多指ハンドによる軸回転プリミティブ物体操作

小 俣 透*

Pivoting Reorientation of an Object with a Multifingered Hand

Toru Omata*

A multifingered hand can reorient an object in a hand. This paper discusses one of primitive ways to reorient a prism object. Two fingers (corresponding to the human thumb and middle fingers) grasp the top and bottom faces of the prism object. Their fingertips form a rotational axis. The third finger (corresponding to the human index finger) exerts a force to the side faces to rotate the object about the axis. Since the workspace of the third finger is limited, the side faces reachable by it are also limited depending on the position of the axis formed by the first two fingers. This paper shows an algorithm for computing the possible region for the axis so that the third finger can rotate the object from start to goal angles. The algorithm also determines the side faces to be pushed by the third finger. It is applicable to any shaped prism objects.

Key Words: Multifingered hand, Regrasp

1. は じ め に

多指ハンドは、指の移動を伴う持ち替え操作により、物体の位置姿勢を手の中で大きく変えることができる。多指ハンドによる持ち替え操作では、例えば Fig.1 のような、各指の役割が定まった特定の持ち替え方法が考えられる。

Fig.1 では、操作される物体の形状は平板状のものを想定している。我々のまわりの物体は平板状もしくはそれにみなせるものが多いからである。(1) は 3 本以上の指で物体の側面を把持し、指を移動させながら回転させるものである [1]。(2) では、指 1、指 2 の 2 本の指で物体をはさみ、それらにより回転軸を構成する。指 3 でその回りにモーメントを与え物体を回転させる。指 3 により並進力を与え、物体を並進させる操作も考えられる。

以下、このような操作をプリミティブ操作と呼ぶ。環境との接触を利用した操作 [2] も考えられるが、これらのプリミティブ操作は空中での操作を前提としている。(1) のプリミティブ操作については、文献 [1] で論じた。Okada [3]、Fearing [4] は棒状物体を 3 本指で回転させる Twirling 操作について議論している。これは (1) のプリミティブの一種と考えることができる。本論文では、(2) のプリミティブ操作（以下、軸回転プリミティブと呼ぶ）を論じる。

持ち替え操作を行うためには、どの指を離しどこに移動すればよいかその手順を適切に決めなければならない。多指ハンドは多くの関節自由度を持つので、三次元的持ち替え操作の指動作を、直接関節自由度空間で計画することは非常に難しいと考えられる。そこで、このような比較的簡単なプリミティブ操作を用意し、それらの組み合わせとして持ち替え操作を実現することを筆者は考えている。まずは、それぞれのプリミティブ操作のための指動作計画法を確立することが課題である。

軸回転プリミティブについては、指 1、2 で構成する回転軸が物体のどこにあるべきか、また指 3 はどの側面を押すべきかという問題について考える。指関節の可動範囲には限界があり、指 3 指先が到達できる物体側面の範囲も限定されている。回転軸の位置を適切に決めなければ、指 3 が物体側面に届かなくな

Fig.1 Regrasp primitives

原稿受付 1995 年 5 月 11 日

*東京工業大学

*Tokyo Institute of Technology

り、物体を回転できなくなることもあり得る。指の到達を考慮に入れた研究は一部を除いて少ない [5]。それを調べるために、本論文では障害物回避問題でよく用いられるコンフィギュレーション障害物の概念を用いる。力学的な条件のみならず、幾何学的な条件を考慮することが本研究の特徴である。

本論文の構成は次のとおりである。第2章では、問題設定について述べる。第3章では、まず指の作業領域を定義しそれを求める方法を示す。次に、物体の姿勢を変化させつつ、その姿勢で物体を回転できる回転軸の領域を求める。第4章では、指定した回転角を回すことができる回転軸領域と、指3が押すべき物体辺を探索する方法を示す。

2. 問題設定

まず以下のような仮定を設ける。

- (1) 物体は断面がL多角形の角柱状のものとする。断面形状は必ずしも凸である必要はない。
- (2) 指1, 2はFig.1(2)のようにそれぞれ側面 S_t, S_b をはさむように掴む。指1, 2は同一の構造の指として、その姿勢は側面 S_t, S_b の中央を鏡面として対称とする。指3は側面 $S_1, \dots, S_L$ を押すものとする。
- (3) 指、物体の運動速度、加速度は十分小さいと考えられるので、動力学的影響は無視できるとする。
- (4) 指1, 2の指先接触はソフト接触とする。すなわち、それぞれ面 S_t, S_b に垂直な方向の力、面内の摩擦力、および面に垂直方向のモーメントを伝えることができるとする。
- (5) 指3の指先接触は摩擦ありの点接触とする。また指の太さは無視できるものとする。

仮定1, 2のもとで、軸回転プリミティブを、Fig.2に示すように指1, 2がはさむ側面 S_t, S_b に平行な面に投影された平面問題として取り扱うことができる。この投影面をPとし、側面 S_t, S_b の面Pへの投影をSとする。指関節の運動も面Pへの投影運動のみを考える。また、指3の発生する力は面Pに平行な成分のみとする。以下、一般性を失うことなく、回転を時計方向と仮定する。

指1, 2指先が構成する回転軸をAとする。ここで考える問題は、面S上で物体を初期姿勢角 β_S から目標姿勢角 β_G まで回転させることができる指1, 2指先（すなわち回転軸A）の領域を求めること、およびそのときの指3が押すべき物体側面 S_i を求めることである。

Fig. 2 Projection of regrasp primitive (2) onto plane P

3. 回転軸領域の計算

3.1 作業領域の計算

物体を回転できる指1, 2指先の領域を求めるために、まず指1, 2指先を原点として、次の条件を満たす指3先端の領域を求める。

- (1) 到達条件：指3先端が到達すること
- (2) 非干渉条件：指3のリンクが物体と干渉しないこと
- (3) 回転条件：回転軸Aにモーメントが加わり物体が回転すること

到達条件を満たす領域を到達領域、さらに非干渉条件を満たす領域を非干渉領域、さらに回転条件を満たす領域を作業領域と呼び、それぞれを W_r, W_n, W とする。 $W \subseteq W_n \subseteq W_r$ の関係にある。

作業領域原点が回転軸Aを構成する指1, 2の指先位置であることに注意しよう。作業領域 W を計算した後、指3が物体のある辺 e を押すことにより物体を回転できる指1, 2指先の領域は、次の式により求めることができる。

$$R_s(e, W) = S \cap R(e, W) \quad (1)$$

ただし、 $R(e, W)$ は作業領域 W が辺 e と交わる作業領域原点の領域とする。3.2節では、物体の姿勢を変化させてこの領域を求める。

回転目標、すなわち、回転する物体と目標姿勢角が与えられる前に計算できる部分は計算しておけば、回転目標が与えられてからの処理を少なくすることができる。そこで、この節で述べる作業領域 W の計算は、できるだけ回転目標が与えられる前に済ませておくようにする。一方、3.2節は回転目標が与えられた後の処理である。

注意：作業領域 W は、以下で示すように実際には指3が押す辺の傾きにより形状が変化する。 α を辺の法線角とすると、 $W(\alpha)$ のようにそれを明示することがある。

以下、条件(1)(2)(3)を順に調べよう。

(1) 到達条件

指1, 2指先から指3の先端までには平面P内に N 個の関節 $\theta_i, i = 1, \dots, N$ があるとする。到達領域 W_r を次のような数値的な方法で求める。数値的な方法を用いる理由は、(2)の非干渉条件を考慮するように拡張しやすいからである。

到達領域を求める数値計算法：

ステップ1) Fig.3のように、 W_r を覆うことが保証されるなるべく小さい長方形 W_{rs} (surrounding box) を見いだす。 W_{rs} を所定の精度のメッシュでセルに細分化する。

ステップ2) 各セルの代表点について逆キネマティクスを解き、関節角 θ_i を求める。すべての関節角が可動角内にあれば、そのセルを W_r に加える。ただし、特異点となるようなセルは除く。この操作をすべてのセルについて行う。

(2) 非干渉条件

指3のリンクは物体と干渉してはならない。最も起こる可能性の高い干渉は先端リンクとそれが押す辺との干渉である。本論文ではそれについてのみ考慮する。 α を辺の外向法線の x 軸

Fig. 3 Numerical calculation of reachable region with the origin being the fingertips of finger 1 and 2 forming the axis A

Fig. 4 Distal link collision avoidance

からの角度（反時計方向を正とする）とし、同様に先端リンクの姿勢角を α_L とする。先端リンクがその辺と干渉しないための条件は、

$$|\alpha - \alpha_L| \geq \pi/2 \quad (2)$$

である (Fig. 4 参照)。

この条件は辺の法線角 α （すなわち傾き）に依存する。回転目標が与えられる前に、非干渉領域の計算を済ませるために、ここでは法線角度を離散化し、

$$\alpha_j = (2\pi/M_\alpha)j, j = 0, \dots, M_\alpha - 1 \quad (3)$$

の角度ごとに非干渉領域 $W_n(\alpha_j)$ を計算して、 M_α 個のデータとして蓄えておく。

到達領域 W_r から非干渉領域 $W_n(\alpha_j)$ を求める方法を次に述べよう。まず、前述の方法によりセル集合で表された到達領域 W_r を計算する。到達条件は辺の傾きには無関係であるから、到達領域 W_r は最初に 1 回計算するだけで済ませられることに注意しよう。このとき、各セルでの先端リンクの姿勢角 α_L を記憶しておく。式 (3) の各法線角 α_j について、その中で、式 (2) を満たすセルのみを選択することにより、非干渉領域 $W_n(\alpha_j)$ が計算できる。

(3) 回転条件

まず、重力の影響を無視した場合について考える。指 3 は物体の辺を摩擦円錐内の力 $\mathbf{f}_3$ で押すものとする。回転軸 A から指 3 接触点までのベクトルを $\mathbf{p}_3$ とする。また、 l を回転軸 A から力 $\mathbf{f}_3$ の作用線への垂線の長さとする。このとき、回転軸 A には $F_t = |\mathbf{f}_3|$ の大きさの力と $M_n = |\mathbf{p}_3 \times \mathbf{f}_3| = l|\mathbf{f}_3|$ の大きさのモーメントがかかる。

この力とモーメントがソフト接触の摩擦限界 [6] を越えると、回転軸 A にすべりが生じる。発生するすべりは、一般に並進す

Fig. 5 Finger 3 region for rotation

べりと回転すべりが混合したものである。しかし、並進力に比べてモーメントが十分に大きいときには、実質的に回転運動が発生するとみなすことができる。そこで、ある定数 l_0 が存在して

$$\frac{M_n}{F_t} > l_0 = \text{constant} \quad (4)$$

のときに純粋な回転が生じるものと仮定する。

これは、指 3 の力の作用線が回転軸 A を構成する指 1, 2 指先から l_0 以上離れなければならないことを意味している。指 3 が作用できる力は摩擦円錐の内側である。ある傾きの辺について、式 (4) を満たす時計方向の回転モーメント（回転方向は時計方向と仮定している）を発生できる指 3 指先の領域は、Fig. 5 のように与えられる。すなわち、(1) 摩擦円錐の右側側辺に平行で回転軸 A から距離 l_0 離れた直線 L_r の外側、または、(2) 左側側辺に平行で回転軸 A から距離 l_0 離れた直線 L_l の外側、または、(3) それらの間の円弧 C の外側である。なお、Fig. 5 の破線で示す辺は、辺の傾きのみを表している。その位置に辺があるという意味ではない。

この条件も辺の傾きに依存するが、すでに非干渉領域を計算する段階で辺の法線角を式 (3) のように離散化して、それぞれに角度ごとに非干渉領域 $W_n(\alpha_j)$ を計算している。作業領域も同様に、その角度ごとに $W(\alpha_j)$ を計算する。 $W(\alpha_j)$ は、非干渉領域 $W_n(\alpha_j)$ の中で回転条件を満たすセルのみを選択することにより計算することができる。ただし、作業領域は物体と指との摩擦係数にも依存するので、それが判明した後計算される。

セルの集合で表された作業領域 W を所定の精度で多角形に近似し、これ以降の図形処理を施す。これを行う画像処理の古典的方法が知られている [7]。ただし、多角形近似せず、セル集合のままで図形処理を施すことも可能である。

なお、重力の影響を考えると、回転軸に加わる力とモーメントの大きさはそれぞれ

$$F_t = |m\mathbf{g} + \mathbf{f}_3| \quad (5)$$

$$M_n = |\mathbf{p}_g \times m\mathbf{g} + \mathbf{p}_3 \times \mathbf{f}_3| \quad (6)$$

となる。この場合も、指 3 の指先位置が上記の領域にあれば、指先力 $|\mathbf{f}_3|$ を大きくすることにより、式 (4) を満たすことができる。

以上まとめると、作業領域を計算する手順は次のようになる。

(1) 到達領域 W_r を計算する。

Fig. 6 Workspaces

- (2) $\alpha_j = (2\pi/M_\alpha)j$, $j = 0, \dots, M_\alpha - 1$ について, 非干渉領域 $W_n(\alpha_j)$ を計算する.
- (3) 指と物体との摩擦係数が与えられたならば $\alpha_j = (2\pi/M_\alpha)j$, $j = 0, \dots, M_\alpha - 1$ について, 作業領域 $W(\alpha_j)$ を計算する.

例 1) 指 3 として, Fig. 6(1) に示すような 2 リンクを考える. リンクの長さは 30 [mm] とする. 指 1, 2 の関節は固定として, 指 1, 2 先端から指 3 基部 B までは固定長 40 [mm] 離れているものとする. 摩擦角を 45° とし, $l_0 = 10$ [mm] とする. 指 3 先端の到達領域 W_r を Fig. 6(1) に示す. 同図 (2)(3)(4) に様々な傾きの辺に対する作業領域 W を示す.

3.2 物体の姿勢を変化させたときの回転軸領域

基準座標系に対する物体の姿勢角を β とする. 回転する物体と目標姿勢角 β_G が与えられたならば, 物体を初期姿勢角 β_S から目標姿勢角 β_G まで回転させて, 各姿勢角について式 (1) で与えられる指 1, 2 指先 (回転軸 A) の領域 $R_S(e, W) = S \cap R(e, W)$ を求める.

まず, 姿勢角 β を十分細かく M 等分散化する.

$$\beta_k = \beta_S + ((\beta_G - \beta_S)/M)k, \quad k = 0, \dots, M \quad (7)$$

$\alpha_{0,i}$, $\alpha_{i,k}$ をそれぞれ, 姿勢角 0, 姿勢角 β_k における辺 e_i の外向法線角とする. この両者は

$$\alpha_{i,k} = \alpha_{0,i} + \beta_k, \quad k = 0, \dots, M \quad (8)$$

の関係にある.

すべての辺 $i = 1, \dots, L$ について, 姿勢角 β_k , $k = 0, \dots, M$ ごとに領域 $R_S(e_i(\alpha_{i,k}), W(\alpha_{i,k}))$ を求める. 作業領域 $W(\alpha_{i,k})$

Fig. 7 The region for A where W intersects edge e

は法線角度により形状が変化することに注意しよう. ここで, 角度 $\alpha_{i,k}$ を式 (3) の角度のなかから最も近い角度 α_j で近似する. 3.1 節で計算した作業領域 $W(\alpha_j)$ を用いて, 領域 $R(e_i(\alpha_j), W(\alpha_j))$ を計算する.

領域 $R(e(\alpha), W(\alpha))$ は, $W(\alpha)$ を 180° 回転させ辺 $e(\alpha)$ の方向にその長さだけ伸ばしたものである (Fig. 7 参照). 多角形をある方向に伸ばさせた図形を求めるアルゴリズムを付録 A に示す. 辺 $e(\alpha)$ を障害物, 作業領域 $W(\alpha)$ を移動ロボットと考えれば, $R(e(\alpha), W(\alpha))$ は障害物回避問題でよく用いられているコンフィギュレーション障害物と同様なものである. また, 領域 $R(e_i(\alpha_j), W(\alpha_j))$ の効率的な計算法を付録 B に示す.

なお, 領域 $R_S(e(\alpha), W(\alpha))$, $R(e(\alpha), W(\alpha))$ は物体座標系から見たものを求めることとする. 物体上のどこに回転軸 A があればよいかを求めることが目的であるからである.

注意: 物体を回転させるためには本来回転軸の位置を定めなければならない. 回転軸の位置により回転後の物体の位置は当然異なる. 今, 回転軸の領域を求めているわけであるから, これは一見矛盾しているように思われる. しかし, 領域 $R_S(e_i(\alpha_{i,k}), W(\alpha_{i,k}))$ は回転とともに移動した物体座標系から見たものであるから, 物体の位置とは無関係である. したがって, 今の場合回転軸の位置はどこでもよく, 便宜的に物体座標系原点に定めればよい. なお, 物体を回転させるかわりに, 作業領域 W の方を逆方向に回転させ, 静止した物体座標系からこの領域を求めてもよい.

例 2) 到達領域等の条件は例 1 と同じとする. 物体を Fig. 8 下側に示すような 6 角形とする. 初期姿勢角 $\beta_S = 0$ 度, 目標姿勢角を $\beta_G = 360^\circ$ とする. 姿勢角を 15° ずつに分割する. すなわち, $M = 24$ とする. 縦軸を姿勢角 β_k として, 物体上に矢印で示した辺に対する $R_S(e_i(\alpha_{i,k}), W(\alpha_{i,k}))$ を示す.

4. 回転軸領域の探索

物体の姿勢角が β_k のとき, $R_S(e_i(\alpha_{i,k}), W(\alpha_{i,k}))$ が空でない辺を用いて, 指 3 は物体を回転させることができる. 和集合

$$U(\beta_k) = \bigcup_{i=1, \dots, L} R_S(e_i(\alpha_{i,k}), W(\alpha_{i,k})), \quad k = 0, \dots, M \quad (9)$$

は, 姿勢角 β_k で物体を回転させることができる回転軸 A の領域である. 積集合

$$V(\beta_S, \beta_G) = \bigcap_{k=0, \dots, M} U(\beta_k) \quad (10)$$

Fig. 8 Rotation axis regions from start to goal angles for some edges

Table 1 The number of set operations

演算	回数
式 (9) 和集合	$(L-1)(M+1)$
式 (10) 積集合	M
式 (11) 積集合	$L(M+1)$

は物体を初期姿勢角から目標回転角まで回転できる回転軸 A の領域である。これが空であれば、回転の途中で回転軸の位置を変更しなければならない。姿勢角 β_k で指 3 が押す辺は、 $i = 1, \dots, L$ のなかで

$$V(\beta_S, \beta_G) \cap R_S(e_i(\alpha_{i,k}), W(\alpha_{i,k})) \neq \phi, k = 0, \dots, M \quad (11)$$

である辺である。

この計算により、物体を回転できる回転軸 A の領域は、もしそれが存在するならば必ず見つかり、またそれが実際に存在しないならば存在しないことが答として求まる（ただし、姿勢角を離散化した影響は除く）。その意味でこの方法は完全と言える。以下この方法を全探索法と呼ぶ。Table 1 に全探索法による和集合演算と積集合演算の回数を示す。

全探索法は、和集合を $(L-1)(M+1) \approx LM$ 回、積集合を $L(M+1) + M \approx LM$ 回求めなければならないので、計算時間がかかる。この点を改良した方法を以下に示す。

式 (9) で L 個の図形 $R_S(e_i(\alpha_{i,k}), W(\alpha_{i,k}))$ の和集合計算を行っているが、この中には面積の小さいものも含まれている。面積の大きいものを優先して選択して、回転軸領域を探索する方が効率的と考えられる。

Table 2 The number of set operations in the area-first method

演算	回数			
	$n = 1$	$n = 2$	$n = 3$	$\dots$
式 (12) 和集合	0	$M+1$	$M+1$	$\dots$
式 (13) 積集合	M	M	M	$\dots$
式 (14) 積集合 (*)	0	0	0	$\dots$
式 (14) 積集合 (**)	0	$2(M+1)$	$3(M+1)$	$\dots$

姿勢角 β_k で $R_S(e_i(\alpha_{i,k}), W(\alpha_{i,k}))$ の面積の最大となる辺を e_{s_1} 、2 番目に大きい辺を e_{s_2} のように表す。そして、面積の n 番目まで大きい領域の和集合を

$$U(n, \beta_k) = \bigcup_{i=1, \dots, n} R_S(e_{s_i}(\alpha_{s_i,k}), W(\alpha_{s_i,k})), k = 0, \dots, M \quad (12)$$

とする。姿勢角を変化させたときの積集合を

$$V(n, \beta_S, \beta_G) = \bigcap_{k=0, \dots, M} U(n, \beta_k) \quad (13)$$

とする。まず $n = 1$ として式 (13) の領域を計算する。それが空でなければ回転軸領域が見つかったことになる。これが空であれば、順次 $n = 2, 3, \dots$ と増加させて領域 $V(n, \beta_S, \beta_G)$ が空でなくなるまで調べていく。姿勢角 β_k で指 3 が押す辺は、 $i = 1, \dots, n$ のなかで

$$V(n, \beta_S, \beta_G) \cap R_S(e_{s_i}(\alpha_{s_i,k}), W(\alpha_{s_i,k})) \neq \phi, k = 0, \dots, M \quad (14)$$

である辺である。この方法を面積優先法と呼ぶ。Table 2 に面積優先法での集合演算回数を示す。

注意：式 (14) 積集合の (*) 欄は、式 (13) 積集合が空のとき、(**) 欄は空でないとき（すなわち回転軸領域が見つかったとき）である。また、式 (12) 和集合が $n \geq 2$ で $M+1$ 回と一定であるのは、それが、

$$U(n, \beta_k) = U(n-1, \beta_k) \cup R_S(e_{s_n}(\alpha_{s_n,k}), W(\alpha_{s_n,k})), n \geq 2 \quad (15)$$

のように計算できるからである。

$n = 1$ で回転軸領域が見つかったとすると、 M 回の積集合演算だけで済む。 $n = n_0 \geq 2$ で回転軸領域が見つかったとすると、和集合を $(n_0 - 1)(M+1) \approx n_0 M$ 回、積集合を $n_0 M + n_0(M+1) \approx 2n_0 M$ 回計算しなければならないが、 $n_0 < L/2$ であれば全探索法よりも効率的と言える。

例 3) 例 2 の問題に対して、面積優先法により物体を回転できる領域を見つけた結果を Fig. 9(1) に示す。ここで、 $n = 1$ により見つけている。同図 (2) に、各姿勢角における $U(1, \beta_k) = R_S(e_{s_1}(\alpha_{s_1,k}), W(\alpha_{s_1,k}))$ を示す。

5. おわりに

2 本の指で物体をはさみ、第 3 の指で物体を回転させるプリミティブ操作について、指動作を生成する方法を提案した。指が物体のどこに届くかという幾何学的条件を考慮することが重

Fig. 9 Intersection (left) of rotation axis regions from start to goal angles (right)

要であることを示した。3本の指があればこのプリミティブ操作により物体を回転できるので、このプリミティブ操作の応用は広いと考えられる。さらに他のプリミティブ操作と組み合わせることにより、三次元的持ち替え操作を実現することが今後の課題である。実際には、回転すべりに並進すべりが伴い、回転軸がずれていく可能性がある。回転軸の位置を監視するには、並進と回転のすべりが検出できるセンサが有効と思われる [8]。そのような点を考慮して実際のハンドに適用することも今後の課題である。なお、本論文の例題では Euslisp を使用してプログラミングした。

参 考 文 献

- [1] 小俣, 永田, 岩槻, 柿倉: “持ち替えによる多指ハンドの物体回転操作の計画”, 日本ロボット学会誌, vol.12, no.5, pp.751-758, 1994.
- [2] 相山, 稲葉, 井上: “ピボット操作を用いた床面上での多面体物体の操り”, 日本ロボット学会第9回学術講演会予稿集, pp.759-760, 1991.
- [3] T. Okada: “Object-Handling System for Manual Industry,” IEEE Trans. on Systems, Man, and Cybernetics, vol.SMC-9, no.2, pp.79-89, 1979.
- [4] R.S. Fearing: “Implementing a force strategy for object re-orientation,” IEEE Conf. on Robotics and Automation, pp.96-102, 1986.
- [5] A.A. Cole, P. Hsu and S. Sastry: “Dynamic Regrasping by Coordinated Control of Sliding for a Multifingered Hand,” IEEE Int. Conf. on Robotics and Automation, pp.781-786, 1989.
- [6] R.D. Howe, I. Kao and M. R. Cutkosky: “The Sliding of Robot Fingers Under Combined Torsion and Shear Loading,” IEEE Int. Conf. on Robotics and Automation, pp.103-105, 1988.
- [7] SPIDER-ii Users' Manual: 情報処理振興事業協会, 1986.
- [8] Y. Yamada and M.R. Cutkosky: “Tactile Sensor with 3-Axis Force

Fig. 10 Extension of a polygon along a vector

and Vibration Sensing Functions and Its Application to Detect Rotational Slip,” IEEE Int. Conf. on Robotics and Automation, pp.3550-3557, 1994.

A. 多角形の伸長図形

多角形をベクトル s の方向に $|s|$ だけ伸長した図形を求めるアルゴリズムについて考察する。多角形が凸の場合は、Fig. 10 (1) のように簡単である。ベクトル s の方向に x 軸を設定し、それから反時計方向 90° 回転した方向に y 軸を設定する。 y 座標の最小の頂点を A とする。頂点 A から時計方向に辺をたどる。頂点の y 座標が増加する辺のつながりを + セグメント、減少する辺のつながりを - セグメントと定義する。凸多角形の場合、辺は + セグメント、- セグメントに分割される。伸長図形は - セグメントの頂点を s 平行移動することにより得られる。

物体が凹の場合は、複数の - セグメントが存在する。例えば、Fig. 10 (2) では、辺は +, -, +, -, +, - のセグメント、 $P_1, M_1, P_2, M_2, P_3, M_3$ に分割される。- セグメントの後に + セグメントが続く場合、- セグメントをベクトル s の方向に平行移動していくと、+ セグメントと交差する。よって、- セグメントの頂点を s 平行移動した点は必ずしも伸長図形の頂点にはならない。この例では - セグメント M_1 を平行移動させていくと、一部分はまず + セグメント P_2 と交差し、次に残りの部分が P_3 と交差する。したがって、どの + セグメントと交差するかを調べる必要がある。

以上より、アルゴリズムの概略は以下のようになる。

- (1) 多角形の辺を + セグメントと - セグメントに分割する。それらを $P_1, M_1, P_2, M_2, \dots$ とする。
- (2) - セグメント M_i をベクトル s の方向に平行移動したと

Fig. 11 -segment intersecting +segments

きに、最初に交差する+セグメントにより、-セグメント M_i を分割する。それらを $M_{i1}, M_{i2}, \dots$ とし、+セグメントの対応する部分を $P_{i1}, P_{i2}, \dots$ とする。例えば Fig. 11 では、-セグメント M_i は、 P_{i1} と最初に交差する M_{i1} と、 P_{i2} と最初に交差する M_{i2} に分割される。

- (3) すべての M_{ij} について以下の処理を行う。

M_{ij} の頂点、 P_{ij} の頂点を y 座標の小さい順にマージする。それらを $q_k, k = 1, 2, \dots$ とする。頂点 q_k を通りベクトル s に平行な直線の、対する + または - セグメントまでの距離 r_k を求める。

- (4) $r_k > |s|$ である頂点は伸長図形の頂点になる。

$$r_k < |s|, r_{k+1} > |s| \text{ または } r_k > |s|, r_{k+1} < |s| \quad (16)$$

であれば、頂点 q_k, q_{k+1} の間で - セグメントと + セグ

メントは交差し、その交点が伸長図形の頂点になる。

B. 領域 $R(e_i(\alpha_J), W(\alpha_J))$ を求める効率的な計算法

辺 e_i の法線角と辺 $e_{i'}$ の法線角は初期姿勢で $|\alpha_{0,i} - \alpha_{0,i'}|$ ずれているだけである。式 (8) より、物体の姿勢角 β_k における辺 i の法線角 $\alpha_{i,k}$ と、姿勢角 $\beta_{k'}$ における辺 i' の法線角 $\alpha_{i',k'}$ は、

$$\beta_{k'} = \beta_k + \alpha_{0,i} - \alpha_{0,i'} \quad (17)$$

であれば、同一の角度に近似される。その角度を α_J とする。すなわち、 $\alpha_{i,k} \approx \alpha_J$ 、かつ、 $\alpha_{i',k'} \approx \alpha_J$ とする。

3.2 節では、領域 $R(e_i(\alpha_J), W(\alpha_J))$ と $R(e_{i'}(\alpha_J), W(\alpha_J))$ を求めるわけであるが、それらの形状は、辺 e_i と $e_{i'}$ の長さにより異なるだけである。ところが、付録 A に示す領域 $R(e_i(\alpha_J), W(\alpha_J))$ を求める計算の大部分のステップ (1)(2)(3) は、辺の長さとは無関係に処理することができる。よってその部分は、各角度 $\alpha_j, j = 1, \dots, M_\alpha$ につき 1 回計算するだけで済ませることができる。すなわち、ステップ (3) までの処理結果を記憶しておき、辺の長さが与えられたら、ステップ (4) を計算する。

小俣 透 (Toru Omata)

1959 年 2 月 10 日生。1981 年東京工業大学制御工学科卒業、1983 年同修士課程修了、1986 年同博士課程修了、同年通産省工業技術院電子技術総合研究所入所。1994 年 4 月より東京工業大学大学院総合理工学研究科精密機械システム専攻助教授、工学博士。知能ロボット、多指ハンドの研究に従事。計測自動制御学会の会員。(日本ロボット学会正会員)