

組立て物体の構造の理解

溝口 博

株式会社東芝 総合研究所

本報告では、ブロック玩具レゴの組立てを行うロボットにおける視覚処理について述べる。このロボットは、組立てられた見本のレゴを見てその3次元構造を認識し、同じものを作る手順を考え出して、組立てを行う。

この問題では、部品となる個々のブロックのモデルは既知であるが、部品の組み合わせや個数は任意であるため、見本に関する先験的なモデルは利用できない。このため、視覚ではモデルなしで見本の3次元構造を認識する必要がある。本報告では、複数の方向から見本を観測し、結果を高さごとの階層に分類し、各階層ごとの部品の2次元的な配置を考えるという方法によって上記の問題を解決した。

Three-dimensional structure recognition for assembled blocks

Hiroshi MIZOGUCHI

Toshiba Research and Development Center
1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki, 210, JAPAN

This paper describes visual processing techniques for a robot which can assemble toy LEGO blocks. The robot recognizes the three-dimensional structure of a sample assembly of LEGO blocks, generates task sequences to build a replica of the sample and does this using dextrous hands.

In this task, models of component blocks are previously defined, but the model of the assembled blocks cannot be given in advance. The problem to be solved by vision is to recognize the three-dimensional structure of the sample without using an a priori model of it.

In our method, the sample is observed with stereo cameras from several viewpoints, in order to obtain front, back and side views. Vertices and edges are extracted from observed images and their descriptions are divided into layers, in terms of their height. At each layer, a two-dimensional composition which satisfies those descriptions is inferred, according to the component models. As a result, the three-dimensional location and orientation of components is determined.

1. はじめに

知能ロボットの生産ラインへの応用を考えた時、一つの理想形として思い浮かべるものは、いちいち組立て手順を教えてやることなく、完成品の見本を見せてやるだけでロボット自らがその構造を理解し、組立て手順を考え出して作業を行う姿であろう。我々は、この姿を最終的な目標として念頭におきつつ、ブロック玩具「レゴ」の組み立てを題材とした、知能ロボットのプロトタイプを開発した。これは、組立てられたレゴの構造を視覚を用いて認識し、それと同じ構造の物を組立てるといったものである。本報告では、開発した知能ロボットプロトタイプにおける視覚機能について、物体の3次元構造の認識法を中心に述べる。

2. 背景および方針

2.1 背景

今回、我々が設定したレゴ組立て作業のシナリオは、以下の通りである。ロボットの前に、ばらばらのレゴブロックを部品として置いておく。また、組立てられたレゴを完成品の見本としてパレットの上に載せ呈示する。この状況でロボットは、

- 1) 見本の3次元的な構造を視覚を用いて認識し、
- 2) 同じ構造の複製を組立てるための作業手順を考え出し、
- 3) 考えた計画に基づきマニピュレータを制御

して実際に組立てを行なう。

図1に、この知能ロボットシステム全体の構成と視覚の位置付けを示す。ここで視覚の役割は、組立てられたレゴの見本を見てその3次元構造を推定し、3次元構造の記述を生成して知能部に渡すことである。組立て手順を考え出す知能部は一種の問題解決器であるから、ここでの視覚の役割は、問題解決器に与えるべきゴールの記述を生成することである、といいかえることができる。

2.2 制約および条件

処理の考え方を述べる前にまず、対象であるレゴに関する制約と問題を解くために設定した条件について述べる。これらの制約および条件は以下の通りである。

- 1) 見本の構造、形状は未知。
- 2) レゴブロックの形状は2種、既知(図2)。
- 3) レゴブロック同士は互いに面が直角、又は平行。
- 4) ブロック間隔は突起間隔 d で離散化(図3)。
- 5) 両眼ステレオを利用。斜め上方から俯瞰。
- 6) 見本はパレット上に固定。
- 7) パレットはテーブル上で平行移動、回転移動可。
- 8) 見本の分解、持ち上げは不可。

1)~4)で述べているように、この問題では、個々の部品(レゴブロック)の3次元形状等は予めわかってはいるが、全体の組み合わせはわからない。言いかえれば、呈示される見本の形状に関しては先験的なモデルが存在し得ず、これを利用することができない。

このことに関連して、8)の制約条件が生ずる。これは、マニピュレータが、形状が既知の物体しかハンドリングできないことに起因している。物体を分解したり、持ち上げたりするためには、対象とする物体の位置・姿勢を目標値としてマニピュレータに与える必要がある。しかし、ここでは見本の形状が未知であるため、適切な目標値を設定することができず、よって見本をマニピュレータでハン

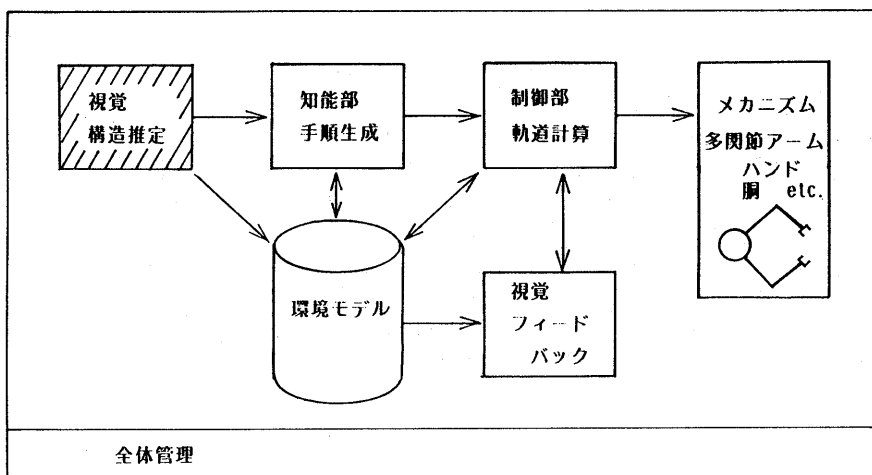


図1. 開発した知能ロボットのシステム構成

ドリリングすることができない。ただし、6)、7)の条件からパレットを動かすことは可能であるから、パレットを動かすことによって、テーブル面上に拘束した間接的な移動を行うことは可能である。

2.3 処理方針

レゴブロックは、上面の突起を無視すれば図4に示すような直交多面体として取扱うことができる。問題はこのような画像から全体を構成する各々のブロックの、1)種類、2)位置、3)姿勢を求めることである。レゴは高さ方向の寸法がブロックの高さ h で離散化されるため、図5のような各段ごとの断面図を考えることができる。このような断面図に相当する情報が得られれば、各々のブロックに関して種類、位置、姿勢を求めることができ、結果として見本全体の3次元的な構造が得られることになる。図5のような断面図を「階層別マップ」と呼ぶ。ここで階層別マップ中の格子点は原画像中では縦方向のエッジに対応し、格子点間を結ぶ線は斜め方向のエッジに対応する。よって、原画像の中から特定の方向のエッジのみを選択的に抽出し、個々のエッジの両端点の3次元座標を測定して高さ方向の成分で分類することにより階層別マップが得られる。

当然のことながら、一方方向から見本を眺めただけでは、裏側の部分は見本自身の陰になるため観測できない。そこで全体を眺めるために、視点を変えて複数の方向から観察を行う。視点の変更は、カメラを移動する代わりに、見本をのせたパレットをロボットで移動させることにより等価的に実現する。現状のインプリメンテーションでは、見本の前後左右像を得るために図6のSE、NE、NW、SWの4方向から観測を行っている。

図7に、視覚における処理の流れを示す。処理は次のような順に進む。

- 1)ある方向からみた見本のステレオ画像を撮る。
- 2)左右それぞれの画像について頂点とエッジを抽出し、2次元的な記述を作る。
- 3)左右の記述同士に対応付けを行い、各頂点の3次元座標を求める。3次元的な記述を作る。
- 4)見る方向を変えて1)~3)を繰返し、各方向ごとの記述を求める。
- 5)観測方向ごとに座標系の異なっている測定値に対して座標変換を施し、統一した座標系表現に揃える。
- 6)頂点とエッジを高さ成分で分類し、断面の外形線情報を得る。
- 7)断面の外形線情報からブロックの存在を認識し、階層別マップを作成する。

これら一連の処理の結果として各ブロックの位置、姿勢、種類が得られ、見本の構造に関する記述が得られることになる。以下では、1)~6)までの処理とデータ表現について3で、7)については4で述べてゆく。

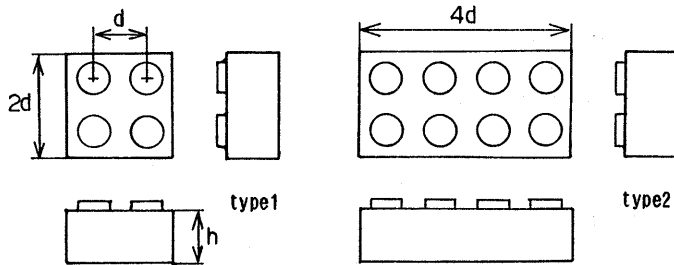


図2. レゴブロックの形状

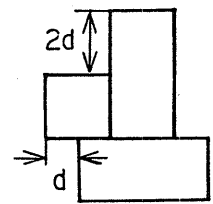


図3. ブロック間の間隔

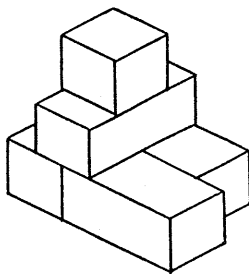


図4. 突起を無視したレゴの外観

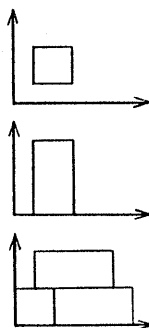


図5. 図4に対する階層別マップ

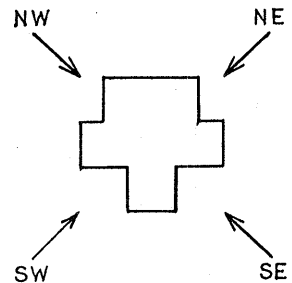


図6. 観測方向

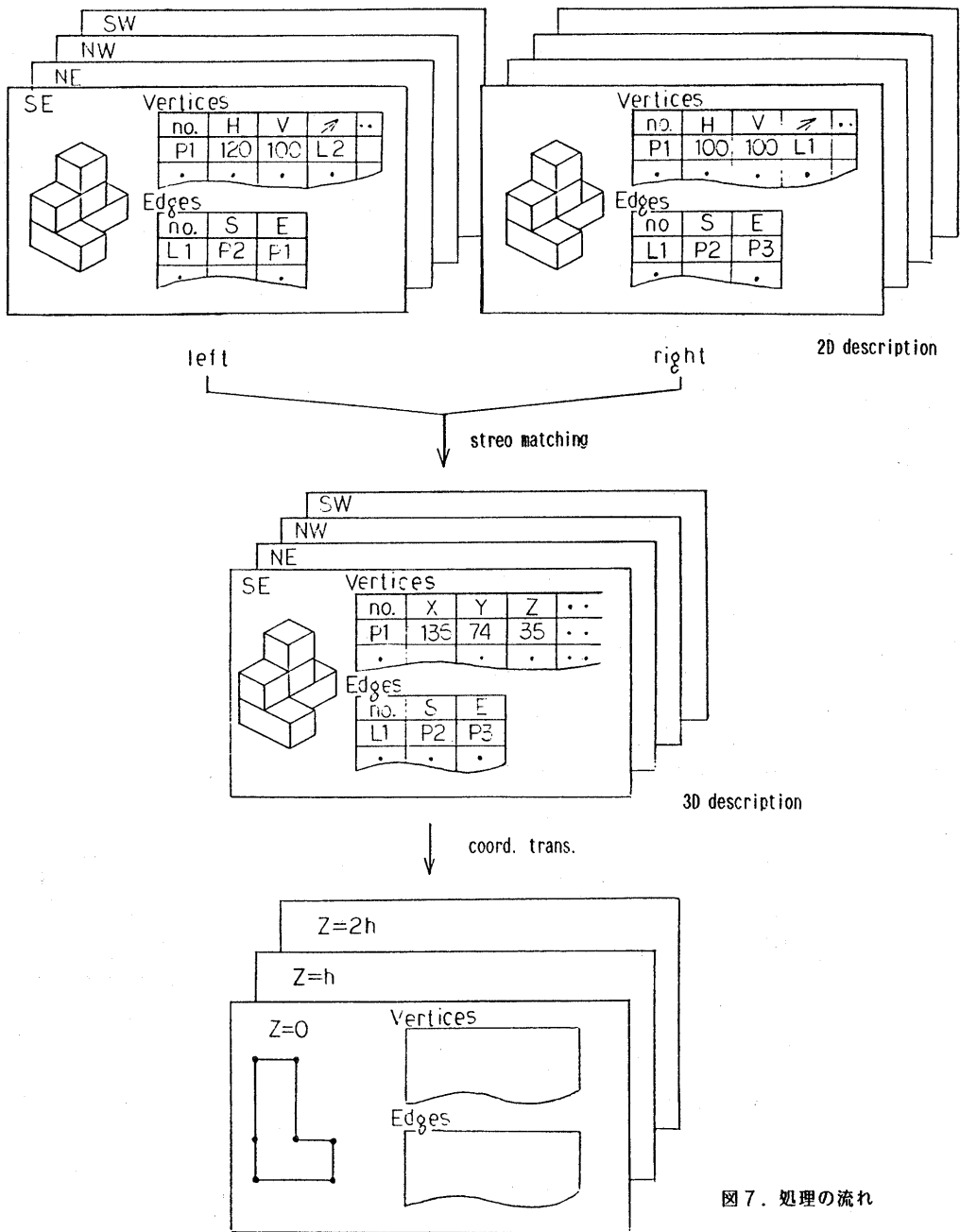


図7. 処理の流れ

頂点テーブル

頂点名	(h v)	ru	rd	lu	ld	up	dn	タイプ
P1	100 80	L1	NIL	L3	NIL	NIL	L2	0x29
P2	80 120	NIL	L3	NIL	NIL	NIL	L4	0x11
.

エッジテーブル

エッジ名	始点	終点	傾き
L1	P1	P4	RIGHT UP
.	.	.	.

図8. 2次元記述の例

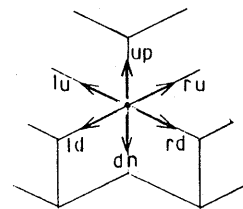


図9. エッジの傾き

3. 頂点とエッジの記述

3.1 2次元記述

頂点とエッジに関する2次元記述の例を図8に示す。各頂点のエントリには、頂点名、画面上の2次元座標値、その頂点に接続する各方向ごとのエッジ名、および頂点タイプのスロットがある。直交多面体の世界で考えているので、頂点に接続し得るエッジは図9に示すように右上がり、右下がり、左上がり、左下がり、上向き、下向きの6通りがあり、それぞれru、rd、lu、ld、up、dnというスロットを割当てて。これらのスロット中、NILは接続するエッジが存在しないことを示す。タイプは、これらの6方向に接続するエッジが存在するかしないかを6ビットにコード化したものである。この値を見れば、どの方向にエッジがつながっているかわかる。

一方、各エッジのエントリには、エッジ名、そのエッジの両端の頂点名およびそのエッジの傾きを保持するスロットがある。ここで、傾きは、画面上での右上がり、左上がり、縦方向を意味し、LEFT UP、RIGHT UP、VERTICALの3種類の値を持つ。

3.2 左右の対応付け

左右それぞれの画像について、前節で述べた2次元記述が得られた後、左右の対応付けを行って各頂点の3次元座標を求める。まず、縦方向のエッジの上端点に注目する。各頂点について、それが縦エッジの端点であるかどうかは、前述の2次元記述中の「タイプ」スロットを参照すればよい。dnに相当するビットが立っていれば、下向きのエッジがその点からつながっていることを意味し、あるエッジに対して上端点であることがわかる。

対応付けはエピポーララインに基づいて行う。左右いずれかの画像中で一点に注目すると他方の画像中でエピポーララインが決まる。理想的には、対応点はこのエピポーラライン上にのるので、タイプが同じでエピポーラライン上にのっている頂点を対応点とする。しかし、現実には観測誤差を含み得るので、エピポーララインまでの距離を求め、それがあるしきい値よりも小さい点を対応点の候補とする。

この場合、画面上の2次元座標を(h,v)とした時のhの値の範囲により探索範囲を限定する。図10に示すように、右画像中の点のh座標をhr、左画像中の対応点のh座標をhlとすると、 $hr < |hl|$ の条件が成り立つので、これを制約条件として探索範囲を制限できる。また、頂点のタイプ情報も対応点決定に際して制約条件として利用できる。

左右の対応付けの結果として得られる頂点の3次元座標値は、カメラに固定のカメラ座標系での座標値である。しかし、レゴの構造を考えるにはレゴに固定した座標系を考えた方が便利である。そこで、[溝口86]で述べたようにパレットに固定した座標系を考え、カメラ座標系で得られる3次元座標値をパレット座標系での表現に変換する。変換行列は、パレット上に設定したマークを観測することにより算出する。

3.3 エピポーララインの算出

エピポーララインは、カメラモデルによって規定される。ここでは、カメラモデルとしてピンホールモデルを用いた。エピポーララインは、撮像面の3次元空間中の位置と傾き、ピンホールの3次元的な位置、撮像面上の点の位置が得られれば求められる。撮像面の位置、傾き、ピンホールの位置は、カメラキャリブレーションにより予め得られているカメラパラメータである。エピポーララインを求めるために、次の2つのプログラムを用意した。一方は、与えられた撮像面上の2次元座標値とカメラパラメータに対して、視線を規定する3次元直線のパラメータを求めるプログラムである。もう一方は、与えられた点の3次元座標とカメラパラメータに対して、投影点の撮像面上での座標を求めるプログラムである。前者を用いて視線を求め、この直線上にのる空間中の異なる2点を適当に選ぶ。そして後者を用いてこの2点の投影点を求めることにより、投影点間を通る撮像面上の直線、すなわちエピポーララインが得られる。

3.4 構造的マッチング

上述した方法で一旦対応付けができれば、その頂点につながるエッジをたどることにより、他の頂点の対応付けを行って対応付けの手間を軽減できる。例えば、図11で左画像のP11と右画像のP13との対応がついたとする。それぞれの頂点の「タイプ」スロットを見

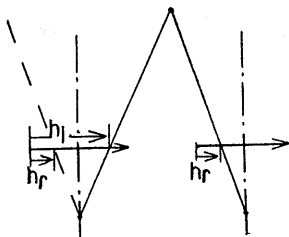


図10. 探索範囲の制限

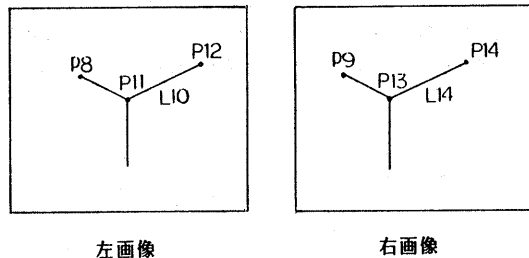


図11. 構造的マッチング

れば、この頂点から右上がりりと左上がりのエッジが出ていることがわかる。そして方向ごとのエッジ名のスロットからエッジ名が得られる。この例で右上がりのエッジは、左画像がL10、右画像がL14である。得られたエッジ名からエッジテーブルを参照すれば、反対側の端点を得られる。この場合はP12とP14である。同様に、左上がりのエッジをたどって左画像中のP8と右画像中のP9との対応付けができる。

3.5 3次元記述

上の例でも明らかなように、頂点名は左右で一致するわけではない。そこで対応付けがとれることに新しい頂点名を生成する。エッジについても同様で、新しいエッジ名を生成して左右画像中のエッジとの関連付けを行う。

採用した3次元記述の例を図12に示す。頂点のエントリが保持するスロットは、新しく生成した頂点名、左右の画像中の頂点名と2次元座標値、算出した3次元座標値、および接続するエッジ名である。エッジに関しては各エントリごとに、新たに生成したエッジ名、左右の画像中のエッジ名、始点、終点、エッジの方向、エッジ長のスロットを設けた。ここで、始点と終点には3次元記述用に生成した新しい方の名前を用いる。エッジの方向は、パレット座標系のx,y,zいずれかを記録する。x方向正をE、y方向正をNとする。SWから眺めた場合、右上がりのエッジがx方向、左上がりがy方向であるのに対し、NWから眺めた場合は右上がりがy方向、左上がりがx方向になる。

3.6 方向別データの統合

頂点やエッジが複数の方向から見える場合には、同じ頂点やエッジに対して方向ごとに別々の記述が作られてしまい、それらを統一する必要がある。このため、各階層ごとに2次元のグリッドテーブルを用意し、頂点のxy座標をプロットしてゆくことにより記述間の同一性のチェックを行う。プロットする時に既にそのグリッドがプロット済みであれば、その頂点の記述が重複していることになる。同一性のチェックに際しては、座標は統一した座標系で表現されている必要がある。座標系の統一は、前述したようにカメラ座標系からパレット座標系への変換により行う。

座標変換を行い、階層に分割し、方向ごとの記述の同一性のチェックを行なった結果として、図7の下方に示したような階層ごとの断面の外形線が得られる。

頂点テーブル

頂点名	左頂点	右頂点	左座標	右座標	(x y z)	ru	rd	...
P3	P4	P5	90 80	80 80	40 30 10	L1	NIL	...

エッジテーブル

エッジ名	左エッジ	右エッジ	始点	終点	方向	長さ
L2	L3	L2	P2	P5	x	64

図12. 3次元記述の例

4. 構造の認識

4.1 考え方

上述したように、構造の認識はレゴの各段ごとの断面図「階層別マップ」を作って考える。複数の方向から見本をステレオ計測し、座標系を統一して階層に分割した結果として図13a)に示すような各段ごとの断面図の外形線が得られる。この外形線データからブロックの存在を推定してゆく。まず、コーナー等、エッジの特徴的な各部分に着目し、その部分を稜線の一部とするようなブロックの存在を決定する(図13b)。これにより、内部にあって外側からは見えないブロックは除き、外から見えるブロックに関しての推定ができる。次に、どこにも接続されていないブロックが存在しないよう、各階層間でチェックを行う。このとき、必要に応じて内部に存在するであろう、外からは見えないブロックを推定してゆく。以上により、図13c)に示すような完全な階層別マップを得る。

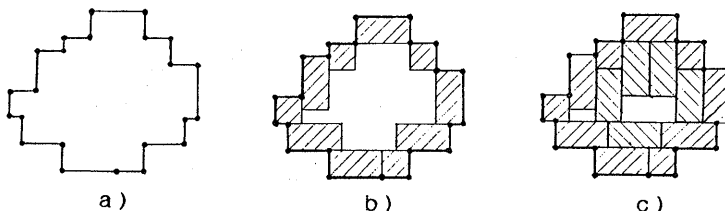


図13. 断面の外形線と階層別マップ

4.2 外周エッジの解釈

図13a)は観測によって得られる各エッジについてXY成分をプロットしたものである。図中の・印は、原画像中の垂直方向のエッジに対応している。図には示していないが、各エッジには座標値だけではなく、それがどの方向から眺めたときに観測されたものであるか、という情報も保持させる。

前述したように、ここで考えているレゴの場合、ブロック同士の位置関係がXY方向では突起間隔dで量子化されているため、エッジの長さは $d \sim 4d$ の4種類に限定される。今、長さ4dのエッジが見えたとすれば、それはタイプ2のブロックの長い方の稜線であり、そこにはタイプ2のブロックの存在を一意に決定できる。

4.3 内部の判定

しかし、単にエッジのみに注目したのでは、図14に示すように、そのエッジのどちら側にブロックが存在するのか決定できない。この問題は、エッジに対しどちら側が物体の「内部」であるかがわかれば回避できる。観測されるエッジは、物体の外形線であるからブロックは必ず「内側」に存在することになる。また仮に「外側」にブロックの存在を仮定したとすれば、推論に矛盾を生じていることの判定にもなる。

エッジのどちら側が物体の内部であるかは、そのエッジがどの方向から眺めた時に見えたものであるか、という情報をもとに決定することができる。今、図15に示すように、X方向のエッジが矢印のいずれかの方向からも見えたとすれば、図にハッチングで示したように内部はエッジの上側ということになる。同様に、Y方向についても観測方向に基づいて判定可能である。

4.4 解釈ルール

上述したように、見えるエッジの長さは $d \sim 4d$ の4種類であり、長さ4dのエッジが見つければ、タイプ2のブロックの存在を一意に決定できる。これに対して、4d以外の長さのエッジの場合には、そのエッジを含むブロックの種類、位置・姿勢が一意には決まらない。しかし、隣接する他のエッジからの制約や、既に決定している他のブロックの制約を利用すれば決定することが可能である。

例えば、図16a)のように、長さ3dのエッジに対し隣接して長さ2dもしくはdのエッジが直交していれば、タイプ2のブロックの存在を確定できる。また、図16b)のように、既に他のブロックが存在しており、タイプ2のブロックの存在を仮定する余地が無ければ、長さ2dのエッジに対してタイプ1のブロックの存在を確定することができる。

4.5 内部ブロックの推定

外形線を解釈した結果として、図13b)に示した不完全なマップが各層ごとに得られる。今、あるブロックに対しその下の層でオーバーラップするブロックが存在すれば、このブロックは下側のブロックによって支持されていることになる。しかし、もし、オーバーラップが無ければ、このブロックはどこからも支持されずに宙に浮かんでいることになってしまう。そこで、下側のマップの空いている部分にブロックを補うことにする。

最上層より最下層に向かって、このようなチェックと推定をくり返してゆけば、上層で新たに仮定したブロックを支えるためにさらに下層でブロックを仮定するという具合に、ブロックの支持、被支持関係の制約が上層より下層へ伝搬してゆき、結果として外側からは見えないブロックの存在を推定することになる。

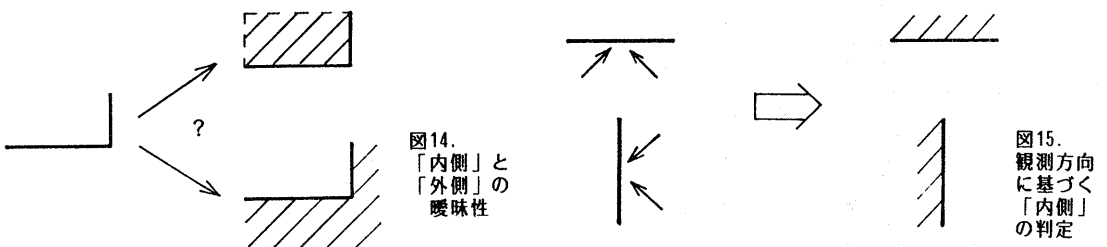


図14. 「内側」と「外側」の曖昧性

図15. 観測方向に基づく「内側」の判定

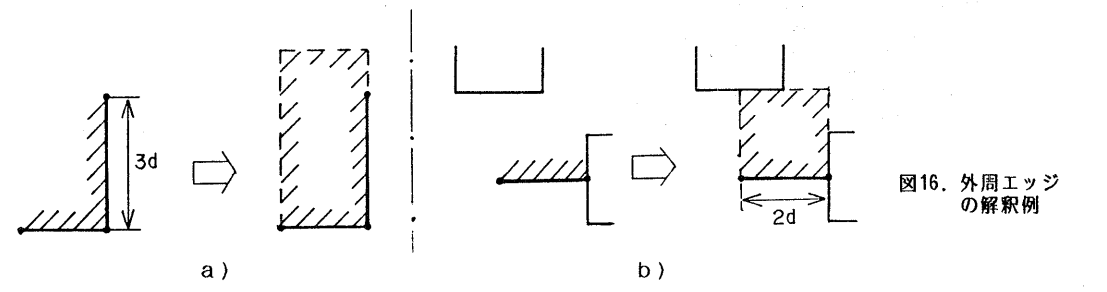


図16. 外周エッジの解釈例

4. 6 構造の記述

3次元構造の最小限の記述は、見本を構成するレゴブロック各々についての、1)種類、2)位置、3)姿勢である。これらは、上で述べた処理の結果として求められるものである。ブロックの位置・姿勢は図17に示すような、ブロックに固定した座標系で表わす。座標系の表現は同次座標系表現を用いる。すなわち、ブロックに固定した座標系の原点位置を p 、approach vector を a 、orientation vector を o 、normal vector を n として、 4×4 行列 ${}^t(n, o, a, p)$ で表わす。図17に、5つのブロックから成るレゴの構造の記述例を示す。ここでは、記述にMinskyフレーム風の表現をS式で表わしたものをを用いた。この記号表現の意味するところは、スロット名より明らかであろう。ただし、行列は行ベクトルのリストのリストとして表わしている。

5. まとめ

本報告では、組立てられたレゴの構造を理解するという問題を例に、組立て物体の3次元構造を視覚によって認識する方法について述べた。ここでは、部品となるレゴブロックの形状、ブロック間の位置関係、接続関係に着目し、これに基づいて画像処理結果の解釈を行い、組立てられたレゴの3次元構造を求めた。

ここで示した方法は、ステレオ対応付けも含めた前段階の画像処理に誤りが無いことを仮定しており、画像処理の失敗にはロバストではない。構造推定の過程で得られる中間結果を画像処理に反映させ、よりロバストにすることが今後の課題である。

参考文献

[Paul81] Paul, R., P., Robot Manipulators, MIT Press, 1981.

[溝口86] 溝口, 他「複数方向ステレオ視による物体の3次元構造の推定(第1報)」, 情報処理学会第33回全国大会, 1986.

[溝口87] 溝口, 他「複数方向ステレオ視による物体の3次元構造の推定(第2報)」, 情報処理学会第34回全国大会, 1986.

```
((block1 (ako block)
  (type ($value (type1)))
  (coordinate ($value (
    ((1 0 0 16)
     (0 1 0 16)
     (0 0 1 38)
     (0 0 0 1)))))))
(block2 (ako block)
  (type ($value (type2)))
  (coordinate ($value (
    ((0 -1 0 48)
     (1 0 0 0)
     (0 0 1 19)
     (0 0 0 1)))))))
(block3 (ako block)
  (type ($value (type2)))
  (coordinate ($value (
    ((1 0 0 0)
     (0 1 0 32)
     (0 0 1 0)
     (0 0 0 1)))))))
(block4 (ako block)
  (type ($value (type2)))
  (coordinate ($value (
    ((1 0 0 32)
     (0 1 0 0)
     (0 0 1 0)
     (0 0 0 1)))))))
(block5 (ako block)
  (type ($value (type1)))
  (coordinate ($value (
    ((0 -1 0 32)
     (1 0 0 0)
     (0 0 1 0)
     (0 0 0 1)))))))
```

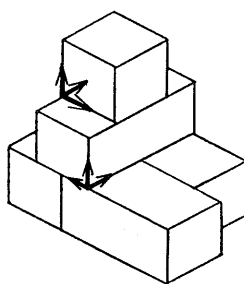


図17. 構造記述例