

シーンコンテキストを考慮したRRTによる経路生成

○平川翼 山下隆義 藤吉弘亘 (中部大学)

1. はじめに

経路生成は、周囲の障害物や歩行者に衝突することなく移動するための移動経路を決定する問題である。従来の経路生成手法では、壁や歩行者などの障害物の情報のみを考慮して、周囲の環境と干渉しない経路を生成する手法が一般的である。しかし、従来の経路生成手法では、人とは異なる移動経路が生成されることが考えられる。人間の生活空間中で動作するロボットの経路を生成する場合、このようなロボットの振る舞いは人間には理解することが難しく、人とロボットが同じ空間内で協調して行動することが難しいと考えられる。そのため、人との協調した行動を獲得するためには、人間と同様の振る舞いを獲得する必要がある。

そこで、本研究では観測した歩行者の経路と類似し、滑らかで無駄のない移動経路の生成手法を提案する。具体的には、図1(b)に示すように、車道や歩道、建物などのシーンコンテキストから生成された特徴マップおよび歩行者の移動経路データから、歩行者と同様の経路を生成するために最適なコストマップを学習する。得られたコストマップを用いて、現在地から目的地までの移動経路を生成する。このとき、RRT*アルゴリズム [1, 2] を用いることにより、コストを最小にするような移動経路を効率的に求める。また、本稿では経路生成のための新たなデータセットを提案する。データセットには障害物の情報に加え、シーンコンテキストおよび歩行者の移動経路データが含まれており、生成した経路の定量的な評価が可能となっている。評価実験により、提案手法が高精度な経路を効率的に生成することを示す。

2. 関連研究

経路生成は古くから取り組まれている問題であるが、大きく二つの問題が存在する。一つは、歩行者の動きなどの動的な環境変化を考慮して、短期の経路を生成する問題である。代表的な手法として、歩行者との相互作用を考慮した Social force model を用いた手法が提案されている [3, 4]。また、深層強化学習を用いて、動的な環境変化に対して最適に振る舞うための方策を学習により獲得する手法が提案されている [5]。

もう一方は、現在地から目的地へ到達するまでの長期の経路を生成する問題である。代表的な手法として、Artificial potential field (APF) [6] や Cell decomposition [7], Probabilistic road map (PRM) [8] などが提案されている。Rapidly-exploring random tree (RRT) [9] は、効率的に経路を探索することができるため、経路生成において広く用いられているアプローチの一つである。また、Karaman ら [1, 2] は、経路の最適性を保障した RRT*アルゴリズムを提案している。Pérez-Higueras ら [10] は、逆強化学習 (Inverse reinforcement learning; IRL) に基づいた経路生成として、RRT アル

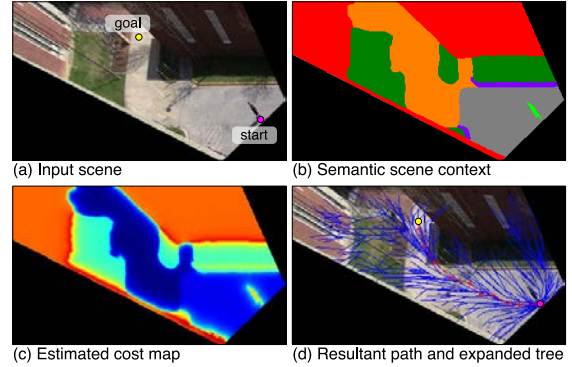


図1 提案手法の概要

ゴリズムを用いて効率的に探索する手法を提案している。これらの手法では、壁などの障害物の情報のみを考慮している。一方、提案手法では単純な障害物領域の情報だけでなくシーンコンテキストをもとに歩行者のような滑らかな経路の生成を目的とする。

IRL に基づく手法も提案されている。IRL ではエキスパートの行動系列を用いて、同様の振る舞いを再現するような報酬を求める。Kitani ら [11] は、IRL に基づいた経路予測手法を提案している。この手法では、動画像から推定したシーンラベルを特徴マップとして使用し、シーン中の歩行者の移動経路を用いて報酬を推定することで、移動経路の予測を実現している。IRL に基づくアプローチには、大規模かつ高次元な空間に対する適用が難しいという問題点がある。一方、提案手法では RRT*アルゴリズムを用いて空間内を探索するため、効率的な経路の生成が可能である。

3. 提案手法

本節では、提案手法の詳細について述べる。

3.1 シーンコンテキストを考慮した経路生成

経路を探索する空間を $X \in \mathcal{R}^2$ とするとき、経路を X 内の座標列 $\mathbf{x} = (x_1, x_2, \dots, x_T)$ と定義する。この時、ある座標 x で得られる特徴ベクトルを $\mathbf{f}(x) = (f_1(x), f_2(x), \dots, f_m)$ とする。ある座標 x で得られるコスト c_r を $\mathbf{f}(x)$ を用いて次のような関数として定義する。

$$c_r(x; \mathbf{w}) = \mathbf{w}^T \mathbf{f}(x) \quad (1)$$

ここで \mathbf{w} は特徴ベクトルに対する重みベクトルである。さらに、経路 \mathbf{x} に対するコスト $c_p(\mathbf{x}, \mathbf{w})$ は各座標で得られるコスト $c_r(x; \mathbf{w})$ を用いて次のように定義される。

$$\begin{aligned} c_p(\mathbf{x}, \mathbf{w}) &= \sum_{t=1}^T c_r(x_t, \mathbf{w}) + \theta \sum_{t=1}^{T-1} \|x_t, x_{t+1}\|_2 \\ &= \sum_{t=1}^T \mathbf{w}^T \mathbf{f}(x_t) + \theta \sum_{t=1}^{T-1} \|x_t, x_{t+1}\|_2. \end{aligned} \quad (2)$$

ここで第一項は経路上の座標で得られたコスト c_r を累積したものを表している。

また、第二項は経路の長さに対する正則化項であり、 θ はスケールパラメータである。第一項のみを用いたコスト関数では、移動距離に対する制約が存在しないため、コストの小さな座標を選択し続けることで、迂回した移動経路が生成されることが考えられる。文献 [10] では、式 (2) と類似したコスト関数を定義しており、特徴ベクトルの要素としてゴールまでの距離が用いられている。しかし、実験では迂回する経路が生成されているため、距離に対する制約が十分に考慮されていない。そこで、提案手法では正則化項として経路の長さに関する情報を導入することで、滑らかな経路の生成を実現する。

提案手法では特徴ベクトル $f(x)$ と重みベクトル w からコスト経路のコストを定義し、コストが最小となる経路を求めることで経路生成を実現する。最小コストの経路を効率的に求めるために、RRT*アルゴリズム [1, 2] を用いる。提案手法のアルゴリズムを Alg. 1 に示す。まず、初期化として初期地点 x_{init} を木構造 T の根ノードに設定する。次に、`Sample` により X 中の座標 x_{rand} をランダムに取得し、 T がノード保有するノードの集合 V から最近傍のノード $x_{nearest}$ を選択する (`Nearest`)。 `Steer` により $x_{nearest}$ から x_{rand} へ距離 η ほど伸ばした先に新たなノード x_{new} を生成する。 $x_{nearest}$ から x_{new} への経路が障害物と衝突しない場合 (`ObstacleFree`)、 x_{new} を T の新たなノードとして追加する。次に、 x_{new} から一定範囲内に存在するノードの集合 X_{near} を求める (`Near`)。 X_{near} を求めるための半径 r は次のように定義される。

$$r = \gamma \left(\frac{\log |V|}{|V|} \right)^{-1/d}, \quad (3)$$

ここで、 $|V|$ は T が保有するノードの数、 d は探索を行う空間の次元数を示しており、 γ はパラメータである。その後、 x_{new} は最もコストが小さくなるノード x_{parent} と連結される。コストを計算する `Cost` では、式 (2) に基づき x_{init} から、対象のノードへ到達するまでに得られる累積コストを計算する。さらに RRT*アルゴリズムでは、`rewire` と呼ばれる処理を行うことで、移動経路のコストを考慮した最適な経路を探索する。この処理では、 x_{new} をノード集合 $x' \in X_{near} \setminus \{x_{parent}\}$ とそれぞれ連結した場合のコストを求め、コストが小さくなる場合にノードを繋ぎ直す。この処理を一定回数反復し経路の探索を行う。探索が終了すると、ノードの中からゴールへのコストが最小となる経路の一つを選択し、最終的な経路を決定する。

3.2 学習方法

提案手法では、特徴ベクトルと歩行者の移動経路データから最適な重みベクトル \hat{w} を推定する。学習には特徴マッチングのアプローチを用いる。すなわち、歩行者の移動経路上で得られる特徴ベクトルおよび、ある重みベクトル w で生成された経路で得られる特徴ベクトルが同じになるように、`exponentiated gradient descent` アルゴリズムを用いて w を更新する。この時の、コス

Algorithm 1 SC-RRT(x_{init})

```

1:  $V \leftarrow \{x_{init}\}; E \leftarrow \emptyset;$ 
2: for  $n = 0$  to  $N$  do
3:    $T \leftarrow (V, E);$ 
4:    $x_{rand} \leftarrow \text{Sample}(n);$ 
5:    $x_{nearest} \leftarrow \text{Nearest}(T, x_{rand});$ 
6:    $x_{new} \leftarrow \text{Steer}(x_{nearest}, x_{rand}, \eta);$ 
7:   if ObstacleFree( $x_{nearest}, x_{new}$ ) then
8:      $V \leftarrow V \cup x_{new};$ 
9:      $x_{parent} \leftarrow x_{nearest}$ 
10:     $X_{near} \leftarrow \text{Near}(T, x_{new}, |V|)$ 
11:    for all  $x_{near} \in X_{near}$  do
12:      if ObstacleFree( $x_{near}, x_{new}$ ) then
13:         $c' \leftarrow \text{Cost}(x_{near}) + c_p(x_{new}, x_{near})$ 
14:        if  $c' < \text{Cost}(x_{new})$  then
15:           $x_{parent} \leftarrow x_{near}$ 
16:        end if
17:      end if
18:    end for
19:     $E \leftarrow E \cup \{x_{nearest}, x_{new}\};$ 
20:    for all  $x' \in X_{near} \setminus \{x_{parent}\}$  do
21:      if ObstacleFree( $x_{new}, x'$ ) and  $\text{Cost}(x') >$ 
22:         $\text{Cost}(x') + c_p(x_{new}, x')$  then
23:         $E \leftarrow E \setminus \{\text{Parent}(x'), x'\}$ 
24:         $E \leftarrow E \cup \{x_{new}, x'\}$ 
25:      end if
26:    end for
27:  end for
28: return  $T = (V, E);$ 

```

ト関数に対する勾配は次のように定義される。

$$\nabla c_w = \bar{f} - \bar{f}_w, \quad (4)$$

ここで、 \bar{f} は歩行者の移動経路で得られる特徴ベクトルの平均値であり、 \bar{f}_w はある重みベクトル w で生成された経路で得られる特徴ベクトルの平均値を示している。この勾配をもとに、 w を次のように更新する。

$$w \leftarrow w e^{-\lambda \nabla c_w}, \quad (5)$$

ここで λ はステップサイズである。この更新を w が変化しなくなるまで繰り返すことで、最適な重みベクトル \hat{w} を学習する。

4. 評価実験

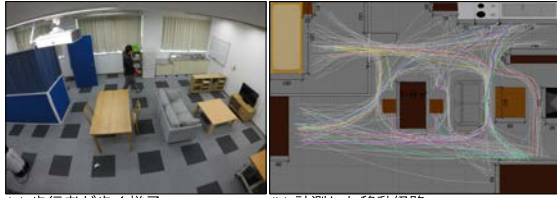
4.1 実験概要

定量的な評価を行うために、本実験では以下のデータセットを用いる。

VIRAT Video Dataset [12]: 本実験では 168 人の歩行者の移動経路データを使用する。また、シーンコンテキストとして、Fully Convolutional Network (FCN) [13] を用いて推定した各シーンラベルの事後確率を特徴ベクトルとして使用する。

Stanford Drone Dataset (SDD) [14]: 本実験では、十分な移動距離の歩行者の経路データのみを使用する。シーンコンテキストとして、手動で付与したシーンラベルを使用する。

Living Space Path Planning Dataset: 定量的な評価のために、新たなデータセットを作成した。このデータセットでは、図 2 に示すような、人間が生活する空間を模した実験室を用意し、その空間中での歩行者の移動経路を収集した。また、空間中に存在する



(a) 歩行者が歩く様子 (b) 計測した移動経路

図2 Living Space Path Planning Dataset の例

表1 各データセットに対する評価結果

	VIRAT	SDD	Living Space
IRL [11]	11.413	19.047	30.109
RRT* [1]	7.327	14.119	24.676
RRT*-IRL [10]	16.845	31.015	36.119
SC-RRT*	25.220	34.579	33.764
SC-RRT*-L2-0.1	8.126	15.825	29.402
SC-RRT*-L2-0.5	7.708	13.609	23.193
SC-RRT*-L2-1.0	6.176	13.099	24.991

壁や机などのシーンラベルの情報および障害物領域の情報も含んでいる。したがって、上記のデータセットと異なり、周囲の環境が完全に観測された状態での経路生成を可能とするデータセットである。

以降の実験では、これらのデータセット中のサンプルの80%を学習に、残りの20%を評価に使用する。

定量的な評価指標として、modified Hausdorff distance (MHD) [15] を使用し、以下の比較手法を用いる。

IRL [11]: IRL では確率分布として経路を生成するため、初期地点から最も確率が高い行動を選択し続けて得られた決定的な経路を用いる。

RRT* [1]: VIRAT および SDD では明示的に障害物の領域を与えることなく経路の生成を行う。一方、Living Space Path Planning Dataset では障害物領域の情報を与えて経路の生成を行う。

RRT*-IRL [10]: 特徴ベクトルとして、シーンコンテキストから生成した特徴ベクトルおよび現在地からゴールまでの距離に関する特徴を用いて経路の生成を行う。

提案手法 (SC-RRT*): 2種類の手法を実験に用いる。一つ目はコスト関数に距離に関する正則化項を導入しない SC-RRT*、もう一つは正則化項を導入した SC-RRT*-L2- θ である。この時、 θ は正則化項に対するスケールパラメータの値を示している。また、提案手法には η および r がパラメータとして存在するが、これらの値は実験的に決定した。一方、 θ は経路の精度に大きく影響すると考え、値を変更した場合の結果を示す。

4.2 実験結果

定量的な評価結果を表1に、生成された経路の例を図3に示す。RRT*アルゴリズムに基づいた手法 (RRT*, SC-RRT*, SC-RRT*-L2- θ) の結果は、1,000回の反復で得られたものを示している。図3(a, e, h) より、IRLの結果では、提案手法と同様にシーンコンテキストを考慮しているものの、直線的な経路が生成されており、歩行者のような滑らかな経路を生成することが難しい。また、RRT*による結果では障害物領域の情報を用いていないため、VIRAT および SDD では目的地向かって

表2 Living Space Path Planning Dataset における反復回数を変化させた場合の MHD (SC-RRT*-L2-1.0)

	VIRAT	SDD	Living Space
0.5k iterations	6.344	13.290	27.142
1.0k iterations	6.176	13.099	24.991
2.0k iterations	6.132	12.981	22.648

表3 各アルゴリズムの処理時間 (単位: seconds)

	VIRAT	SDD	Living
IRL [11]	491.8	1,144.6	2,894.5
RRT* (1.0k iter.) [1]	5.1	5.3	3.8
SC-RRT*-L2-1.0 (0.5k iter.)	5.5	7.6	8.5
SC-RRT*-L2-1.0 (1.0k iter.)	19.0	28.3	35.3
SC-RRT*-L2-1.0 (2.0k iter.)	74.5	118.6	132.7

直線的な経路を生成している (図3(a-f))。Living Space Path Planning Dataset では障害物を避けた経路を生成しているが、経路の長さのみを考慮しているため、壁や机などの物体に接近した経路を生成している。経路の長さに対する正則化項を導入していない SC-RRT* では、大きく迂回した経路を生成していることがわかる。これは、経路の長さを考慮していないために、コストの小さな座標を選択し続けたためと考えられる。同様に、RRT*-IRL でも迂回した経路を生成している。RRT*-IRL では、ゴールまでの距離を特徴ベクトルの要素として定義している。しかし、この特徴に対する重みが学習により小さな値となっているために、経路の長さが考慮されず、SC-RRT*と類似した結果が得られたと考えられる。

一方、提案手法 (SC-RRT*-L1- θ) では、経路の長さに対する正則化項を導入することで、歩行者の移動経路と類似した経路が生成されており、全てのデータセットに対して精度が向上している。また、 θ の値を変化させることで、精度が変化していることがわかる。 θ を0.1のような小さな値に設定した場合、経路の滑らかさが失われるために、精度が低下している。また、データセットに応じて最適な θ の値は異なることがわかる。そのため、最適な値の設定は今後の課題の一つである。

4.3 収束性に対する評価

表2に異なる反復回数で生成された経路に対するMHDを示す。500回の反復でMHDはある程度収束しており、その後の反復で得られる改善はわずかなものとなっている。したがって、1,000回程度の反復で十分な精度が得られることが確認できる。

4.4 処理時間の比較

提案手法はRRT*アルゴリズム、すなわちランダムサンプリングに基づいている。そのため、最適な経路を生成するためには十分な反復回数が必要となる。本節では、反復回数の変更による精度の変化と計算時間の関係について議論する。表3に計算時間を示す。各手法はPythonにより実装されている。IRLでは、多くの処理時間を要することがわかる。一方で、提案手法およびRRT*では、IRLに比べて処理時間が短い。従来のRRT*と比較すると提案手法では計算時間が増加しているが、これは単純な経路の長さを計算するRRT*

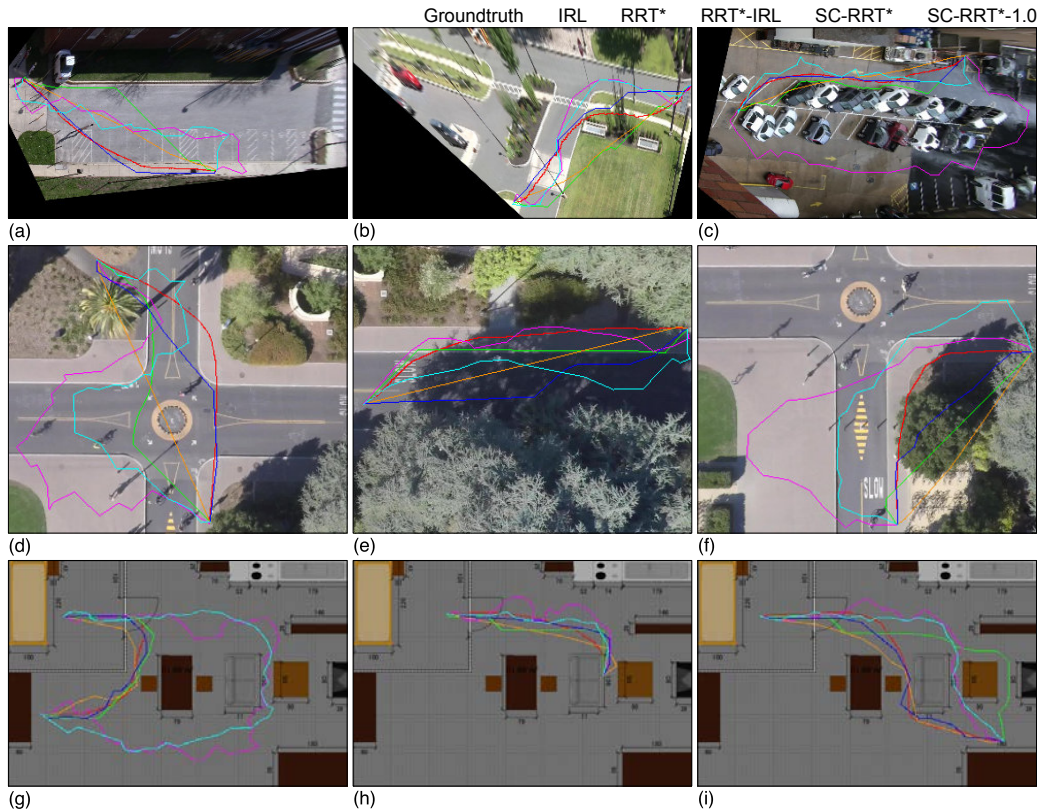


図3 経路生成結果. 上段: VIRAT Video Dataset, 中段: SDD, 下段: Living Space Path Planning Dataset の結果.

に比べて、コストの計算に時間を要するためと考えられる。表2の結果を考慮すると、1,000回の反復で十分な精度の経路が生成されているため、MHDと比較して効率的な経路生成が可能である。

5. おわりに

本稿では、シーンコンテキストを導入したRRT*による経路生成手法を提案した。提案手法では、シーンコンテキストから生成された特徴マップおよび歩行者が移動した経路データからコストマップを推定する。推定したコストマップに対してRRT*アルゴリズムを適用することでコストが最小となるような経路を生成した。実験により、短時間かつ高精度な経路を生成することを示した。

今後の予定としては、より高次元な空間における経路の生成および、動的な環境を考慮した経路生成などが挙げられる。

謝辞 本研究は科研費(JP16H06540)の補助を受けたものである。

参考文献

- [1] S. Karaman, E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," in *RSS*, 2010.
- [2] S. Karaman, E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int J Rob Res*, vol. 30, no. 7, pp. 846–894, 2011.
- [3] D. Helbing, P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, no. 5, pp. 4282–4286, 1995.
- [4] G. Ferrer, A. Garrell, A. Sanfeliu, "Robot companion: A social-force based approach with human awareness-navigation in crowded environments," in *IROS*, 2013.
- [5] Y. F. Chen *et al.*, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *ICRA*, 2017.
- [6] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int J Rob Res*, vol. 5, no. 1, pp. 90–98, 1986.
- [7] R. A. Brooks, T. Lozano-Pérez, "A subdivision algorithm in configuration space for findpath with rotation," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 2, pp. 224–233, 1985.
- [8] L. Kavraki, J. C. Latombe, "Randomized preprocessing of configuration space for path planning: articulated robots," in *IROS*, 1994.
- [9] S. M. LaValle, J. James J. Kuffner, "Randomized kinodynamic planning," *Int J Rob Res*, vol. 20, no. 5, pp. 378–400, 2001.
- [10] N. Pérez-Higueras, F. Caballero, L. Merino, "Learning Robot Navigation Behaviors by Demonstration Using a RRT* Planner," in *ICSR*, 2016.
- [11] K. M. Kitani *et al.*, "Activity forecasting," in *ECCV*, 2012.
- [12] S. Oh *et al.*, "A large-scale benchmark dataset for event recognition in surveillance video," in *CVPR*, 2011.
- [13] J. Long, E. Shelhamer, T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.
- [14] A. Robicquet *et al.*, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *ECCV*, 2016.
- [15] M. P. Dubuisson, A. K. Jain, "A modified hausdorff distance for object matching," in *ICPR*, 1994.