

# $\theta$ - $\rho$ パラメータによる多角形の物体検出

学生員 奥苑 大成\* 正員 脇迫 仁\*

## Object Detection Using Straight Line Matching in $\theta$ - $\rho$ Space

Taisei Okuzono\*, Student Member, Hitoshi Wakizako\*, Member

The contours of many industrial parts contain straight lines and positions of the lines are therefore information for object detection. This paper presents a matching technique of straight lines in  $\theta$ - $\rho$  space. Any lines in 2D space are represented with parameters  $\theta$  and  $\rho$  by Hough transform. To find an object to find combination of line parameters in  $\theta$ - $\rho$  space. Our matching method contains edge detection, detection and matching processes. At first, we perform edge detection for model and scene image to detect contour of objects. Next, we extract the straight lines using Hough transform for provided edge image. In the matching process, we perform matching with parameters of straight lines. Matching process contains  $\theta$  matching,  $\rho$  matching and pose estimation. In the  $\theta$  matching, we use the relative  $\theta$  values of corresponding lines. In the  $\rho$  matching, we compute parameters of transfer and deviations of  $\rho$  values. Finally, in the pose estimation we compute transfer parameters using corresponding intersection points of the straight lines. experimental results show our method is robust for rotation, occlusion and scaling of objects.

キーワード：物体検出, ハフ変換, 直線検出, 画像解析

**Keywords:** object detection, hough transform, line detection, image analysis

### 1. まえがき

画像処理技術は物体検出や検査など製造ラインにおいて広く利用されている。特に物体検出は、産業用ロボットのハンドリング作業での対象ワークの位置ずれ検出や、半導体製造装置での位置決めマーク検出など様々な用途で使われている<sup>(1)(2)</sup>。ここでの画像処理は主に正規化相関によるパターンマッチングが使われているが、対象物の回転やオクルージョンが課題となっている。そのため、実際のラインにおいては対象物の向きを揃えるなど、お互いが重ならないような工夫がなされている。一方、回転やオクルージョンに対応するため、一般化ハフ変換<sup>(3)</sup>などの手法が提案されている。一般化ハフ変換では、対象物の輪郭の接線方向が投票空間のパラメータとなるが、実際のシーンにおいて輪郭の接線方向を精度よく求めることは難しく、最終的な検出精度に影響する。また、カメラと対象物の距離が変動すると対象物の大きさが変わり、スケールもパラメータとなり4次元の投票空間での探索となるため処理時間への影響が考えられる。上記の処理において対象は任意形状であ

るが、産業用の部品では直線部を含むパーツが多く、半導体のパターンでも多くは直線部から構成されているため、直線部による検出機能だけでも広い用途があると思われる。直線部に基づく物体検出では、直線検出が重なる。直線検出では、直線検出用のハフ変換<sup>(4)</sup>が知られているが、大きなメモリ空間が必要で計算量が多いと考えられてきた<sup>(5)</sup>。しかしながら近年の半導体技術の進歩により処理速度やメモリ容量が向上してきており、従来の課題実用化の課題とはいえなくなりつつある。

そこで本論文では、直線検出用ハフ変換によって検出された直線に基づく物体検出について述べる。本手法は、物の回転やオクルージョン、拡大・縮小にも対応した物体検出方式である。ここでは得られた直線の2次元投票空間でのパラメータを用いたマッチングを行っている。ハフ変換の投票空間を利用したマッチングでは回転や並進を求める報告<sup>(6)</sup>や、さらに拡大縮小のスケール変換に拡張した報告<sup>(7)</sup>などがある。これらの報告では、対象を任意形状として、投票空間におけるヒストグラムのマッチングによって回転成分を求め、さらにスケールや並進成分を求めているが、ここではハフ変換による $\theta$ - $\rho$ 平面での分布そのものを処理しているため、対象以外のものがあつたり、対象が隠蔽されると分布状態が変わりマッチングに影響が出てくると考えられる。一方、本手法では、対象物の直線に限定し、

\*九州工業大学  
804-8550 福岡県北九州市戸畑区仙水町 1-1  
Kyushu Institute of Technology  
1-1 Sensui-cho, Tobata-ku, Kitakyushu shi, Fukuoka, 804 8550

それらの  $\theta$ - $\rho$  平面での点同士のマッチングにより、マッチングの信頼性が上がり、高速化やロバスト性の向上が期待できる。本手法では、モデルとシーンの画像から直線を検出し、その中から対応する組合せを求めている。まず、カメラからの入力画像に対してエッジ画像を得る。次に、得られたエッジ画像よりハフ変換を用いて直線検出を行う。ハフ変換によってエッジ画像中の直線は二つのパラメータの組  $(\theta, \rho)$  で表される。あらかじめ登録しておいた検出対象 (モデル) と入力画像 (シーン) の直線パラメータを比較することにより、対応する直線の組み合わせを見つけ物体検出を行う。

概要

〈2・1〉 直線の  $\theta$ - $\rho$  平面への変換  $x$ - $y$  平面上の直線  $L$  上の点  $(x, y)$  は、ハフ変換によってパラメータ  $(\theta, \rho)$  で表される。また、 $\theta$  と  $\rho$  の関係は式 (1) で表される。

$$\rho = x \cos \theta + y \sin \theta = \begin{bmatrix} \cos \theta & \sin \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \dots\dots(1)$$

ここで、 $\rho$  は直線と原点との距離、 $\theta$  は直線と原点との垂角である。さらに Fig.1(a) に示す直線  $L$  が平行移動  $(t_x, t_y)$ 、回転角  $\alpha$ 、拡大率  $s$  で Fig.1(b) に示す直線  $L'$  に移動したとき、直線  $L'$  上の点  $(x', y')$  は (2) 式で表される。

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = s \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \dots\dots(2)$$

また、直線  $L'$  のパラメータ  $(\theta', \rho')$  は (3) 式で表される。

$$\begin{aligned} \rho' &= x' \cos \theta' + y' \sin \theta' \\ &= \begin{bmatrix} \cos \theta' & \sin \theta' \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} \dots\dots(3) \end{aligned}$$

$$\begin{aligned} &\frac{\rho' - t_x \cos \theta' - t_y \sin \theta'}{s} \\ &= \begin{bmatrix} \cos(\theta' - \alpha) & \sin(\theta' - \alpha) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \dots\dots(4) \end{aligned}$$

を得ることができる。(1) 式と (4) 式を比較して、(5) 式および (6) 式のように移動前と移動後の直線パラメータの関係を得ることができる。

$$\theta = \theta' - \alpha \dots\dots(5)$$

$$\rho' = t_x \cos \theta' + t_y \sin \theta' + s\rho \dots\dots(6)$$

Fig.2 および Fig.3 に例を挙げる。Fig.2(a) は三角形の像で、それぞれの直線を  $L_1, L_2, L_3$  とする。Fig.2(b)  $L_1, L_2, L_3$  を  $\theta$ - $\rho$  平面に変換しており、直線  $L_1, L_2, L_3$  が点  $P_1, P_2, P_3$  に対応し、 $\beta_1$  は  $L_1$  と  $L_2$  の相対角度、 $\beta_2$  は  $L_2$  と  $L_3$  の相対角度を表す。Fig.3(a) は Fig.2(a) が移動した状態で、回転角度は  $\alpha$  である。直線  $L'_1, L'_2, L'_3$

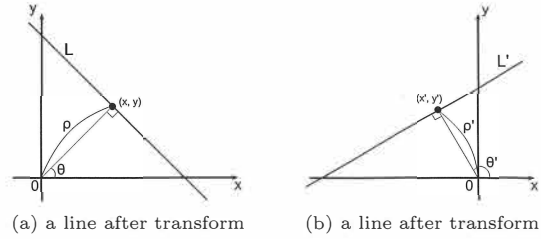


Fig. 1. A line in  $x$ - $y$  space before and after transform

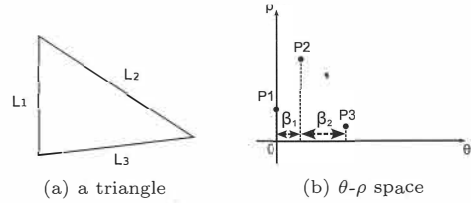


Fig. 2. Before Transfer

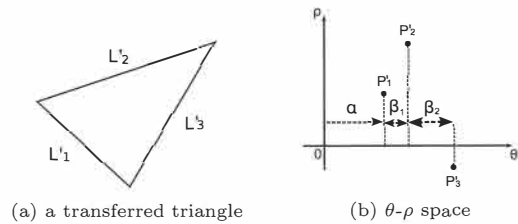


Fig. 3. After Transfer

はそれぞれ Fig.2(a) 中の直線  $L_1, L_2, L_3$  に対応する。直線間の相対角度である  $\beta_1$  および  $\beta_2$  は移動後も変わらないため、Fig.3(b) に示すように  $\theta$ - $\rho$  空間中におけるそれぞれの直線パラメータの  $\theta$  方向の相対位置も変わらず、回転角  $\alpha$  だけ  $\theta$  方向にずれるだけである。そのため、移動前と移動後の直線パラメータの  $\theta$  を比較することにより回転角  $\alpha$  を推定することができる。

〈2・2〉 物体検出 本論文で提案する手法は、物体検出を行うためにモデルとシーンで対応する直線部を見つけ、平行移動量、回転角、拡大率を推定する方法である。最も単純な方法は、すべての直線パラメータの組合せについて計算を行う方法であるが、それだと多くの処理時間を必要とする。そこで本手法では、複数のマッチング処理を設け、各マッチング処理で解と判断された直線の組合せのみ後段のマッチング処理に進めることで処理時間の短縮を図っている。Fig.4 に提案する物体検出法の概略を示す。提案する物体検出法は、エッジ検出、直線検出、マッチングの三つに分けられる。さらにマッチング部は、 $\theta$  パラメータによるマッチング、 $\rho$  パラメータによるマッチング、直線の交点によるマッチングを含んでいる。エッジ検出部では、Canny Edge Detection<sup>(8)</sup> を用いて入力画像からエッジ画像を得ている。直線検出部では、ハフ変換によって入力画像中の直線部を検出し、それぞれの直線はパラメータ  $(\theta, \rho)$  によって表される。マッチング処理を行う前にシーンの直線パラメータ空間を拡張する。拡張することによって検出対象の回転角  $\alpha$  が大きい場合でも検出できるようにしてい

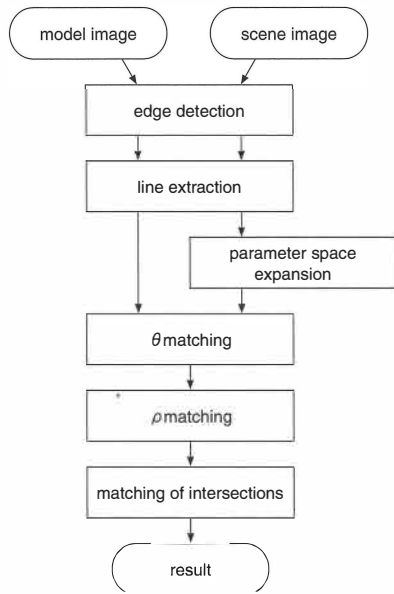


Fig. 4. Flow diagram of our method

る。 $\theta$  マッチングでは、シーンから選び出した直線の相対角度と対応するモデルの直線の相対角度を比較することによってマッチングを行う。 $\rho$  マッチングでは、式 (6) を用いて導出した直線の  $\rho$  パラメータの理論値と実際の値との偏差によってマッチングを行う。直線の交点によるマッチングでは、推定した検出対象の平行移動量、回転角、拡大率を用いて直線の交点座標の理論値と実際の値を比較することによってマッチングを行う。

### 3. 原理

〈3・1〉 ハフ変換による直線検出 提案する物体検出法では、Canny Edge Detection により得られたエッジ画像に対してハフ変換を行い直線部を検出する。ハフ変換は投票と多数決による直線検出法で、画像中にノイズが含まれている場合、また直線の一部が欠落している場合でも検出可能であるとされている。ただし、ハフ変換による直線検出法が安定して使用できるのはパラメータ空間上の十分に離れた長い直線群の場合であり、本論文では、パラメータ空間のピークを消し込むことによって直線検出の精度を高めている。

(1) 式に示すように、ハフ変換によってエッジ画像中のひとつの点は  $\theta$ - $\rho$  平面上のサイン曲線に変換される。x-y 平面中の一直線は複数の点からなり、 $\theta$ - $\rho$  平面に複数のサイン曲線を描くことができ、それらは一点で交わる。x-y 平面上の直線を見つけるには、ハフ変換を行った後、 $\theta$ - $\rho$  平面上で複数のサイン曲線の交点を見つけることになり、この処理をプログラムで実現するには、サイン曲線を描く代わりに二次元配列を用意し対応するセルをインクリメントする。このときサイン曲線の交点は投票数のピークに対応する。入力画像中に複数の直線が含まれている場合、二次元配列中に複数のピークが現れる。これらのピークを検出

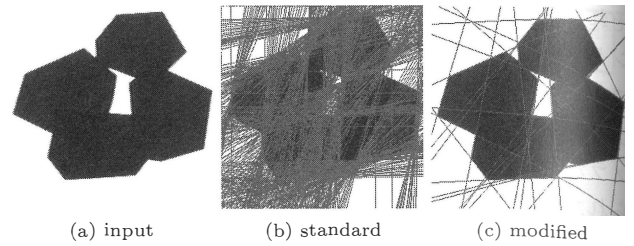


Fig. 5. Line Detection

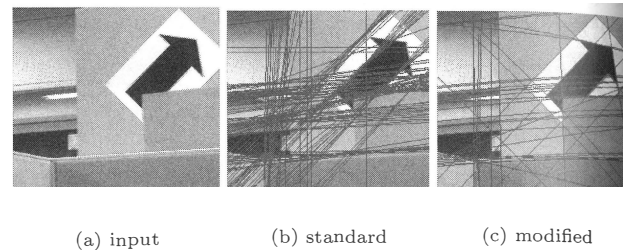


Fig. 6. Line Detection

する手法は多く提案されている<sup>(9)(10)</sup>。

一本の直線を構成する画素が  $n$  個の場合、それに対応する  $\theta$ - $\rho$  平面上の点を  $n$  本のサイン曲線が通り投票数は  $n$  となる。一般に直線検出では、投票数が与えられた閾値より大きい点を直線とするが、ここで課題となるのが、投票数がピークとなる点の周囲の点も大きな投票数を持つため、これらの点も直線として検出されてしまう。そのためピークの周辺を無視することも考えられるが、近接する別の直線がある場合、それが検出できなくなる。ピークとなる点の周囲の投票数が大きくなるのはピークを通るサイン曲線が寄与しているためであり、本方式ではピーク点を検出した後、そのピーク点を通る曲線を消して次のピーク点を探索している。具体的には以下の手順である。

- (1) 二次元配列から最大投票点を探し出し、直線部と判断する。
- (2) 最大投票点を通るサイン曲線上のセルをデクリメントする。
- (3) 二次元配列中に新たな最大投票点が現れる。

これらのステップを、最大投票点の投票数がしきい値以下となるまで繰り返す。通常のハフ変換による直線検出結果と  $\theta$ - $\rho$  空間でピークを消し込んだ場合の直線検出結果を、Fig.5 および Fig.6 に示す。

〈3・2〉 パラメータ空間の拡張 Fig.7(a) と Fig.7(b) はモデルとシーンのパラメータ空間である。ハフ変換での  $\theta$  の範囲は  $0 \sim \pi$  であり それ以上の回転量に対応するため、本手法では、Fig.7(c) に示すようにシーンのパラメータ空間を  $\theta$  方向に拡張している。この後の  $\theta$  パラメータによるマッチング処理ではモデルの直線を表す点  $P_1 \sim P_3$  とシーンの点との組み合わせを求めると、その際、モデルの点の方向の並びを考慮しており、例えば Fig.7(a) と Fig.7(c) おいてモデルの点  $P_1$  と  $2\pi$  に近いシーンの点  $Q'_4$  の組み合わせを選ぶとそれ以降の  $P_2, P_3$  の対応点の候補は  $2\pi$  よ

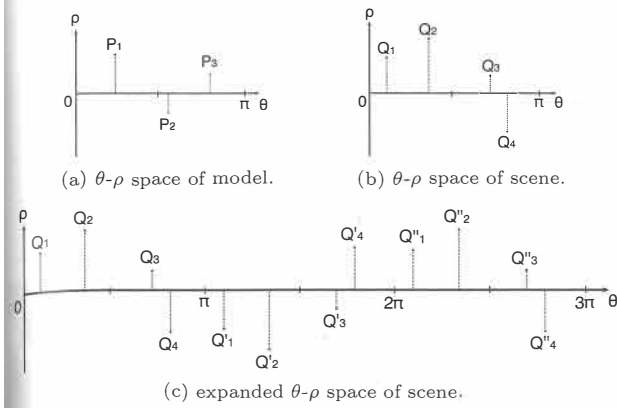


Fig. 7. Expanding θ-ρ space

大きい点  $Q'_1 \sim Q'_4$  との組み合わせとなるため、実際に求める回転量は  $0 \sim 2\pi$  であるがここでは 3 倍に拡張している。

シーンに  $n$  本の直線が含まれているとし、 $S_j$  を  $j$  番目の直線、 $\theta'_j$  および  $\rho'_j$  をそれぞれ直線  $S_j$  の  $\theta$  パラメータおよび  $\rho$  パラメータとすると、パラメータ空間の拡張は以下の式で表される。

$$S_{n+j} : \theta'_{n+j} = \theta'_j + \pi, \rho'_{n+j} = \rho'_j \quad (7)$$

$$S_{2n+j} : \theta'_{2n+j} = \theta'_j + 2\pi, \rho'_{2n+j} = \rho'_j \quad (8)$$

**〈3・3〉 θ パラメータによるマッチング** Fig.2(b) および Fig.3(b) に示したように、回転によって直線パラメータは  $\theta$  方向にシフトされるが直線間の相対角度は一定である。 $\theta$  パラメータによるマッチングではシーンからモデルと同数の直線を選び出し、それぞれの相対角度がモデルの直線間の相対角度と一致する直線の組み合わせをマッチングの解の候補とする。実際には、直線検出での量子化誤差を考慮してしきい値を設け、それぞれの相対角度に若干のずれがあってもマッチングの候補とする。また、解と判断された直線の組み合わせに関して、モデルとシーンで直線パラメータの  $\theta$  の差分を計算することによって回転角  $\alpha$  を得ることができる。

**〈3・4〉 ρ パラメータによるマッチング** シーン中の直線の  $\rho$  パラメータは (6) 式で表される。 $\rho$  パラメータによるマッチングでは、 $\theta$  マッチングで候補とされたモデルとシーンの対応する直線の組み合わせから検出対象の平行移動量および拡大率を推定し、直線の  $\rho$  パラメータの理論値と実際の値を比較することにより対応する直線を絞り込む。モデルの直線パラメータ  $(\theta_i, \rho_i)$  とシーンの直線パラメータ  $(\theta'_i, \rho'_i)$  が対応する場合、評価値  $E_\rho$  は式 (9) のように定義される。

$$E_\rho = \sum (\rho'_i - t_x \cos \theta'_i - t_y \sin \theta'_i - s \rho_i)^2 / s \quad (9)$$

ここで  $t_x$  および  $t_y$  はそれぞれ  $x$  方向および  $y$  方向の平行移動量、 $s$  は拡大率で、

$$\begin{bmatrix} \sum \cos \theta_i'^2 & \sum \cos \theta'_i \sin \theta'_i & \sum \rho_i \cos \theta'_i \\ \sum \cos \theta'_i \sin \theta'_i & \sum \sin \theta_i'^2 & \sum \rho_i \sin \theta'_i \\ \sum \rho_i \cos \theta'_i & \sum \rho_i \sin \theta'_i & \sum \rho_i^2 \end{bmatrix} \begin{bmatrix} t \\ t_y \\ s \end{bmatrix} = \begin{bmatrix} \sum \rho'_i \cos \theta'_i \\ \sum \rho'_i \sin \theta'_i \\ \sum \rho'_i \rho_i \end{bmatrix} \quad (10)$$

を解くことにより以下のように求まる。

$$t_x = \frac{D(CH - G^2) + E(FG - BH) + I(BG - CF)}{ACH + 2BFG - CF^2 - AG^2 - B^2H} \quad (11)$$

$$t_y = \frac{D(FG - BH) + E(AH - F^2) + I(BF - AG)}{ACH + 2BFG - CF^2 - AG^2 - B^2H} \quad (12)$$

$$s = \frac{D(BG - CF) + E(BF - AG) + I(AC - B^2)}{ACH + 2BFG - CF^2 - AG^2 - B^2H} \quad (13)$$

ただし、

$$\begin{aligned} A &= \sum \cos \theta_i'^2 & B &= \sum_{i=1}^n \cos \theta_i' \sin \theta_i' \\ C &= \sum \sin \theta_i'^2 & D &= \sum_{i=1}^n \sin \theta_i' \cos \theta_i' \\ E &= \sum \rho'_i \sin \theta'_i & F &= \sum_{i=1}^n \rho'_i \cos \theta'_i \\ G &= \sum \rho_i \sin \theta'_i & H &= \sum_{i=1}^n \rho_i \cos \theta'_i \\ I &= \sum \rho_i \rho'_i \end{aligned}$$

である。(9) 式に示した評価値  $E_\rho$  は、シーンの直線の  $\rho$  パラメータの理論値と実際の値の偏差で、ここでは評価値  $E_\rho$  がしきい値以下となる直線の組み合わせを候補とする。

**〈3・5〉 直線の交点によるマッチング** 物体検出の解として平行移動量、回転角、拡大率は  $\rho$  マッチングで求めた。しかし、コンピュータの量子化誤差などが原因で誤検出を行う場合がある。そこでマッチングの最終段階として直線の交点によるマッチングを行うことにより、より高精度な検出を実現している。モデルの直線  $M_p$  および  $M_q$  がそれぞれシーンの直線  $S_p$  および  $S_q$  に対応しているとする、 $M_p$  と  $M_q$  の交点も  $S_p$  と  $S_q$  の交点に対応する。これらの交点座標をそれぞれ  $(x_k, y_k)$ ,  $(X_k, Y_k)$  とすると、ここでの評価値  $E_c$  は次のように定義される。

$$E_c = \sum (x_k - X_k)^2 + (y_k - Y_k)^2 \quad (14)$$

ただし、

$$\begin{bmatrix} x'_k \\ y'_k \end{bmatrix} = s \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} X_k \\ Y_k \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (15)$$

直線の交点によるマッチングでは、評価値  $E_c$  をしきい値以下かつ最小となるシーンの直線の組み合わせを解とする。

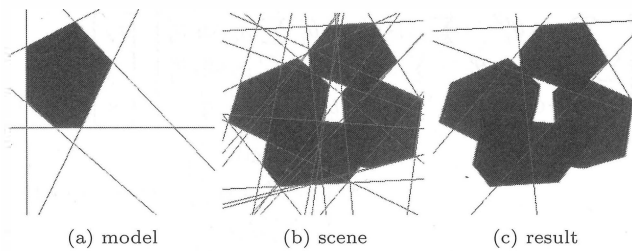


Fig. 8. Detection of polygonal shape

Table 1. Result of Fig.8

Item	Value
Translation $t_x$	-5.15 pixel
Translation $t_y$	261.93 pixel
Rotation angle	274.17 degree
Scale	1.12

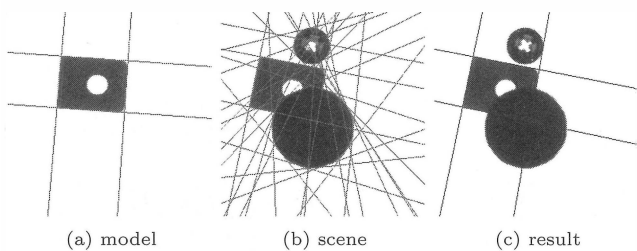


Fig. 9. Detection of square part

#### 4. 実験結果

本論文で提案した物体検出法を用いて、種々の入力画像に対して物体検出を行った結果を示す。各画像は左上を原点とし、大きさは  $256 \times 256$  pixel である。また、モデルの直線はハフ変換による直線検出を行った後、オフラインで登録した。

Fig. 8 に示すのは、厚紙を切り抜いて作成した多角形を対象物とした場合である。Fig. 8(a) はモデル画像で、6本の直線部をモデルとして登録した。Fig. 8(b) はシーン画像で、4個の多角形が含まれており、検出対象と他の多角形に重なりが生じている。Fig. 8(c) は検出結果で、モデルと対応するシーン中の直線の組を表示している。Table 1 に検出対象の移動量を示す。また、検出実験のしきい値は、 $\theta$  パラメータによるマッチングでは  $\pm 4$  [degree] の範囲、 $\rho$  パラメータによるマッチングでは評価値  $E_\rho$  をモデルの直線の数でマッチングでは評価値  $E_c$  を交点の数で割ったときの値が  $5$  [pixel] 以下とした。これらの値は以下の実験でも共通である。

Fig. 9 に示すのは、中央部に穴があり角が丸い四角形の産業部品である。Fig. 9(a) はモデル画像で、4本の直線部をモデルとして登録した。Fig. 9(b) はシーン画像で、検出対象の他に円形の部品とねじ穴の空いた部品が含まれている。Fig. 9(c) は検出結果で、モデルと対応するシーン中の

Table 2. Result of Fig.9

Item	Value
Translation $t_x$	11.93 pixel
Translation $t_y$	5.85 pixel
Rotation angle	-7.00 degree
Scale	1.01

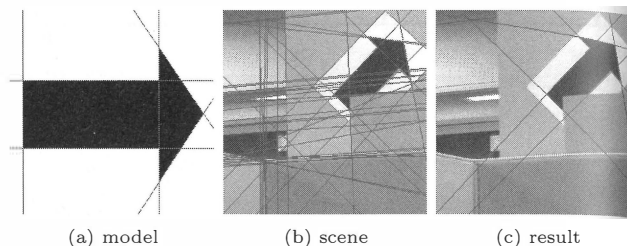


Fig. 10. Detection of arrow mark in indoor environment

Table 3. Result of Fig.10

Item	Value
Translation $t_x$	-29.46 pixel
Translation $t_y$	-275.54 pixel
Rotation angle	-134.33 degree
Scale	-2.11

直線の組を表示している。Table 2 に検出対象の移動量を示す。

Fig. 10 に示すのは、矢印が含まれる風景である。Fig. 10(a) はモデル画像で、6本の直線をモデルとして登録した。Fig. 10(b) はシーン画像で、モデルの矢印を印刷した紙を風景中に張った状態を入力画像とした。Fig. 10(c) は検出結果で、モデルと対応するシーン中の直線の組を表示している。Table 3 に検出対象の移動量を示す。

#### 5. 処理時間

第4章で示した結果以外に、いくつかの画像を用いて実験を行った。それぞれの画像においてマッチングに要した時間は、シーンやモデルの直線の数に関わらずおよそ  $15 \sim 20$  ms であった。しかし、直線検出に要した時間は入力画像中のエッジ点の数によって変化した。その結果を Fig. 11 に示す。実験に用いたコンピュータの仕様およびプログラミング言語は Table 4 の通りである。

マッチングの処理時間は、シーンに含まれる直線の数に依存するが、 $\theta$  マッチングで  $\theta$  軸の相対位置を基にほとんどの組み合わせが除かれるため、最も時間を要すると思われる (9) 式~(15) 式の計算が、シーン中の直線の数が多い場合でも数回しか実行されず、処理時間に差がなかったと考えられる。一方、直線検出処理は、Canny Edge Detection によるエッジ点の抽出、(1) 式によるハフ変換、そして  $3 \cdot 1$  節で述べたピーク点の検出がある。いずれも処理時間がエッジ点の数に依存しており、特に (1) 式の計算で

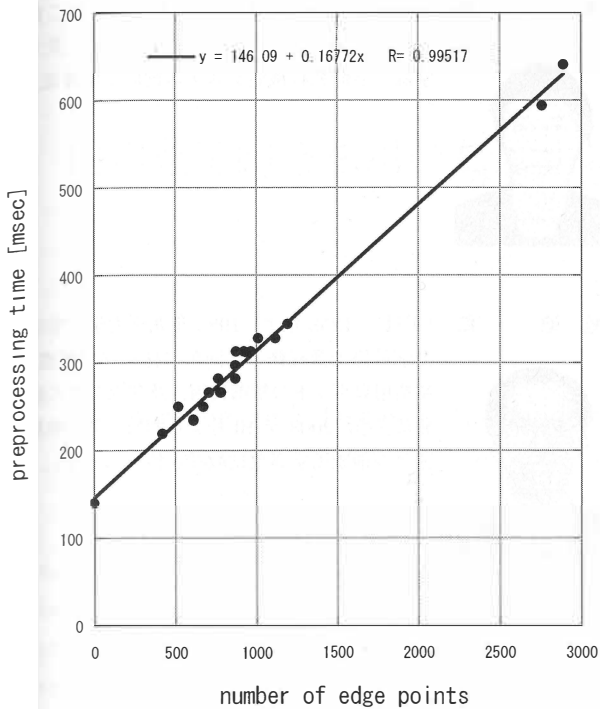


Fig. 11. Preprocessing time

Table 4. Computer environment

Item	Spec
OS	Microsoft Windows XP Professional SP2
CPU	Intel Pentium4 Processor 2.4GHz
Memory	1GB RAM
Programing language	Microsoft Visual Basic .NET 2003

は一つのエッジ点で 360 回の乗算を行っており、処理時間の大半を占めていると考えられる。

実用化を想定したとき、移動ロボットや産業用ロボットなどの用途では、ロボットの動作速度を考慮すると処理時間は 500[msec] 以下で十分であると考えられる。前述のように処理時間はエッジ点の数に依存しており、Fig.10 に示した通常のシーンではエッジ点の数が 3000 近くあり、600[msec] 以上かかっている。現在この処理は Microsoft 社の Visual Basic .NET 2003 を用い、中間言語形式で実行しているが、C++ に移植することによって実行ファイル形式になるため、処理時間は経験上、半減以下になることが見込める。

## 6. まとめ

本論文では、産業用ロボットへの応用として、検出対象の直線部に注目した物体検出法の提案および検討を行った。

まず、直線検出法としてハフ変換を改良し  $\theta$ - $\rho$  空間でピークを消し込む方法を述べた。通常のハフ変換による直線検出では、入力画像中の一本の直線部に対して複数の直線が検出され、また短い直線部や近い位置にある直線部を検出できないことがあった。一方、消し込みを行った場合には、一本の直線部に対して多くとも三本程度の直線しか検出されず、また通常の方法では検出できなかった短い直線部や

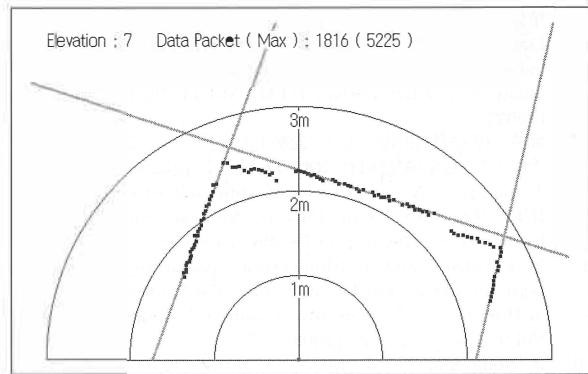


Fig. 12. Sensor data of mobile robot

近い位置にある直線部も検出することができた。消し込みを行う方法は通常の方法と比較して、高精度な直線検出ができるといえる。

更に、本論文で提案した直線パラメータを用いた物体検出法により、検出対象に回転、拡大、隠蔽が生じている場合でも検出を行うことができた。また、バックライト等の撮影条件が整っていない風景を入力画像とした場合においても物体検出を行うことができた。

現在この検出方式では、モデルの直線がシーン中にすべてであることを前提としており、一つの直線が完全に隠れている場合、検出ができない。これはマッチング処理においてモデルの全ての直線に対して、シーン中の直線に対応させて評価しているためである。今後の課題の一つとして、(14) 式で与えられる評価値において満足する結果が得られない場合は、モデルの直線の一つだけ無視して、同様なマッチング処理を実行する処理方式を検討中である。これによって満足する評価値が得られた場合、無視した直線がシーン中では隠れているものと解釈できる。

本論文で提案した物体検出法は産業用ロボットを対象としているが、移動ロボットへの応用についても検討している。Fig.12 に示すように、移動ロボットに使用されるレーザを用いた位置検出センサーは、ロボット周辺の状況を図中の点で示すように感知する。このセンサーデータから、本論文で提案した物体検出法を用いてロボット周辺の壁などの直線形状を検出できると考えている。

(平成 19 年 3 月 30 日受付, 平成 19 年 9 月 14 日再受付)

## 文 献

- (1) 江尻正員監修:「画像処理産業応用総覧上巻 基礎・システム技術編」, フジテクノシステム (1994)
- (2) 江尻正員監修:「画像処理産業応用総覧下巻 応用技術編」, フジテクノシステム (1994)
- (3) D.H. Ballard: “Generalizing the Hough Transform to Detect Arbitrary Shapes”, Pattern Recognition, Vol.13, No.2, pp.111-122 (1981)
- (4) Paul V. C. Hough: “Method and means for recognizing complex patterns”, U.S. Patent 3069654 (1962)
- (5) 和田俊和・松山隆司:「Hough 変換に基づく図形検出方法の新展開」,

---

情処学誌, Vol.36, No.3, pp.253-263 (1995)

- (6) D.C. Pao, H.H. Li, and R. Jayakumar: "Shape Recognition using the Straight Line Hough Transform: Theory and Generalization", *IEEE Trans., PAMI*, Vol.14, No.11, pp.1076-1089 (1992)
- (7) 坂井 厚・野村由司彦:「ハフ変換平面を利用した相似変換画像のマッチング」, 信学論 (D-II), **J80-D-II**, 11, pp.2969-2975 (1997)
- (8) J. Canny: "A computational approach to edge detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.8, No.6, pp.679-698 (1986)
- (9) Y. Furukawa and Y. Shinagawa: "Accurate and robust line segment extraction by analyzing distribution around peaks in Hough space", *Computer vision and image understanding*, Vol.92, No.1, pp.1-25 (2003)
- (10) J. Song and M. R. Lyu: "A Hough transform based line recognition method utilizing both parameter space and image space", *Pattern recognition*, Vol.38, No.4, pp.539-552 (2005)
- (11) T. Okuzono and H. Wakizako: "A matching method of straight lines in  $\theta$ - $\rho$  space for object detection", In *AISM2006 Second Asia International Symposium on Mechatronics* (2006)

奥 苑 大 成 (学生員) 1983 年生。2005 年 3 月九州工業大学工学部電気工学科卒業。同年 4 月九州工業大学大学院工学研究科機能システム創成工学専攻入学。



脇 迫 仁 (正員) 1958 年生。1983 年九州大学工学部電子工学科修士課程修了。同年 (株) 安川電機入社。産業用ロボットの制御技術, 及びセンサの研究開発に従事。2003 年 10 月より九州工業大学機能システム創成工学専攻助教授。博士 (工学)。

