

松田 多嘉哉, 久喜 友博, 巽 久行, 徳増 眞司
 神奈川工科大学 工学部 情報工学科

1. はじめに

距離場空間モデル^{[1],[2]}は、物と物、物とボイド空間とのスペースの取り合いの問題を扱うための基礎となる空間表現の機能を有する。このスペースの取り合いが主要なテーマの一つとして、ロボットナビゲーションの問題がある。この場合、ロボットが走行を許された路面を障害物を避けながら最も効率良く移動するための経路探索が必要である。この際、障害物や路面についての情報、即ち、走行ドメインとしての環境が、既知である場合と未知の場合がある。筆者らの方法は世界モデルとして距離場空間モデルを用いるものであり、2次元であろうと3次元であろうと、環境が既知であろうと未知であろうと、有効に対応できる。本論文は、問題を2次元に限定して、自律走行ロボットの経路探索を環境既知のもとで行う方法を提案し、距離場空間モデルの有効性を示す。

2. 距離場空間モデルの概念とファジィ論的解釈

距離場空間モデル^{[1],[2]}は、距離場と呼ばれる、目標物と参照点との位置的な関係を示すベクトル場として表現される。走行ロボットは、局所的な情報を距離場の形で収集し分析することで、対象物を認識し理解する。

まず、空間上に物体Gと参照点Pが与えられているものとする。このとき、参照点Pにおける物体Gの距離場は次式で定義される(図1参照)。

$$(IO, d, Q, KIND) \quad \dots(1)$$

- 但し、IO : 点Pの内外属性 (IN / OUT)
- d : 点Pと図形境界δGとの距離またはその下側近似値
- Q : 点Pに対する境界δG上の最短距離点, 求まらないとき, NIL
- KIND : dとQが求まるとき1, 求まらないとき2.

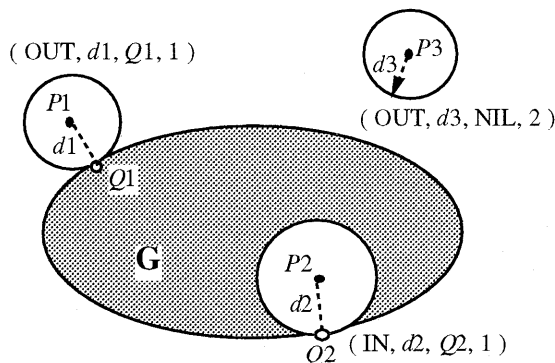


図1 距離場データ

空間推論を展開するために、距離場空間モデルが一種のファジィ空間モデルとして解釈できることを示す。

まず、ユークリッド距離空間 E^n 上で定義された通常の点集合をAとし、集合Aの補集合、内部、外部、境界および閉包を、それぞれ記号 $A^c, A^i, A^e, \bar{A}, \overline{A}$ で表すものとする。次に、 E^n に関わるA上のファジィ集合[A]を、次のようなメンバシップ関数で定義する。

$$\mu_{[A]} : E^n \rightarrow R, \text{ 但し, } \dots(2)$$

$$\begin{cases} \mu_{[A]}(x) > 0 \ \& \ B(x, \mu_{[A]}(x)) \subset A^i, & \text{if } x \in A^i \\ \mu_{[A]}(x) < 0 \ \& \ B(x, -\mu_{[A]}(x)) \subset A^e, & \text{if } x \in A^e \\ \mu_{[A]}(x) = 0 & \text{if } x \in \bar{A} \end{cases}$$

ここで $B(P,r)$ は、 $B(P,r) = \{Q \mid \overline{PQ} < r\}$ で定義された点Pのr近傍と呼ばれる開球である。

次に、式(2)で構成されるファジィシステムが、距離場空間モデルのファジィ論的解釈になっていることを示す。最初に、 E^n 上に構築された距離場空間モデルのもとで、集合Aの距離場 df_A が定義されているものとする。[A]のメンバシップ関数 $\mu_{[A]}$ は、 df_A と次式の様に関係づけられる。

$$\mu_{[A]}(x) = \begin{cases} \text{distance datum of } df_A(x), & \text{if } x \in A \\ -\text{distance datum of } df_A(x), & \text{if } x \in A^c \end{cases}$$

このように、上記ファジィシステムは距離場空間モデルから導出され、物体Aの形状が未知ないし不確定であるときに、空間上の参照点xが集合Aに属するという確かさの程度であるとして解釈できる。結局、ファジィシステムと距離場空間モデルには、互いに一方から他方を誘導するメカニズムが構成できるので、これらの間には自然な対応が存在していると言える。

3. 外部環境が既知である場合の最適ルート探索

外部環境が既知である場合、距離場空間モデルによって、障害物の補空間となる走行可能な領域をトレースすることが可能である。本論では、走行ロボットのドメインは2次元空間で、その外部環境は距離場空間モデルとして予め与えられていることを前提とする。そこで問題は、この距離場空間に於て、走行可能領域に属するスタート点からゴール点に至る最適な経路を求めることである。最適な経路とは、経路の長さ、経路と障害物との間の距離に関わるコスト(ペナルティ)を最小にする経路である。以下では、経路探索にA*アルゴリズムを用いるが、この場合の候補経路はA*アルゴリズムが逐次生成する両端固定の折線群である。従って本論でいう最適とは、A*アルゴリズムに依存した局所的なものとなるが、十分強力で説得力あるものとして構成できる。

最初に、 P_0 (スタート点: P_S), P_1, P_2, \dots, P_n (ゴール点: P_g)を障害物Oの補空間 O^e に属する任意の点列とすると、経路Lは、点列を順次線分で結んで得られる折線で表される(図2参照)。

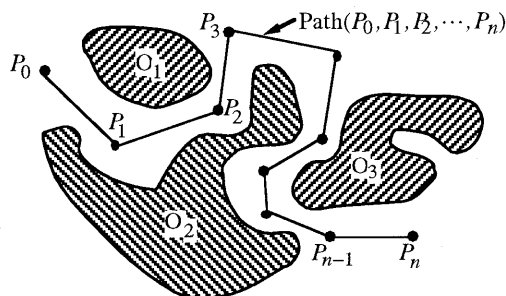


図2 経路探索のドメイン：障害物とパス

経路の要素である各線分，即ち P_i, P_{i+1} を結ぶ線分 $L_{i,i+1}$ には，次式の区間コスト $w_{i,i+1}$ が付与される。

$$w_{i,i+1} = \begin{cases} 1/d_{i,i+1}^2 & \cdots R_{\min} > d_{i,i+1} \\ d_{i,i+1}/R_{\min}^3 & \cdots R_{\min} \leq d_{i,i+1} \end{cases} \quad \dots(3)$$

ここに， $d_{i,i+1}$ は線分 $L_{i,i+1}$ の長さであり， R_{\min} は走行ロボットの大きさを考慮した有効な行動半径の下限として解釈されるパラメータである。経路 L のコストはこの区間コストの総和として，次式で表される。

$$\tilde{g}(L) = \sum_{i=0}^{n-1} w_{i,i+1} \quad \dots(4)$$

経路探索問題は上記コストを最小化する経路 L ，即ち，

$$\arg \min(\tilde{g}(L)) \quad \dots(5)$$

を求める問題である。

一般に A* アルゴリズムは，各枝のコストが定義された経路ネットワークの下で適用されるが，経路探索問題では，このようなものが予め存在するわけではない。そこで A* アルゴリズムの活用を考える場合，経路 L を逐次生成しながら，解の改善を図るという手段が必要になる。本論では，これを距離場空間モデルによって実現し，以下，距離場 A* アルゴリズムと呼ぶことにする。

まず，逐次生成される経路 L によって形成されるネットワークを，次式で定義する。

$$N = (V, E, g, f) \quad \dots(6)$$

但し， V ：節点集合， E ：枝集合，

$g: V \rightarrow R$ ： $g(i)$ はスタート点から，点 i に至る最適経路のコスト，

$f: V \rightarrow R$ ： $f(i)$ はスタート点から点 i までは最適経路を通り，そこを経由してゴールに至るコストの推定値： $f(i) = g(i) + h(i)$

とする。 $h(i)$ はヒューリスティック関数であり，点 i からゴール点までの直線距離を dig とすると，

$$h(i) = \text{dig}/R_{\min}^3 \quad \dots(7)$$

によって評価するものとする。これから距離場 A* アルゴリズムは，次のようにして構成される。

【距離場 A* アルゴリズム】

[Step 1] $V = \{P_s\}$, $E = \phi$, $g(P_s) = 0$,

$f(P_s) = \text{distance}(P_s, P_g)/R_{\min}^3$ として N を作成する。

[Step 2] N における未展開の節点に着目し，最小のコスト推定値 f を有する節点を i とする。点 i における距離場を計算し，その距離場データ d_i を求める。また点 i とゴール点との距離 dig を求める。

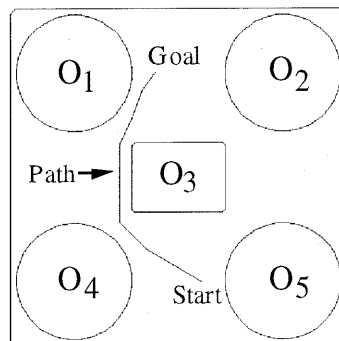


図3 距離場 A* アルゴリズムの実行結果

[Step 3] $d_i \geq \text{dig}$ ならば，ゴール点に到達する線分が存在し， $V \leftarrow V + \{P_g\}$, $E \leftarrow E + \{(i, g)\}$,

$g(P_g) = \text{distance}(i, g)/R_{\min}^3$, $f(P_g) = g(P_g)$ ，として N を更新しステップ 5 へ。

[Step 4] 節点 i の親節点を k (存在しない場合， ϕ) とする。節点 i を中心として半径 d_i の円を描き，その周上に一定個数 ($= N_{\text{ang}}$) の点を等間隔に取る。これらの点 j の内，ベクトル (i, j) とベクトル (k, i) との内積が正となる点 j ($k = \phi$ の時はすべての点 j が対象) に対し，

$$V \leftarrow V + \{j\}, E \leftarrow E + \{(i, j)\},$$

$$g(j) = g(i) + w_{i,j}, f(j) = g(j) + h(j)$$

を行って N を更新する。そしてステップ 2 へ。

[Step 5] N において，ゴール点より親節点を逆に辿ってスタート点に至るルートが，求める経路となる。

[Step 6] 得られたパスの区分点を減らして，直線部が長い滑らかなパスに整形し，評価値を改善する。□

4. 数値実験と考察

前節で述べた経路探索アルゴリズムをプログラミングし， $N_{\text{ang}} = 8$, $R_{\min} = 1$ として数値実験を行った。図 3 は，自律走行ロボットの走行空間を一辺 30 の正方領域と仮定し，周囲の囲いを含めて図のような障害物を配置した場合の，スタート点からゴール点までの経路探索結果を示す。実験結果から，得られたパスは滑らかであり，特に不自然な部分は生じていないことが分かる。処理性能については測定していないが，ほぼ瞬時に結果が得られるのでリアルタイム処理上の問題はないと思われる。

5. おわりに

自律走行ロボットのナビゲーションに関して，障害物を含む周辺環境が既知であるという状況のもとで，2 点間の最適経路を求める問題に対するアルゴリズムの構築と評価を行った。本方法は，従来のようなグリッドによる空間分割ないし障害物との交点計算をベースとしたものではないので，3 次元にも容易に拡張できる。

参考文献

- [1] Nonaka S., Tokumasu S., et al., ; A new geometric model, Proc. of the 15th (ASME) Design Automation Conference on Advances in Design Automation, Vol. 1, pp. 135-141 (1989).
- [2] 徳増, 野中, 他, : 距離場に基づく形状表現法, 情報処理学会論文誌, Vol. 30, No. 1, pp. 50-59 (1998).