットがいる位置から解候補までの簡略された経路を生成し、計算される [8]. 以下に適応度の式を示す。

$$fit_i^{path} = \sum_{l=1}^L \sum_{j=0}^{N-1} \begin{cases} N-m & \text{if } (map_k^{act}(x,y) = 0 \wedge \mu_k^{emp}(x,y) = 0) \\ 0 & \text{otherwise} \end{cases} \quad (21)$$

ここで、 L は経路の長さを表す。この適応度が最大となる解候補を次の目標地点とする。また、ロボットが一定距離を進むごとにノードを配置していき、全ての解候補の適応度が 0 となる地点では、一つ前のノードに戻り、次の解候補の決定を行う。ま

た、終了条件はロボットがノードの初期位置に戻り、初期位置における解候補の適応度が全て 0 となった場合とする。

5. 実験結果

本稿における実験では、提案手法に基づく SLAM を用いて構築した、 $8 \times 8m$ の室内の地図を用いてシミュレーション上における実験を行った。図 5 に用いた地図と初期位置を示す（初期位置：青丸）。また、本実験における各パラメータとして、構築された地図の説のサイズは $16mm$ 、探索に用いた解像度 $k=5$ 、行動地図の更新値 $N=3$ を用いた。また、各ノードの配置間隔は、 $3m$ として実験を行った。図 6 に各 time step における解候補の数と、図 7 に探索を行った実験結果を示す。図 7 の(a)のように初期段階では、適応度の値はロボットがまだ訪れた事ない方向が最大値となるため、ロボットは後戻りをする事なく効率よく、次の探索ポイントを決めていく。ロボットが安全に行動できない領域まで探索を行うと、図 6 のように解候補の数は 0 に近づいていく、このように解候補の数が全て 0 となる地点に到達すると、1 個前に配置されたノードに戻り、まだ訪れていない未知領域の探索を行っていく（図 7 (b),(c)）。最終的に $t=57$ の時、ロボットがスタート地点のノードに戻り探索を終了した（図 7 (d)）。このように、行動地図を用いることにより、まだロボットが探索していない領域を決定することができるため、効率のよい探索が行えたと考えられる。また、 $\mu_k^{Emp}=0$ で $map_k^{act}>0$ とならなかった領域は 1 セルのみだった。さらに、図 6 における解候補の数も step が進むにつれ、減衰し収束していった。よって、ロボットが行動できるほぼすべての領域を 1-確実にすることができ、未知領域の探索を十分に行なえたのではないかと考えられる。このように、multi-resolution map を用いることにより、ロボットが安全に行動することのできるセルを値として得ることができる。また、セルの解像度を落とすことにより、探索の際にセルの参照回数の削減を行えることができ、さらに行動地図を用いることで次の探索地点を効率よく決定することが可能となり、未知環境における探索が行うことができた。



Fig.5 Experimental environment

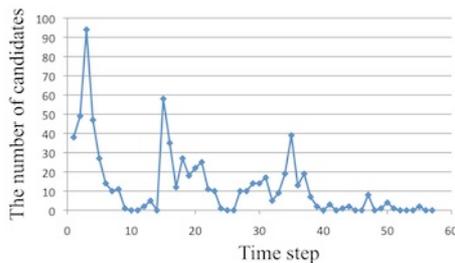


Fig.6 The number of candidates

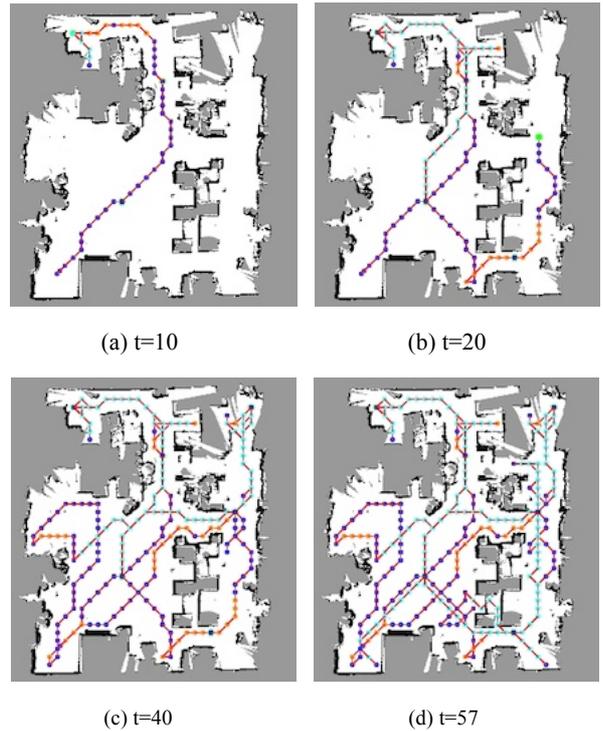


Fig. 7 Experimental result of exploration

6. 結言

本稿では、未知領域の探索を行うために、占有格子空間に基づく SLAM を提案した。また、効率のよい探索を行うために multi-resolution map の構築、multi-resolution map を用いた探索方法の提案を行った。実験では、本提案手法を用いて構築された地図上でのシミュレーション実験を行い、地図のほぼ全ての領域を効率よく探索できることを確認した。

今後の課題として、実機における実験を行わなければならない。また、より正確な地図を構築するために、探索終了後、不確実と未探索を合わせ持つセルの再探索を行うアルゴリズムの開発を行っていく。

文 献

- [1] S.Huang, G.Dissanayake, "Convergence and Consistency Analysis for Extended Kalman Filter Based SLAM", IEEE Transactions on Robotics, vol.23 (5), pp1036-1049, Oct. 2007.
- [2] T.Bailey, J.Nieto, J.Guivant, M.Stevens, E.Nebot, "Consistency of the EKF-SLAM Algorithm", Proc. of 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.3562-3568, Oct. 2006.
- [3] J. Folkesson, H.I. Christensen, "Closing the Loop With Graphical SLAM", IEEE Transactions on Robotics, vol.23 (4), pp731-741, Aug. 2007.
- [4] M. Kaess, A. Ranganathan, F. Dellaert, "iSAM: Incremental Smoothing and Mapping", IEEE Transactions on Robotics, vol.24 (5), pp1365-1378, Dec. 2008.
- [5] 友納正裕, "エッジ点追跡に基づくステレオカメラを用いた三次元 SLAM", 日本ロボット学会誌, vol.27-7, pp.759-767, 2009
- [6] S. Thrun, W. Burgard, and D. Fox, "Probabilistic Robotics", The MIT Press, 2005.
- [7] 宮崎和光, 山村雅幸, 小林重信, "k-確実探索法: 強化学習における環境同定のための行動選択戦略", 人工知能学会誌, vol.10-4, pp.454-463, 1995.
- [8] D.M.Bourg, G.Seemann, "AI for Game Developers", O'Reilly Media, 2004