知識工学と人工知能 44－7
（１９８６．１．２０）

オクトツリーを用いた経路決定アルゴリズム
Path planning algorithm using the octree

登尾　啓史　　福田尚三　　有本　卓
Hiroshi NOBORIO　Syozo FUKUDA　Suguru ARIMOTO

ABSTRACT A conventional direct or indirect teaching/playback method which has been adopted for controlling industrial robot requests human operators to elaborate a path plan. Furthermore, it is difficult to use the method particularly in extremely cluttered environment. In place of the teaching/playback method, this paper proposes a path planning algorithm based upon the octree representation of a three-dimensional space (a room). The algorithm generates a continuous path for the moving object which begins an initial position to a goal position without collision with obstacles. The algorithm can be used in the task level robot programming.

１．はじめに

複雑な三次元環境のもとで、多様な仕事を行なうことが要求される、知能ロボットの機構は、ＮＣ工作機械などに比べてはるかに複雑であるから、その設計においては、ビジュアルシミュレータなどで動作教示を行なう必要がある。

その際、その動作を人間が教示することは、手間のかかる作業である。そこで、仕事の目標位置と姿勢を人間が与えると、逐次、障害物に対する移動物体の干渉を調べ、障害物回避を考慮した、目標までの経路の自動的な決定が必要となる。この考えは、タスクレベルプログラミングにも通ずるものである。

また、知能ロボットの制御においても、経路が外乱により乱れる場合がある。そのとき現在位置がわかれば、前述のことは、障害物回避と目標への到達を柔軟に選択し、有効な経路を作成することにあたるので、その経路を、各関節軸のサーボ機構を使って実行すればよいことになる。

以上のことより、本研究では、初期位置と最終位置を与えることによって、障害物との接触を避け、あるコスト（現在は距離）に対して準最適なルートの決定を、自律的に行なうという意味での、経路決定の自動化について述べる。

本研究では環境情報を、実空間をブロック分割し、それを階層構造で表現したオクトツリーで、また移動物体を、回転移動を表現しやすいサーフェイスモデルで保持している。このことにより、干渉チェック、最短距離にある物体の認識と距離計算、任意方向の物体の認識と距離計算などを行なうのに、環境内の物体の個数やその形状の複雑さにではなく、移動物体を構成する面の位置と数に依存した、効率的なアルゴリズム([8])を利用できる。

阪大．基礎工．機械．〒５６０豊中市待兼山町１丁１番　TEL．(06)844-1151 ext.4506

それらを使った干渉チェック、及び、柔軟な平行移動や回転移動により、複雑な動きが可能な、実時間処理の出来る、径路決定アルゴリズムを提案する。

このアルゴリズムは、大別して

① 大局的な三次元情報を考慮して、大局的に最適な解ルートを作成する。（４章）

② 周囲の形状にもとずき、解ルートにそって、移動物体の非干渉動作を決定する。（５章）

の２つの処理からなっている。

この方法は、現在までに行なわれている代表的な径路決定問題の研究と比較して、以下のような利点がある。

① 三次元環境を表わすオクトツリーを作成するのに要する時間が、面の数に比例する。

② 三次元空間内の物体の移動にもとずくデータの挿入、削除が、オクトツリー上で容易に行なえる。

③ 誤動作によりあいまいになった現実の位置を、センサによって得られる距離情報から、容易に算出できる。

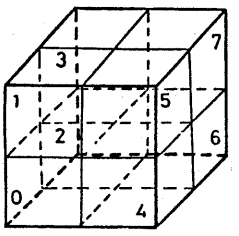
④ 環境表現の精度やアルゴリズムに使える計算時間に対して、動作の複雑度を変えることができるので、必要に応じた柔軟な動作の選択ができる。

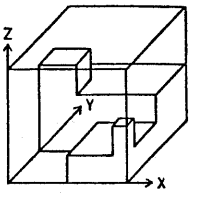
⑤ 移動物体の自由度に制約がない。

２．オクトツリーとDF-表現

①オクトツリー

これは、三次元空間を図２－１のように８分割して、その各々の領域を元の領域のノードに対する子ノードとして表現したものである。物体が存在していない領域は白ノードで表現し、又物体で占有されている領域は黒ノードで表現する（これらは終端ノード）。物体は存在しているが、占有はされていない領域はミックスノードで表現し（これは内部ノード）、再帰的に８分割される。ミックスノードには、黒ノードの占める割合（黒ノードの密度）を記入する。このオクトツリーで、三次元空間の領域情報を表現する。





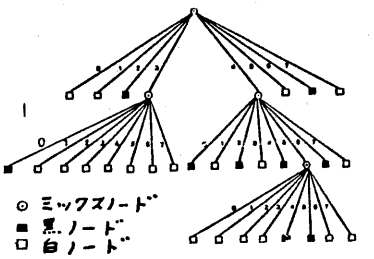


図２－１．８分割の方法　　図２－２．環境領域のオクトツリー変換（位置番号と頂点番号）

②DF-表現

これはオクトツリーを、線形に変換した表現である。その方法は、ツリーのレベルが１つ増すと、”(” を、１つ減少すると”)” を記述し、黒ノードを１、白ノードを０で記述する。例えば、図２－２のオクトツリーは(001(10000000)(10101(00001100)10)010)で表現される。

DF-表現でオクトツリーを表現すると高圧縮となるので、実際に利用する場合には、この陰な形態で持っておき、必要な部分をオクトツリーとして陽な形態で表現する。

また縦型探索の場合は、DF-表現上でアルゴリズムを容易に達成することができる。

３．サーフェイスモデル

サーフェイスモデルは物体を、頂点と面の集合で表わしたものである。頂点座標より、面の方程式Pi
=ai*(x-xi)+bi*(y-yi)+ci*(z-zi)=(ai,bi,ci)*(x,y,z)
-(ai,bi,ci)*(xi,yi,zi)=(ai,bi,ci)*(x,y,z)
-d (i=0,...,n)を得る。

移動物体は面Ｐｉとそれを構成する頂点座標の２つの集合として表現し、平行移動は(xi,yi,zi)を、回転は(ai,bi,ci)を更新することによって行なう。

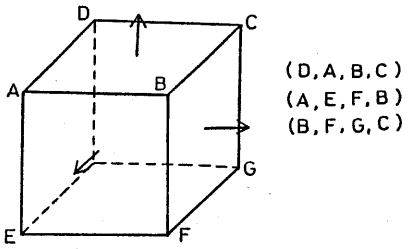


図３－１．サーフェイスモデル

４．大局情報によるルートの決定

移動物体に対する初期位置，最終位置を与え、それをもとにオクトツリーを探索し、解グラフを作成する。物体として移動ロボットを考えるとき、位置．姿勢は重心において考え、マニュプレータを考えると、まず手先が通過できないと動作不可能なので、位置．姿勢はその手先において考えることとする。

コスト（今は距離）を準最適にする解ルート（パスの集合）が得られるまで、解グラフの終端ノードのうち最も妥当なノードを、発生するパスが非干渉となるように、オクトツリーを探索することにより拡張する。

妥当性の判断は、各ノードにおける、現在までの非干渉ルートの距離g 、現在の目標位置までにかかると推定されるコストh の合計f で評価する。コストを表わすheuristic 関数h を楽観的に作成することにより、A*アルゴリズムで最適解が得られる。

また、移動物体が全体空間の軸方向に平行移動できる領域より　大きい最小キューブのレベルを探索レベルと呼ぶことにし、そのレベルまで、３次元環境を表わした前述のDF- 表現から、オクトツリーを作成する。したがって、移動物体によりこのレベルは変化する。

はじめの目標位置は最終位置とし、非干渉パスの決定は、解グラフ作成アルゴリズム、目標位置の変更は目標位置設定アルゴリズムを、作成したオクトツリーに用いることによっておこなう。

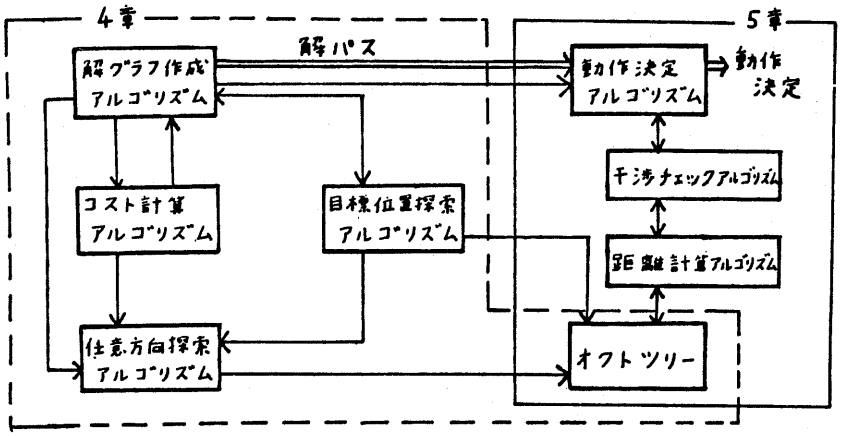


図４－１．アルゴリズム全体のブロック図

（解グラフ作成アルゴリズム）

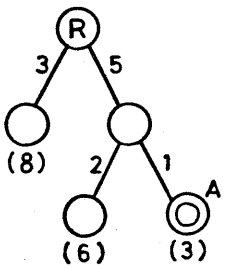
これは、環境の大局的な情報を基にして、移動物体が移動できる複数ルート内から、コストを最小にする可能性のあるルートを選ぶアルゴリズムである。

(1) A*アルゴリズムにより解グラフの内から拡張ノードを選択する。始めは初期位置を表わすノードとする。

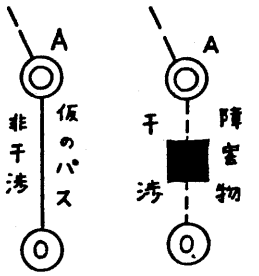
(2) 任意方向探索アルゴリズムを利用して、目標位置に対して拡張ノードからパスを作成しようと試みる。もし、黒ノードやミックスノードを通過しないなら、そのパスを仮に選ぶ。そうでなければ、目標位置設定アルゴリズムを利用して、その障害物に対する回避点を設定し、現在位置から回避点までのパスを仮に選ぶ。

(3) (2) で作成された新たな数個のパスに対しての、拡張ノードにおける移動可能性を５章の移動物体の動作決定法で調べ、移動可能なら、そのパスを解グラフに加える。その本数は、任意の値Mにより制限する。このことにより、解パスの複雑度が決まる。

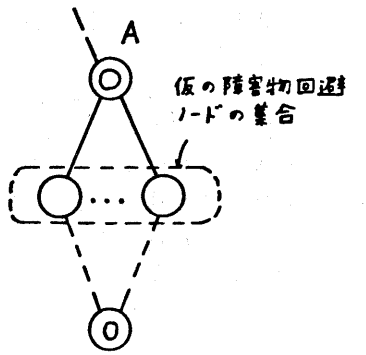
(4) (3) で得られた目標位置が最終位置と一致したとき終了すれば、満足解が得られ、A*アルゴリズムによると、最適解が得られる。



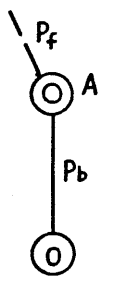
①解グラフからの拡張ノードの選択（Ｒは初期位置）



②仮のパスの決定（Ｏは目標位置）



パスPfとPbに対するノードＡの移動可能性を調べ、可能である場合、パスPbを解グラフに付加する。



③解グラフの拡張選択

④解パスの作成終了（満足解）

図４－２．解グラフの作成（Ｇは最終位置）

（コスト計算アルゴリズム）

これは、解グラフの終端ノードが作成されたとき、それに対して、これからかかるコストを推測する、heuristic 関数を与えるアルゴリズムである。

(1) 終端ノードから目標位置に対して、任意方向探索アルゴリズムを使い、障害物をさがす。

(2) 障害物がなければ、現在位置から目標位置までの距離l を、h の値とし、そうでなければ、まず解グラフのノードに、障害物の番号を記述しする。次に、探索レベルの黒ノード，ミックスノードを通過する距離の和l1と、それ以外の距離の和l2を計算し、h=l2+c*l1 をh の値とする。

(3) (2) の過程で、解グラフのあるルートにおいて、n 回続けて同じ障害物が表われた場合、さきにそれを回避するための位置を目標位置設定アルゴリズムで探索し、それを前の問題の目標位置，また後の問題の初期位置とする。ここで問題分割がおこる。

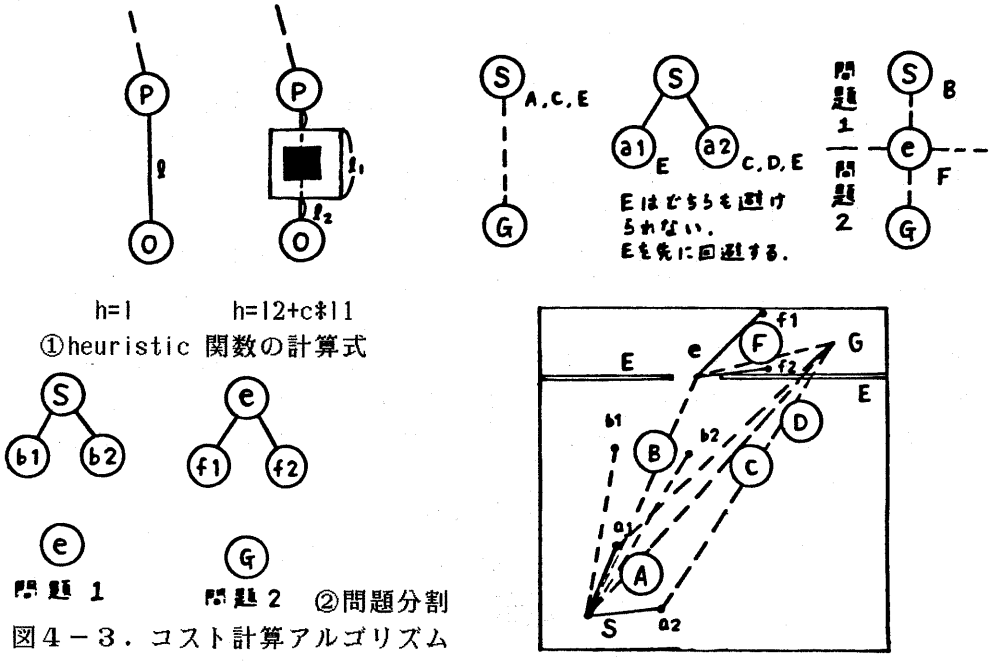


図４－３．コスト計算アルゴリズム

（任意方向探索アルゴリズム）

これは、ある辺Ejを辿り、その上にあるノード（キューブ）を連続的に調べるアルゴリズムである。

(a) 辺Ejの始点座標(x0,y0,z0) ,終点座標(x2,y2,z2)を与える。始点の属するノードを現在ノード、終点の属するノードを終了ノードとよぶ。

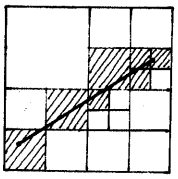
(b) 辺Ejの方向ベクトルをg=g(g0,g1,g2),頂点位置 i （８進数）を２進数で表わしたものをベクトルξ(i)=(i0,i1,i2)とする。g*ξ(i) を最大にする頂点 i を得る。

(c) 現在ノードでの頂点 i の座標(x1,y1,z1)を得る。辺Ejは、頂点 i に隣接する３つの面の領域を通過する。

(d) t0=(x1-x0)/g0 , t1=(y1-y0)/g1 , t2=(z1-z0)/g2 を比較し、最小の値を得るパラメータを調べる。各々、現在キューブでの頂点i を通るX,Y,Z 軸に垂直な面Px,Py,Pzの通過を意味する。

(e) (d) で得られた隣接ノードを探索する。それが、現在ノードと比べてレベルが同じか高い単色ノードであれば、そのノードを現在ノードとする。そうでないミックスノードであれば、その中のどのノードと辺Ejが最初に交わるかを、座標比較で再帰的に、探索レベルまで調べ、現在ノードを得る。

(f) 得られた現在ノードに対して、(c) から(e) の処理を再帰的に行なう。終了条件は、現在ノードが終了ノードと一致した場合である。



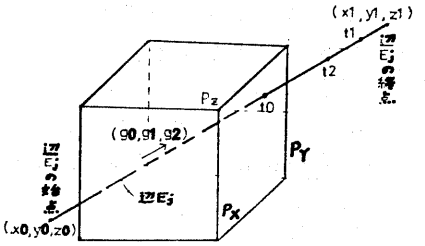


図４－４．任意方向探索アルゴリズム　　図４－５．通過面の決定法

（目標位置探索アルゴリズム）

このアルゴリズムは、ある障害物に対する回避点を決定するものである。

(1) 障害物Ａを取り囲む最小キューブを表わすノードをノードＮとする。その子ノードのうち白であるものの重心に仮の回避点を設定する。白ノードがなくなると、ミックスノードのうち、濃度（２章参照）の小さいものからノードＮとし、同様の処理を行なう。

(2) 仮の回避点から、現在位置，目標位置に対し、任意方向探索アルゴリズムを用いて障害物Ａと交差しないかを探索し、交差しなければ、障害物Ａに対する回避が確定したとして、仮の回避点を回避点と決定する。交差したら、(1) に戻って同様の処理を行なう。

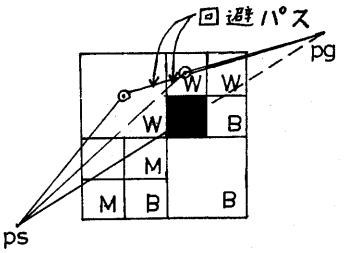


図４－６．回避点の設定

５．移動物体の動作決定

このアルゴリズムは、オクトツリーの白キューブの大きさ（レベル）は、障害物（黒キューブ）までの距離を表現するという事実に基ずき作成されている。したがって、そうなっていない領域に対して、予め、オクトツリーを正規化する必要がある。

この利用法は、

① 拡張ノードにおける２つのパス間の移動可能性を、４－３において調べる。

② ４．で得られた解ルートを構成するパスに対し、移動物体がスムーズに移動するための動作（回転、平行移動に対する自由度の変化）を決定する。

の２点である。

(1) [8] で提案した干渉チェツクアルゴリズムを使い、移動物体の干渉を調べると同時に、移動物体がある大きさより小さい白キューブと接触していれば、移動物体中のその置x0をチェックする。

(2) (1) で検出された位置x0から最短距離ｄを与える、障害物上の位置x1を得る。そして、ベクトルvk(=x0-x1)を算出し、それを回避ベクトルvkと呼ぶ。

(3) 現在の目標位置y1に対して、現在位置y0における進行ベクトルvsを計算する。マニピュレータでは、回避ベクトルは手先に与える。
(4) 進行ベクトルvsと、(2) で作成された回避ベクトルvkに従い、移動物体の保持する自由度に対して、その移動量を以下の方法で割りふる。
(5) 移動物体として本研究では、移動ロボットやマニピュレータを考えているが、後者の方がより一般的に記述できるので、ここではマニピュレータを考えることとする。
まず、マニピュレータの運動学方程式をDenavit-Hartenberg記法[3] で表わす。

$$T_E = \prod_i A_i E \quad , \quad T_F = \prod_j A_j + \Delta \qquad (1)$$

Ａｉ：各リンク間の関係を表わす４×４行列　Ｅ　：エンドエフェクタに関する変換
ＴＥ：手先位置におけるy0の位置と姿勢を表わす行列
ＴＦ：リンク内の位置x0の位置と姿勢を表わす行列
Δ　：リンクの大きさを考慮した補正項

ＴＥ，ＴＦ は、以下のような位置と姿勢を表わす行列である。

$$T = \begin{bmatrix} Nx & Ox & Ax & Px \\ Ny & Oy & Ay & Py \\ Nz & Oz & Az & Pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (2)$$

この要素を用いると、位置と姿勢を表現する９次元ベクトルx0,y0 は、
$(Ox, Oy, Oz, Ax, Ay, Az, Px, Py, Pz,)^T$　　と定義できる。　(3)
これを用いて、次のようなポテンシャル関数を考える。

$$Vs = \frac{1}{2}(y0-y1)^T W(y0-y1) \ , \quad Vk = 1/2(x0-x1)^2 \quad V = \alpha Vs + \beta Vk \qquad (4)$$

W：正定対称な重みずけ行列　α，β：２つのポテンシャル間の重みずけパラメータ

マニピュレータの各関節角からなるベクトルを$\Theta^T=(\theta 1, \theta 2, \theta 3, \theta 4, \theta 5, \theta 6)$ (5)
とすると

$$\frac{\partial Vs}{\partial \theta i} = \frac{\partial y_0}{\partial \theta i} W (y_0 - y_1) \ , \quad \frac{\partial Vk}{\partial \theta i} = -\frac{\partial x_0}{\partial \theta i} \frac{1}{(x_0 - x_1)^3} \ , \quad \frac{\partial V}{\partial \theta i} = \alpha \frac{\partial Vs}{\partial \theta i} + \beta \frac{\partial Vk}{\partial \theta i}$$

$$\theta_i^{(k+1)} = \theta_i^{(k)} - \frac{\partial V}{\partial \theta i} \qquad k:繰り返し回数 \qquad (6)$$

（４）式のポテンシャル関数の単峰性は、一般には言えないので、停留点問題は解決されていないが、
①大局情報の利用により、目標位置y1が適当な位置に逐次変化する。
②障害物と真に接近した、移動物体内の位置x0を、逐次決定しポテンシャルにより反力をかける。
ことにより、移動物体は、適宜必要な動作が行なえる。

6. さいごに

三次元空間をオクトツリーで表現することによって、各種情報を位置に依存した高速アルゴリズムで得ることができる。ここでは、それを利用して障害物を避け、目標に進む経路を、自律的に作成する実時間処理が可能な径路決定アルゴリズムを提案した。今後は、本研究のアルゴリズムによって得られた、三次元形状と現在の目標に対する、動作に関する情報を整理することにより、動作の組織的な学習を考えている。これが達成されると、より高速に仕事が行なわれる柔軟なシステムとなるであろう。

[参 考 文 献]

[1] Brooks,R.A.,Lozano-Perez,T.,A subdivision algorithm in configuration space for findpath with rotation ,in:Proceedings Eighth International Joint Conference on Artificial Intelligence , Karlsruhe,W.Germany , 1983

[2] Brooks,R.A.,Solving the findpath problem by good representation of free space,IEEE Trans.Systems Man Cybernet. 13 ,pp.190-197,1983.

[3] Denavit,J.,Hartenberg,R.S.,Journal of Applied Mechanics,pp.215-222,1955-6

[4] 比留川 博久，北村 新三，人見 信，”知識ベースロボットシステムＫａｒｌの基本構成”，第３回日本ロボット学会学術講演会，１２０６、１９８５．

[5] Jackins C.L.,Tanimoto S.L.,"Octrees and their use in represerting three dimensional objects",Computer Graphics & Image Processing,Vol.14,No.3,pp. 249-270,Nov.,1980.

[6] Lozano-Perez, T.,Spatial planning: a configuration space approach, IEEE Trans.,Comput. 32, PP.108-120,1983.

[7] Nillson N.J., Problem-solving methods in artificial intelligence. New York : McGraw-Hill,1971.

[8] 登尾啓史，野田哲男，有本 卓，オクトツリーを用いた径路決定のための各種アルゴリズムの提案，情報処理学会研資，知識工学と人工知能研究会，４２－９，１９８５．

[9] 登尾啓史，野田哲男，有本 卓，”領域情報を利用した経路決定アルゴリズム”，第３回日本ロボット学会学術講演会，１４１３，１９８５．