

## デジタル直線の性質に基づく高速多角形近似を用いた 移動物体の認識†

向井信彦\*\* 有本卓\*\*\*

動画像データから移動物体を識別し、その位置ベクトルおよび速度方向を実時間で計測するシステムを開発した。移動物体の識別には、処理を高速に行うために新しく考案した多角形近似アルゴリズムを用いた。従来の多角形近似アルゴリズムは、あらかじめ物体の輪郭を形成する画素列を抽出する前処理と、近似を行う本処理の二段階構成になっているが、考案した方式は物体の輪郭を追跡しながら、同時に近似も行うという特徴を有する。一般に局所的なデータを基にした逐次処理を行う場合は、データ全体を大局的に処理する方法に比べて、近似精度の点で劣化が生じる。そこで、二次元直線を量子化して得られるデジタル直線を定義し、その性質を詳しく調べ、この性質を用いて簡単に精度よく近似直線を生成するアルゴリズムを導出した。さらに、ユークリッド距離を用いた通常の近似誤差計算を加減算および比較演算のみで行い、処理の高速化を図った。最後に二つの移動物体を認識・識別し、一方の物体の姿勢を制御することにより、他方の物体を追跡する実験を行い、システムの有効性を確かめた。

### 1. ま え が き

近年、工業分野では光センサや位置センサあるいはまた触覚センサ等を用いて自律的に仕事を行うロボットの開発が試みられてきた<sup>1)~3)</sup>。ところが、このようなセンサ類では取り得る情報が非常に少なく、人間が視覚能力として持つ“目”にはまだまだ及ばない。抽出できる情報量の点から見れば、ロボットの視覚には画像を用いる方が良いのであるが、画像のデータ量は非常に多く、すべての情報を逐一処理していたのではとてもリアルタイムで行うことはできない。

そこで、移動物体の実時間認識をする場合、認識対象によってはエッジ抽出のための空間微分<sup>4)</sup>や細線化<sup>5)</sup>等の時間のかかる前処理をやめ、物体の輪郭を直接追跡<sup>6)~8)</sup>した方が良いと思われる。また、認識の基本となる多角形近似についても、輪郭を抽出した後近似を行う二段階処理ではなく、輪郭を追跡しながら同時に近似も行えれば、より速く処理することができる。

本論文では、二次元平面に限定されたアナログ直線を量子化して得られるデジタル直線を定義し、この性質を用いることにより物体の輪郭を一回り逐次追跡

するだけで、簡単にしかも高速に精度よく近似するアルゴリズムを導出した。幾つかの静止画像を例に取り、近似精度と処理時間を調べ従来の近似方法<sup>9)~13)</sup>の中で代表的と思われる方法<sup>9),10)</sup>について比較検討したところ、良い結果が得られたので、この方法を応用して移動物体の認識を行い、本論文で提案した方法が動画像のリアルタイム処理にも有効であることを示す。

2章ではデジタル直線を定義し、その性質について詳しく調べる。3章ではデジタル直線の性質を利用した高速多角形近似アルゴリズムを説明し、従来の近似方法との比較結果を示す。最後に、4章では認識対象を限定して、その範疇にある二つの移動物体を認識・識別し、一方の物体の姿勢を制御して他方の物体を追跡させた実験結果について報告する。

### 2. デジタル直線の性質

#### 2.1 デジタル直線の定義

二次元平面に限定された幅のない数学的なアナログ直線を量子化して、格子点列であるデジタル直線をつくるには幾つかの方法が存在する。本論文では、この格子点列を表現する方法としてチェーン符号を用いる。チェーン符号<sup>14)~17)</sup>とは画素の連なりを符号化したもので、その基本となる方向および符号は図1に示すように8の剰余系により定められ、このチェーン符号化列を得る方法は次のように規定される。

まず、図2に示すように直線の始点を原点とした直交座標系で第一象限のうちX軸からの角度が45度以下の領域について考える。X座標は整数値のみを取

† Recognition of Moving Objects by Using a Fast Polygonal Approximation Based on Some Properties of Digital Line by NOBUHIKO MUKAI (Systems Development Department, Information Systems and Electronics Development Laboratory, Mitsubishi Electric Corporation) and SUGURU ARIMOTO (Department of Mechanical Engineering, Faculty of Engineering Science, Osaka University).

\*\* 三菱電機(株)情報電子研究所情報システム開発部

\*\*\* 大阪大学基礎工学部機械工学科

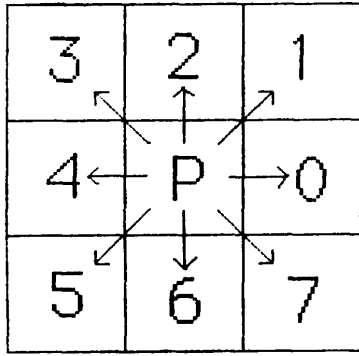


図 1 チェイン符号の方向および符号  
Fig. 1 Directions and codes of chain code.

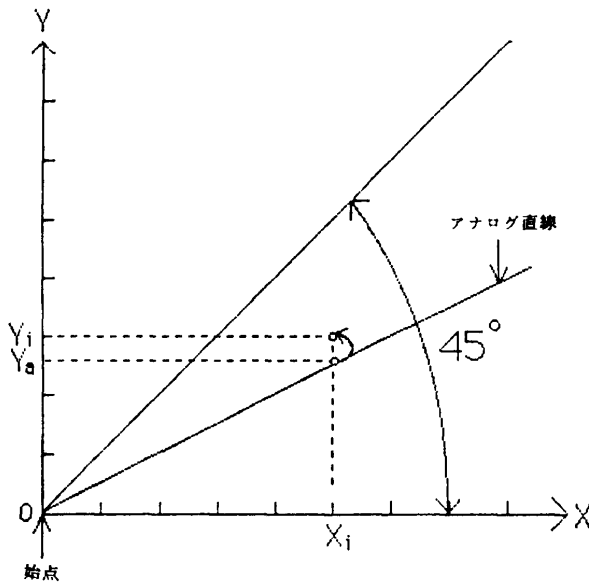


図 2 量子化の方法  
Fig. 2 A scheme of quantization.

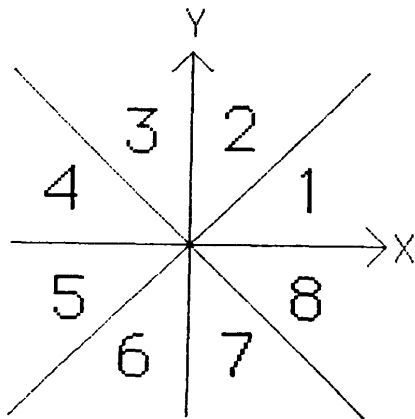


図 3 量子化の分類  
Fig. 3 A classification of quantization.

り得るように量子化し、この量子化された  $X$  座標  $X_i$  に対応するアナログ直線上の  $Y$  座標  $Y_a$  の絶対値を四捨五入し、 $Y_a$  と同符号を与えることにより量子化された  $Y$  座標  $Y_i$  を定める。他の象限および領域における直線の量子化についても、図形の対称性を考慮して同様に定める。つまり、

- ① 図 3 で 1, 4, 5, 8 の領域における直線は  $X_i$  に対応するアナログ直線上の  $Y$  座標  $Y_a$  の絶対値を四捨五入し、 $Y_a$  と同符号を与えることにより  $Y_i$  を定める。
- ② 図 3 で 2, 3, 6, 7 の領域における直線は  $Y_i$  に対応するアナログ直線上の  $X$  座標  $X_a$  の絶対値を四捨五入し、 $X_a$  と同符号を与えることにより  $X_i$  を定める。

このようにして作成されたデジタル直線は、直線を構成している画素列の方向性から次の三つのタイプに分けることができる。タイプ 1 のデジタル直線は図 4 a) に示すように、画素の中心を通る始点から始めて次の画素が水平 ( $X$  軸) 方向あるいは垂直 ( $Y$  軸) 方向に並んでいる。幾つか (二つ以上) の画素が水平あるいは垂直方向に並んだ後、斜め 45 度方向に一画素の補正を施して、再び水平あるいは垂直方向へ画素が並ぶ、ということの繰り返しによりタイプ 1 の直線は構成されている。タイプ 2 のデジタル直線は図 4 b) に示すように、タイプ 1 と画素の並ぶ方向が逆になっているだけで同じ構成 (斜め 45 度方向に画素が並んだ後、水平あるいは垂直方向へ一画素の補正を施し、再び斜め 45 度方向に画素が並ぶという構成) をとっている。また、タイプ 1 とタイプ 2 の境界線は  $Y=(1/2)X$  で、この直線上のデジタル直線はタイプ 2 に属する。タイプ 3 の直線は、図 4 c) に示すように誤差の補正を施す必要がないので、ただ同じ方向 (水平、垂直あるいは斜め 45 度方向) に繰り返し画素が並ぶという構成になっている。

2.2 デジタル直線の解析

2.1 で定義したデジタル直線のチェイン符号化列に対しては、次の性質がある<sup>10)</sup>。

- ① 一本のデジタル直線に対応するチェイン符号化列にはたかだか、二種類のチェイン符号しか存在せず、それらは 8 の剰余系で 1 だけ異なる。
- ② 二種類のチェイン符号のうち、一方は常に孤立して現れる。
- ③ 孤立して現れるチェイン符号どうしの間隔は、

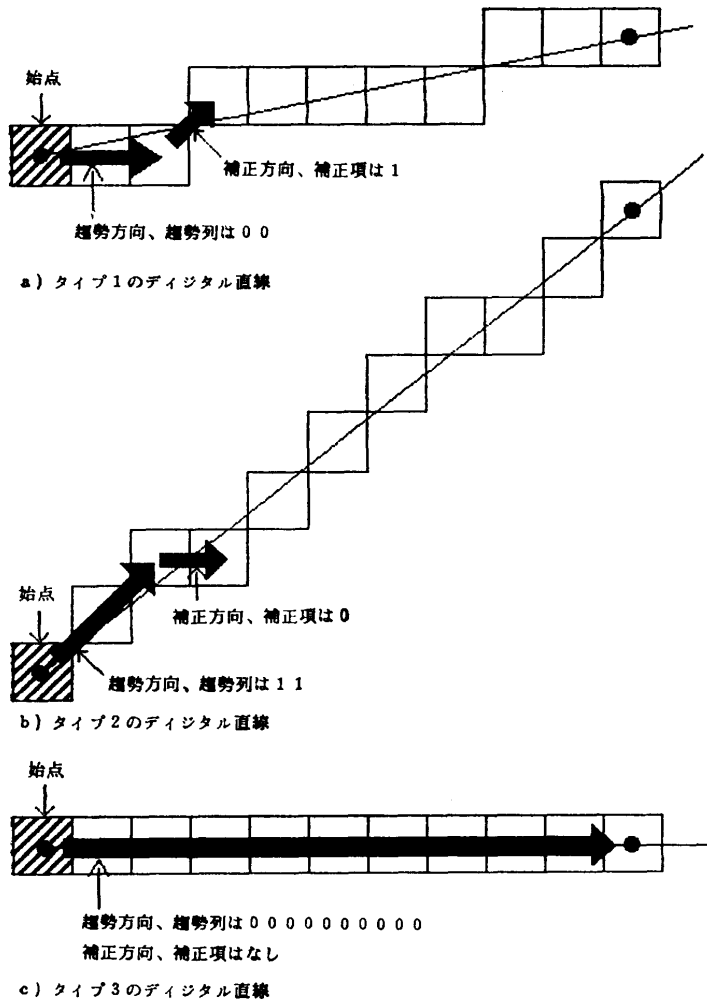


図4 デジタル直線の例  
 Fig. 4 Examples of digital line.

ほぼ一定である。

そこで、デジタル直線の始点から始まる同一のチェーン符号の連なりが直線の傾きの趨勢を表すので、この符号化列を趨勢列、孤立して現れるチェーン符号がアナログ直線からの誤差の補正を施すので補正項（長さは常に1）と呼ぶ。さらに、趨勢列および補正項のチェーン符号に対応する方向を、それぞれ趨勢方向および補正方向と呼ぶことにする（図4参照）。なお、この趨勢列の長さについては、次に示すようなさらに詳しい性質がある。

〈趨勢列の長さについての性質〉

2.1 で述べたように、デジタル直線は直線を構成している画素列の方向性から図4に示す三つのタイプに分けることができ、さらにタイプ3の直線には補正項がないので、ここでは元になるアナログ直線の傾き

$\alpha$  が  $0 < \alpha < 1$  ( $\alpha$  は有理数) である場合についてのみ考察する。2.1 でも述べたように図形の対称性を考慮すれば、次の結果は一般性を失うことはない。図5に示すように、与えられた始点および終点が画素の中心点であると考へ、それらをつなぐ直線を2.1の規則に従って量子化を行い、デジタル直線を作成する。既約分数で表された傾き  $\alpha = n/m$  に対して、デジタル直線の始点から終点までの画素数が  $m$  以下であれば、直線が画素の中心を通る点は現れないが、 $m+1$  以上であれば、途中に必ず直線が画素の中心を通る点（この点を中継点と呼ぶ）が現れる。また一般に、画素数が  $m+1$  以上の直線は  $m$  以下の直線の繰り返しによって構成されている。そこで、デジタル直線の始点から始まり一定方向に連なる画素列数を  $L_k$ 、直線途中における中継点を含まない一定方向に連なる画素列数を  $L'_k$ （ここで、 $k$  は  $1 \leq k \leq n-1$  で、 $L'_k$  が趨勢方向に対して垂直に  $L_k$  から  $k$  画素離れた位置に存在することを示す指標である）、中継点を含む一定方向に連なる画素列数を  $L''_k$ 、終点に至る最終的な画素列数を  $L_n$  とすると、これらの画素列数（同じチェーン符号を構成する画素の

個数で、趨勢列数よりも必ず1大きい）の間には次の関係式が成立する。

- ①  $0 < \alpha = (1/h) < 1/2$  ( $h$  は正の整数)  
 $L_k = \lceil h/2 \rceil$ ,  $L'_k = h$ ,  $L''_k = (h+1) - \lceil h/2 \rceil$   
 $2L_k - 1 \leq L'_k \leq 2L_k$ ,  $L_k \leq L''_k \leq L_k + 1$
- ②  $0 < \alpha = (n/m) < 1/2$  ( $n \neq 1$ ,  $n, m$  は互いに素な正整数)  
 $L_k = \lceil m/2n \rceil$ ,  $L'_k = \lceil (2k+1)m/2n \rceil - \lceil (2k-1)m/2n \rceil$   
 $L''_k = \lceil m/2n \rceil - \lceil (n-1)m/2n \rceil + m$   
 $L_k = m+1 - \lceil (2n-1)m/2n \rceil$   
 $2L_k - 2 \leq L'_k \leq 2L_k$ ,  $2L_k - 2 \leq L''_k \leq 2L_k$ ,  $L_k = L_n$
- ③  $1/2 \leq \alpha = (n/m) \leq 1$  ( $n, m$  は互いに素な正整数)  
 $L_k = \lfloor m/2(m-n) \rfloor + 1$

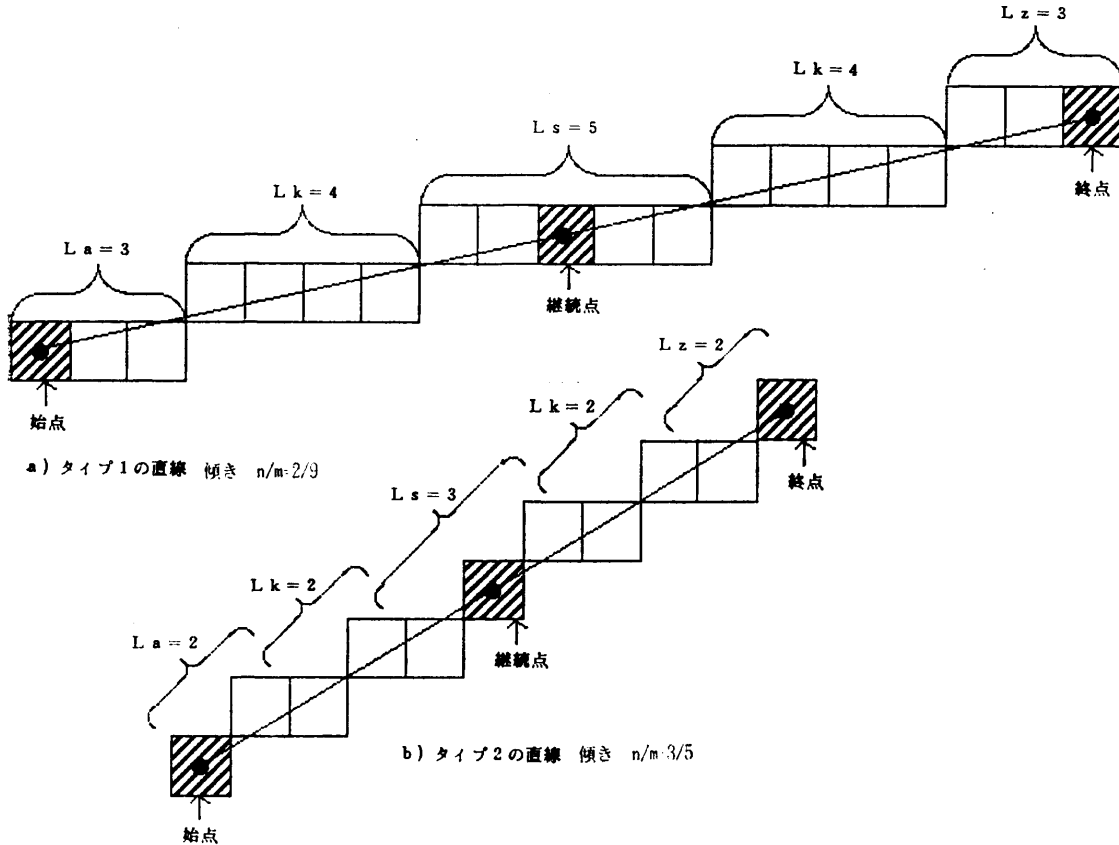


図5 デジタル直線の性質  
Fig. 5 Properties of digital line.

$$L_k = \lfloor (2k+1)m/2(m-n) \rfloor - \lfloor (2k-1)m/2(m-n) \rfloor$$

$$L_s = \lfloor m/2(m-n) \rfloor - \lfloor 2(m-n) - 1 \rfloor m/2(m-n) + m$$

$$L_z = m - \lfloor 2(m-n) - 1 \rfloor m/2(m-n)$$

$$2L_a - 2 \leq L_k \leq 2L_a$$

$$2L_a - 2 \leq L_s \leq 2L_a - 1$$

$$L_a - 1 \leq L_z \leq L_a$$

ただし、「 $\lceil X \rceil$ 」は  $X$  以上の整数の中で最も小さい数を、「 $\lfloor X \rfloor$ 」は  $X$  を超えない最大整数を表すものとする。

また、これらの性質を用いて実際にグラフィック画面上に作図された図形（多角形）に対して、高速にその頂点を抽出した実験結果も報告されている<sup>18)</sup>。

### 2.3 アナログ直線までの距離と許容誤差

本節では、直線近似に必要なアナログ

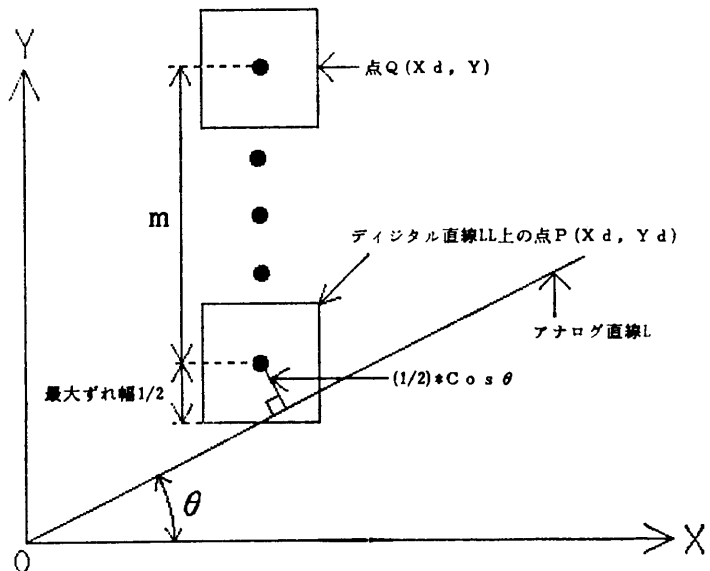


図6 アナログ直線までのユークリッド距離  
Fig. 6 Euclidean distance to analog line.

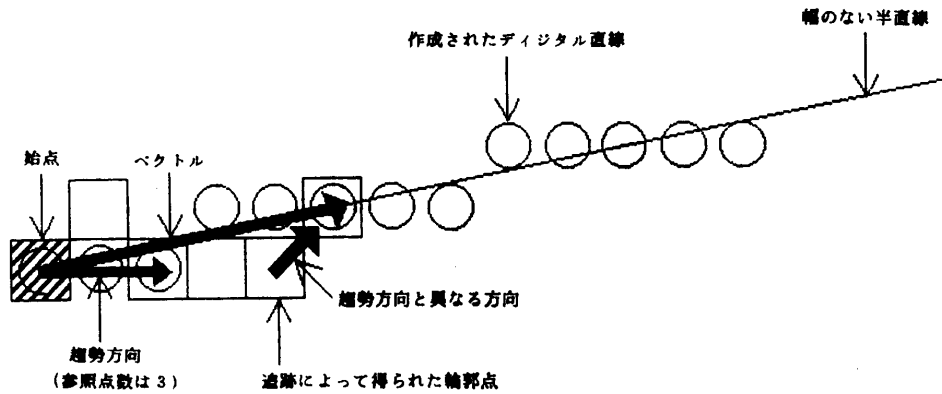


図7 デジタル直線の作成  
Fig. 7 A drawing of digital line.

直線までのユークリッド距離とデジタル直線を用いた近似（デジタル直線近似）における許容誤差との関係について述べる。デジタル直線  $LL$  上の点を、図6に示すように  $P(X_a, Y_a)$  とすると、アナログ直線  $L$  とデジタル直線  $LL$  の最大ずれ幅は  $1/2$  なので、 $P$  から  $L$  までのユークリッド距離  $D(P, L)$  は、

$$D(P, L) \leq (1/2) \times \cos \theta \quad (1)$$

となる。ただし、 $\theta$  はアナログ直線  $L$  の傾きである。

したがって、デジタル直線上の点  $P$  からさらに軸方向へ  $m$  だけ離れた点  $Q$  からアナログ直線  $L$  までのユークリッド距離  $D(Q, L)$  は、

$$m \times \cos \theta \leq D(Q, L) \leq \{m + (1/2)\} \times \cos \theta \quad (2)$$

となる。近似誤差の許容範囲を  $\varepsilon$  と設定した場合、 $m$  に関しては常に、

$$\{(1/2) + m\} \times \cos \theta \leq \varepsilon \quad (3)$$

が成立しなければならない。よって、ユークリッド距離での許容誤差  $\varepsilon$  をデジタル直線近似における許容誤差  $m$ （軸方向の許容ずれ幅）に変換する式は次のようになる。

$$m = \lfloor (\varepsilon / \cos \theta) - (1/2) \rfloor \quad (4)$$

### 3. デジタル直線近似

#### 3.1 多角形近似アルゴリズム

本章では、前章で述べたデジタル直線の性質を利用したデジタル二値物体の高速多角形近似アルゴリズムについて考察し、さらに実験結果について報告する。なお、図形は近似誤差の閾値を越える大きさあるいは幅を持つ物体を対象とする。アナログ直線を2.1で述べた方法によって量子化した直線（デジタル直線）を考えることにより、多角形の頂点座標を正確かつ高速に求めることができる。つまり、デジタル直

線の趨勢列の長さについては厳密な関係式が成立しているから、この性質を用いれば頂点抽出を容易に行うことができる。しかしながら、実在の多角形物体をデジタル画像として計算機に入力する場合、物体の輪郭部は量子化誤差の影響を受けるため、2.2で示した性質を厳密に保っているデジタル直線は得られない。直線の趨勢方向も単に始点と次の画素だけでは決定できず、始点から始まる数画素を参照する必要が生じてくる。そこで、この参照すべき画素を参照画素、その個数を参照点数と呼ぶことにし、始点から最後の参照画素に至るベクトルによって趨勢方向を決定する。

二値物体の輪郭を追跡し、ベクトルを作成して趨勢方向が決定されると、この趨勢方向と異なる方向のチェーン符号が現れるまで無条件に物体の輪郭を追跡し、始点からのベクトルを更新して行く。趨勢方向と異なる方向のチェーン符号が現れた時点で、始点を通り現在作成されているベクトルと同じ傾きをもつ半直線を量子化することによって、デジタル直線を作成する（図7参照）。

これ以後、追跡によって得られた輪郭点と作成されたデジタル直線上の点とを、2.1で述べたデジタル直線作成の定義に従って比較する。つまり、 $X_i$  に対応して  $Y_i$  を定めた場合は同一の  $X$  座標に対する  $Y$  座標の比較を、逆に  $Y_i$  に対応して  $X_i$  を定めた場合は同一の  $Y$  座標に対する  $X$  座標の比較を行う。そして、二点の座標差の絶対値が2.3で求めた許容誤差  $m$  内であれば同じデジタル直線により近似を続け、 $m$  を超えていれば一つ前の点で同一直線による近似を中止し、新たに別のデジタル直線を考える。多角形の頂点は、デジタル近似直線を作成する基に

なった近接二本のアナログ直線（始点の逆方向にも延長）の交点を四捨五入することにより、整数化した点とする。

この方法は、物体の輪郭点とデジタル直線上の点との  $X$  座標あるいは  $Y$  座標の差を求めるだけで近似が行え、輪郭点から近似直線までのユークリッド距離を逐一計算する必要がない。また、物体の輪郭を追跡しながら局所的かつ逐次的に処理ができるので、あらかじめ輪郭点を抽出し保存しておく必要がなく、省メモリで効率のよい高速な処理を行うことができる。

### 3.2 抽出すべき頂点の選択

前節で述べたアルゴリズムでは、二つのアナログ近似直線の交点を頂点としていた。ところが、二つの近似直線が平行なときは交点が存在しないし、また平行

に近い場合は頂点として不適当な位置に交点が見れる（図 8）。このような場合、ただ単に近似直線の両端点を頂点として採用すると、復元された図形の輪郭部が階段上になり、近似効率も悪くなる。そこで頂点抽出には少し工夫を加え、交点が存在しない場合は、各近似直線の許容範囲にある物体輪郭上の端点（始点あるいは終点）から他の近似直線まで垂線を下し、二つの垂線の足の midpoint を四捨五入により量子化した点を頂点として抽出する。また不適当な位置に交点が見れる場合は、近似直線の端点（始点）を頂点として抽出する。

こうすることにより、近似結果としての頂点を基に復元された図形の歪み（輪郭点の階段上効果）がなくなり、また抽出点数も減少するので近似効率も良くなる。しかし、元の近似直線と復元された直線とが一致

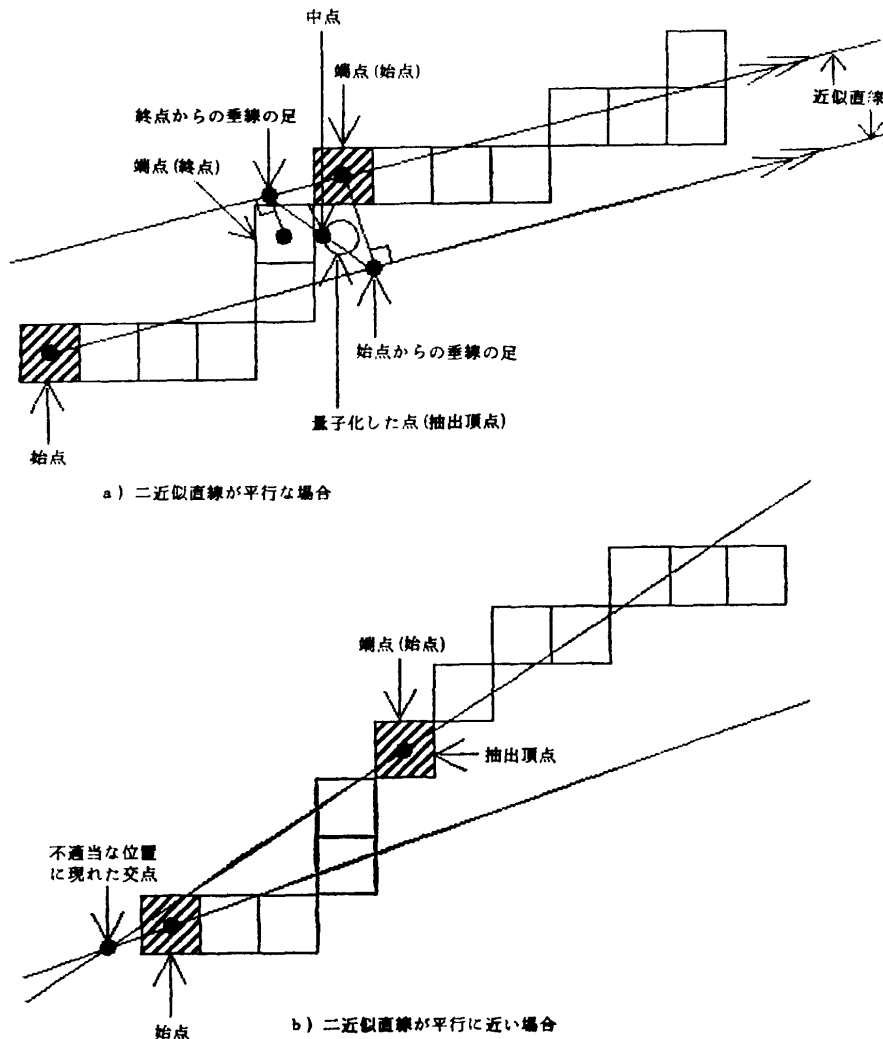


図 8 抽出頂点の選択  
Fig. 8 A selection of extracted vertex.

表 1 3方法で近似した結果  
Table 1 Results approximated by three methods.

	近似方法	頂点数	面積	近似時間	E1	E2	E1+E2	E1-E2	圧縮率	誤差
1. ティーポット 輪郭点数: 811画素 面積: 38,530画素 輪郭抽出時間: 1分56秒	MM	49	37,847	33分06秒	819	136	955	683	0.060	0.018
	RA	38	37,660	2分26秒	880	10	890	870	0.047	0.023
	DL	47	37,356	3分04秒	1,233	59	1,292	1,174	0.057	0.030
		(57)*								(0.070)**
2. モンキーバナナ 輪郭点数: 697画素 面積: 10,886画素 輪郭抽出時間: 1分43秒	MM	53	10,838	29分35秒	391	343	734	48	0.076	0.004
	RA	23	10,849	1分42秒	197	160	357	37	0.033	0.003
	DL	30	10,830	2分30秒	322	266	588	56	0.043	0.005
		(32)*								(0.046)**
3. 電話の受話器 輪郭点数: 571画素 面積: 11,504画素 輪郭抽出時間: 1分23秒	MM	43	11,462	22分35秒	321	279	600	42	0.075	0.004
	RA	23	11,394	1分31秒	188	78	266	110	0.040	0.010
	DL	29	11,191	2分03秒	436	123	559	313	0.050	0.027
		(40)*								(0.070)**
4. コーヒーミル 輪郭点数: 1,120画素 面積: 30,796画素 輪郭抽出時間: 2分37秒	MM	51	30,818	50分22秒	418	462	880	-44	0.046	-0.001
	RA	48	30,762	3分23秒	271	237	508	34	0.043	0.001
	DL	54	31,496	3分48秒	278	978	1,256	-700	0.048	-0.023
		(161)*								(0.144)**
5. フライパン 輪郭点数: 825画素 面積: 33,184画素 輪郭抽出時間: 2分06秒	MM	53	33,057	30分36秒	478	351	829	127	0.064	0.004
	RA	28	33,150	2分13秒	197	164	361	33	0.034	0.001
	DL	32	32,618	2分56秒	625	59	684	566	0.039	0.017
		(52)*								(0.063)**
6. ハサミ 輪郭点数: 985画素 面積: 11,294画素 輪郭抽出時間: 2分33秒	MM	55	11,400	50分02秒	405	501	906	-96	0.056	-0.009
	RA	42	11,025	2分58秒	371	102	473	269	0.043	0.024
	DL	49	11,109	3分23秒	561	376	937	185	0.050	0.016
		(113)*								(0.115)**
7. 数字の5 輪郭点数: 695画素 面積: 8,951画素 輪郭抽出時間: 2分40秒	MM	53	8,898	24分40秒	378	325	703	53	0.076	0.006
	RA	42	8,922	2分20秒	186	157	343	29	0.060	0.003
	DL	44	8,896	2分30秒	381	326	707	55	0.063	0.006
		(76)*								(0.100)**

近似方法 MM: ミニマックス近似 (Kurozumi and Davisの方法)

RA: 反復近似 (Ramerの方法)

DL: デジタル直線近似 (提案した方法)

E1: 元の図形の内点であるが近似図形の外点であるものの画素数

E2: 元の図形の外点であるが近似図形の内点であるものの画素数

(圧縮率) = (頂点数) / (輪郭点数), (誤差) = (E1 - E2) / (元の面積)

\* デジタル直線近似で近似直線までの距離が閾値を超える輪郭点数. ただし, 閾値はすべて2.0画素.

\*\* (精度) = (\*の数) / (輪郭点数)

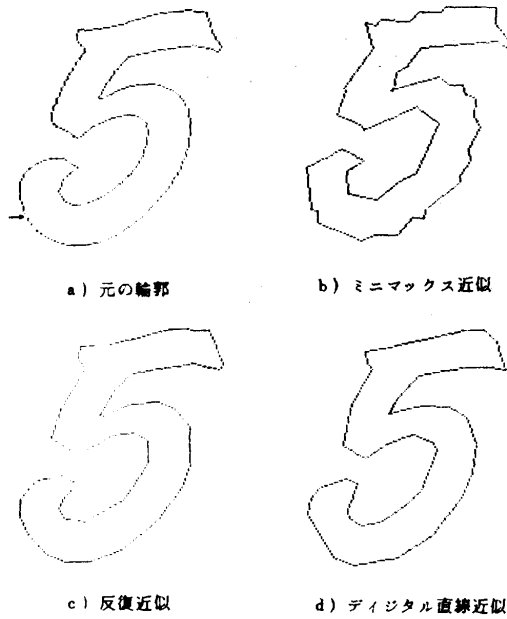
しなくなるため, 物体の輪郭点から復元された直線までの距離が最初に定めた閾値内に収まっていることは, 保証されない.

### 3.3 デジタル直線近似の結果

本論文で提案した近似方法 (以下デジタル直線近似と呼ぶ) を他の代表的な方法である, ミニマックス近似<sup>9)</sup> および反復近似<sup>10)</sup> と比較した結果を表1に, またその一例を図9に示す. なお, プログラムは BASIC (安立電気 Packet II e) で組んであり, 直線の趨勢方向を求めるための参照点数はすべて3とした. 許容誤

差  $m$  は(4)式で与えられるが, 前述したように近似精度を保証しないので近似効率を考慮して  $(\epsilon/\cos\theta) - (1/2)$  を四捨五入し,  $m = \lfloor \epsilon/\cos\theta \rfloor$  として近似を行った.

図9を見ると, デジタル直線近似した結果は反復近似に比べて少し見劣りするが, 正確に元の輪郭を保っていることがわかる. また表2に示すように, 圧縮率は0.05で, 誤差は1.8%であった. 3.2で述べたように, デジタル直線近似は図形の歪みをなくして圧縮率 (近似効率) の向上を図るために, 距離誤差の



但し、→は境界追跡開始点を表す

図 9 近似結果の比較

Fig. 9 Comparisons of approximation results.

表 2 平均した近似結果

Table 2 Comparisons of averaged approximation results.

	MM	RA	DL
圧縮率	0.06	0.04	0.05
誤差	0.007	0.009	0.018
精度	0	0	0.09
頂点数の比	1.25	0.86	1
輪郭抽出時間	2分10秒	2分10秒	0秒
近似時間	34分25秒	2分21秒	2分53秒

精度を犠牲にした。表 2 (精度) を見れば確かに約 9% ほどの点が閾値距離 2.0 を超えているが、大部分の点は距離 4.0 以内に収まっている。さらに処理時間の考察を行うと、プログラムは BASIC で組んであるため厳密な比較はできないが、他の二つの方法ではあらかじめ抽出された輪郭点を基に近似を行うため、抽出時間として約 2 分 10 秒を要し、近似するのにミニマックス近似では約 34 分 25 秒、反復近似では約 2 分 21 秒を要する。これに対してデジタル直線近似では、局所的に抽出された輪郭点だけから近似を行えるので、近似のために約 2 分 53 秒を要するだけで、輪郭点を抽出する時間を必要としない。

したがって、本論文で提案した近似方法は近似誤差の距離が正確に一定の閾値内に収まっていることは保証されないが、適当な誤差範囲内で元の輪郭を正確に保ち、しかもアルゴリズムの記述が容易で高速に物体の輪郭を多角形近似できることが示された。

#### 4. 移動物体の認識

##### 4.1 認識対象の限定

3 章で述べたアルゴリズムは動画像にも適用できる。しかし、動画像を扱う場合、画像のふれや残像およびこれらに伴うノイズ等が問題となり、任意形状の物体を取り扱うのは困難となる。そこで本章では、安価なプロセッサを用いて動画像のリアルタイム認識を実現するために、移動物体の形状を以下の条件を満足する多角形に限定し、デジタル直線の性質を用いて物体の輪郭を追跡しながら同時に近似 (頂点抽出) も行うという基本方針のもとに、3 章のアルゴリズムに多少の変更を加えた。その上、高速にしかも正確に処理を行うために、多角形の形状を詳しく調べることなく、頂点の個数だけを用いて認識・識別できる図形を扱うことにした。

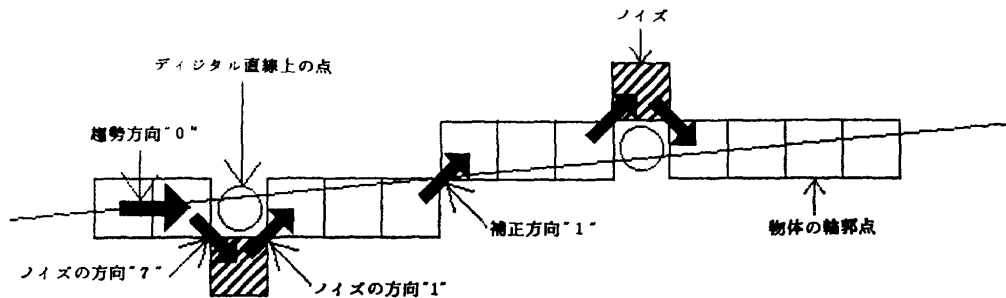
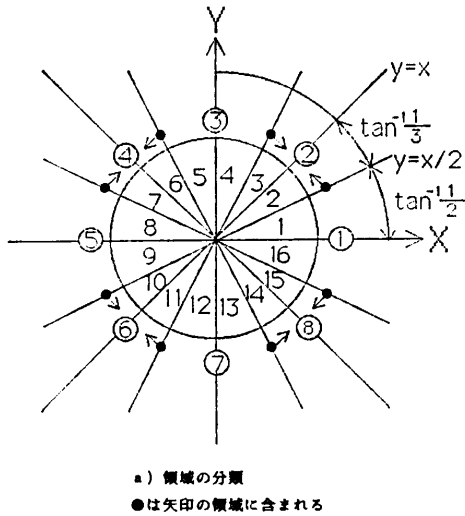


図 10 ノイズの方向

Fig. 10 Direction of noise.





領域	趨勢方向	補正方向	ノイズの方向	領域	趨勢方向	補正方向	ノイズの方向
1	0	1	1, 7	13	6	7	7, 5
2	1	0	0, 2	14	7	6	6, 0
3	1	2	2, 0	15	7	0	0, 6
4	2	1	1, 3	16	0	7	7, 1
5	2	3	3, 1	①	0	なし	7, 1
6	3	2	2, 4	②	1	なし	0, 2
7	3	4	4, 2	③	2	なし	1, 3
8	4	3	3, 5	④	3	なし	2, 4
9	4	5	5, 3	⑤	4	なし	3, 5
10	5	4	4, 6	⑥	5	なし	4, 6
11	5	6	6, 4	⑦	6	なし	5, 7
12	6	5	5, 7	⑧	7	なし	6, 0

b) 許容方向

図 11 領域と許容方向の関係  
Fig. 11 Relation between area and allowable directions.

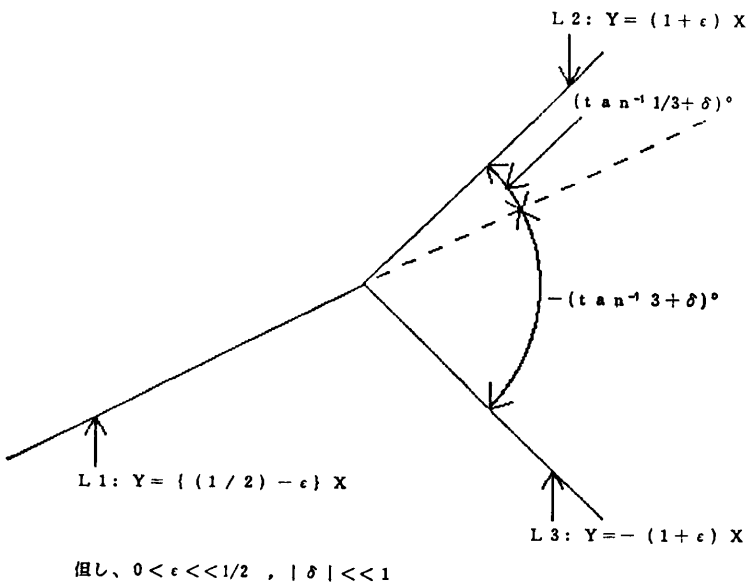


図 12 直線の識別と角度  
Fig. 12 Direction of line and angle.

- ① 外角が  $\tan^{-1} 3$  以上ある多角形（詳細は後述）で、凸多角形でも凹多角形でも構わない。
- ② 画像上における物体の個数制限はないが、頂点数の同じ物体が二つ以上存在してはならないし、各々の物体が重なることもないとする。
- ③ 姿勢を制御する物体は、制御しやすいように前方を識別できる特徴を持たせる（例えば、二等辺三角形  $\Delta$  や矢印形  $\uparrow$  等）。
- ④ 適当な閾値により容易に二値化できる二次元平

面画像のみを扱う。

ここで、特に①の条件を設けた理由を以下に述べる。画像上のノイズを詳しく調べてみると、図 10 で示すような 2.1 で定義したデジタル直線上の点から一画素ずれる点がよく現れ、その方向はチェイン符号で趨勢方向と“1”だけ異なることがわかる。そこで、この方向をノイズの方向と呼ぶ。デジタル直線の補正方向も趨勢方向とチェイン符号で“1”だけ異なる方向を持つから、一本のデジタル直線で許容される方向（許容方向）を趨勢方向  $\pm 1$  の 3 方向とする。この様子を図 11 に示す。

本節では、リアルタイムで多角形を識別するために、デジタル直線を構成しているチェイン符号の方向のみを

用いて直線を識別する方法について考察する。図 12 に示すように、 $Y = \{(1/2) - \epsilon\} X$  である直線  $L_1$  を考えると  $L_1$  は図 11 の領域 1 の直線であるから、許容方向として (0, 1, 7) の 3 方向を持っている。すべての直線がノイズの方向を含む 3 方向を持っていれば、図 11 に示す 16 の領域のうち 8 領域内にある直線は他の領域内にある直線と許容方向だけを用いて識別することができる。しかし、必ずしもすべての直線がノイズの方向を持つとは限らない。また図 11 の丸で囲んだ

領域番号の直線は補正方向を持たない。そこで、 $Y=(1+\varepsilon)X$  の直線  $L_2$  を考えると図 11 の領域 3 に属するから、この直線は補正方向として必ず“2”の方向を持つ。 $L_1$  の許容方向は(0, 1, 7)の3方向なので、 $L_1$  と  $L_2$  は許容方向のみを用いて識別することができる。また、 $L_1$  に関して  $L_2$  と反対側にある直線  $L_3$  として  $Y=-(1+\varepsilon)X$  を考えると、この直線は補正方向として必ず“6”の方向を持つから、やはり許容方向のみを用いて  $L_1$  と識別することができる。

よって、許容方向のみを用いて直線  $L_1$  と識別するためには二直線の角度が  $+(\tan^{-1} 1/3)$  あるいは  $-(\tan^{-1} 3)$  を超える必要がある。同様に、他の領域に属する直線についても考察してみると、二直線を識別することができない可能性のある角度  $\theta$  は次のようになる。

$$\begin{aligned} &-(\tan^{-1} 3) \leq \theta \leq +(\tan^{-1} 1/3) \\ &(\text{or } -(\tan^{-1} 1/3) \leq \theta \leq +(\tan^{-1} 3)) \\ &-(\tan^{-1} 2) \leq \theta \leq +(\tan^{-1} 1/2) \\ &(\text{or } -(\tan^{-1} 1/2) \leq \theta \leq +(\tan^{-1} 2)) \end{aligned}$$

以上の事柄より、多角形の隣接する二辺を識別するためには、二辺の角度(多角形の外角)の絶対値は最低  $\tan^{-1} 3$  以上必要となる。また、許容方向以外のチェイン符号が現れた点を物体の頂点として抽出する。物体の輪郭を一回りした時点で頂点の個数が判定されるので、この個数のみを用いて物体を認識・識別する。位置ベクトルは各頂点の平均値(代表点と呼ぶ)として定め、この位置ベクトルから追跡すべき物体の速度方向を決定する(後述)。

#### 4.2 物体認識の実験結果

上記の考えを基にして、実際に約 33 ms で次々と入力してくる動画像(解像度は  $128 \times 128$ )の認識を行った。境界追跡をしながら頂点を抽出し、各々の物体を認識・識別して追跡物体(姿勢制御を行い、目標物を追跡する物体)に姿勢制御命令(前進、右回転、左回転の3命令)を送る。ここまでの処理を 33 ms 以内に行い、A/D 変換と同期をとるために信号待ちをしてから、次の画像処理を行っている。

システムの構成を図 13 に示す。CCD カメラから

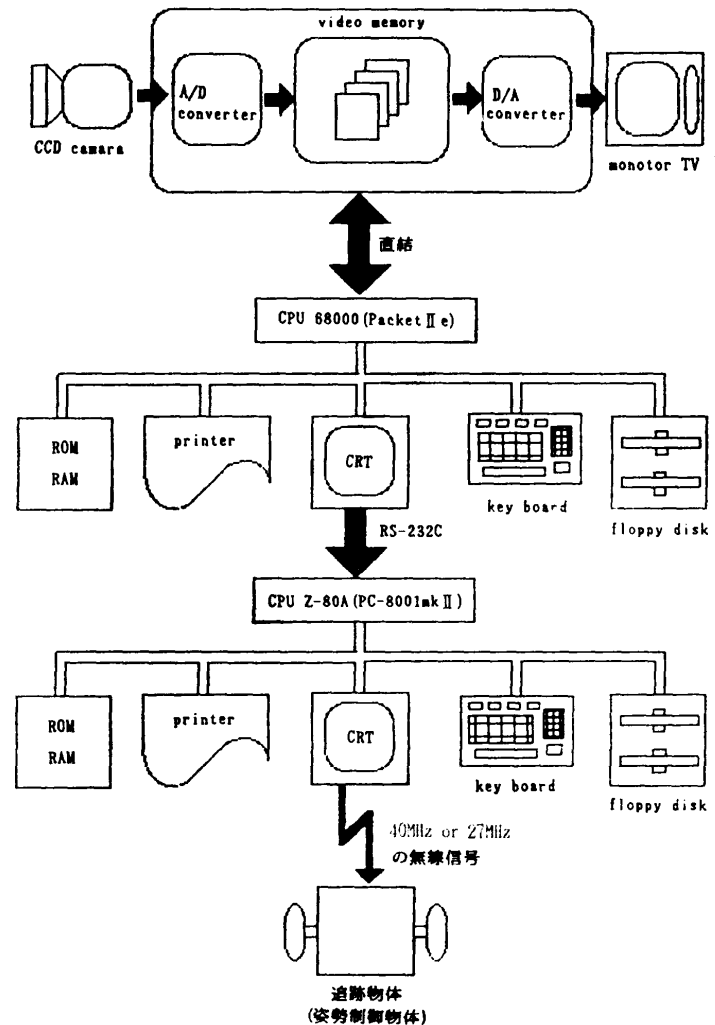


図 13 システムの構成

Fig. 13 Composition of the system.

入力された画像情報は CPU 68000 を備える Packet II e で処理され、姿勢制御命令が RS-232C を通して CPU Z-80A を備える PC-8001mk II に送られる。そして、PC-8001mk II から 40 MHz あるいは 27 MHz の無線信号により、追跡物体に命令が送られて、目標物体(目標となる物体)を追跡する。

使用した平面領域(カメラの視野)は、縦×横 =  $130 \times 136 \text{ cm}^2$  の長方形でカメラを地上 229 cm の位置に設置した。また容易に二値化できるように背景を黒画用紙で、移動物体を白画用紙で覆った。追跡物体の形状は底辺 15 cm 高さ 20 cm の二等辺三角形で移動速度は約 20 cm/s、目標物体の形状は一辺 15 cm の正方形で移動速度は約 15 cm/s であった。なお、直線の趨勢方向を求めるための参照点数は 5 とし、姿勢制御

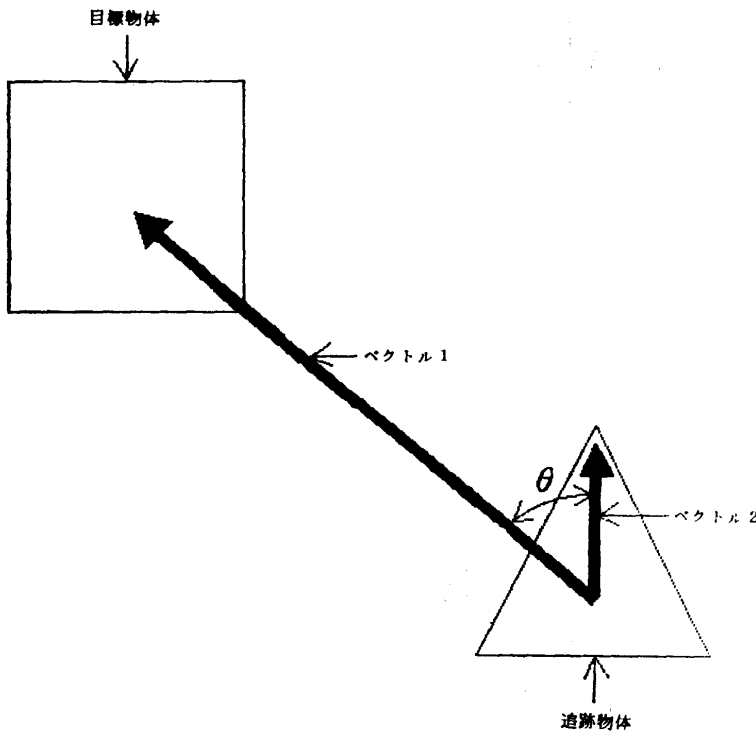


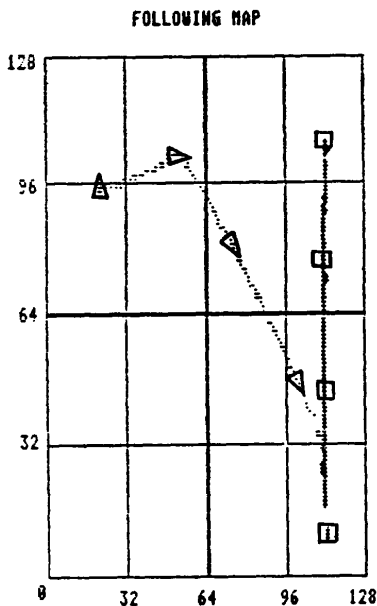
図 14 制御命令の与え方  
Fig. 14 A method of giving control command.

命令（速度方向）は次のようにした。図 14 に示すように、追跡物体（二等辺三角形）の代表点から目標物体（正方形）の代表点までのベクトル 1 と二等辺三角形の代表点から頂角を与える点までのベクトル 2 とのなす角度  $\theta$  が、40 度を超えていれば回転（右回転あるいは左回転）、40 度以下であれば前進を制御命令として追跡物体に与えた。実験結果を図 15 および図 16 に示す。

5. むすび

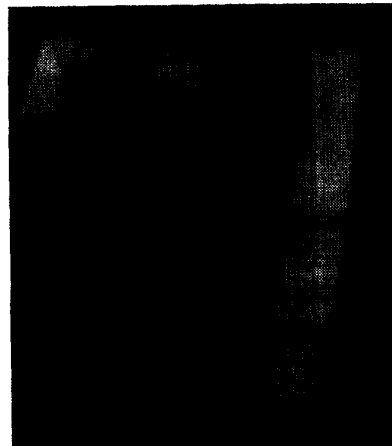
本論文では、デジタル直線の性質を調べてこの性質を利用することにより、物体の輪郭を追跡しながら高速に精度よく多角形近似するアルゴリズムを提案し、この手法を用いて移動物体の認識をリアルタイムで行った。提案した方法は、正確に元の輪郭を保ち圧縮率も良く、しかも高速に近似できることが示された。

距離誤差が必ずしも閾値内に収まっていることは保証されないが、大きく歪むこともないので大雑把な高速近似には有効である。また移動物体の認識も各頂点の対応付けを行うことなく、一画面ごとに再認識して姿勢制御を行っているので、ノイズに対しても頑強である。したがって、本論文で提案した方法は有効であ



目標物体：□，追跡物体：△  
目標物体は上から下へ移動している

図 15 移動物体の軌跡  
Fig. 15 Trace of the moving bodies.



目標物体：□，追跡物体：△  
目標物体は上から下へ移動している

図 16 追跡中の様子  
Fig. 16 Tracing scene.

り、今後この方法を用いて画像を利用したロボットの視覚を作製することが期待される。

**謝辞** 常日頃から適切な助言をしていただいた豊橋技術科学大学の森田啓義氏、大阪大学の橋本猛氏、また有本研究室の皆様深く感謝します。

### 参 考 文 献

- 1) 谷内田：視覚を用いた自動検査および自動組み立てシステムの動向，日経エレクトロニクス，1982年6月7日号，No. 292，pp. 169-203 (1982)。
- 2) 谷内田，辻：機構部品の学習・認識システム，信学論 (D)，Vol. J59-D，No. 3，pp. 149-156 (1976)。
- 3) 溝口，村田，井上：マルチウィンドウ方式ロボット認識システム—試作システムを用いた応用実験—，第21回日本ロボット学会学術講演会，pp. 261-262 (1984)。
- 4) Barr, A. and Feigenbaum, E. A.: *The Handbook of Artificial Intelligence*, Vol. 3, pp. 206-224, W. Kaufman, Cal. (1982)。
- 5) Hilditch, C. J.: Linear Skeleton from Square Cupboards, *Machine Intelligence*, Vol. 4, pp. 403-420 (1969)。
- 6) Rosenfeld, A. and Kak, A. C.: *Digital Picture Processing*, pp. 333-403, Academic Press, N. Y. (1976)。
- 7) Pavlids, T.: *Algorithm for Graphic and Image Processing*, pp. 1-165, Computer Science Press, Md. (1982)。
- 8) Freeman, H.: *Picture Processing and Psychopictorics*, Lipkin, B. S. and Rosenfeld, A. (eds.), pp. 241-266, Academic Press, N. Y. (1970)。
- 9) Kurozumi, Y. and Davis, W.: Polygonal Approximation by the Minimax Method, *Comput. Gr. Image Process.*, Vol. 19, pp. 248-264 (1982)。
- 10) Ramer, U. E.: An Iterative Procedure for the Polygonal Approximation of Plane Curves, *Comput. Gr. Image Process.*, Vol. 1, pp. 244-256 (1972)。
- 11) Williams, C. M.: An Efficient Algorithm for the Piecewise Linear Approximation of Planar Curves, *Comput. Gr. Image Process.*, Vol. 8, pp. 286-293 (1978)。
- 12) Momtanari, U.: A Note on Minimal Length Polygonal Approximation to a Digitized Contour, *Comm. ACM*, Vol. 13, No. 1, pp. 41-

47 (1970)。

- 13) Betley, J. L. and Faust, M. G.: Approximation Algorithms for Convex Hulls, *Comm. ACM*, Vol. 25, No. 1, pp. 64-68 (1982)。
- 14) Freeman, H.: Computer Processing of Line Drawing Images, *Comput. Surv.*, Vol. 6, No. 1, pp. 57-97 (1974)。
- 15) 木村，矢代，今井：デジタル画像における線図形の変ベクトル符号化法，信学論 (D)，Vol. J66-D，No. 1，pp. 73-80 (1983)。
- 16) 木本，安田：Hilditch法を用いた細線化図形の方法類別チェーン差分符号化方式，第4回情報理論とその応用研究会資料，pp. 238-247 (1983)。
- 17) 中島，安居院：デジタル輪郭線のデルタ符号化について，信学論 (D)，Vol. J64-D，No. 2，pp. 109-115 (1981)。
- 18) 向井，森田，有本：デジタル直線を利用した境界追跡による高速特徴点抽出アルゴリズム，第29回情報処理学会全国大会講演論文集，pp. 1043-1044 (1984)。

(昭和61年1月31日受付)

(昭和61年11月5日採録)



向井 信彦 (正会員)

昭和35年生。昭和58年大阪大学基礎工学部機械工学科卒業。昭和60年同大学院修士課程修了。同年三菱電機(株)入社。同社情報電子研究所情報システム開発部所属。現在ワークステーションの研究・開発に従事。画像処理，コンピュータ・グラフィックスに興味をもつ。



有本 卓 (正会員)

昭和11年生。昭和34年京都大学理学部卒業。同年沖電気(株)入社。電子計算機の開発に従事。37年東京大学工学部助手，42年講師，43年大阪大学基礎工学部助教授，48年教授，42年工学博士。この間，情報理論，制御理論，デジタル信号処理の高速アルゴリズムに従事する一方，ロボット工学に興味をもち，そのインテリジェント化の研究開発を行っている。電子情報通信学会，計測自動制御学会，日本機械学会，ロボット学会各会員。IEEEのFellow会員。