【論文】

# ロボットの視覚処理に関する研究

黒野　繁*1、荒牧重登*2 、聶　伝洋*3

# Study on the Image Processing for Robots

Shigeru Kurono, Shigeto Aramaki, Joh Denyou

**Abstract:** Up to this time, we have developed a real time image processor called "OIP:Outline Image Processor"[1] and utilized it for many robot application systems[2] in our labolatory. But,OIP has several defects such as being short of reliabilities in the circumstances with changing brightness and being weak in noises. In this study,we developed a reliable real time image processing system using a ITV color camera in the market. In this system,we introduced several processing methods such as “exclusive image processing”, elimination of noises by "labeling process","outline image processing" and "edge detecting process" to the images input from the ITV color camera. Furthermore, we constructed a objects handling system with an industrial robot and the image processing system mentioned above. In this paper, we illustrate the outline of software of image processing system and objects handling system.

**Keywords:** exclusive image processing,labeling process,outline image processing

## 1．はじめに

我々の研究室ではこれまでにアウトラインプロセッサ（ＯＩＰ：Outline Image Processor）と呼ぶ画像処理装置[1]を開発し、各種ロボットのリアルタイム制御に応用してきた[2]。ＯＩＰはモノクロ画像の輪郭情報（アウトライン）をシリアル通信によりホストコンピュータに送信するもので、使い勝手の良いリアルタイム画像入力装置であった。しかし、この装置は背景の明るさの変動やノイズに弱いということと、対象物の切り分け（Separation ）が難しい（不可能ではないが、信頼性が低い）という欠点があった。本研究では、市販のＩＴＶカラーカメラを用いて入力した画像に後述する「排他的画像処理」、ラベリング処理による「ノイズ除去」、「アウトライン処理」および、「エッジ検出処理」等の画像処理を施すことにより、信頼性の高い画像処理システムを開発した。このシステムはＩＴＶカメラが捉えた画像から多数の対象物の位置、形状、寸法をリアルタイムで認識することができる。さらに、本研究ではこの画像処理装置を用いて工業用ロボットによるハンドリング制御を行い、本画像処理装置の実用性を確かめたので報告する。

＊１本学電気情報工学科　＊２福岡大学工学部
＊３本学大学院電気工学専攻修了

## 2．実験システム全体の構成

実験システム全体の構成を図１に示す。今回使用したＩＴＶカラーカメラは㈱ソニー製の

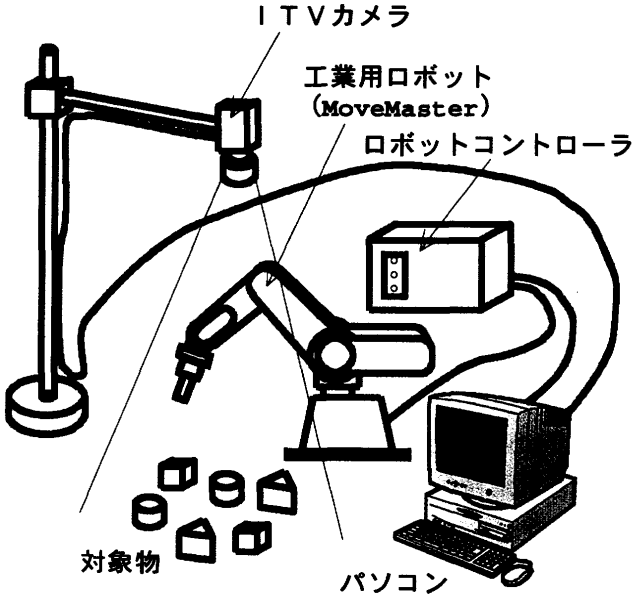



図１　実験システム全体の構成

ＥＴＶ－３０、画像入力装置は㈱マイクロテクニカ製のＭＴＰＣＩ－ＤＣボードで、ファクトリオートメーションでの目視検査の自動化等で多用されているボードである。工業用ロボットは㈱三菱電機製の MoveMasterEX（５自由度＋ハンド)、パソコンは㈱ＩＢＭの NetVista (Pentium Ⅳ，メモリ 512MB) である。なお、パソコンのＯＳは WindowsXP、画像処理に用いた言語はＧＵＩ（Graphical User Interface）に特徴のある Visual Basic Ver.6 である。

## ３．画像処理全体の流れ

理解を容易にするために、まず、画像処理全体の流れをフローチャートで図２に示し、簡単な対象物に関する画像処理の具体例を図３に示す。以下、それぞれの画像処理について説明する。

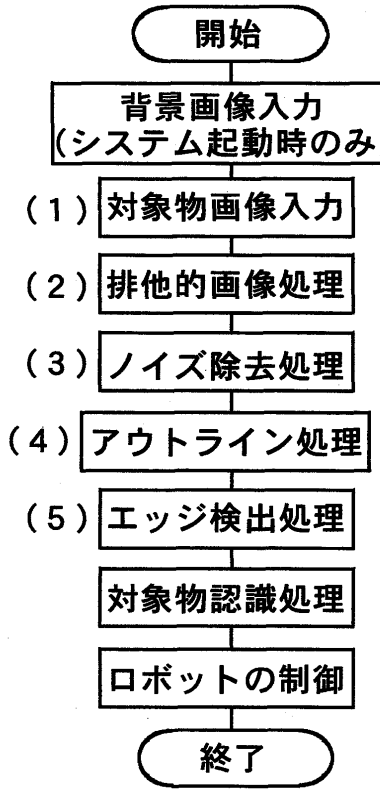
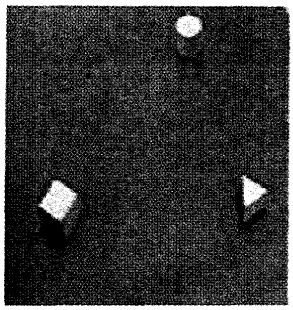



図２　画像処理全体の流れ

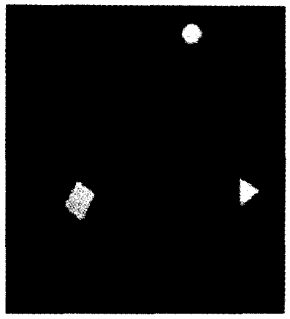
## ４．排他的画像処理

排他的画像処理とは背景画像（Buffer0）と対象物画像 (Buffer1) の第ｉピクセルのＲＧＢ成分をそれぞれ (r0i,g0i,b0i)、(r1i,g1i,b1i) としたとき、各成分の差の２乗和ｄを次式により求め、

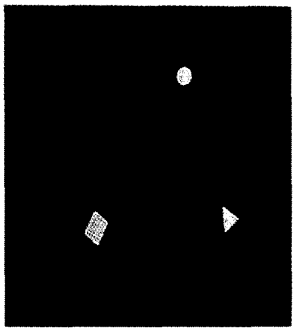
（１）**対象物の画像入力**



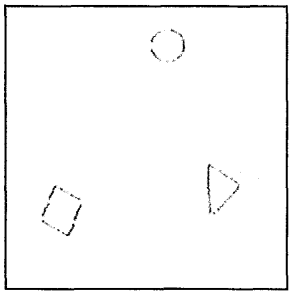
（２）**排他的画像処理**
背景画像と対象物入力画像との排他的画像処理（影や汚れによるノイズを含む）



（３）**ノイズ除去**
２値化とラベリングによるノイズ除去



（４）**アウトライン**
画像のアウトラインを求める



（５）**エッジ検出処理**
対象物が円のときは中心と半径、多角形のときは各頂点の座標を求める。右の図はこれらの情報を元にグラフィック関数で描画したものである。

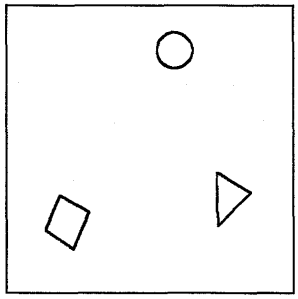


図３　画像処理の具体例

$d = (r1i-r0i)^2+(g1i-g0i)^2+(b1i-b0i)^2$　・・・(4.1)

この値がある閾値より小のときは背景、つまり対象物がない、大のときは対象物があると判断する訳である。具体的には、対象物がないときはＲＧＢバッファ(Buffer2)のＲＧＢ成分を

$r2i=g2i=b2i=0$　・・・・・・・・(4.2)

として背景を黒で塗りつぶす。このとき後の処理に備えて２値バッファ(BBuffer1)の第ｉ成分B1i に 0 をセットしておく。対象物があるときは、

$r2i=r1i$ , $g2i=g1i$ , $b2i=b1i$　・・・・・・(4.3))

として対象物のＲＧＢ成分をいれる。このとき、B1i には 1 をセットしておく。この処理によって図３（１）の画像から、同図（２）の画像がBuffer2 に得られる。一般にこの画像には対象物の汚れとか影の影響で多くのノイズが含まれる。

## ５．ノイズ除去

排他的画像処理の後に含まれるノイズを上記２値バッファ(BBuffer1)で表すと、例えば図４のようになる。

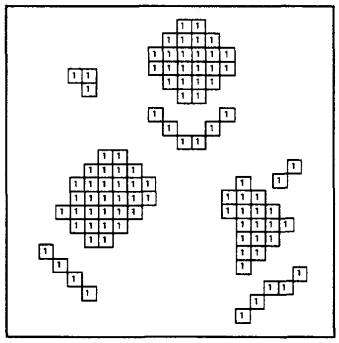


図４　２値画像におけるノイズ（例）

これらのノイズを除去するためにラベリングと呼ばれる処理を行う。ラベリング処理は、画像内に多数のオブジェクトが存在する場合、対象とする領域を識別する手法として有効である。この処理では、連結している画素に同じラベル（番号）を付加することで複数の領域をグループとして分類することができる。まず、２値画像において、ピクセルの値が１である領域について、同一の対象物として連結した領域を割り出して、それに１から始まる番号を付ける。異なる対象物にはプラス１しながら順次番号を付けていく。実際のプログラムでは、図５に示すように現在のピクセルを中心とした8近傍を再帰的に探索して値が１のピクセルがなくなるまで処理を行う。このとき、対象物の特徴量として、次の３つの情報を獲得しておく。

（１）同一の対象物と見なせる領域のピクセルの数（面積：Area）
（２）領域の縦横比（Aspect）
（３）領域を囲む矩形の面積と実際の面積との比（密度：Density）

図４の２値画像に対してラベリング処理した結果を図６に示す。

| | | |
|---|---|---|
| (x-1,y+1) | (x,y+1) | (x+1,y+1) |
| (x-1,y) | 基準点 (x,y) | (x+1,y) |
| (x-1,y-1) | (x,y-1) | (x+1,y-1) |

図５　８近傍の図

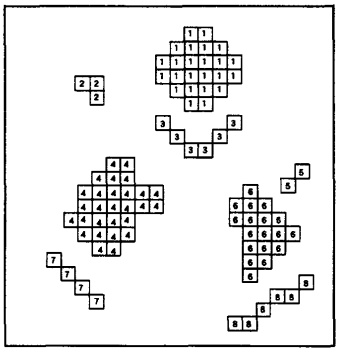


図６　図４の画像にラベリング処理した結果

取り扱う対象物の特徴量として、

Area ＞ 10 、 Aspect ＞ 0.5 、 Density ＞ 0.2

等と設定しておけば対象物か否かが判定できる。また、排他的画像処理した後の画像（図３（２））の色情報も対象物の判定に利用できる。

上記特徴量の条件をすべて満足する領域だけを残せば図７に示すようにノイズを除去できる。

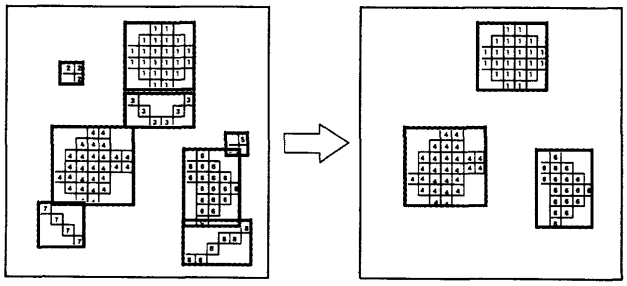
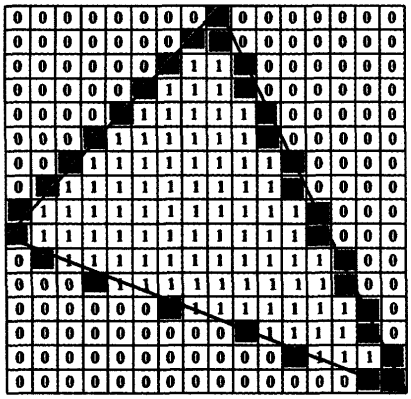


図７　特徴量判定によるノイズ除去

## 6．アウトライン処理

前ステップまでの処理で対象物が存在する領域が得られているので以後の処理は対応する領域のピクセルについてのみ行えばよい。ここでは、個々の対象物の形状、例えば、円のときは中心の座標と半径、多角形のときは各頂点の座標等を求めるため対象物のアウトライン（輪郭）の座標を求める。具体的には、図８に示すような２値画像について、すべての水平走査線（ｙ座標）に対して０から１に変化するピクセルのｘ座標（ｘ１）と１から０に変化するピクセルのｘ座標（ｘ２）のテーブルを作成する。（表１参照）アウトライン処理結果の例については図３（４）を参照されたい。



■ アウトライン(左)　■ アウトライン(右)

図８　対象物のアウトライン

| 配列 インデックス | y | x1 | x2 |
|---|---|---|---|
| 0 | | | |
| 1 | | | |
| 2 | | | |
| ⋮ | | | |

図９　アウトライン座標のテーブル

## 7．エッジ検出処理

対象物の形状認識の最終段階として、特徴点の座標を求める。その前にまず、対象物の形状判別をする必要がある。その方法としてパターンマッチングがあるが、これは取り扱う対象物の種類とサイズが多様化するとコンピュータの計算負荷（特に、画像の回転とマッチング処理）が大きくなり、リアルタイム性が悪くなる。本研究では、リアルタイム性を優先するため、アウトライン座標の標準偏差により形状判定することとした。三角形を例として図１０で説明する。まず、領域の中心座標 $(G_x,G_y)$ を求め、そこからすべての点（$(x_i,y_i)$ :i=1 ～ N）までの平均距離 $\bar{r}$ を求める。

$$\bar{r}=\sqrt{\frac{\sum_{i=1}^{N}\{(x_i-G_x)^2+(y_i-G_y)^2\}}{N}} \quad \cdots\cdots (6.1)$$

次に標準偏差ｅを求める。

$$e=\sqrt{\frac{\sum_{i=1}^{N}(d_i-\bar{r})^2}{N}} \quad \cdots\cdots (6.2)$$

実際のアウトラインデータで標準偏差を求めた結果、次のような閾値で判別できることが分かった。

円：e<3 、四角形：3<e<10 、三角形：e>10

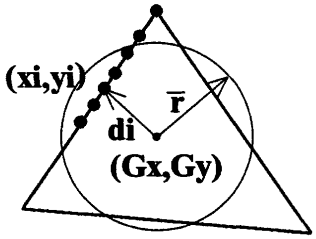



図１０　標準偏差の説明図

さて、対象物の形状が判定されると、その特徴点の座標と寸法を求めるのは比較的容易である。

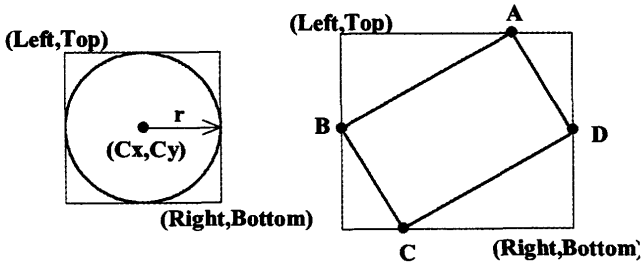
円のときは、中心の座標と半径ｒは

Cx= (Left+Right) /2, Cy= (Top+Bottom) /2

r= (Right-Left) /2

で求まる。（図１１（ａ）参照）

長方形または正方形のときは、図１１（ｂ）の頂点Ａ，Ｃの座標は図９のテーブルの先頭と最後の点として求まる。Ｂ，Ｄの座標は同テーブルで x1 が最小の点と、x2 が最大の点として求まる。




（ａ）円　　（ｂ）長方形

図１１　円と長方形の特徴点

次に、３角形の場合は図１２のように２つの見え方があるが、これは頂点Ｂ（またはＣ）が対辺ＡＣ（またはＡＢ）のどちら側にあるかで判断できる。同図（ａ）の頂点Ａ、Ｃ、（ｂ）の頂点Ａ、Ｂは長方形の場合と同様に図９のテーブルの先頭と最後の点として求まる。図１２（ａ）の頂点Ｂの座標は同テーブルx1が最小の点、（ｂ）のＣの座標はx2が最大の点として求まる。ただし、円以外の場合に、上記の方法で特徴点の座標を求める前に、図１３の様に各辺が水平または垂直となるケースをチェックするアルゴリズムが必要であるが、これも図９のテーブルを用いれば可能である。

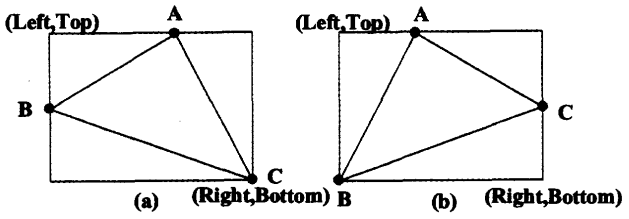



図１２　三角形の特徴点

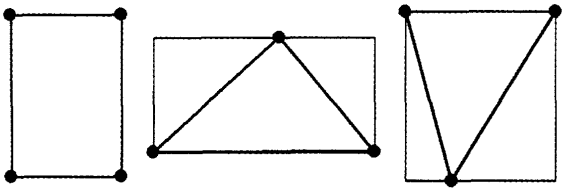


図１３　三角形または四角形の特殊ケース

## ７．ロボット制御システム

### ７．１　座標変換

ＩＴＶカメラからの画像を処理して得られた対象物の位置、姿勢、寸法を用いてロボットによるハンドリングを行うときに問題となるのがカメラ座標とロボット座標との座標変換の問題である。図１４にロボット座標 $O_r$-$x_r$-$y_r$ とカメラ座標 $O_c$-$x_c$-$y_c$ との関係を示す。本研究では、比較的簡単で精度の高い座標変換法を考案し、ハンドリングロボットに応用し、満足できる結果を得たので報告する。図１５に示すように、ロボットの作業空間とカメラの視界が重なる領域にほぼ４角形の領域を想定し、その４つの頂点 (P1,P2,P3,P4) の座標をカメラ座標 (Xci,Yci) (i=1 ～ 4) とロボット座標 (Xri,Yri,Zri,Pri,Rri) (i=1 ～ 4) でそれぞれ求めておく。ここで、ロボット座標の (X,Y,Z) ) はロボットハンド先端の直交座標、(P,R) はハンドのピッチ角とロール角を表す。これら４つの代表点のロボット座標とカメラ座標を求める簡単で確実な方法は、まず、ロボットに予めP1 ～ P4 の４点の位置を記憶させておいて、次に、例えば円柱の様なブロックをロボットに持たせて上記４つの位置へ置かせる。その４つのブロックをカメラで見て、その中心座標を求める訳である。これらの作業を自動化するプログラムを作成しておけば、ロボットとカメラの位置関係が変更されたときも容易に再現することができる。さて、実際にロボットを制御するときに得られる情報は図１５の点Ｑの座標 (Xcq,Ycq) である。これからロボット座標 (Xrq,Yrq,Zrq,Prq,Rrq) を求めるのがここでの課題である。具体的には４点 P1 ～ P4 のロボット座標とカメラ座標を用いた補間法である。つまり、点 Q を通る Xr 軸に平行な直線と直線 P1P4 との交点を Q1、直線 P2P3 との交点を Q2 とし、Q を通る Yr 軸に平行な直線と直線 P1P2 との交点を Q3、直線 P3P4 との交点を Q4 とすると座標変換の手順は次の通りである。

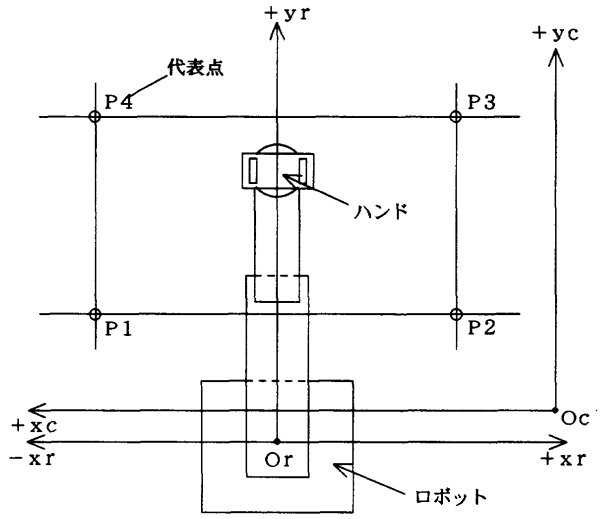



図１４　ロボット座標とカメラ座標

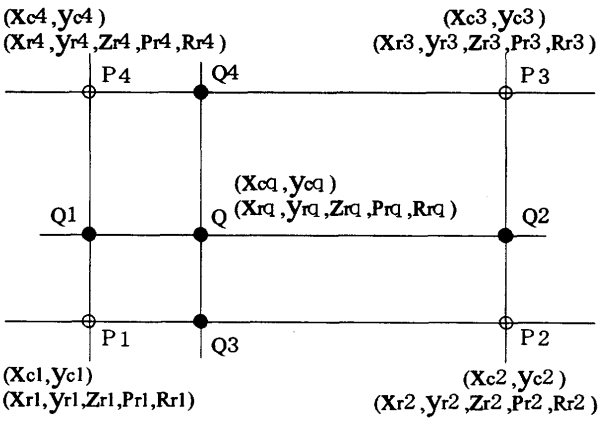



図１５　代表点の座標の表記方法

（1）点 P1、P4 と Ycq による補間で Q1 のロボット座標とカメラ座標を、例えば、Xrq1 は、

$$Xrq1=Xr1+(Xr4-Xr1)\frac{Ycq-Yc1}{Yc4-Yc1} \quad \cdots\cdots (7.1)$$

点 P2、P3 と Ycq による補間で Q2 のロボット座標とカメラ座標を

$$Xrq2=Xr2+(Xr3-Xr2)\frac{Ycq-Yc2}{Yc3-Yc2} \quad \cdots\cdots (7.2)$$

のように求める。補間式は以下同様。

（2）次に、点 P3、P4 と Xcq による補間で Q4 のロボット座標とカメラ座標を、点 P1、P2 と Xcq による補間で Q3 のロボット座標とカメラ座標を求める。

（3）次に、点 Q1、Q2 と Xcq による補間で Q のロボット座標を、点 Q3、Q4 と Ycq による補間で同じく Q のロボット座標を求める。この両者はほぼ、近い値になるが念のため平均を取って座標変換が終了する。本研究では、簡単のため Q1、Q2 から Q の X 座標 (Xrq)、Q3、Q4 から Q の Y 座標 (Yrq) を計算してロボットを制御しているが非常に精度の高い座標変換が可能となった。

なお、ロボットの他の自由度 P（ピッチ）と R（ロール）についても同様の補間処理を行うが、Z（高さ）については一定としているので変換は不要である。また、補間は内挿のみでなく外挿も同じ処理で行えるのでロボットの作業領域は図１５の矩形領域の外でもハンドリングできることは言うまでもない。補間法はカメラ画像の中心からずれた位置でのひずみの影響を軽減するという特徴がある。

## 7．2　ハンドの姿勢制御

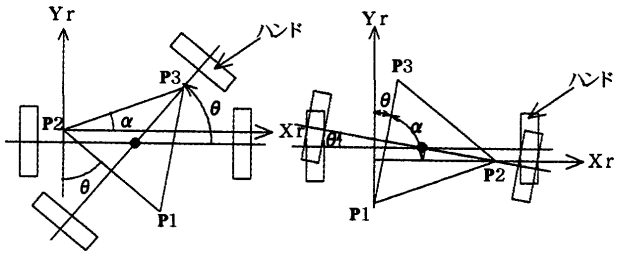
本研究で使用した対象物は円柱、正三角柱、正四角柱のブロックである。これらの対象物を動かさないで、垂直方向からその中心を正確につかむためには、対象物が置かれた方向に応じて、ハンドのロール角 (Rr) を制御する必要がある。図１６は、三角柱をつかむときのハンドのロール角制御の方法を説明したものである。カメラから三角柱の形状を認識したとき、3点P1、P2、P3 の座標の取り込み方には、図１６の (a), (b) のような2つの場合がある。つまり、P1、P2のXr座標をX1、X2としたとき、X1＞X2の場合とX1＜X2の場合がある。同図において、三角柱の方向 $\alpha$ が 0° ～120° の範囲で変化するとき、回転させるべきロール角 $\theta$ の大きさは

$\alpha$=0° ～30° のとき、　$\theta$= －($\alpha$+30° )

$\alpha$=30° ～120° のとき、$\theta$= 90° －$\alpha$

で良いことが分かる。

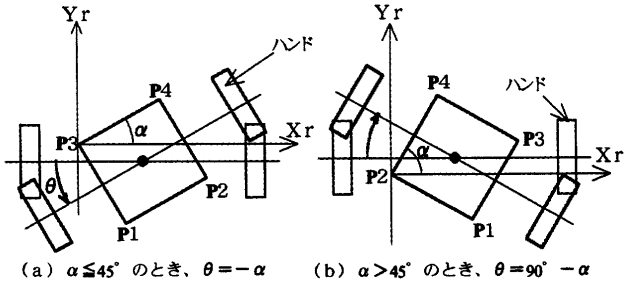
ただし、$\theta$ には正負があり、正のとき右回り、負のとき左回りを意味する。また、ロール角の回転角度は必ず60° 以下となるように制御される。




(a) X1＞X2, α=0° ～30° のとき、θ＝－(α＋30° )　(b) X1＜X2, α=30° ～120° のとき、θ＝90° －α

図１６　三角柱のハンドリング

四角柱の場合は、向い合った2辺が平行であるため、ロール角の制御は比較的簡単である。このとき、回転角度は45° 以下となるように制御される。(図１７ 参照)




(a) α≦45° のとき、θ＝－α　(b) α＞45° のとき、θ＝90° －α

図１７　四角柱のハンドリング

## 8．まとめ

本研究では、市販のＩＴＶカラーカメラを使って対象物の形状、位置、姿勢等の視覚情報を入力し、工業用ロボットでそれを正確にハンドリングするシステムを構成し、実験により視覚情報処理の実用性を確認した。ここで導入した排他的画像処理、ノイズ除去、アウトライン処理等の手法は一般的な画像処理の前処理として有用である。また、本論文で提案した、ロボット座標とカメラ座標の座標変換方式は比較的簡単であるが十分実用的な方法である。

［参考文献］

1) S.Kurono et al."The Development of an Outline Image Processor" Proceedings of IECON '93 in HAWAII （IEEE） vol.3 of 3 (pp.2251 ～ 2256)

2) 黒野他「高速リアルタイムビジョンの応用」本学工学部研究報告第 32 号 (pp.113 ～ 118)