

歩行者が存在する通路でのロボット経路計画に関する研究

Research on Robotic Path Planning Based on Passengers' Location Information on a Sidewalk

情報工学専攻 中島 巧乃

NAKAJIMA Takuno

概要

本研究では、歩行者が通行している通路を移動するロボットの経路計画問題について考える。まず、数フレーム先の各歩行者の位置を予測し、それを楕円で表す。その後、それらの楕円に対する近似 Voronoi 図を生成する。近似 Voronoi 図を用いて作成した経路候補のグラフにおいて、なるべく歩行者に接近し過ぎたり、高速ですれ違ったりしないものを選択するために、辺のコストの求め方を考え、その重み付きグラフ上で経路を求めた。

キーワード：経路計画問題, Voronoi 図, Dijkstra 法

1 序論

日本では少子高齢化により労働人口が減少することが予想され、様々な産業でロボット活用の期待が高まっている。これに先立ち、羽田空港では国土交通省並びに経済産業省との連携のもと“Haneda Robotics Lab”を開設、実証実験を行った。この実験では、2020年のオリンピック開催時に実際に運用できるような清掃、移動支援、そして案内ロボットの開発を目標としている。このように、歩行者が存在するような動的環境下での補助ロボットに関する研究は今日盛んに行われている。

このような研究では、通路にロボットの通る道を予め用意することは難しい。何故ならば、ロボットのコースを用意してもそこが荷物や人などで妨げられる可能性があり、一方そのコースを柵などで分離してしまえば、ロボットの活用が難しくなる。そのため、走行ガイドなどの補助システムを使わずにロボットを行動させる技術開発が求められる。

そこで本研究では、歩行者がいる環境下でのロボットの経路計画について考える。この経路計画の目標は、歩行者との衝突を回避し、且つ歩行者の妨げとなるような行動はしないようにすることである。即ち、歩行者の目の前を横切ったり、高速ですれ違ったりするといった、歩行者の軌道に干渉するような経路は極力選択をしないようにする。その上で、可能な限り遠回りをしないような経路を選択することを目標としている。

1.1 関連研究

ロボットの経路計画に関する研究には、Weng の手法 [3] がある。これは、回避すべき歩行者集合を一つの大きな障害物であるとみなし、その障害物を迂回するような経路を作成する。この方法はシンプルな考え方であるため、歩行者が多く存在しても計算することができる一方、歩行者一人一人の要素を無視しているため、障害物の範囲は大きくなる。そのため、通路などの狭い環境下では大きな障害物となり避けることができず、経路計画することができなくなる。また、歩行者に囲まれている場合には移動計画すること自体が出来なくなる。そこで本研究は、Walker と Dodd の手法 [2] を用いる。これは歩行者が移動する場所を予測して、それを楕円にて表現する。そして、その楕円に対する Voronoi 図を構築することで、歩行者から離れたところの経路を選択することができる。この手法をベースに用いて効率の良い経路計画を考えることにする。

2 問題設定

本研究では、歩行者が存在する通路でのロボット経路計画を行う。前提として、歩行者の位置情報を各フレームごとの時刻で正確に取得できるものとする。この経路計画問題を、[2] を基に定義する。

入力は、移動させたいロボット R の現在地 R_s 、目標到達地点 R_d 、経路予測をしたいフレーム数 f 、そして歩行者の現在の位置データ H_t 、及び1フレーム前の歩行者の位置データ H_{t-1} である。

出力は、ロボットの通るべき経路である。ここで、1.1 節で述べたように、歩行者はロボットに影響されないで移動可能なようにする必要がある。即ち、次の要件を満たす必要がある。

- ロボットは歩行者と接触しない。
- ロボットは歩行者の目の前を横切らない。
- 歩行者がより安全に歩けるよう離れた所を通る。

最適な経路とは、各歩行者の移動地点 H_{t+1} を推定して、ロボットがその歩行者との衝突を回避できるものである。これは上述の三点を達成するためには欠かせない。そのために潜在的な衝突可能性を計算して回避し、尚且つ歩行者に接近するような位置にロボットを移動しない経路を作成し選択する必要がある。

2.1 ロボットについて

ロボット R は半径が r の単純な円形をしていて、かつホロノミック運動学を満たす(即ち、瞬間加速性と全方向への運動が可能である)と仮定する。また、ロボットの最高速度は v_R であり、断りのない場合はこの速度で航行するものとする(図 1(a))。

2.2 歩行者について

歩行者 h は半径 r_h の単純な円形をしている。通路を歩いている、または止まっているものとし、歩行者同士は衝突をしない。

また、歩行者はマルコフ性を持つと仮定することで、統計データ [1] に基づいた移動予測を行うこととする。このデータを用いることで、次のフレーム時に h が存在し得る範囲を、図 1(b) のような楕円 E_h で表現することが可能である。この楕円は歩行者の急停止だけではなく速度や横方向への変化を考慮しており、二自由度のカイ二乗分布に従い楕円を作成し、90%の信頼区間に歩行者の存在可能性を閉じ込めている。尚、止まっている歩行者には半径 r_h の円を与えるものとする。

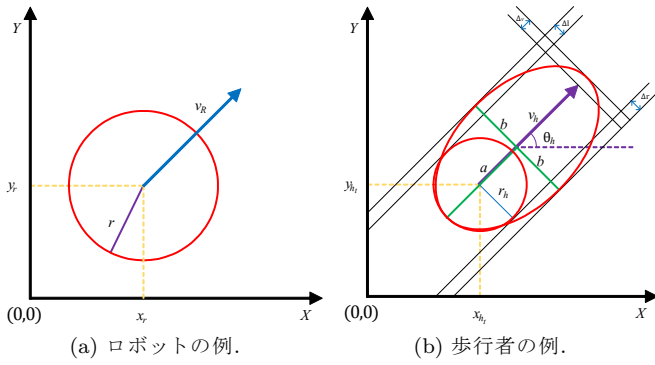


図 1. ロボットと歩行者のモデル.

3 既存手法

ベースとなる Walker と Dodd の手法 [2] は、大きく分けて 3 つのステップに分かれている。まず、各歩行者の位置情報から歩行速度を算出し、歩行者の移動予測範囲を作成する。次にその楕円に対する Voronoi 図を作成し、経路候補となるグラフ (経路候補グラフ) $G = (V, E)$ を作成する。但し、 V は頂点集合で、 E は有向辺の集合である。最後に、そのグラフの各辺に重みを与え、総和が最も小さくなるような経路を選択する。

3.1 歩行者の移動予測

H_t と H_{t-1} を基に各歩行者 h の速度 v_h を計算し、[1] に基づき移動予測範囲を表す楕円 \mathcal{E}_h を作成する。計算量削減のため \mathcal{E}_h を、 H_t での位置と、そのままの速度で f フレーム直進した位置にそれぞれ半径 r_h の円 (図 2 赤円, 青円), 及び短径の一方の端点を通り、2 円が内接するような円を両端点に作成し (図 2 緑円), この 4 つの円の弧を用いた近似楕円 (Pseudo Ellipse, PE) ρ_h に置換する (図 2)。そして $\mathcal{P} = \{\rho_1, \dots, \rho_n\}$ を PE の集合 \mathcal{P} とする。但し、 $n = |H|$ である。

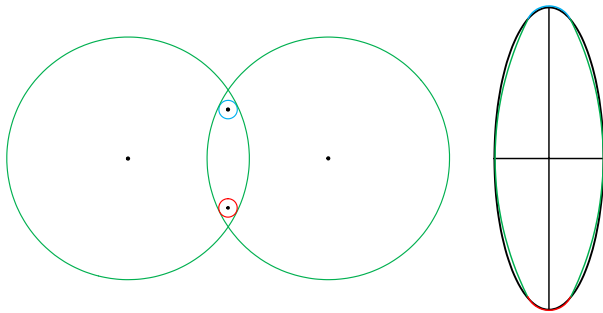


図 2. 近似楕円の例.

3.2 経路候補グラフの作成

前節で説明した歩行者の移動予測を基に、その移動予測範囲と交差しない R_s から R_d までの経路を作成する。そのために、歩行者の移動予測範囲を表現する全ての PE の集合 \mathcal{P} に対する Voronoi 図を作成する。厳密な \mathcal{P} に対する Voronoi 図を求めるには計算量が多くなってしまいうため、各 PE に対する中点、及び Voronoi 頂点を計算後、その頂点同士を結んだ近似 Voronoi 図を作成する。

まず各歩行者の PE の四つの円の中心を用いた四角形の重心を頂点とした Delaunay 三角形分割を行い、

Voronoi 図を計算する三つの PE の組合せ \mathcal{P}_i ($i = 1, \dots, m$ かつ $\bigcup_i \mathcal{P}_i = \mathcal{P}$) を求める。ここで、 n 個の点に対して Delaunay 三角形分割を行った際にできる三角形の数は高々 $2n - 2$ であることが知られているため、 $m \leq 2n - 2$ である。続いて、各 \mathcal{P}_i に対して、二つの PE の最短距離線分の中点 (Middle Point, MP) と三つの PE から等距離な Voronoi 頂点 (Voronoi Point, VP) をそれぞれ計算する (図 3(a))。三点の MP と、一点の VP を作成したならば MP と VP を V に追加した後、各 MP と VP をそれぞれ有向辺で結び E に追加する。これが \mathcal{P}_i に対する近似 Voronoi 図である (図 3(b))。これを全ての PE の組合せに対して計算し統合させたものが、PE の集合 \mathcal{P} に対する近似 Voronoi 図である。また、PE と交差しない、MP 同士を結んだ線分をショートカットとして E に追加する (図 3(c))。この近似 Voronoi 図とショートカットを合わせたものを経路候補グラフとして用いる (図 3(d))。

そして、 R_s, R_d を V に追加した後、 R_s と R_d を各 MP と接続可能であるかを確認し、接続できる辺を全て有向辺で接続して、 E に追加する。ここで、点 s と t が接続可能であるとは、線分 st 上に歩行者の PE が存在しないことを指す。このグラフ全体 $G = (V, E)$ を経路候補のグラフとして考える。

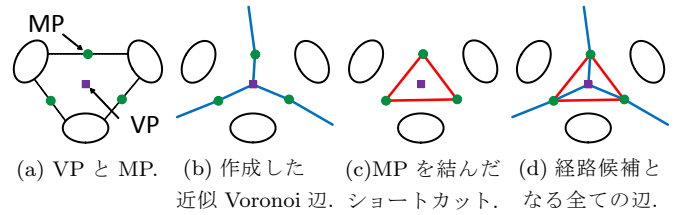


図 3. 三つの PE に対する経路計画例.

3.3 経路の選択

3.2 節で作成した全ての有向辺 $uv \in E$ にコスト cost_{uv} を与える。ここでは

1. 有向辺の長さ (length_{uv})
2. 歩行者との距離 (dist_{uv})
3. 歩行者に接近する角度 (ang_{uv})

の三点を考慮して次式のように与える。

$$\text{cost}_{uv} = \text{dist}_{uv}^{-2} \times \text{ang}_{uv} \times \text{length}_{uv} \quad (1)$$

これを全ての有向辺に対して計算を行う。そして、この経路候補グラフ G に対して Dijkstra 法を用いて総コスト最小の経路を選択する。

4 提案手法

既存手法 [2] では多くの歩行者が存在する環境下であっても効果的な計算ができる一方で、接近する角度が同じであれば速度に依ることなく同等であるというコストの与え方を式 (1) では行っている。また歩行者同士が密集している場合であっても、その歩行者の間を通るようにしてしまうため、危険性の高い経路計画となることも考えられる。更には、一度に予測したフレームの時間内にすれ違う可能性のない歩行者とすれ違う可能性のある歩行者が、同じように考慮されてしまう。これらの問題を解決した経路計画を行うような手法を本研究では提案していく。但し、経路候補グラ

フ $G = (V, E)$ は 3.2 節で説明したものをそのまま用いた.

4.1 各辺に与えるコスト

経路作成を行うにあたり、経路候補グラフの各辺に、より多くの情報を加味してコストの計算を行うこととする. そのため、各経路候補グラフの有向辺 $uv \in E$ のコスト cost_{uv} を

1. 歩行者との距離 (dist_{uv})
2. 歩行者との相対速度 (rel_{uv})
3. 歩行者との位置関係に応じたコスト (posi_{uv})
4. uv の通過に要する時間 (time_{uv})

の四点を考慮して次式のように与える.

$$\text{cost}_{uv} = \text{dist}_{uv}^{-2} \times \text{rel}_{uv} \times \text{posi}_{uv} \times \text{time}_{uv}$$

1 は有向辺 uv とそれに一番近い歩行者 h_{near} との距離を表す.

2 は、有向辺 uv に h_{near} との相対速度を計算する. ここで有向辺 st における**相対速度**とは、 st から h_{near} を観測した際の速度である.

3 は、ロボットが有向辺 uv を通過する際に、最も近い歩行者 h_{near} との位置関係に応じてコストを与える. ここで uv における**歩行者との位置関係**に応じたコスト posi_{uv} とは、以下の四通りに分類される.

- i. uv が歩行者の後ろを通過する場合
- ii. uv が歩行者の横で追い越す場合
- iii. uv が歩行者の横ですれ違う場合
- iv. uv が歩行者の前を横切する場合

ここで、 posi_{uv} の値は $i < \text{ii} < \text{iii} < \text{iv}$ とすることで、歩行者と接触しにくい経路を選択するようになっている.

4 は、有向辺 uv の通過に要する時間 time_{uv} をコストとして与える. 提案手法ではロボットの速度を変えることがあるため、距離ではなく時間を用いることで有向辺の長さに応じたコストを与える.

4.2 減速メソッド

歩行者と接近している際、高速ですれ違うことがないようにするため次のような減速メソッドを追加する.

まず有向辺 uv におけるコスト cost_{uv} を計算するために、 dist_{uv} , rel_{uv} , posi_{uv} 及び time_{uv} を計算する. その後、歩行者に関するコスト near_{uv} を

$$\text{near}_{uv} = \text{dist}_{uv}^{-2} \times \text{rel}_{uv} \times \text{posi}_{uv}$$

として与える. この near_{uv} を予め設定した閾値 $\text{near}_{\text{limit}}$ と比較し、超えている場合はロボットの速度を落とすことで閾値を下回らないようにする. もし落とすべき速度がロボットの最高速度を超えていたならば、 $\text{near}_{uv} = \infty$ とすることでその有向辺を選択されないようにする. 以上を実現させるためにアルゴリズム 1 を用いる.

4.3 経路選択手法

歩行者に接近し過ぎないながらも不必要な大回りを避けるために、移動可能な範囲と範囲外とでコストに傾斜をつけることを考える. 具体的には、移動可能な範囲内は歩行者に関するコスト near_{uv} を重視したコストで計算を行い、範囲外ではそこから移動目的地 R_d までに掛かる時間を優先するよう計算を行うようにする. ここで、 R_s から頂点 v までの時間に関するコスト

アルゴリズム 1 減速メソッド

```

1:  $D \leftarrow \text{dist}_{uv}$ 
2:  $R \leftarrow \text{rel}_{uv}$ 
3:  $P \leftarrow \text{posi}_{uv}$ 
4:  $T \leftarrow \text{time}_{uv}$ 
5:  $H \leftarrow D^{-2} \times R \times P$ 
6: if  $H > \text{near}_{\text{limit}}$  then
7:    $R' \leftarrow \text{near}_{\text{limit}} \div (D^{-2} \times P)$ 
8:   if  $R - R' > v_R$  then
9:      $\text{rel}_{uv} \leftarrow R'$ 
10:     $\text{near}_{uv} \leftarrow \text{near}_{\text{limit}}$ 
11:     $\text{time}_{uv} \leftarrow T \times \frac{R}{R'}$ 
12:  else
13:     $\text{near}_{uv} \leftarrow \infty$ 
14:  end if
15: else
16:    $\text{rel}_{uv} \leftarrow R$ 
17:    $\text{near}_{uv} \leftarrow H$ 
18: end if
19:  $\text{cost}_{uv} \leftarrow \text{near}_{uv} \times \text{time}_{uv}$ 
20: return  $\text{cost}_{uv}, \text{time}_{uv}$ 

```

の合計が $0.4 \times f$ 以下の場合は頂点 v は**予測範囲内**と呼ぶ. 但し、有向辺 $uv \in E$ に対して、頂点 u までの時間に関するコストの和は $0.4 \times f$ 未満であるが、 time_{uv} を追加すると $0.4 \times f$ を超える場合は、頂点 v までを予測範囲内であるとする. もし頂点 v が予測範囲内であれば、頂点 v は**予測範囲外**と呼ぶ. 以上を実現するために、次の流れで経路選択を行う.

まず、事前にパラメータ $\text{par}A < \text{par}B$ を与える. これはコストを計算する際に、 time_{uv} の指数部に与える. そして、予測範囲内では $\text{par}A$ を、予測範囲外では $\text{par}B$ をそれぞれ与えることで、有向辺の通過に要する時間の重みを変動させることができる. 即ち、有向辺 $uv \in E$ のコスト cost_{uv} を

$$\text{cost}_{uv} = \text{dist}_{uv}^{-2} \times \text{rel}_{uv} \times \text{posi}_{uv} \times \text{time}_{uv}^{\alpha} \quad (2)$$

と定義する. そして、予測範囲内、範囲外にてそれぞれ $\alpha = \text{par}A$, $\alpha = \text{par}B$ で与えることで実現をする.

まず、各有向辺 E に式 (2) のコストを与える. 但し、ここでは $\alpha = \text{par}B$ である.

次に各頂点 $v \in V$ から、終点 R_d への最小コストを Dijkstra 法で算出する. これは、Dijkstra 法の計算方法から、ある地点から各頂点への最小コストの経路を求めることができる性質を用いることで計算可能である.

最後に、ロボットの現在地 R_s から予測範囲内の頂点を全列挙する. この時通過した経路に対して、式 (2) を用いてコストの和を計算する. 但し $\alpha = \text{par}A$ とする. そして予測範囲内の各頂点から、予め計算しておいた R_d への最小コストをそれぞれ加算し、最もコストの小さい経路を選択する.

4.4 迂回路作成手法

4.3 節での経路選択において、コストの最小値が ∞ になることがある. これは即ちどのような経路を選択しても歩行者と接近し過ぎてしまったり、高速ですれ違うことが発生してしまうため、最小コストの経路も ∞ となってしまうことを意味する. この場合は歩行者

全体を大回りするような迂回路の経路候補グラフ $G' = (V', E')$ を新たに作成する。

まず各歩行者の PE に用いる円のうち、短軸の延長線上に存在する円の中心と、その半径と同じ長さだけ長軸上に存在する円の中心から外側へ配置した点を頂点集合 V_{can} に追加する (図 4(a)). 続いて V_{can} に対する凸包 $Con(V_{\text{can}})$ を考える (図 4(b) 青線). $Con(V_{\text{can}})$ の各辺を二本の有向辺に置き換え、 $Con(V_{\text{can}})$ の頂点とその有向辺を新しいグラフ $G' = (V', E')$ とする. もし V' に R_s が含まれていない場合、 V' の各頂点と接続可能であるかを判定する. もし R_s と接続可能な頂点 $v \in V'$ が存在した場合は、 V' に R_s を、 E' に有向辺 $R_s v$ を追加する. V' に R_d が含まれていない場合も同様に、 V' の各頂点と接続可能であるかを判定し、 R_d と接続可能な頂点 $v \in V'$ が存在した場合は、 V' に R_d を、 E' に有向辺 $v R_d$ を追加する. そして $G' = (V', E')$ を経路候補グラフとして、4.3 節の手法を用いてコストの計算を行う. 但し、 V' に R_s, R_d の一方でも含まれていなければ、迂回路のコストも ∞ とする.

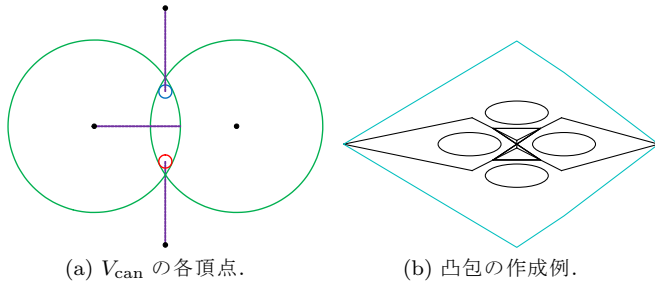


図 4. 迂回路の計算を行う凸包の作成.

4.5 停止メソッド

4.4 節で作成した迂回路でもコストが ∞ になってしまう場合、これはどのような経路を選択しても歩行者に接近しすぎた経路を通ることになる. そこで、そのまま経路を選択せずにロボットを経路候補グラフ上の頂点で停止させることで、衝突する可能性を減らす.

ここで停止させる場所は、止まる場所でのロボットと歩行者の位置関係、及び目標到達地点 R_d との距離を考慮することで、歩行者の目の前で停止することを回避しつつ、 R_d から大きく離れることを回避する. これを実現するために、次の流れで経路計画をおこなう.

まず 4.3.4.4 節で作成した経路候補グラフ G, G' から、コストが ∞ となっている有向辺を削除する. 削除したことにより G, G' は複数の部分グラフに分割される. R_s を含むような部分グラフをそれぞれ $G_s = (V_s, E_s)$, $G'_s = (V'_s, E'_s)$ とする. そして G_s, G'_s の各頂点 $v \in V_s, V'_s$ に対してコストを与え、その中の最小コストの頂点で停止させる. ここで、各頂点 v のコストは以下の三点を考慮して与える.

1. 頂点 v における一番近い歩行者 h_{near} との距離
2. h_{near} との位置関係によるコスト posi_v
3. R_d との距離

この三点を用いてコストを与えることで、可能な限り歩行者の障害とならないようにしつつ、大きく目的地から離れない場所で停止するようにする.

5 計算機実験

これらの手法を実装し、計算機実験を行った. 図 5 内の四角が R_s , 三角が R_d , 破線が既存手法 [2] での経路計画で実線が提案手法での経路計画であり、太線が予測ステップ内の経路である. 既存手法と異なり、相対速度が低くなるような経路を多く選択している. また、歩行者に接近し過ぎない範囲で距離も短くなっている.

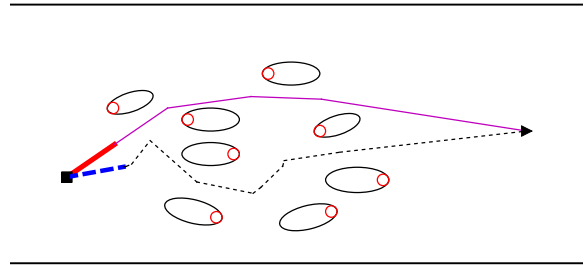


図 5. $\text{par}A = 1.5$, $\text{par}B = 2.5$, $f = 2$, 8 人の歩行者 (赤円が現在位置) に対して経路を作成した例.

6 まとめと今後の研究課題

本研究では、提案のアルゴリズムを用いることで歩行者に接触することなく経路計画を行うことが確認できた. また既存のものと比較して、より歩行者に対して干渉しにくい経路計画を行うことができた.

今後の課題としては、ロボット制約を厳格にした上で経路計画が行えるようにする. 現在ロボットの仮定として、ホロノミック運動学を満たすものとしている. しかし、これは現在開発されているロボットでは満たすことが難しく、また歩行者から見ても動きが予測し辛く危険に感じる恐れがある. そのため、急加速や急旋回をしないよう、非ホロノミック拘束を条件とした経路計画が行えるようにする必要がある.

謝辞

本研究を進めるにあたり、適切な御指導、御指摘をいただきました中央大学理工学部情報工学科の今井桂子教授、鈴木寿教授、並びに明治大学先端数理科学インスティテュートの森口昌樹特任講師に心から感謝致します. また、今井研究室および情報工学専攻の学生各位の協力なしには、この論文は完成しませんでした. 深く感謝致します.

参考文献

- [1] M. Huber, Y.-H. Su, M. Krüger, K. Faschian, S. Glasauer, and J. Hermsdörfer, "Adjustments of speed and path when avoiding collisions with another pedestrian," *PLoS ONE*, Vol.9(Issue 2), e89589, 2014.
- [2] R. Walker and T. J. Dodd, "A novel path planning approach for robotic navigation using consideration within crowds," In *Proceedings of Towards Autonomous Robotic Systems 2015*, pp. 270–282, 2015.
- [3] S. Zeng and J. Weng, "Obstacle avoidance through incremental learning with attention selection," In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, Vol.1, pp. 115–121, 2004.