The University of Osaka
Institutional Knowledge Archive

Title	リスト表現形状モデルと生産加エシステムへの応用に関する研究
Author(s)	大久保, 雅史
Citation	大阪大学, 1996, 博士論文
Version Type	VoR
URL	https://doi.org/10.11501/3110144
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

リスト表現形状モデルと生産加工システムへの応用に関する研究

平成７年11月

大久保　雅史

リスト表現形状モデルと生産加工システムへの応用に関する研究

平成７年11月

大久保　雅史

目　　次

第1章　序論　1
1.1　本研究の背景および目的　1
1.2　本研究の構成　2

第2章　リスト表現を用いた形状モデルと図形処理　4
2.1　緒論　4
2.2　形状モデルの計算機内部表現法と幾何学的意味　4
2.2.1　ワイヤーフレームモデル　4
2.2.2　サーフェイスモデル　5
2.2.3　ソリッドモデル　5
2.2.3.1　CSG　6
2.2.3.2　B-reps　6
2.3　生産加工システムにおける形状モデルのあり方　7
2.3.1　形状モデルの簡潔性　8
2.3.2　形状モデルの対象との整合性・信頼性　8
2.3.3　マスマトリックス計算性　8
2.3.4　図形処理の簡便性　9
2.3.5　形状モデルのシステムに対する柔軟性・拡張性　9
2.3.6　形状モデルのあり方　9
2.4　符号化形態表現（CBR）法　10
2.4.1　形状モデルと形状リスト　10
2.4.2　定性的特徴抽出　12
2.4.3　基本図形処理とリスト操作の関係　13
2.4.3.1　回転操作について　13
2.4.3.2　共役操作について　14
2.4.3.3　鏡像操作について　14
2.4.3.4　合成操作について　15
2.4.3.5　分解操作について　17
2.4.4　CBR法の柔軟性・拡張性　19
2.4.4.1　簡易型符号化形態表現（SCBR）法　19
2.4.4.2　改良型符号化形態表現（MCBR）法　20
2.5　結論　21

第3章　リスト表現形状モデルの計画問題の画像化への応用　22
3.1　緒論　22
3.2　スケジューリング問題の定義とPERTによる解法　22
3.2.1　取り扱うスケジューリング問題の定義　22

3.2.2　PERTによる解法　23
3.3　スケジューリング問題における形状モデル　25
3.3.1　簡易型符号化形態表現（SCBR）法　26
3.3.2　作業と枠の図形モデル　27
3.4　図形処理を用いたスケジューリング問題解法のアルゴリズム　28
3.4.1　優先順位決定法　28
3.4.2　スケジューリング・アルゴリズム　28
3.5　スケジューリングテスト　29
3.5.1　スケジューリング例　29
3.5.2　PERT山崩し法との比較　32
3.6　所要人数変更問題への適用　34
3.6.1　所要人数変更問題の定義　34
3.6.2　形状モデル　35
3.7　スケジューリングテスト　35
3.8　結論　37

第4章　リスト表現形状モデルの画像の変換への応用　38
（MCBR法を用いたコンピュータグラフィックスの生成）
4.1　緒論　38
4.2　MCBR法による対象の記述　39
4.3　MCBR法を利用する3次元CADモデルからの　39
2次元画像データデータの生成
4.3.1　前処理　40
4.3.1.1　2つの2次元多角形の位相関係　41
a　辺と点の関係　41
b　辺と辺の交差条件　42
c　面と点との位相関係　43
4.3.1.2　2つの稜線と視点との位置関係　44
4.3.2　隠面処理を施した2次元データの作成　46
4.4　結論　49

第5章　リスト表現形状モデルの画像の認識への応用　50
5.1　緒論　50
5.2　画像認識システムの構成　51
5.3　画像からの線分の抽出　52
（コーナー検出と直線成分抽出の協調処理）
5.3.1　Hough変換と補完アルゴリズム　52
5.3.2　Hough変換の問題点　52
5.3.3　画像からのコーナー検出　54
5.3.3.1　コーナー検出の基本アルゴリズム　54

5.3.3.2　コーナー検出による端点抽出　56
5.3.4　Hough変換とコーナー検出との融合による線分抽出　56
5.3.4.1　融合アルゴリズム　57
5.3.4.2　線分抽出実験　58
5.4　面心を用いた画像マッチングアルゴリズムの開発　59
5.4.1　面心検出　59
5.4.1.1　面データ作成アルゴリズム　59
5.4.1.2　面心抽出アルゴリズム　61
5.4.1.3　面心抽出結果　62
5.4.2　面心を用いた面のマッチングアルゴリズム　63
5.4.3　パターンマッチング実験　65
5.4.4　適用限界　67
5.5　結論　68

第6章総括　69

謝辞　72

参考文献　73

発表論文　76

第1章　序　論

1.1　本研究の背景および目的

最近，産業界においては，円高や東南アジア諸国さらに欧米諸国との販売競争の激化等に対処するため，よりいっそうの製品のコストダウンを目的として，これまで以上の生産の自動化や新しい生産システムの開発，さらに経営までも含めた生産システムの統合化が叫ばれている[1_1].

このような状況下で，システムは益々巨大・複雑となりシステム自体の設計・開発が非常に煩雑となっている．さらに，システムの運営にもかなりの労力を要することは明白である．自立分散型のシステム[1_2]は，このように巨大化・複雑化するシステムへの一つのアプローチであると考えられるが，分散化されたサブシステム同士のインタフェースやサブシステムを統括するシステムとのインタフェースの設計・開発が煩雑になることに変わりはない．従って，このようなシステムにおいては，システム全体で用いられる製品の基本モデルや基本的なアルゴリズムを統一することにより，これら巨大化するシステムの開発・運営の労力を軽減することが必要となる[1_3].

一方，最近のインターネットをインフラストラクチャーとしたマルチメディアの発展[1_4]やバーチャル・リアリティー技術の進歩[1_5]，人工知能の分野での'図による推論'への注目[1_6]を見てみると，図面や画像データの重要性が増し，これらの技術の生産システムへの導入が望まれている[1_7].

本研究では，このような2つの観点から，生産システムでの要素技術である形状モデルの開発とその形状モデルを用いた'画像'データの生産システムへの適用を目的とした.

まず，新しく開発した形状モデルは，3次元のCADによく利用されているB-repsを基本とし，形状を複数のリストで表現する．このモデルの特徴として，形状をリスト表現することで様々な画像変換を簡便なリスト処理に置き換えることができる，リストの一つが形状を構成する辺の方向を示しており，形状が持つトポロジカルな性質の抽出が簡便に行える，いくつかのリストの添付・削除やリストの内容の入れ替えが可能で様々な状況・局面に対処できる柔軟性を持つ，等が挙げられる.

つぎに，開発した形状モデルの生産システムの3つの状況・局面での適用について検討する．まず，計画問題の画像化を目的とし，スケジューリング問題を取り上げ，形状モデルと図形処理を用いたスケジューリング・システムの開発について述べる．さらに，この形状モデルを適用した3次元形状モデルの2次元画像化と形状モデルの画像認識への適用を示し，本形状モデルの生産システムへの適用の有効性について検討する．

1.2　本研究の構成

本論文は，本章を含めて6章から構成されている．図1_1に本研究の構成図を示す．第2章では，まず，既存の形状モデルについて触れ，本論文で対象としている生産システムにおける形状モデルのあり方について論ずる．さらに，本研究を通して開発・適用した形状モデルと図形処理について述べる．第3章から第5章までは，第2章で

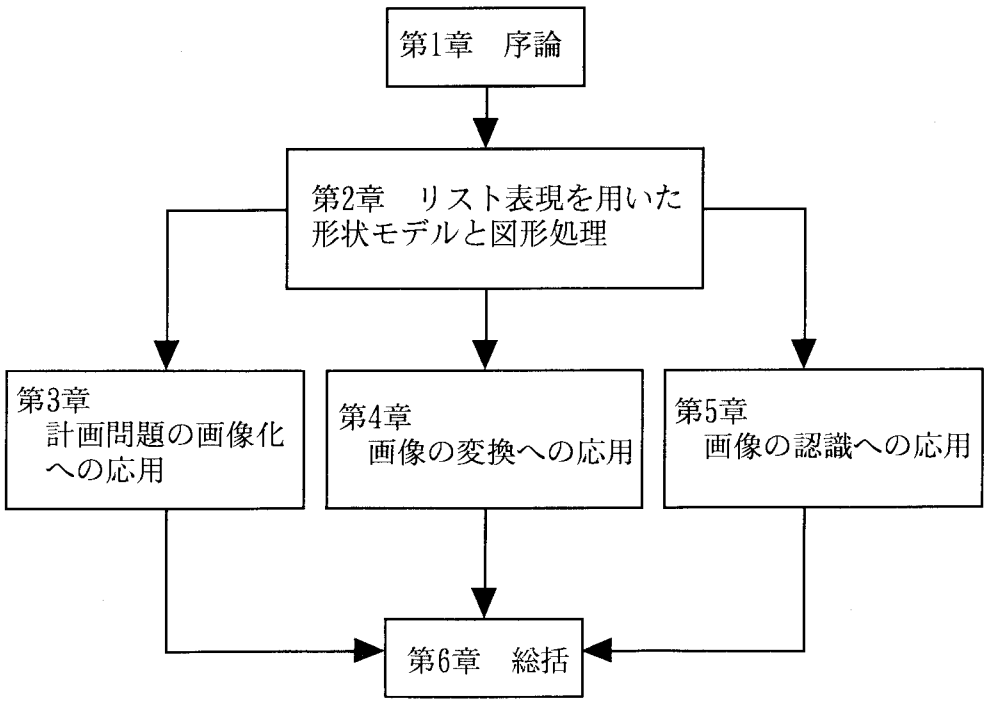


図1_1　本研究の構成.

述べた新しい形状モデルを生産システムの様々な状況・局面に適用した例について述べる．第3章では，形状モデルを計画問題に適用し，図形処理を用いたスケジューリング・システムの開発について述べる．ここでは図形処理を用いることで，既存のスケジューラが適用困難な問題を取り扱うことができることを示す．第4章では，新しい形状モデルを用いて，3次元CADモデルの情報から2次元の画像情報を得るシステムの開発とその有効性について述べる．またこの際，隠面処理が必要となるが，ここでは新しい形状モデルの特徴を生かした隠面処理について詳しく述べる．得られる2次元の画像データは，加工・組立工程や検査・補修工程で実際の画像と比較・検討され対象の認識や対象の位置の把握に用いられる．続く第5章では，新しい形状モデルの画像認識への適用について述べる．ここでは，ステレオ立体視や第4章で生成された3次元形状モデルからの2次元画像と実際の画像とのマッチングで問題となる対応検索に新しい形状モデルを利用し，対象を構成する面同士のマッチングから稜線・頂点と徐々に小さな部位のマッチングを行うトップダウン的手法について記す．またこの際，画像から正確な線分を抽出する必要があるが，直線成分の抽出に一般的に良く用いられているHough変換の欠点を補うためにコーナー検出から得られるデータを用いる手法についても述べる．最後に第6章で本研究を総括する．

第2章　リスト表現を用いた形状モデルと図形処理

2.1　緒論

現在，CAD/CAMやCAEの分野ばかりでなく様々な分野で図形処理が用いられている[2_1]．図形処理において核となるのは，計算機内部で対象物を幾何学的にどのように表現するかを示す形状モデルである[2_2]．これまでに開発された形状モデルはワイヤーフレーム，サーフェイス，ソリッドモデルの3つに大別され，目的により使い分けされたり併用されたりしている[2_3][2_4][2_5]．その中でCSG（Constructive Solid Geometry）[2_6]やB-reps（Boundary Representation）[2_7]等のソリッドモデルは，計算機内部のモデルと実際の形状とが一対一の関係にあり形状を正確に表現できることから最近のCAD/CAMによく用いられている[2_8]．本研究で開発した符号化形態表現法（Coded Boundary Representation 以下CBRとする）も元来は2次元簡易CADを目的として作られたソリッドモデルである．

本章では，これまで開発されてきた様々な形状モデルの概略を説明し，本研究の対象となる生産加工システムにおける形状モデルが具備すべき条件について考察する．さらに，本研究で提案するCBR法について述べ，生産加工システムへの適用性を検討する．

2.2　形状モデルの計算機内部表現法と幾何学的意味

これまでに開発されてきた形状モデルは，基本的な考え方によって，ワイヤーフレームモデル，サーフェイスモデル，ソリッドモデルの3つに大別される．基本的にこれらのモデルでは，モデルが持つ情報量が多いほど，実際の形状を正確に表現可能であるが，図形処理に要する計算機資源や処理時間が大きくなる[2_9]．ここでは，この3種類の形状モデルについて，その特徴等について述べる．

2.2.1　ワイヤーフレームモデル

このモデルは形状を稜線の3次元座標のみで表現する．言い方を替えれば，計算機は，その形状の稜線しか取り扱わない．図2_1（a）にその例を示す[2_10]．この形状の特徴として，

・形状のデータ量が少なく，図形処理に要する計算量も少ない．

・対象物の特徴をよく表現できる.

ことが挙げられる. しかし, 稜線のみのデータで形状を表現しているために, 対象物を構成している面が一意に決まらない. 例えば図2_1（a）で表されたモデルは, 図2_1（b）～（d）のいずれの形状を表現しているか特定できない. したがって体積や重心を求めるようなマスマトリックスの計算や断面図の作成等を行うことが一般的に不可能である.

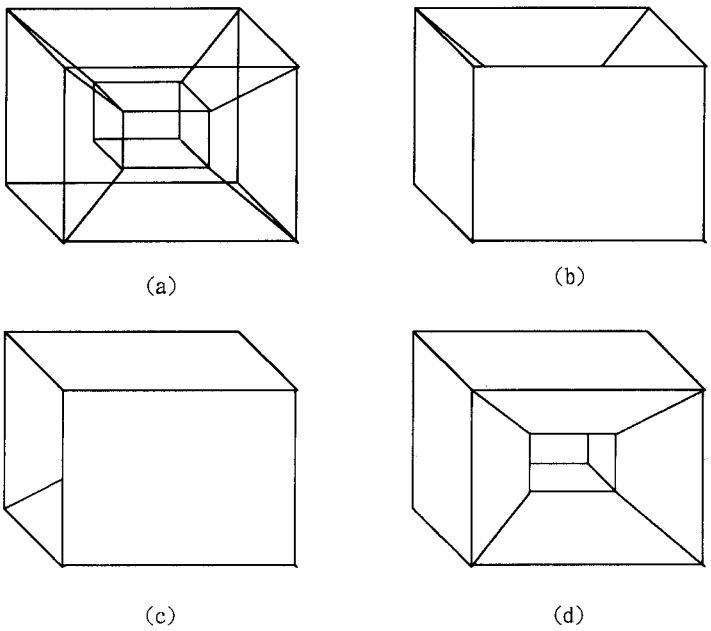


図2_1 ワイヤーフレームモデル.
(a) 形状モデル
(b)～(d) (a)から考えられる実体

2.2.2　　サーフェイスモデル

ワイヤーフレームモデルに面の情報をつけ加えたものである. データの情報量は比較的少なく, また, 図形処理に必要な情報を殆ど完備している. しかしワイヤーフレームモデルと同様に曖昧性が残る. つまりサーフェイスモデルでは面に対する実体の存在方向が定まらない. このことから, マスマトリックスの計算を行うことは依然として不可能である.

2.2.3　　ソリッドモデル

面に対する実体の方向が定まらなかったため, 対象物を完全に復元できなかった上記2つのモデルに対して, 対象物を完全に定義できるのがソリッドモデルである.

ソリッドモデルはそのデータ構造によりCSG（Constructive Solid Geometory）と境界表現（B_reps : Boundary representation）法に大別される. 実際のCADシステム

ではこの二つを複合化した表現形式を用いていることが多い．例えば，計算機内部では形状のマスマトリックスの計算が容易なB-repsで表現し，インターフェイスでは設計者の設計感覚に比較的近いCSGの形式で入力するよう工夫されているシステムもある[2_11]．以下に両者の表現形式について述べる．

2.2.3.1　　CSG

あらかじめ定義した基本的な形状（プリミティブ）の集合演算で対象物の形状を表現する方法である．具体的には，図2_2に示すように，プリミティブに大きさを与えて所定の大きさの立体（オブジェクト）を定義し，これに和・差・積などの集合演算を施すことにより対象物を表現する方法である．集合演算以外に回転などの操作も施せる．

このCSGは既存の形状を使用することから，形状の入力や修正が簡単なこと，データに要する計算機の記憶容量が少なくてすむことが長所として挙げられる．しかし逆に言えば，稜線データを必要とする処理には不向きといえる．またこの方法では，一つの立体を表現するために複数の表現方法が存在する可能性があり，形状の比較等が難しくなる．

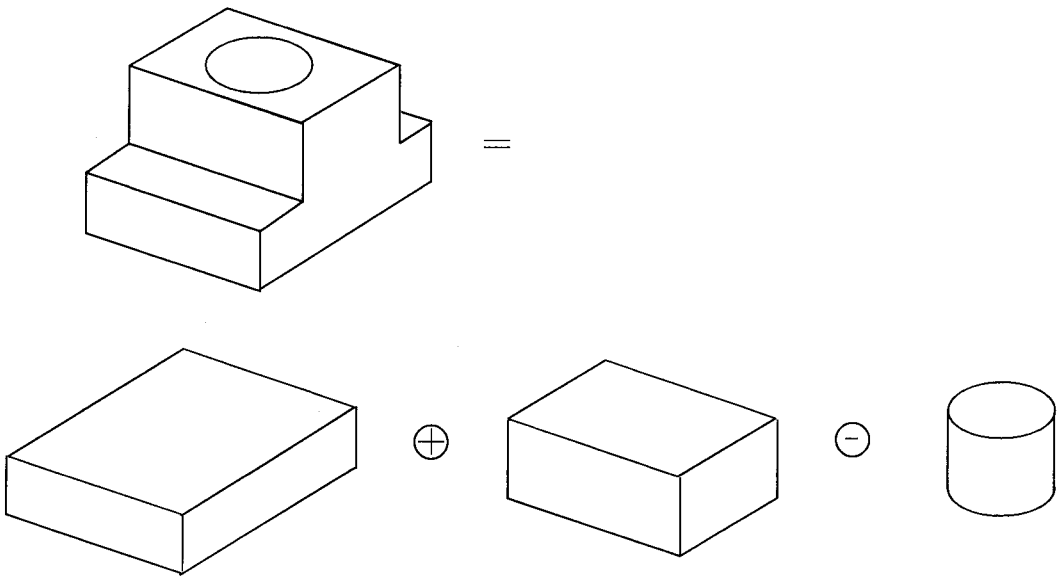


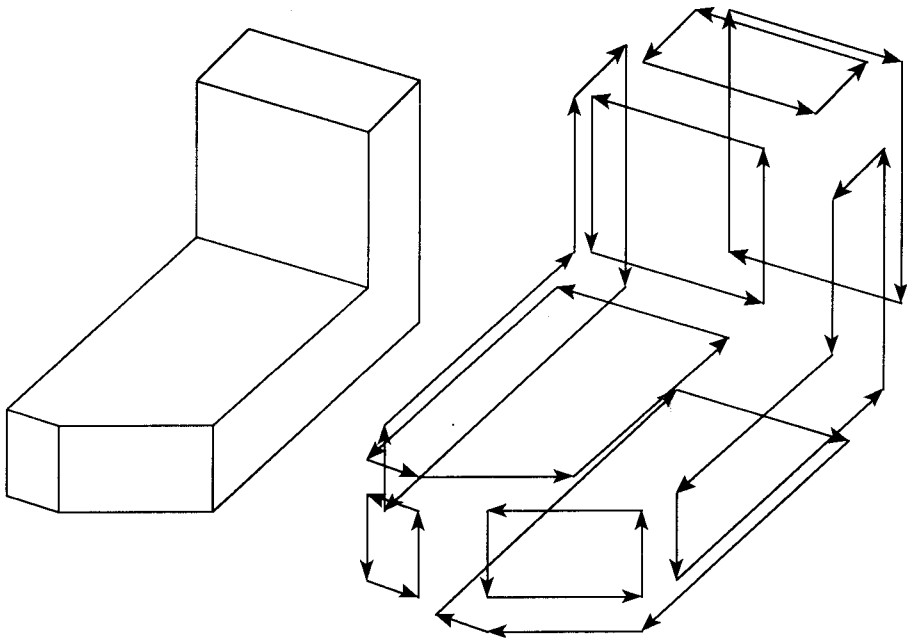
図2_2 CSGモデル．

2.2.3.2　　B-reps

基本的にはサーフェイスモデルを発展させたものである．つまりサーフェイスモデルに面に対する実体の存在方向を示すデータをつけ加え，完全な立体を復元できるようにしている．実体の存在方向の明示には面に対して右ネジの進む方向を法線ベクトルとす

るデータを使用する（図 2_3）.

このB-repsでは，形状データが対象物の実際の数値列で直接表されていることから，マスマトリックス計算が簡単に行える．しかしその反面，面を構成する稜線の数が不定であるため，計算機内部での表現がめんどうになる欠点もある．さらにこのことによって図形処理等のプログラミングが複雑になる．



(a) オブジェクトの例　　(b) (a)のB-reps表現

図2_3 B-repsモデル.

2.3　生産加工システムにおける形状モデルのあり方

前節で，既存の形状モデルについての概略を述べたが，ここでは，生産加工システムにおける形状モデルが具備すべき条件について考える．生産加工システムには多様な要素，様々な局面があり，それに応じて形状モデルを適用する目的も異なったものになる．CADで利用される製品の設計モデルやCAEで利用される製品の解析モデル等，目的に応じた形状モデルをそれぞれ用意することも可能であるが，生産加工システムの統合を考えた場合，各状況に適応するように形状モデルのアーキテクチャーから変えてしまうよ

りも，柔軟性・拡張性のある一つのアーキテクチャーの基に構築された形状モデルを利用し，図形処理の統一化やそれに伴う所要計算機資源の縮小化等を図ることができればその価値は大きいと考えられる．そこで，本節では，システム全体を通して利用可能な形状モデルを対象に議論を進める．

2.3.1 形状モデルの簡潔性

ワイヤーフレームやサーフェイスモデルさらにB-repsでは，データのフォーマットが比較的簡単に表現されており，データから頂点，稜線，面の情報が比較的高速に求められる．しかしながら，B-repsの場合一つの面がいくつの稜線から構成されているかが不明なため，メモリーの割り当てを静的に行うことが難しい．一方，CSGでは，対象物の形状は陽には表現されず，プリミティブの集合演算等による空間の点の集合として表されているため，頂点，稜線，面等の情報を取り出すための何らかの前処理を行う必要がある．上述のように，多様な対象に対して，異なった目的で適用できる形状モデルを考えるならば，データ構造はできるだけ簡潔であることが求められる．さらに，形状に対して，寸法精度・材質・色・質感等付加情報を簡単に付加可能であることが望ましい．

2.3.2 形状モデルの対象との整合性・信頼性

前述のように，ワイヤーフレームモデルやサーフェイスモデルでは，形状モデルと実際の対象物が一対一の関係にならない．しかし，これらのモデルがソリッドモデルより劣っているとは言えない．表現すべき対象物がある限定した範囲に限られた場合や一意性が成り立たない場合に補助的な情報を付加することで一対一の関係を保つようにシステムを工夫した場合は，これらのモデルの簡潔さと形状操作時の計算量の少なさは非常に利点となる．このような観点から，ここでは，形状モデル適用対象と必ずしも一対一の関係を保証されていなくても良いとの考えで，形状モデルを開発する．

2.3.3 マスマトリックス計算性

CADでは，製品の体積や，重心，モーメントを求めることがしばしば行われる．また，ロボットの未知の対象のハンドリングの問題でもカメラから得られた画像から対象の重心等のデータを解析する必要がある．このように，形状モデルに対してマスマトリックスの計算を必要とする場合には，その計算アルゴリズムの簡便さ，従って計算量の少なさが必要となる．上に挙げたソリッドモデルにおいて，B-repsは，形状の稜線や，頂点の情報を明示的に保持しているために，この条件を満たしている．一方，CSGは，マス

マトリックス計算を行う前に，形状と空間の境界を検索する必要があるが，形状モデルを構成する各プリミティブに対してマスマトリックスを予め計算しておくと対象物のマスマトリックス計算が非常に簡単に行える．マスマトリックスを計算する必要がある対象において，その対象で用いるモデルのマスマトリックスが容易に求められれば良い．

2.3.4　図形処理の簡便性

図形処理の手続きの煩雑さの度合いは，形状モデルがどのような形式で計算機内部に保存されているかに大きく依存する．また，図形処理が単独のオブジェクトに対して行われるか，複数のオブジェクトに対して行われるかによっても変わってくる．つまり，単独のオブジェクトに対して，回転・鏡像・平行移動等の処理が行われる場合は，基本的には座標変換のみでデータ量の増減は起こらない．しかし，複数のオブジェクトに対して，合成や分解等の処理が行われる場合，形状モデルの計算機内部表現によってデータ量の増減が起こる可能性がある．処理を実際に始める前に，データ量の変化を予測することが難しい場合，データの取り扱いが煩雑となる．従って，図形処理によるデータ量の変化を前もって予測できるような形状モデルを用いるか，データ量の変化に対して柔軟なデータ構造を持っている必要がある．

2.3.5　形状モデルのシステムに対する柔軟性・拡張性

全適用対象に対して，同一のアーキテクチャーを持った形状モデルを開発する場合，形状モデル自体が各対象において最適化されるような柔軟性を持っている必要がある．例えば，形状が常に形状内部に穴を持たない場合とその可能性がある場合，さらに形状モデルの’形’が重要である場合や’形’とそれに付加されている情報が同じように重要である場合に全く同じ形状モデルを利用することは計算機資源のむだである．従って，各対象において形状モデルが，そこで利用される情報に基づいた形式になっていることが望ましい．

2.3.6　形状モデルのあり方

以上のことから，汎用の形状モデルをすべての対象に対して適用することは非常に難しい，あるいはむだが多いことが分かる．一般に対象を限定したCADやCAMが良く利用されている割に汎用のCADやCAMがあまり使われていないことも以上の理由からだと考えられる．しかし，生産システムの統合を行うためには形状モデルや図形処理が統一されたアーキテクチャーの基に構築されていることが望ましい．従って柔軟性・拡張性を

持った形状モデルを構築することが必要と考えられる.

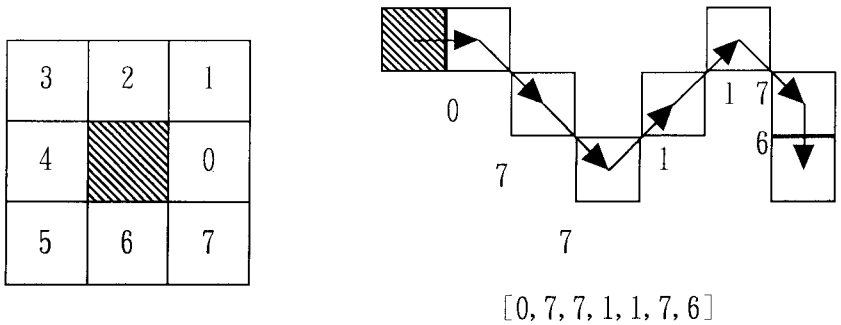
2.4　符号化形態表現（CBR）法

前節までで述べた形状モデルのあり方に基づき，新たな形状モデルを開発した．開発したCBR法は基本的には，B-repsと同様の考えに基づいているが，形状をリストで表現している点，さらに方向コードという考え方を導入し，形状の各頂点に対して座標値を明示的に持たせず，代わりに形状の各辺に方向の情報を付加している点等が異なる．このため図形の合成・分解，共役・鏡像・回転などの図形処理をリスト操作として扱える，形状のトポロジカルな特徴を定性的に把握できる等の特徴を持つ．さらにCBR法ではこの方向コードを4方向に規定し形状を簡単化することによって各種の図形処理を数値演算を用いず，記号処理のみで行うことを実現している．ここでは，このCBR法について詳細に述べる.

2.4.1　形状モデルと形状リスト

CBR法は形状を構成する稜辺に対して方向を付与し，形状をこの方向を示す記号を含めたリストで表現する方法である.

Freemanは，画像処理において線画に対し図2_4（a）に示す8近傍の方向コードを規定し，図2_4（b）に示すような線画を方向コードの列（チェインコード）として表現することで線画データを圧縮することを提案した[2_12].

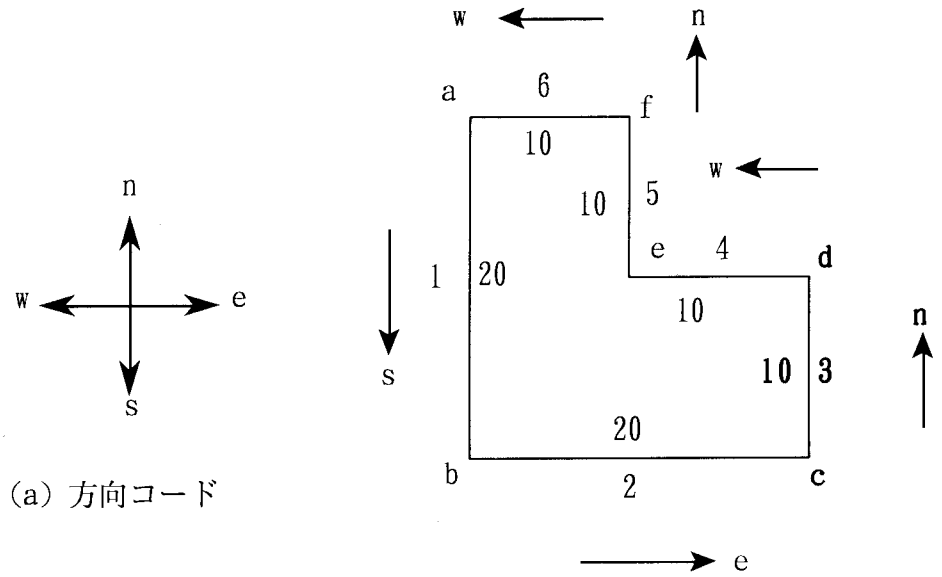


（a）方向コード　　（b）チェインコード

図2_4 Freemanのチェインコード

この方向コードの列がリスト表現そのものであることに着目して開発した形状モデルがCBR法である．CBR法では，東（e）・西（w）・南（s）・北（n）の4方向を辺の方向コードに規定し（図2_5（a）），形状データを5つのリスト，辺リスト（E_I_list），始

点リスト（S_P_list），終点リスト（E_P_list），方向リスト（E_D_list），長さリスト（E_L_list）で表現する．E_I_listは各辺に与えられたID番号であり，各辺を特定する．CBR法では，各辺は，実体を左手にみる方向に進む有向辺として規定している．このような定義のもとで，S_P_list・E_P_listは，それぞれ辺の始点・終点名を，E_D_listは前述の方向コードにより規定された辺の方向を示している．最後のE_L_listは，各辺の長さを表している．図2_5（b）に示した図形について各形状リストを図2_5（c）に示す．



（a）方向コード

（b）オブジェクトの例

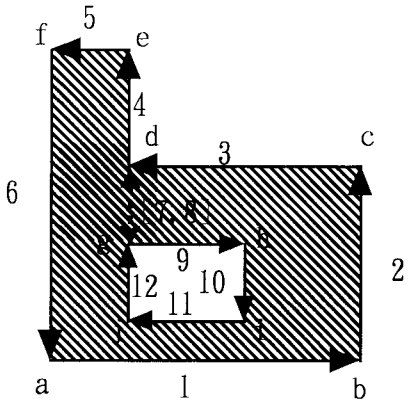
E_I_list[1, 2, **3**, 4, 5, 6]
S_P_list[a, b, **c**, d, e, f]
E_P_list[b, c, **d**, e, f, a]
E_L_list[20, 20, **10**, 10, 10, 10]
E_D_list[s, e, **n**, w, n, w]

（c）（b）の形状リスト

図2_5 符号化形態表現(CBR)法.

また，内部に空間（穴）があるような図形に対しては，外部を構成する線群と内部を表現する線群を実際には存在しない二本一組の内部線と呼ばれる線で結ぶことにより表

現する．その際，内部辺は，E_I_listの中で2つの数字によって構成されたリストとなる．図2_6（a）に示した図形について各形状リストを図2_6（b）に示す．ここで，内部辺［7,8］と－［7.8］は，同じ辺で方向が逆向きであることを表現している．



(a) 内部に穴を有する
オブジェクト

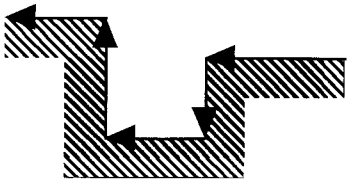
E_I_list[1, 2, 3,[**7,8**],9,10,11,12,-[**7,8**], 4, 5, 6]
S_P_list[a, b, c, **d**, g, h, i, j, **g**, d, e, f]
E_P_list[b, c, d, **g**, h, i, j, g, **d**, e, f, a]
E_D_list[e, n, w, **s**, w, s, w, n, **n**, n, w, s]
E_L_list[40,25,30,**10**, 15,10,15,10, **10**, 15,10,40]

(b) (a)の形状リスト

図2_6 内部ループを有するオブジェクトと
その形状リスト．

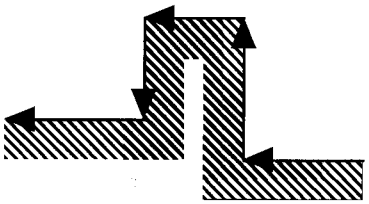
2.4.2　　定性的特徴抽出

このCBR法は上で述べたように形状をリストで表現している，さらに明確に座標値を持っていないため，図形の位置にとらわれず，そのトポロジカルな特徴を把握しやすく，そのため図形操作が簡単に行えるという特徴を持つ．方向コードを4方向と定めたのも図形操作をすべて記号処理のみで実現するためである．例えば，ある形状が図2_7に示すような凹凸を持っているか否かは，E_D_listに特定の記号列を含んでいるかどうかを調べれば判る．この例では，E_D_list中に，[w,s,w,n,w]，[w,n,w,s,w] の連続した要素を持つ形状は，それぞれ凹部，凸部を有することを示している．



[...,w,s,w,n,w,...]

(a) 凹部



[...,w,n,w,s,w,...]

(b) 凸部

図2_7 凹凸の検出．

2.4.3　基本図形処理とリスト操作の関係

CBR法では，対象を計算機内部においてリスト表現しているため，様々な図形処理はリスト操作を行うことで実現する．ここでは，CBR法における基本的な図形処理の例を示す．ただし，以下の説明では，処理に直接関係のないリストの操作については省略する．

2.4.3.1　回転操作について

CBR法では，対象を構成する辺の向きが4方向しかない．従って，回転可能な角度も4となる．回転操作の場合図形の位相的な変化はない．従って，辺の方向を示すE_D_listの要素だけが変化する．

例えば，図2_8（a）に示すように図形を反時計方向に90度回転させるためには，各辺の方向コードDを図2_8（b）の指標に従って，

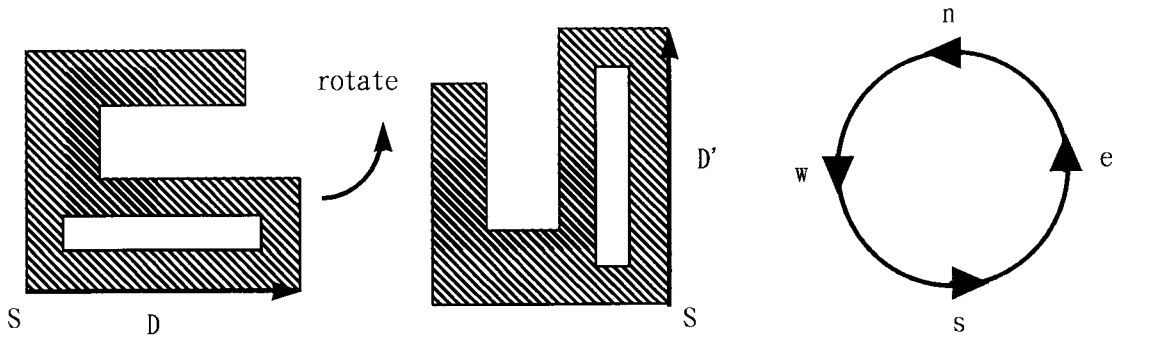
D'=n　if D=e

D'=w　if D=n

D'=s　if D=w

D'=e　if D=s

と書き換えることで実現できる．



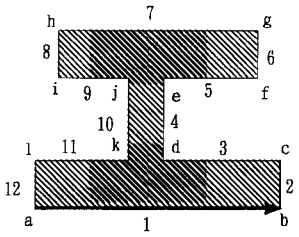
(a) $\pi/2$ 回転

(b) 回転前後の方向コードの関係

図2_8 回転操作.

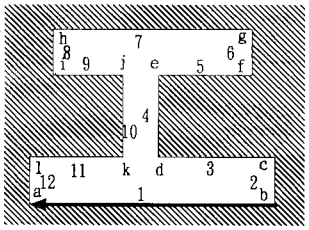
2.4.3.2　共役操作について

共役操作とは図2_9(a)(b)に示すように図形の実体部分と空間部分を入れ換える操作である．CBR法では図形の実体は辺の左側に存在すると定義していることから辺の方向を逆にすれば実体の位置は変化する．従って，方向リストにおける辺の方向を変化させる．さらに，これにより辺の順番が逆となるので全リストを反転し，始点リストと終点リストを入れ換える．



E_I_list[1, 2, 3, 4, 5, 6, 7, 8, 9,10,11,12]
S_P_list[a, b, c, d, e, f, g, h, i, j, k, l]
E_P_list[b, c, d, e, f, g, h, i, j, k, l, a]
E_D_list[e, n, w, n, e, n, w, s, e, s, w, s]

(a)　オブジェクトの例



E_I_list[12,11,10, 9, 8, 7, 6, 5 ,4, 3, 2, 1]
S_P_list[a, l, k, j, i, h, g, f, e, d, c, b]
E_P_list[l, k, j, i, h, g, f, e, d, c, b, a]
E_D_list[n, e, n, w, n, e, s, w, s, e, s, w]

(b)　(a)の共役形状

図2_9 共役操作.

2.4.3.3　鏡像操作について

鏡像操作とは図2_10のように鏡となる線分に対して線対称となる図形を求める操作である．CBR法の方向コードの制約から，考えられる鏡は水平・垂直の二種類である．鏡像操作は，上の共役操作とほとんど同じ操作を行うが，方向リストに関してだけは若干異なる操作を行うことになる．

すなわち，すべてのリストを反転し，S_P_listとE_P_listを入れ換えた後，

・垂直な鏡の場合

各辺の方向コードで，垂直方向を表す（n, s）だけを反対方向のコードにする（図2_10(b)）．

・水平な鏡の場合

各辺の方向コードで，水平方向を表す（e, w）だけを反対方向のコードにする（図2_10（c））.

以上の操作で，対象の鏡像図形を得ることができる.

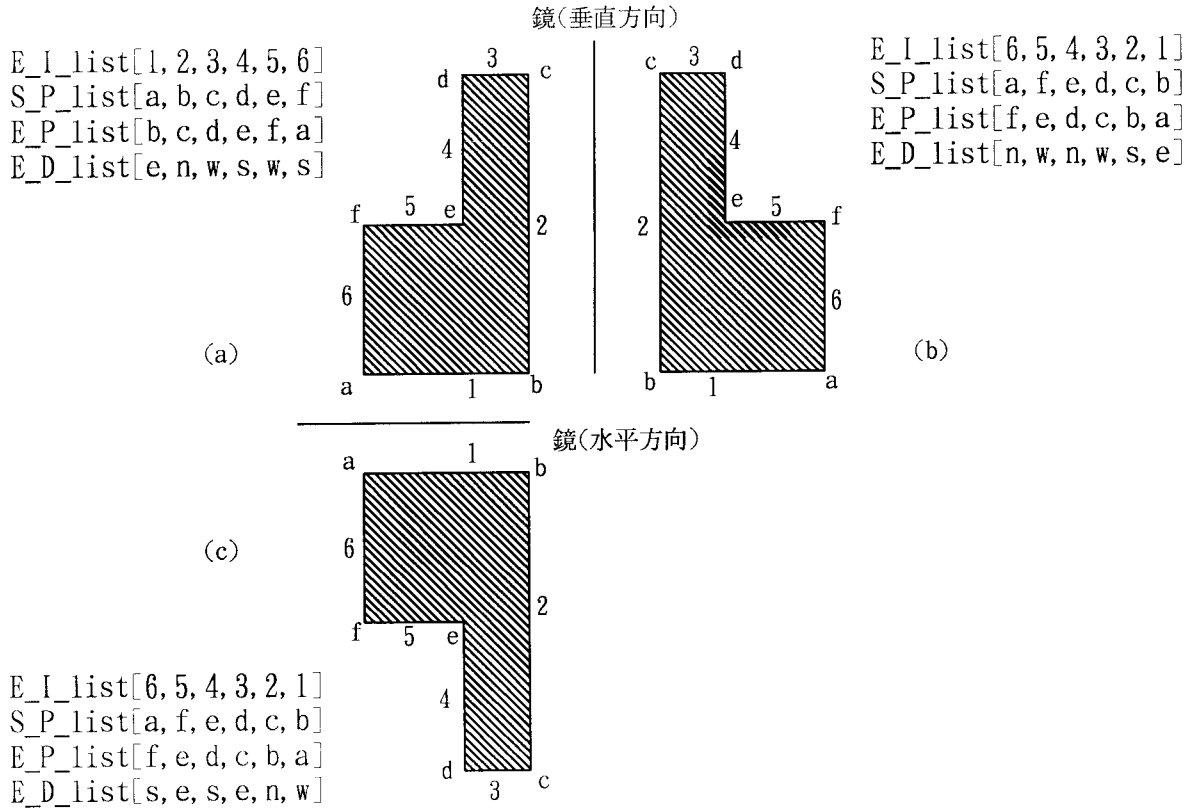


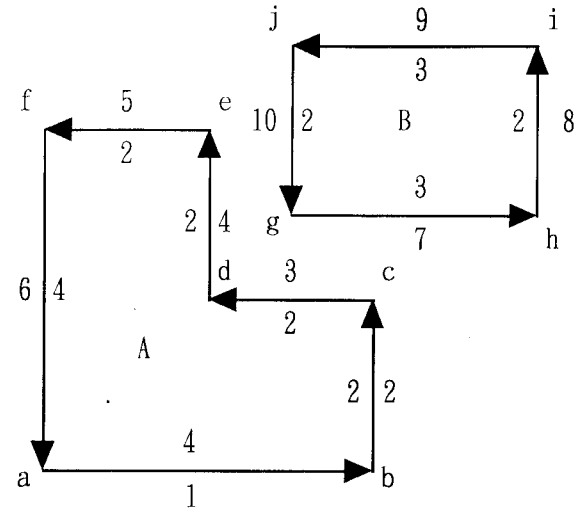
図2_10 鏡像操作.

（a） オブジェクトの例
（b） 垂直方向の鏡による鏡像
（c） 水平方向の鏡による鏡像

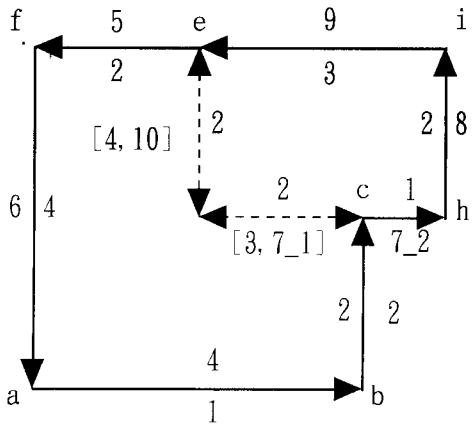
2.4.3.4　合成操作について

ここでは，図2_11（a）に示した2つのオブジェクトA, Bの頂点d, gでの接合を例に図形の合成をどのように行うかについて述べる．まず，図形A, Bに対応するリストと接

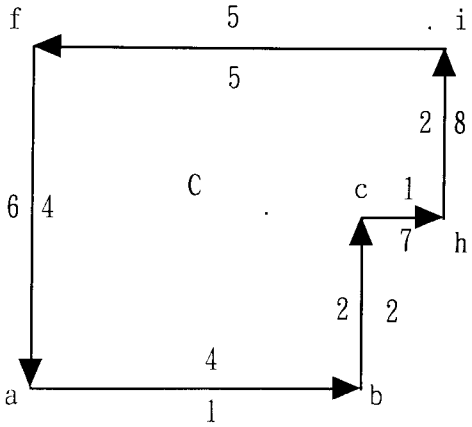
合点d, gが与えられるとシステムは以下のステップによる処理をシーケンシャルに実行する.



(a) 処理前



(b) 処理途中



(c) 処理後

Object A E_I_list[1, 2, 3, 4, 5, 6].
S_P_list[a, b, c, d, e, f].
E_D_list[e, n, w, n, w, s].
E_L_list[4, 2, 2, 2, 2, 4].
Object B E_I_list[7, 8, 9, 10].
S_P_list[g, h, i, j].
E_D_list[e, n, w, s].
E_L_list[3, 2, 3, 2].

↓ リストの回転

Object A E_I_list[4, 5, 6, 1, 2, 3].
S_P_list[d, e, f, a, b, c].
E_D_list[n, w, s, e, n, w].
E_L_list[2, 2, 4, 4, 2, 2].
Object B E_I_list[7, 8, 9, 10].
S_P_list[g, h, i, j].
E_D_list[e, n, w, s].
E_L_list[3, 2, 3, 2].

↓ 辺の分割

Object A E_I_list[4, 5, 6, 1, 2, 3].
S_P_list[d, e, f, a, b, c].
E_D_list[n, w, s, e, n, w].
E_L_list[2, 2, 4, 4, 2, 2].
Object B E_I_list[7_1, 7_2, , 8, 9, 10].
S_P_list[d, c, h, i, e].
E_D_list[e, e, n, w, s].
E_L_list[2, 1, 2, 3, 2].

↓ 内部辺の生成

Object A E_I_list[[4, 10], 5, 6, 1, 2, [3, 7_1]].
S_P_list[d, e, f, a, b, c].
E_D_list[n, w, s, e, n, w].
E_L_list[2, 2, 4, 4, 2, 2].
Object B E_I_list[-[3, 7_1], 7_2, 8, 9, -[4, 10]].
S_P_list[d, c, h, i, e].
E_D_list[e, e, n, w, s].
E_L_list[2, 1, 2, 3, 2].

↓ リストの接合

Object C
E_I_list[[4, 10], 5, 6, 1, 2, [3, 7_1], -[3, 7_1], 7_2, 8, 9, -[4, 10]].
S_P_list[d, e, f, a, b, c, d, c, h, i, e].
E_D_list[n, w, s, e, n, w, e, e, n, w, s].
E_L_list[2, 2, 4, 4, 2, 2, 2, 1, 2, 3, 2].

↓ 辺の縮退

Object C
E_I_list[5, 6, 1, 2, 7, 8].
S_P_list[i, f, a, b, c, h].
E_D_list[w, s, e, n, e, n].
E_L_list[5, 4, 4, 2, 1, 2].

(d) 形状リストの変化

図2_11 合成操作.

(1) S_P_listにおいて結合する頂点が先頭にくるようにすべてのリストを回転する.
(2) 2つのオブジェクトのE_D_listで，結合する頂点の前後の辺の要素（オブジェクトAの先頭とオブジェクトBの最後の要素あるいはその逆）を比較し，nとs，eとwの組み合わせであれば接合可能辺であるとする．接合する辺の長さを比較し，同じであれば一方の頂点名に名前を統一する．長さが異なれば，長い方の辺をもつオブジェクトに短い方の辺の頂点を挿入し，2つの辺に分解する．双方のE_I_listで対応する辺の名前をリストで表す（図2_11（b)).
(3) すべての対応するリストを結合する（図2_11（c)).
(4) E_I_listを循環リストと考え，同じ辺が続いていれば辺を削除する（図2_11（c)).
以上の処理過程での形状リストの変化の様子を図2_11（d）に示す．このような一連の自動処理シーケンスはProlog言語で簡潔に記述できる．図2_12に上記の処理（2）の一部をProlog言語で記述した例を挙げる．ここでは，Boxとbaseという2つのオブジェクトの結合に際し，結合される辺の長さが異なる場合の処理について記している．retractとassertaは，それぞれデータベースからのデータの削除とデータベースへの挿入を行うコマンドで，Boxの形状リストに新たな頂点V3と辺Nxを挿入している．

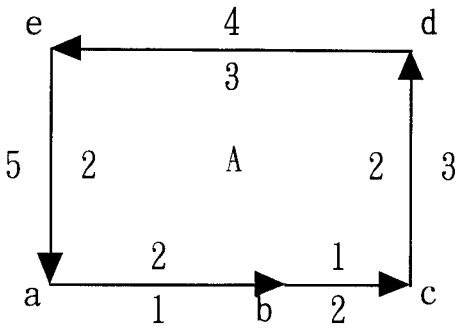
```
pack_box32(Box,V,Num2,L2,Lb):-
        L2 < Lb,
        s_p_list(base,Slist),
        num_of_list(V3,Slist,Num2),
        Lx is Lb - L2,
        retract(e_i_list(Box,[Na,Nb,Nc,Nd])),
        retract(s_p_list(Box,[V ,Vb,Vc,Vd])),
        retract(e_d_list(Box,[e ,n ,w ,s ])),
        retract(e_l_list(Box,[La,Lb,La,Lb])),
        call_n(Nx),
        asserta(e_i_list(Box,[Na,Nb,Nx,Nc,Nd])),
        asserta(s_p_list(Box,[V ,Vb,V3,Vc,Vd])),
        asserta(e_d_list(Box,[e ,n ,n ,w ,s ])),
        asserta(e_l_list(Box,[La,L2,Lx,La,Lb])),
      pack_box4(Box,V).
```

図2_12 Prologコード.

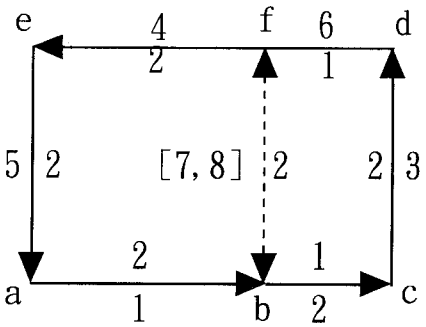
2.4.3.5　分解操作について

次に，図2_13（a）に示したオブジェクトAを頂点bを通る垂線で2つに分けるプロ

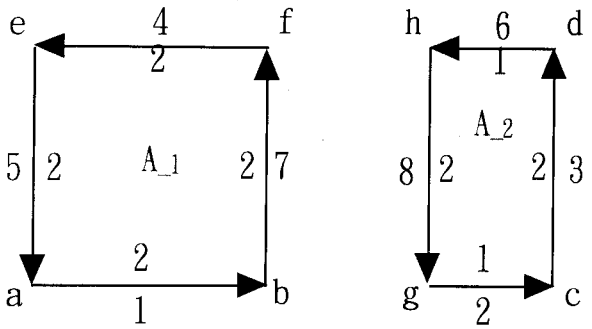
セスの例を示す．前例と同様に分解対象Aのリストと分解点bが与えられ，分解操作が指示されると，システムは以下の処理を実行する．



(a) 処理前



(b) 処理途中



(c) 処理後

Object A E_I_list[1, 2, 3, 4, 5].
S_P_list[a, b, c, d, e].
E_D_list[e, e, n, w, s].
E_L_list[2, 1, 2, 3, 2].

↓ 頂点の生成

Object A E_I_list[1, 2, 3, 6, 4, 5].
S_P_list[a, b, c, d, f, e].
E_D_list[e, e, n, w, w, s].
E_L_list[2, 1, 2, 1, 2, 2].

↓ 内部辺の挿入

Object A E_I_list[1, [7, 8], -[7, 8], 2, 3, 6, 4, 5].
S_P_list[a, b, f, b, c, d, f, e].
E_D_list[e, n, s, e, n, w, w, s].
E_L_list[2, 2, 2, 1, 2, 1, 2, 2].

↓ リストの回転

Object A E_I_list[-[7, 8], 2, 3, 6, 4, 5, 1, [7, 8]].
S_P_list[f, b, c, d, f, e, a, b].
E_D_list[s, e, n, w, w, s, e, n].
E_L_list[2, 1, 2, 1, 2, 2, 2, 2].

↓ リストの分割

Object A_1
E_I_list[4, 5, 1, [7, 8]].
S_P_list[f, e, a, b].
E_D_list[w, s, e, n].
E_L_list[2, 2, 2, 2].
Object A_2
E_I_list[-[7, 8], 2, 3, 6].
S_P_list[f, b, c, d].
E_D_list[s, e, n, w].
E_L_list[2, 1, 2, 1].

↓ リストの修正

Object A_1
E_I_list[4, 5, 1, 7].
S_P_list[f, e, a, b].
E_D_list[w, s, e, n].
E_L_list[2, 2, 2, 2].
Object A_2
E_I_list[8, 2, 3, 6].
S_P_list[h, g, c, d].
E_D_list[s, e, n, w].
E_L_list[2, 1, 2, 1].

(d) 形状リストの変化

図2_13 分解処理.

(1) 分解点bと時間座標が同じ点を通り，かつ分解点bを始点とする辺2と図2_13(b)に示した方向が逆となる辺4上に新たな頂点fを生成する．同時に辺4を点fで2つの辺4と6に分解する．

(2) 分解点bと新たに生成された頂点を結ぶ2つの内部辺を生成し，頂点bを終点とする辺の後に挿入する．この際，形状リストが一般性を失わずにループを形成するように辺の順番を決定する（図2_13(b)）．

(3) E_I_listにおいて，2つの内部辺がリストの先頭と最後尾にくるようにすべてのリストを回転する．

(4) S_P_listにおいて先頭の要素と同じ要素が出現する前で，すべてのリストを分割する．

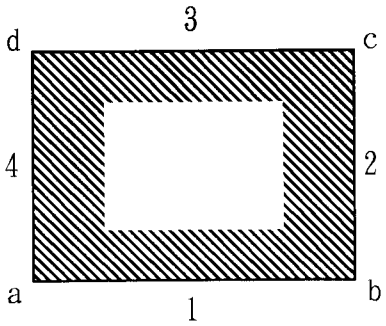
(5) E_I_listにおいて内部辺表現されている辺を正規の表現に修正する（図2_13(c)）．

2.4.4 CBR法の柔軟性・拡張性

前述のようにCBR法は，5つのリストから構成される形状モデルである．しかし，多様な対象に対する適用を考えたとき，すべてのリストが必要とはならない場合や，逆に形状の表現力をあげたい場合もあり得る．そこでCBR法を基礎とした2つの形状モデルを開発した．一つは簡易型符号化形態表現（Simplified Coded Boundary Representation 以下SCBR）法で，内部ループ（穴）を持たない形状だけを取り扱う．もう一つは改良型符号化形態表現（Modified Coded Boundary Representation 以下MCBR）法で，CBRよりも形状の表現力を高めた．

2.4.4.1 簡易型符号化形態表現（SCBR）法

SCBR法は，CBR法で用いた5つのリストのうち形状を構成する線分の終点を表す



(a)オブジェクトの例

E_I_list[1 ,2 ,3 ,4]
S_P_list[a ,b ,c ,d]
E_D_list[e ,n ,w ,s]
E_L_list[20,10,20,10]

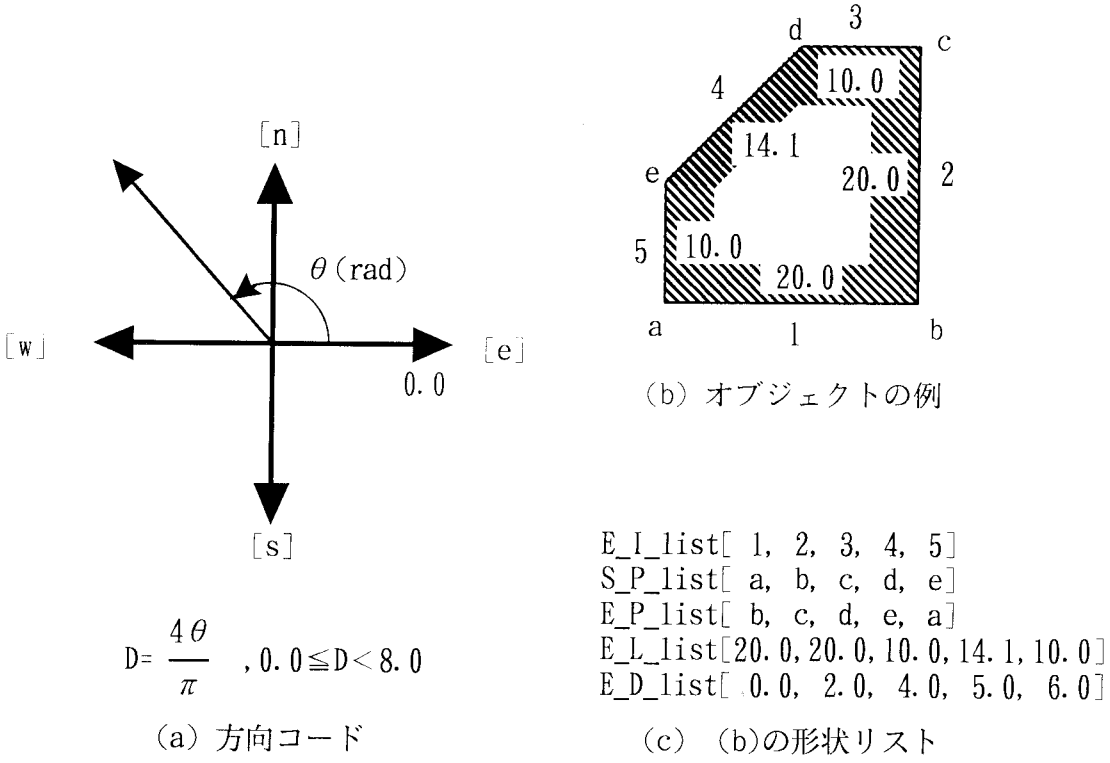
(b) (a)の形状リスト

図2_14 簡易型符号化形態表現（SCBR）法.

E_P_listを省いたモデルである（図2_14）．このモデルでは，各線分の終点は，線分の始点を表すS_P_listの次の要素となる．また，最後の線分の終点は，S_P_listの最初の要素として表される．このため，内部に別のループを有するような形状は，表現することができない．しかし，CBR法よりもデータ量が少なく，従って図形処理によるリストの操作回数も少ない．

2.4.4.2　改良型符号化形態表現（MCBR）法

CBR法は方向を表すコードが4方向しかない．これは形状を簡略に表現することで，物体形状の特徴を大まかに判断するためと，定性的な図形処理を可能にするためである．しかし，4方向で様々な形状を表現することは対象によっては非常に無理がある．そこでCBR法の形状表現力を上げるために次の改良を加えた．　CBR法では方向リスト（E_D_list）の要素が取り得る値はs, e, n, wの4方向であったが，MCBR法では図2_15に示すように，CBR法においてe方向を0とした反時計回りの角度θ（rad）をπ/4で除した実数D（$0.0 \leqq D < 8.0$）で表現した．



$$D = \frac{4\theta}{\pi}, \quad 0.0 \leqq D < 8.0$$

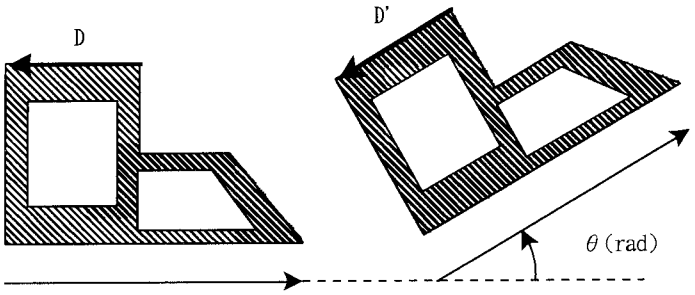
(a) 方向コード

(b) オブジェクトの例

(c) (b)の形状リスト

図2_15 改良型符号化形態表現(MCBR) 法.

この改良にともなってE_D_listの要素を用いる図形処理にも改良を加えた．例えば図形の回転を図2_16に示す．この例では，左の図形の任意の辺の方向コードDは，回転後（右の図形）D'で表される．この操作によって方向リスト（E_D_list）のすべての要素に対して上の処理を施すが，他のリストにはまったく操作を行わなくて良い．このように，改良型CBR法は，画像上の任意方向の線画を表現することができる．また後に示すように，複数の線画図形の位相的な関係を等式または不等式で表現できる．従って，CBR法より柔軟な図形処理を定義できる．



$$D' = D + \frac{4\theta}{\pi}$$

図2_16 MCBR法における回転操作.

2.5 結論

本章では，これまで開発された形状モデルを基に様々な対象に利用される形状モデルの考え方について述べ，提案したCBR法の有効性について検討した．その結果

・CBR法は，形状をいくつかのリストで表現しているため，形状の操作を簡単なリスト操作に置き換えることが可能である．

・CBR法は，方向コードの導入により形状の定性的な取り扱いが容易に可能である．

・リスト表現であるため，図形処理によるデータ量の変化に対して柔軟に対処できる．

・CBR法は，対象とする問題によってモデル自体の改良が容易に可能である．

等から，生産加工システムにおける各種の問題に対応し得る統一的なモデルとして有効であると考えられる．

第3章　リスト表現形状モデルの計画問題の画像化への応用

3.1　緒論

前章で，生産加工システムで用いる形状モデルのあり方とCBR法の基本的な概念・特徴について述べた．本章以降では，実際に多様な対象に対してCBR法を用いた形状モデルと図形処理の利用可能性について述べる．まず本章では，対象を計画問題とし，スケジューリング問題へのCBR法の適用について述べる．

製品の多様化，複雑化やライフサイクルの短縮，さらにCIM等の導入に伴い，生産の工程管理が一段と重要になってきている．　その中でスケジューリング問題は，整数変数を含む線形計画法を解く必要があり，最適解を求めることが非常に困難とされている．これまでスケジューリングを直感的に行ってきた専門家の代わりをするため，近似的に解を求める様々なアルゴリズムやシステムが開発されてきた[3_1][3_2]．

しかし，既存のシステムで生産現場のスケジューリングを実際に行う場合，直面している問題の仮定がシステムに合わないことが考えられる．例えば，上に挙げたシステムでは，各仕事で必要とされる作業員や機械の数が一定であるという条件が必要となるが，溶接施工等では，ある仕事に必要な作業員等が時間の経過とともに変化する場合が考えられる[3_3]．

本章ではこのような見地から，上で述べたようなプロジェクトに適したスケジューリングシステムの開発を目的とし，図形処理手法を応用したスケジューラを作成した．ここでは，PERTの山崩し法で取り扱う問題を図形処理を用いて解いていく計算機モデルとアルゴリズム，さらに，既存のシステムでは取り扱いが難しい問題への本手法の適用について述べる．具体的には，各作業を箱に，また各作業をスケジューリングしていく過程を表現する枠も大きさのない箱と定義し2次元の箱詰め問題に置き換えて解く．このように各作業を2次元の箱と定義することで，作業途中で必要となる作業員の数が変化するような作業を含むプロジェクトの取り扱いが可能となる．

3.2　スケジューリング問題の定義とPERTによる解法

3.2.1　取り扱うスケジューリング問題の定義

本章で取り扱ったスケジューリング問題はPERT山崩し法が対象としている問題で，以下のような前提条件をもつ[3_4].

・一つのプロジェクトは複数のそれ以上分割不可能な作業に分割できる.

・各作業は一度開始すると途中で中断することはできない.

・各作業にはそれを遂行するのに必要な時間と人数が与えられる.

・作業は他の作業と従属関係にある．たとえば，ある作業Bを開始するためには作業Aが終了している必要がある．このとき作業Aを作業Bの先行作業と呼び，逆に作業Bを作業Aの後続作業と言う．作業によっては先行作業や後続作業を複数もっているものもある.

・単位時間あたりにプロジェクト全体で利用できる作業員や機器の数が限られている.

このような条件下で，プロジェクト全体にかかる時間を最適化することを目標とする．プロジェクト全体で利用できる作業員や機器の数の制約がなければ解はPERTを利用することにより，比較的簡単に求められる．しかし，上記の制約を与えることで解析的に最適解を求めることが非常に困難になる．そこで簡易な山崩し法を利用して近似解を求めることがよく行われている.

3.2.2 PERTによる解法[3_5][3_6]

表3_1で与えられたプロジェクトを例に，PERTの山崩し法によるスケジューリングを述べる．表3_1では，作業とその各作業に対する先行作業さらにその作業を遂行するの

表3_1 プロジェクトのプロファイル例1.

Activity Name	Predecessor Name	Necessary Days	Necessary Workers
A	-	3	4
B	-	5	3
C	B	4	1
D	A	4	2
E	A	7	3
F	C, D	5	3
G	E, F	4	5

に必要な日数と作業員の数だけが与えられている．この表よりたとえば，作業Cを開始するためには，作業Bが終了していることが条件で，この作業を遂行するためには1人の作業員が4日かかることが判る．この表を用いて図3_1に示すようなアローダイアグラムを描く．図3_1では，矢印が各作業を表し，○で示された点はイベントと呼ばれ作業の開始または終了を表現している．各イベントで四角で囲まれた数字は，上段が最早結合点時刻（t_s）でそのイベントから開始されるすべての作業が最も早く開始可能となる時間を，下段が最遅結合点時刻（t_n）でプロジェクト全体にかかる時間を変えないためにそこを起点とするすべての作業が遅くとも開始しなくてはならない時間を表している．したがって，つねに$t_s \leqq t_n$の関係が成り立つ．また$t_s = t_n$となるイベントを両端にもつ太線のアローは，クリティカルパスと呼ばれ，このパス上の作業の開始及び終了時間を少しでも遅くするとプロジェクト全体にかかる時間が長くなる．

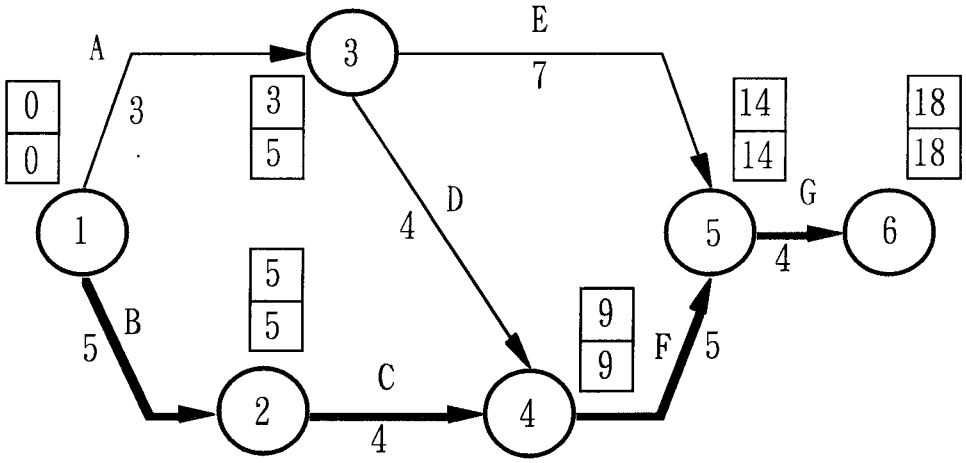


図3_1 アローダイアグラム.

図3_2は横軸に日数を，縦軸に作業員の数をとり，図3_1の各作業を矩形表現したも

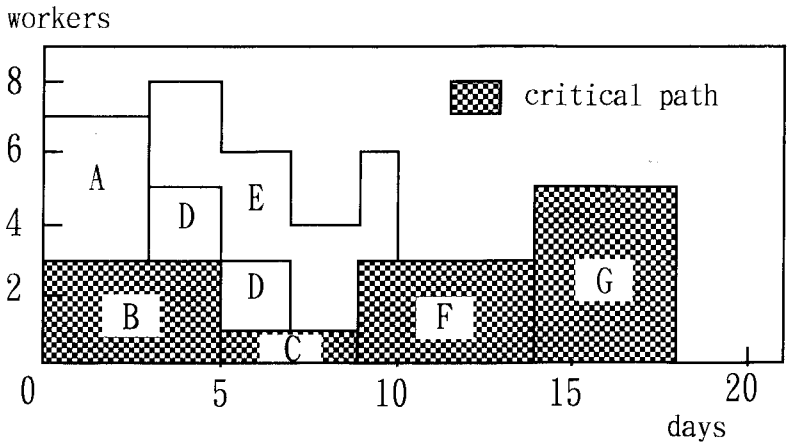


図3_2 PERTを利用したスケジューリングの結果.

のである．この図からプロジェクトを最短時間で終わらせるためには最大8人の作業員が必要で最短工期が18日になることが判る．

ここでプロジェクト全体で参加できる作業員が，7人という制限を受けると作業Eを2日遅らせて開始することで全体にかかる日数を変えずに対処できるが，6人以下に制限されるとプロジェクト全体の日数を増やす必要がある．PERTではこのような場合，山崩し法を用いてスケジューリングを行う．ここでは5人の制約を受けた場合について山崩し法の概略を説明する．図3_1で時刻0（プロジェクト開始日）において，開始可能な作業のなかで最遅開始時刻（L_S= 作業の終了点の最遅結合点時刻 - 作業の所要時間）がもっとも早い作業を優先する．この時刻0では，作業A（L_S=2）と作業B（L_S=0）が開始可能であるが，最遅開始時刻の早い作業Bが優先される．このように各時点で優先順位が高い作業から5人の制約条件を満たせばその作業を開始するようにスケジュールしていく．開始できる作業がなければ時刻を一単位増やして同様の操作を繰り返す．例に挙げた問題に対する山崩しの結果の図形表示を図3_3に示す．

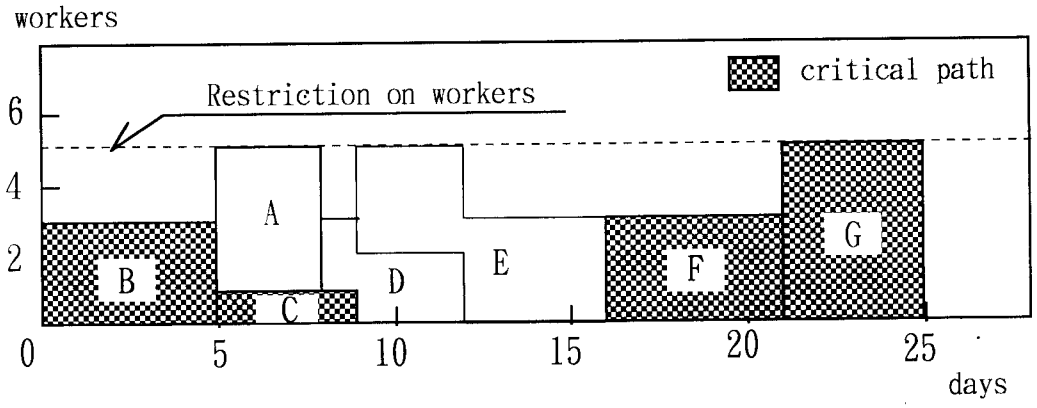


図3_3 PERT山崩し法によるスケジューリングの結果.

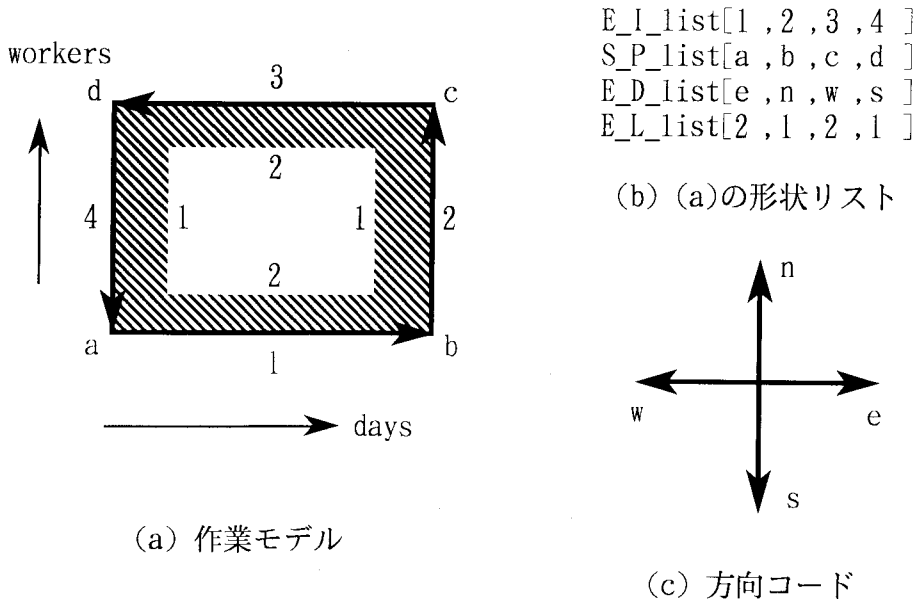
3.3　スケジューリング問題における形状モデル

図3_2，図3_3は，PERTおよび山崩し法で得たスケジューリングの結果を図形表示したものであり，各仕事が矩形表現された図形を用いてスケジューリングを行ったものではない．本項では，より柔軟なスケジューリングを目的とし，形状モデルと図形処理を用いたスケジューリングシステムについて述べる．

本システムでは形状モデルとして前章で概要を述べた簡易型符号化形態表現（SCBR）法を用いた．符号化形態表現法では，ある形状を表現するのに5つのリストを用いているが，SCBR法では4つのリストで表現している．ここで取り扱う2次元の箱詰め問題では，外形を表すループの中に別のループを持つ，つまりドーナッツ状の形状を扱うことがない．そこで，SCBR法では，E_P_listを考えずに，頂点の連結をS_P_listのみで表現する．本節では簡易型符号化形態表現法と箱詰め問題への適用に必要な図形処理について記す．

3.3.1　簡易型符号化形態表現（SCBR）法

本章で用いた図形モデルは，簡易型符号化形態表現（SCBR）法である．これは2.4.4.1に記したように，図3_4（a）に示された形状では，図3_4（b）の4つのリストで表現される．ここでもう一度各リストが表す内容について記すと，図3_4（b）で，E_I_listは各辺に与えられたユニークな番号で，図中の各有向辺の右側の数字を示しており，辺の識別に用いられる．また，辺番号がリストで表現されるものは内部辺と呼ばれ可視化はされないが，形状の操作時に利用される．S_P_listは，各辺の始点を表し，そのリストは形状を構成している各辺の連結を示している．E_D_listは各辺が形状の実体を左に見たときに図3_4（c）に示された4つの方向のうちの一つを示している．最後のE_L_listは各辺の長さを示している（図中の各辺の左側の数字）．



（a）作業モデル

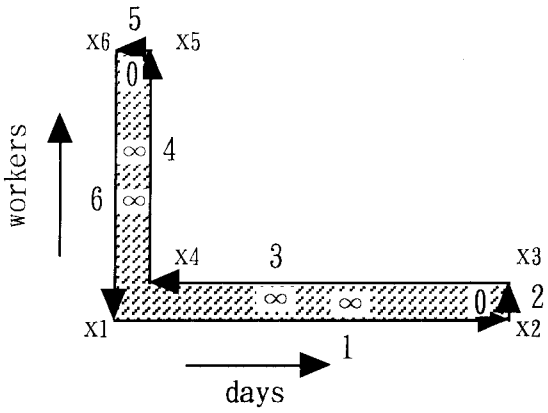
（b）(a)の形状リスト

（c）方向コード

図3_4 SCBR法による作業の形状モデル．

3.3.2　作業と枠の図形モデル

本システムでは，各作業は図3_4 (a) に示した矩形のオブジェクトを2次元の箱と考え，横方向を所要日数，縦方向を所要人数として表現するものとする．また各作業をスケジューリングしていく土台となるフレームは図3_5 (a) に示すように面積を持たない箱として定義している．図3_5 (b) にこの形状のリスト表現を示す．このように定義することで，システムが用いる図形処理が，後で述べる合成と分解のみに集約できる．



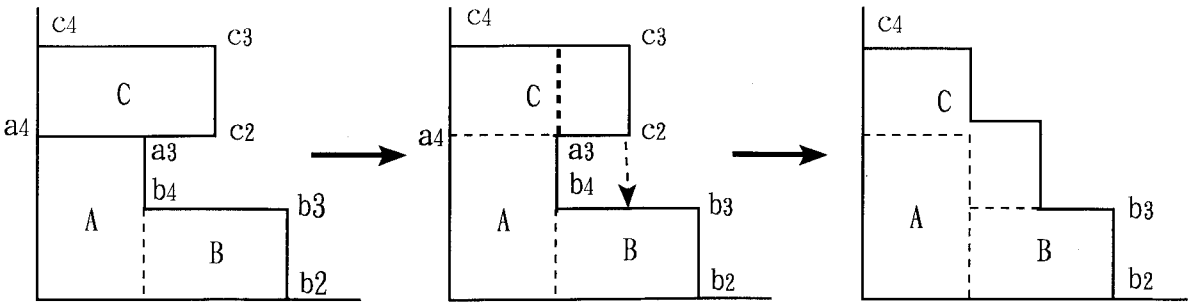
(a) フレームのモデル

E_I_list[1 ,2 ,3 ,4 ,5 ,6]
S_P_list[x1, x2, x3, x4, x5, x6]
E_D_list[e ,n ,w ,n ,w ,s]
E_L_list[∞, 0, ∞, ∞, 0 , ∞]

(b) (a)の形状リスト

図3_5 フレームの形状モデル.

以上のモデルをスケジューリングに適用した場合のスケジューリング過程の一例を図3_6に示す．図3_6 (a) は，作業AとBがスケジュールされた後に，作業Cが選ばれ，a_4に詰められる場合を想定している．その際，作業BとCの間に余剰空間ができる．この



(a) 余剰空間の発生　(b) 形状の分割　(c) 余剰空間の削除

図3_6 箱詰め手順.

余剰空間を削除するために，作業Cをa_3で分解し（図3_6（b）），スケジューリング空間に浮いている部分を余剰空間を作らないように，b_4に詰め直す処理（合成）が自動的に実行される（図3_6（c））.

このように形状モデルと図形処理を定義することで，スケジューリングを実行していくために必要な処理は，基本的に前章で述べた合成操作（2．4．3．4）と分解操作（2.4.3.5）のみである.

3.4　図形処理を用いたスケジューリング問題解法のアルゴリズム

これまでに，スケジューリングされる作業の優先順位を決定するアルゴリズムが数多く研究されてきたが，適用されるプロジェクトによってその成績が異なる[3_7]．そこで本システムでは図形処理の特徴を生かせる優先順位の決定法を採用した．本節ではシステムのスケジューリングアルゴリズムを説明し，さらにいくつかの問題に適用したときの結果とPERT山崩し法による結果との比較について述べる.

3.4.1　優先順位決定法

本システムでは，各作業をスケジューリングするための優先順位を以下のように決定する.

(1) 各作業について，その作業から最終作業に至るまでの所要日数の総和を求め最早終了日数とする．ただし，最終作業に至るまでの行程が複数ある場合には，各行程に対して所要日数の総和を求め，最大値を最早終了日数とする.

(2) 最早終了日数の大きい作業から高い優先順位を与える．最早終了日数が同じ場合には，それらの作業の中で所要日数の短い作業を優先する．さらに，所要日数も同じであれば，所要人数の少ない作業を優先する.

3.4.2　スケジューリング・アルゴリズム

本システムでは，表3_1のようなプロジェクトが与えられると，各作業に対し形状リストの作成，優先順位の決定を行い，基本的には優先順位の高い作業からできるだけ早く作業を開始するような以下のスケジューリングアルゴリズムにより，処理を実行する.

処理1　作業リストより，各作業の最早終了日数を求め優先順位を決定する．

処理2　3.3.2で記したように各作業とフレームの形状モデルを作成する．

処理3　優先順位の最も高い作業をフレームに箱詰め（合成）し，フレームの形状リストを更新する．さらに，箱詰めした作業の終了時刻を記録する．

処理4　優先順位にしたがって作業を取り出す．

処理5　注目している作業の先行作業がすべて終了している時刻以降に制約条件を充たし，できるだけ早く開始できる時刻を検索する．

処理6　作業をフレームの上記の位置に箱詰め（合成）する．その際，フレームと作業の間に余剰空間が発生したり，干渉したりする場合，注目している作業を分解し，それぞれをフレームに箱詰めする．その作業の終了時刻を記録する．

処理7　スケジューリングが終わっていない作業があれば処理4から繰り返す．

このように本手法では，優先順位が高い作業から人数の制約を充たしつつ，最も早い時間に開始されるようにスケジューリングされる．

3.5　スケジューリングテスト

本システムはPrologで記述されており，EWS（Sun Sparc Station 2相当）上に実装されている．システムは，スケジューリングの過程を図でCRT画面上に描画していく．本節では，与えられたプロファイルに対して，計算機が行う処理について順を追って述べる．さらに，本手法とスケジューリングの代表的な手法であるPERT山崩し法とを比較し，本手法の有効性を検討する．

3.5.1　スケジューリング例

ここでは，表3_1に示したプロジェクトのプロファイルに対し人数制限が5人という制約条件を与えられた問題のスケジューリングを例として，システムが行う過程を順を追って述べる．

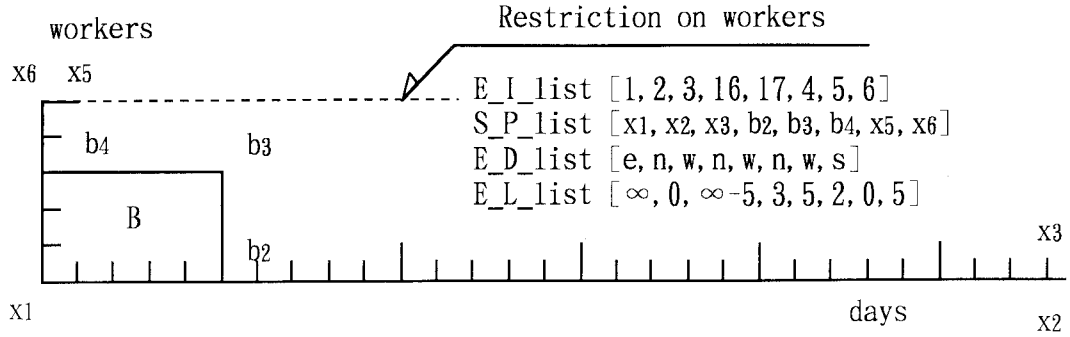
まず，表3_1から各作業の最早終了日数を求め優先順位を決定する（表3_2）．この表を作成する際，作業Bから作業Gに至る行程が18日で最も長く，クリティカルパスであることが判る．また，この時間が人数の制約を受けない場合の最短時間である．前述の

処理に従い，優先順位の高い作業からスケジューリングを行っていく．この過程を図

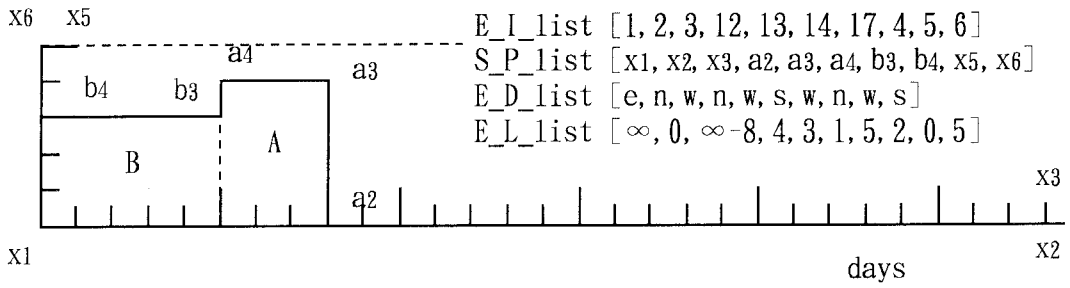
表3_2 作業の優先順位.

Activity Name	Fastest Finish Time	Priority Level
A	16	2
B	18	1
C	13	3
D	13	4
E	11	5
F	9	6
G	4	7

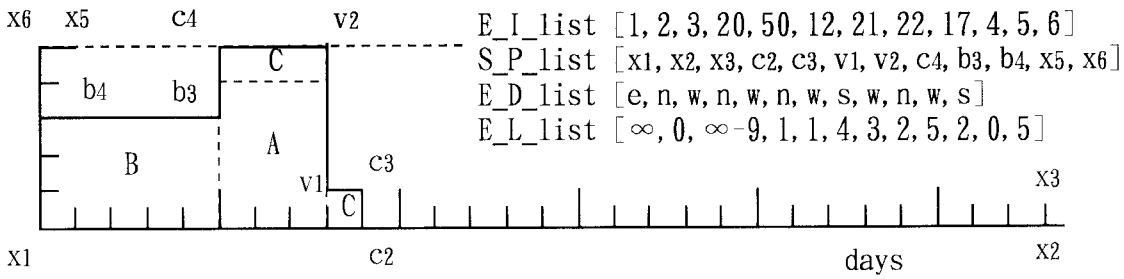
3_7 (a) ～ (c) に示す．ただし，フレームの目盛りは表示の際に加えられるものであって，形状リスト等の内部では表現されていない．また，図中のリストは，フレームの形状リストを示しており，作業がスケジューリングされる度に更新されていることが判る．図3_7 (a) では，優先順位により作業Bが選択され，フレームのx_4に箱詰めされる．つぎに図 (b) では，作業Aが選択され，箱詰めの候補点として頂点b_4, b_2が抽出されるが，b_4では作業員の制約を満たさないので，b_2が選択され箱詰めされる．さらに図 (c) では，作業Cが選択され，候補点として頂点b_4, a_4, a_2が抽出されるが，b_4では先行作業Bが終了していないことからa_4が選択され箱詰めされる．このような操作が優先順位に従って順次行われ，すべての作業のフレームへの箱詰めが終了した時点でスケジューリングが完了する．すべての作業をスケジューリングした結果を図3_7 (d) に示す．



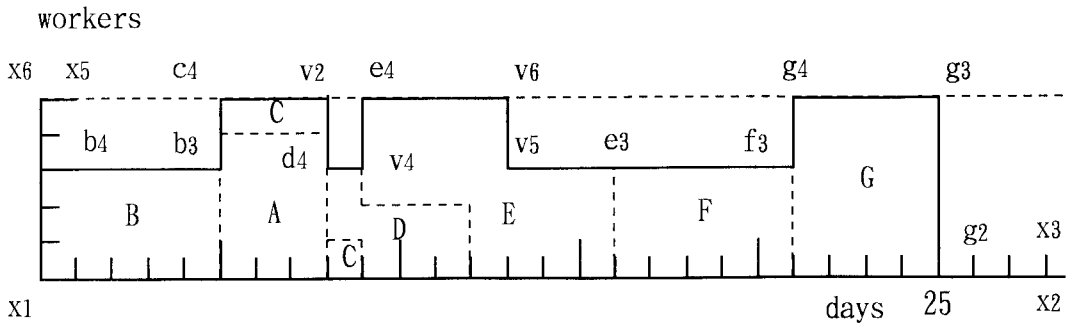
(a) Work Bの箱詰め



(b) Work Aの箱詰め



(c) Work Cの箱詰め



E_I_list [1, 2, 3, 36, 37, 38, 33, 53, 54, 29, 30, 25, 12, 21, 22, 17, 4, 5, 6]
S_P_list [x1, x2, x3, g2, g3, f3, e3, v5, v6, e4, v4, d4, v2, c4, b3, b4, x5, x6]
E_D_list [e, n, w, n, w, s, w, w, n, w, s, w, n, w, s, w, n, w, s]
E_L_list [∞, 0, ∞-25, 5, 4, 2, 5, 4, 3, 3, 2, 1, 2, 3, 2, 5, 2, 0, 5]

(d) 結果

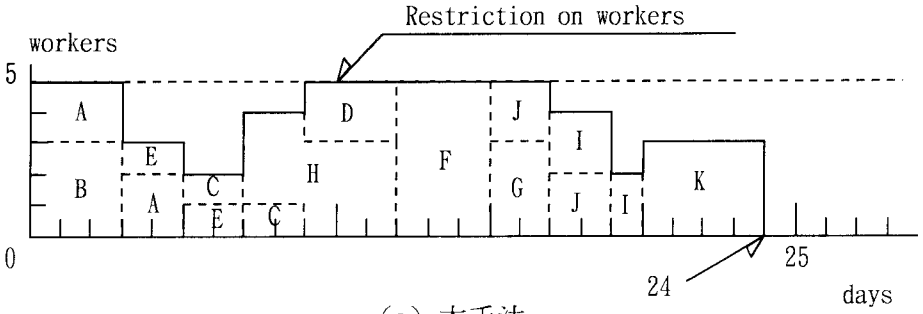
図3_7 スケジューリング手順と結果.

3.5.2　PERT 山崩し法との比較

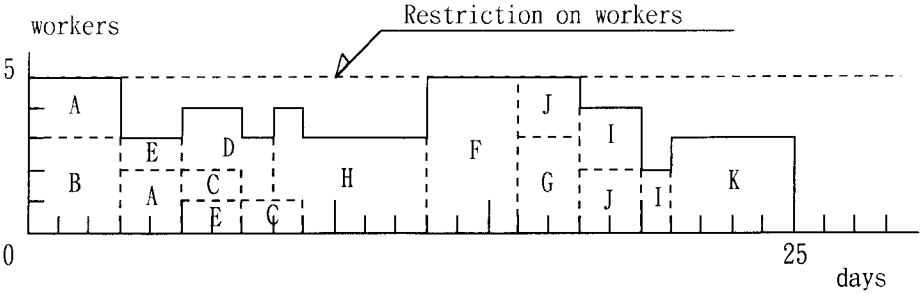
本手法によるスケジューリング結果図 3_7（d）と PERT 山崩し法による結果を図形で表示した図 3_3 では，まったく同じ結果となった．しかしながら，表 3_3 で与えられたプロジェクトに対する本手法と PERT 山崩し法による結果は図 3_8（a）（b）にそれぞれ

表3_3 プロジェクトのプロファイルと作業の優先順位.

Activity Name	Predecessor Name	Necessary Days	Necessary Workers	Priority Level
A	-	5	2	2
B	-	3	3	1
C	A	4	1	4
D	A	3	2	6
E	B	4	1	3
F	C	3	5	7
G	D	2	3	8
H	E	5	3	5
I	F, G	3	2	10
J	H	4	2	9
K	I, J	4	3	11



（a）本手法



（b）PERT山崩し法

図3_8 PERT山崩し法と本手法の比較.

示すように，少し異なったものになっている．

図3_8（a）（b）を用いてアルゴリズムの違いを検討する．まず，本手法での作業の優先順位は図3_8（a）に示すように，B，A，E，C，H，D，F，G，J，I，Kとなる．一方，PERT山崩し法によるスケジューリングの順序は図3_8（b）に示すように，B，A，E，C，D，H，F，G，J，I，Kとなり，本手法とは作業HとDの順番が入れ替わっている．この理由は以下のように考えられる．

本手法では，静的に優先順位を決定し，その後，優先順位に従って各作業の実施時刻を制約条件を満たすようにスケジューリングしている．つまり，図3_8（a）に示すように作業Hの開始時刻が決定されてから，Dの作業開始時刻が模索される．一方，PERTにおいては，作業Aが終了した時点では，すでに実施されている作業Eをのぞいて，CとDが作業実施可能となり，人数の制約条件を満たすので，双方ともに実行される．次に，作業Eが終了する時点において作業C，D，Hの中ではHのスケジュール全体の時間に与える影響が最も大きいが，人数の制約上作業Dが終了するまで実施が見送られている．このため本手法との順位の入れ替えが起こっている．このようにPERTの山崩し法では，時間を順に追っていくことでスケジューリングを行うのに対して，本手法では優先順位のみによって行っているので，局所にとらわれることなく大局的なスケジューリングが

表3_4 比較に用いたプロジェクト．

Project	1	2	3	4	5	6
Number of Activity	9	12	12	13	15	19
Earliest Scheduling Time	30	19	25	23	23	28

表3_5 本手法とPERT山崩し法の結果比較．　（days）

Project		1	2	3	4	5	6
Restriction on 5 Workers	This Method	39	30	29	33	32	40
	PERT	39	30	29	33	33	44
Restriction on 6 Workers	This Method	39	25	29	30	26	35
	PERT	39	25	29	30	28	33

可能となっている．表3_4に示した6つのプロジェクトに対して，作業員の制約条件が5人の場合と6人の場合のスケジューリングのPERT山崩し法による結果と本手法による結果を表3_5に示す．6番目のプロジェクトに6人の制約条件を設けた場合以外，本手法がPERT山崩し法と同等あるいはそれより良い結果を得た．このことより，本手法の有効性が示された．

3.6 所要人数変更問題への適用

本節では，PERT山崩し法では取り扱いが難しい問題（所要人数変更問題）に対する本手法の適用について述べる．まず取り扱った問題の定義と解法のための図形モデル，システムアルゴリズムについて述べ，さらに所要人数変更問題に対する本手法の適用例について述べる．

3.6.1 所要人数変更問題の定義

本節で取り扱うスケジューリング問題も基本的にはPERTの山崩し法が対象としている問題で，以下のような前提条件をもっている．

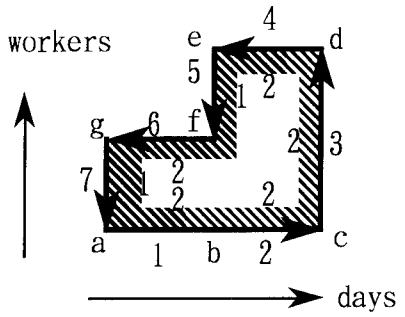
・一つのプロジェクトは複数のそれ以上分割不可能な作業に分割できる．
・各作業は一度開始すると途中で中断することはできない．
・各作業にはそれを遂行するのに必要な時間と人数が与えられる．
・ある作業は他の作業と従属関係にある．
・単位時間あたりにプロジェクト全体で利用できる作業員や機器の数が限られている．

一般的には一つの作業において，時間の経過に従って必要とされる作業員の数が変更されることが許されない．これは，「各作業はプロジェクトを構成する最小単位であるので，その作業においては必要とする作業員の数が変わらない」という立場に立っている．しかし実際の溶接施工等では，ある作業が，必要な作業員の数が異なる先行作業の終了直後に開始されなければならない，あるいは一つの作業でも時間の経過とともに作業人員の数が変わるなど，各作業で必要とされる作業員の数が変更されるような状況は十分考えられる．そこで本節では，このような問題を所要人数変更問題と呼び，従来のスケジューリング問題の定義の拡張を行う．本手法では，スケジューリングに図形処理

を用いることでこのような問題の取り扱いが容易となる.

3.6.2　形状モデル

本システムでは前節と同じく形状モデルとして簡易符号化形態表現（SCBR）法を用いた．所要人数変更問題においては，作業は，例えば図3_9（a）のような形状と（b）に示すリストで表現される．ここで形状としては，aからcは一つの直線で表現できるが，スケジューリングを考えた場合，頂点bを後の処理で利用するので，辺の縮退を行わない.



E_I_list [1, 2, 3, 4, 5, 6, 7]
S_P_list [a, b, c, d, e, f, g]
E_D_list [e, e, n, w, s, w, s]
E_L_list [2, 2, 2, 2, 1, 2, 1]

（a）作業モデル　　（b）(a)の形状リスト

図3_9 所要人数変更問題における作業の形状モデル.

3.7　スケジューリングテスト

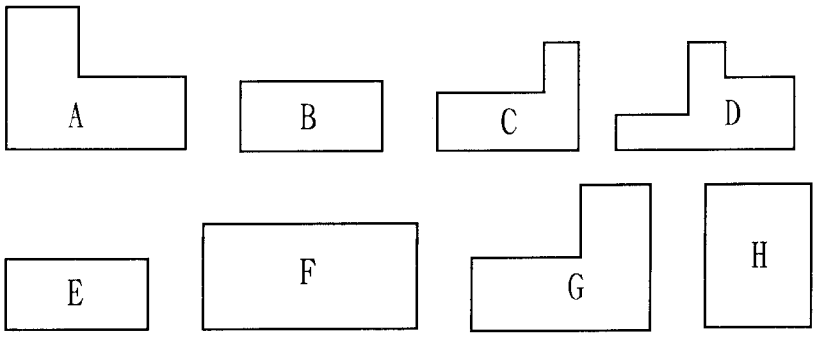
所要人数変更問題をスケジューリングする場合も3.4節で述べたアルゴリズムをそのまま適用することが可能である．3.4節で述べた手順は，各作業の形状モデルの形にとらわれない.

所要人数変更を含むプロジェクト（表3_6）のスケジューリング例を図3_10に示す．このプロジェクトの各作業の形状モデルを（a）に，またこのプロジェクトに5人という制約条件を与えた場合の本手法によるスケジューリング結果を（b）に示す.

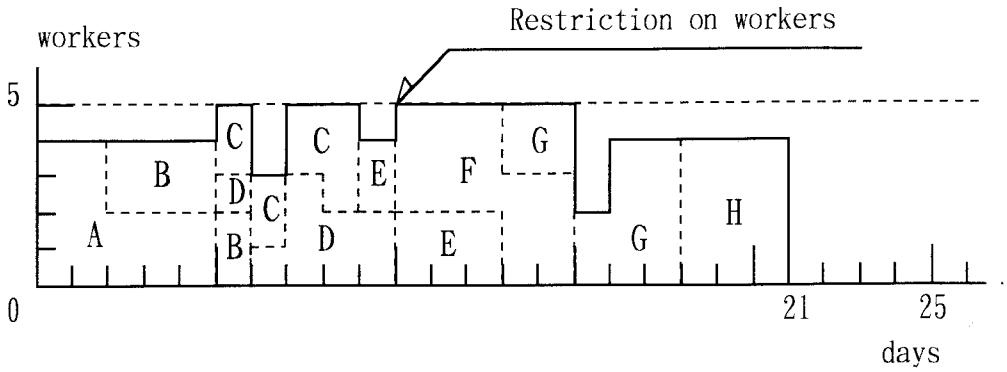
前述のように本手法では，スケジューリングを始める前に決定された作業の優先順位に従って各作業がどの時点から開始できるかを箱詰め問題に置き換えて検索していくために，前節で取り扱ったオーソドックスな問題に対するアプローチと同じ手法を用いることができる.

表3_6 所要人数変更問題のプロファイル例.

Activity Name	Predecessor Name	Necessary Days	Necessary Workers	Priority Level
A	-	2	4	1
		3	2	
B	-	4	2	2
C	A	3	2	4
		1	3	
D	A	2	1	3
		1	3	
		2	2	
E	B	4	2	5
F	C	5	3	6
G	D, E	3	2	7
		2	4	
H	F, G	3	4	8



(a) 作業の形状モデル



(b) スケジューリング結果

図3_10 所要人数変更問題のスケジューリング例.

本手法の適用結果を定量的に評価するために，次式で平均労働率を定義する．

$$Aw = \frac{\sum_{i} \mathrm{Area}(i)}{(\mathrm{Total\ days}) \times (\mathrm{workers})} \qquad \cdots (3_1)$$

ただし，Area（i）は，i番目の仕事の面積（人×日）である．

この例では，作業員の平均労働率が84.8%もの高い値を示し，本法の実用性，有効性を表わしている．

一方，PERT山崩し法では，ある時点において開始できる作業を検索するため，作業の開始時と作業進行途中で人数の変更がある作業を含むような問題の取り扱いが難しくなる．

3.8　結論

本章ではリスト表現形状モデルSCBR法を計画問題に適用し，新しいスケジューリング・システムを提案した．その結果について述べる．

1）プロジェクトの各仕事やフレームを形状モデルで表現することで，スケジューリング問題を2次元の箱詰め問題に置き換えることができる．

2）既存の手法であるPERT及び山崩し法が扱い得る問題に対して，同等ないしはそれ以上の良い結果が得られる．

3）PERT山崩し法が対応できない優先順位の決定法に対処できる．

4）所要人数変更問題等，既存のスケジューリングシステムが対処できなかった問題への対応が可能である．

さらに，形状モデルの拡張により，時間に対して線形的に必要な作業員の数が増減する作業を含んだプロジェクト等にも対応可能である．

第4章　リスト表現形状モデルの画像の変換への応用
（MCBR法を用いたコンピュータグラフィックスの生成）

4.1　緒論

コンピュータグラフィックス（以下CGと言う）は，各種計算機シミュレータやバーチャルリアリティーばかりでなく，画像認識にも利用されている．モデルベースドビジョン[4_1]，CADベースドビジョン[4_2]は，入力画像から物体の個々の頂点や面，境界線等の特徴を抽出し，あらかじめ計算機内部に記憶している物体のモデルから予想される特徴と照合し，物体の種類，位置，姿勢を決定する手法である．これらの手法はロボットの作業する環境があらかじめ限定され，また対象物の幾何学的なモデルを獲得することが容易である場合に非常に有効である．さらに，アスペクトグラフを利用した画像認識手法は，あらかじめ対象物を様々な方向から見た「見え方」（アスペクト）を計算機内部に保存しておき，カメラからの画像と比較することで対象物体の認識やカメラと対象の相対的な位置関係を求める手法である[4_3][4_4]．

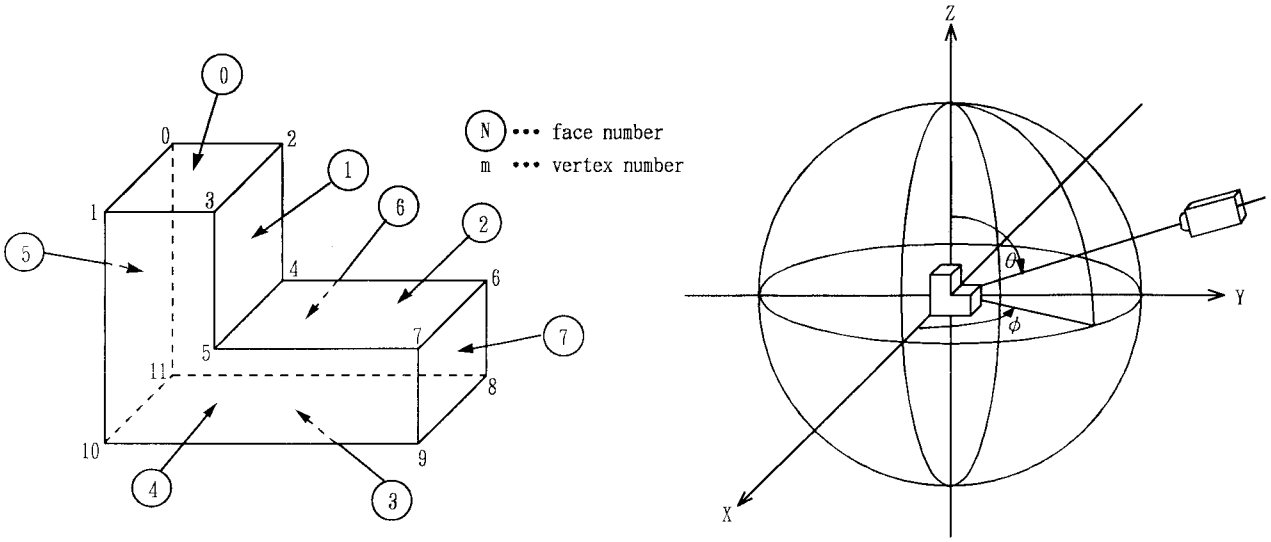
本章では，CBR法を改良したMCBR法を2次元の形状モデルとして用い，3次元のCADモデルから2次元のアスペクトを自動生成する手法の開発について述べる．具体的には，3次元の形状モデルで記述された対象物を計算機内部の仮想的な空間中で固定し，カメラを対象物を中心においた球面上を等間隔で移動させたときの対象物の「見え方」をMCBR法で記述した2次元の形状モデルとして作成する．このようにして，3次元から2次元画像面へ投射された図形は，3次元モデルの各頂点の2次元座標として与えられる．この際，実際には見えない頂点・辺・面の情報が含まれているために，隠面処理が必要となる[4_5]．最近，CG等で良く利用されているZ-buffer法[4_6]をはじめとする隠面処理はそのほとんどがピクセル単位の計算を行っているため，大局的な要素（辺・面）を容易に抽出することが困難で，必ずしも画像認識用のアスペクト生成に適しているとは言えない．画像から抽出した線分とアスペクトを比較する際，アスペクト側の情報として線分や面のデータが必要となる．そこで本章ではMCBR法を利用した新たな隠面処理手法について述べる．この手法は，各面の位相関係とその面を構成する線分同士の関係に着目し，結果として面の情報を得るものであり，画像認識用として望ましいものである．

4.2　MCBR法による対象の記述

まず，3次元モデルからのアスペクトや画像から得られたデータを基に，作成する対象物体の2次元形状をMCBR法によって記述する．この場合，物体を構成する各面に対して，それぞれ第2章で示したリストを作成する．これらのリスト中には，同じ辺で方向が逆の辺が現れることもあるがこの場合，E_I_listのID番号の符号を変えて区別する．これらのリストは，この対象物体の形状のもつ位相的な特徴（面の数や形，面の相互関係，平行な辺）を有していることになる．

4.3　MCBR法を利用する3次元CADモデルからの2次元データの生成

本項ではMCBR法を2次元の形状モデルとして用い，3次元のCADモデルから2次元のアスペクトを自動生成する手法について述べる．具体的には，図4_1（a）に示すような3次元の形状モデル（B-reps）で表現された対象物体を図4_1（b）に示す仮想的な空間中で固定し，カメラを球面上を移動させたときの対象物の「見え方」を2次元の形状モデルとして作成する．このとき，3次元空間での点（X,Y,Z）は，式（4_1），（4_2）に



（a）3次元モデル　　（b）仮想空間

図4_1 アスペクトグラフの概念図.

より2次元画面上の点（x, y）に投射される[4_7]．ここで，ϕは視線とx軸とのなす角，θは視線とz軸とのなす角，ρは視点から原点までの距離，dは画像面から原点までの距離とする．

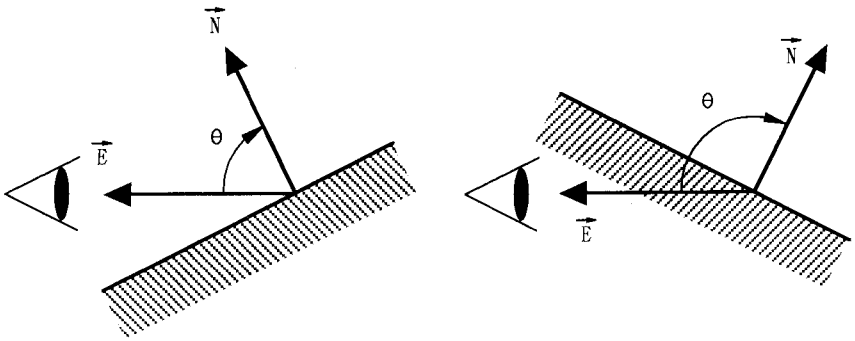
$$(V_x, V_y, a) = (X, Y, Z, 1)\begin{bmatrix} (d-\rho)\sin\theta & (d-\rho)\cos\phi\cos\theta & -\sin\phi\cos\theta \\ -(d-\rho)\cos\theta & (d-\rho)\cos\phi\sin\theta & -\sin\phi\sin\theta \\ 0 & -(d-\rho)\sin\phi & -\cos\phi \\ 0 & 0 & \rho \end{bmatrix} \quad \cdots (4_1)$$

$$x = \frac{V_x}{a}, y = \frac{V_y}{a} \quad \cdots (4_2)$$

4.3.1　前処理

仮想空間に置いた3次元モデルを式（4_1），（4_2）を利用して2次元に変換し，MCBR法で表現されたデータが与えられているとする．このデータには，全部あるいは一部分が実際には見えない面のデータもすべて含まれている．対象物体が凸多面体の場合，図4_2に示すように3次元空間上での面の法線ベクトルを$\vec{N}$，その点から視点方向に向かうベクトルを$\vec{E}$とすると，その面の可視・不可視の判定は式（4_3）で行える[4_8]．

$$\vec{N}\cdot\vec{E}\begin{cases} > 0 \text{ visible} \\ \leq 0 \text{ unvisible} \end{cases} \quad \cdots (4_3)$$



（a）可視面　　（b）不可視面

図4_2 面の法線ベクトルを用いた前処理.

この式はまた凹多面体においても，ある面が不可視となるための必要条件となる．この処理を行うことで，図4_3に示すような形状を得る．このデータから，まず隠面処理を施す必要がある面の検索を行い，続いて形状データの修正を行う．

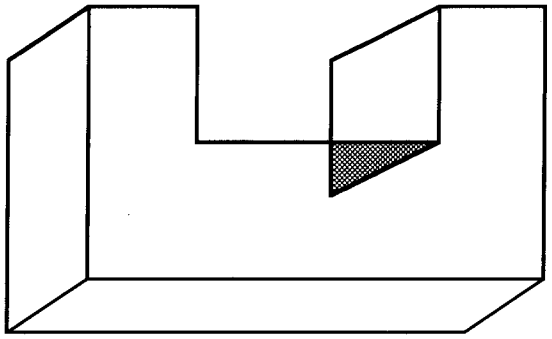


図4_3 面の法線ベクトルを用いた隠面処理の結果.

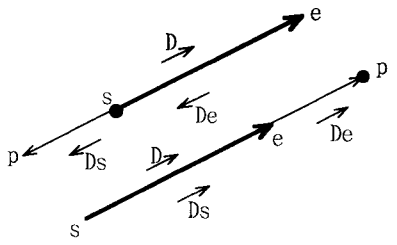
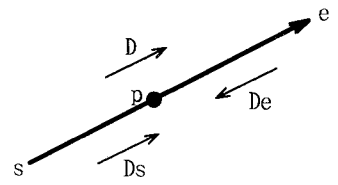
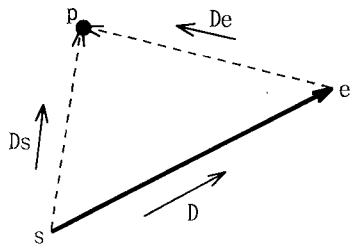
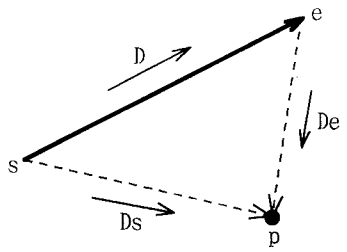
4.3.1.1　　2つの2次元多角形の位相関係

隠面処理を施す必要がある面の検索を行うためにまず，得られた形状データのすべての面に対して他の面との位相関係（交わらない・接する・交わる）を調べる．検索にはMCBR法のデータを利用する．ここでは，図形を構成する要素（頂点，辺，面）の位相関係から面同士の位相関係を調べる手法について述べる．

a　辺と点の関係

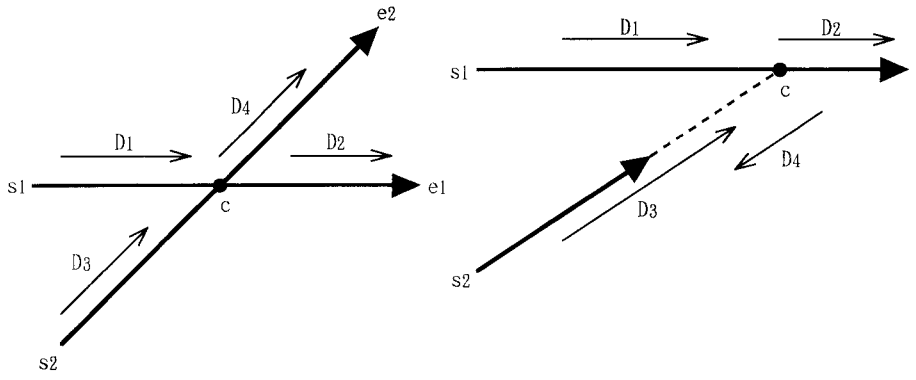
この関係は，面と点の位相関係を判定するための基準の一つとして用いる．辺と任意の点の関係を辺の左側，右側，辺上，始点上，終点上，延長線上の6つに分類する．特に始点，または終点上に対象の点があるときは直ちに判定できる．残りの4つの関係に分類するためにE_D_listの要素である辺の方向Dと始点から対象の点pに向かう方向Ds，終点から点pに向かう方向Deを求め，表4_1の関係を用いて判定を行う．

表4_1 線分と点の関係.

(a) 点が線分の延長上にある場合	
D=Ds±4.0, D=De±4.0 or D=Ds, D=De	
(b) 点が線分上にある場合	
D=Ds, D=De±4.0	
(c) 点が線分の左側にある場合	
If 0.0≦D≦4.0 Ds>D, De>D, |Ds-D|<4.0, |De-D|<4.0 If 4.0<D<8.0 Ds<D, De<D, |Ds-D|>4.0, |De-D|>4.0 or Ds>D, De>D	
(d) 点が線分の右側にある場合	
If 0.0≦D≦4.0 Ds<D, De<D or Ds>D, De>D, |Ds-D|>4.0, |De-D|>4.0 If 4.0<D<8.0 |Ds-D|<4.0, |De-D|<4.0	

b　辺と辺の交差条件

この関係は，2つの面が交差する条件で使われると共に面と点の位相関係を判定するための基準の一つとしても用いられる．図4_4に示すように，始点s_1，終点e_1の辺と始点s_2，終点e_2の辺が与えられている．また，2つの辺の交点をcと置き，s_1からcに向かう方向コードをD_1，cからe_1に向かう方向コードをD_2，s_2からcに向かう方向コードをD_3，cからe_2に向かう方向コードをD_4とする．このとき，2つの辺が交差する条件は



(a) 交差する　　(b) 交差しない

図4_4 2本の線分の交差条件.

式（4_4）で与えられる.

$$D_1 = D_2 \wedge D_3 = D_4 \qquad \cdots (4_4)$$

c　面と点との位相関係

MCBR法で記述された形状において，ある面ループと任意の点が与えられた場合，対象の点は，面ループの内部，外部，面ループを構成する辺，あるいは頂点上のいずれかに存在する．このような分類をするためには，面ループを構成する各辺と対象の点の関係を調べる必要がある．凸多角形の場合，すべての辺に対して対象となる点が左側に存在すれば面の内部，一つの辺でも点が右側にあれば外部にあると判定可能である．また，凹多角形の場合は更に辺と辺の関係を用いて判定する．図4_5に示すように，ある辺の

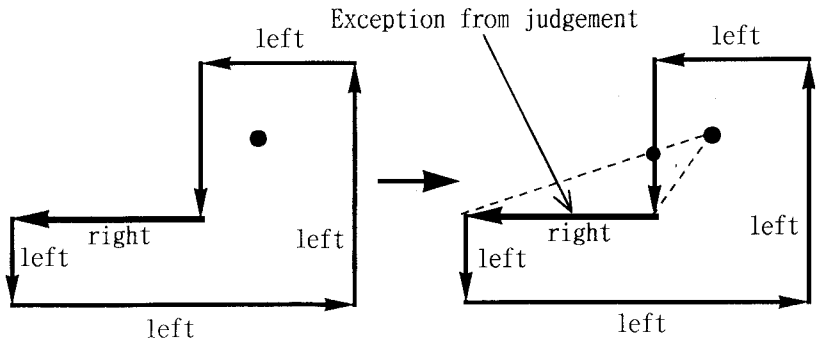


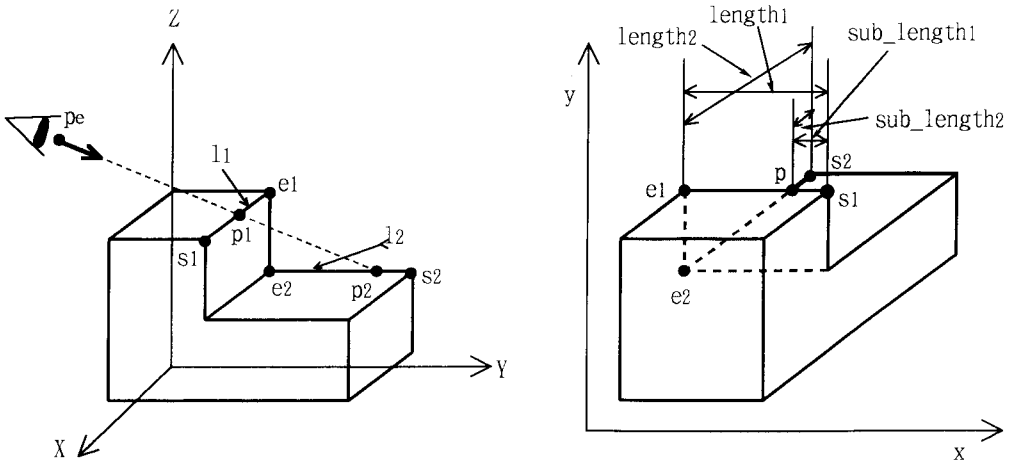
図4_5 点の包含条件.

右側に対象となる点が存在する場合，その辺の始点，終点から対象となる点に線分を引き，いずれの線分も辺と交わらなければ，対象の点は面ループの外側にあると判定する．また，いずれかと交わる場合にはこれを点の内外判定から除外し，それ以外の辺と点の関係を用いて判定を行う．以上の操作で面ループと任意の点の関係が得られる．

さらに，2つの面の位相関係は，bとcの関係を2つの面を構成する要素（辺・頂点）について行うことで判定可能である．

4.3.1.2　　2つの稜線と視点との位置関係

つぎに，2つの面が交差すると判定された場合，3次元空間中での視点から見た2つの面の前後関係を求める必要がある．しかし，2面の前後関係を2次元データだけから判断するのは難しい．そこで以下のような処理を行い，交点の3次元空間上での視点との位置関係から面の前後関係を判断する．図4_6（a）のように始点s_1，終点e_1からなる稜線をl_1，始点s_2，終点e_2からなる稜線をl_2とする．視点Peから見た対象モデルは図4_6（b）に示すような2次元画像になる．このときの図4_6（b）における交点pは，図4_6（a）のp_1, p_2が重なった点であるから交点pでのl_1とl_2の稜線の奥行きを判定すればよい．しかし，p_1，p_2の正確な3次元座標を計算するのは処理的に無駄が多い．つまり，実際に必要となるのはp_1とp_2と視点との位置関係であり，p_1とp_2の絶対的な3次元座標ではない．そこで簡便な稜線の奥行きの判定方法を用いる．



（a）3次元空間　　（b）2次元画面上

図4_6 視点と2直線の交点との関係.

2次元の座標より交点p（x_p, y_p）を求める．交点pと各辺の始点s_1（x_{s1}, y_{s1}），s_2（x_{s2}, y_{s2}），終点e_1（x_{e1}, y_{e1}），e_2（x_{e2}, y_{e2}）より$length_1$, $length_2$, sub_length_1, sub_length_2を式（4_5）より求める，さらに式（4_6）に示す比を求める．

$$\begin{cases} length_1 = \sqrt{(x_{e1}-x_{s1})^2+(y_{e1}-y_{s1})^2} \\ sub_length_1 = \sqrt{(x_p-x_{s1})^2+(y_p-y_{s1})^2} \\ length_2 = \sqrt{(x_{e2}-x_{s2})^2+(y_{e2}-y_{s2})^2} \\ sub_length_2 = \sqrt{(x_p-x_{s2})^2+(y_p-y_{s2})^2} \end{cases} \quad \cdots (4_5)$$

$$H_1 = \frac{sub_length_1}{length_1}$$
$$H_2 = \frac{sub_length_2}{length_2} \quad \cdots (4_6)$$

式（4_6）より求めた比を3次元空間にそのまま適用する．つまりl_1, l_2の各始点と終点の3次元座標s_1（X_{s1}, Y_{s1}, Z_{s1}），e_1（X_{e1}, Y_{e1}, Z_{e1}），s_2（X_{s2}, Y_{s2}, Z_{s2}），e_2（X_{e2}, Y_{e2}, Z_{e2}）よりp_1, p_2のおおよその3次元座標p_1^*（X_{p1}^*, Y_{p1}^*, Z_{p1}^*），p_2^*（X_{p2}^*, Y_{p2}^*, Z_{p2}^*）を求める．

$$\begin{pmatrix} X_{p1}^* \\ Y_{p1}^* \\ Z_{p1}^* \end{pmatrix} = \begin{pmatrix} X_{s1} \\ Y_{s1} \\ Z_{s1} \end{pmatrix} + H_1 \cdot \begin{pmatrix} X_{e1}-X_{s1} \\ Y_{e1}-Y_{s1} \\ Z_{e1}-Z_{s1} \end{pmatrix}$$
$$\begin{pmatrix} X_{p2}^* \\ Y_{p2}^* \\ Z_{p2}^* \end{pmatrix} = \begin{pmatrix} X_{s2} \\ Y_{s2} \\ Z_{s2} \end{pmatrix} + H_2 \cdot \begin{pmatrix} X_{e2}-X_{s2} \\ Y_{e2}-Y_{s2} \\ Z_{e2}-Z_{s2} \end{pmatrix} \quad \cdots (4_7)$$

式（4_7）で求められたp_1^*（X_{p1}^*, Y_{p1}^*, Z_{p1}^*），p_2^*（X_{p2}^*, Y_{p2}^*, Z_{p2}^*）と視点の3次元座標からp_1^*と視点までの距離およびp_2^*と視点までの距離を求め，これらを比較して 面の前後関係を判定する．

4.3.2　隠面処理を施した2次元データの作成

前項までの操作で隠面処理を施すべき面とその前面さらに，2つの面の交点が明らかとなった．本項では，MCBR法で記述された形状データの修正手順について述べる．ここでは，これら2面が2つの交点を有する場合についてのみ記すが，その他の場合もほぼ同様の手順か，その繰返しで隠面を処理可能である．

手順1　2つの交点を含む面と辺を特定する．

手順2　両面の形状データにおいて該当する辺上に頂点（交点）を加え，辺をそれぞれ交点で2分する．

手順3　隠面側のデータで，2つの交点に挟まれたデータを削除する．

手順4　前側の面のデータで，処理する2つの交点に挟まれたデータを隠面側のデータに加える．

（手順5　必要であれば，前側の面のデータを処理前に戻す．）

図4_7は，上の手順を計算機が行った形状データの修正過程を示している．

以上の操作によりMCBR法で表現された適切なアスペクトが得られる．

また，図4_1（a）に示した形状をある視点から見たときの隠面処理の例を図4_8（a）～（c）に示す．このように複数の面が1つの面を隠す場合でも上の操作を逐次行うことで処理可能である．

図4_8の形状リストから，この視点からの対象物体のアスペクトは，0,2,4,5の4面が視認可能で，各面の形状リストの長さより3角形，2つの4角形，6角形から構成されていることが判る．また，E_D_listから平行な線分，E_I_listから各面の繋がり，E_L_listから各辺の長さ等も容易に抽出可能である．

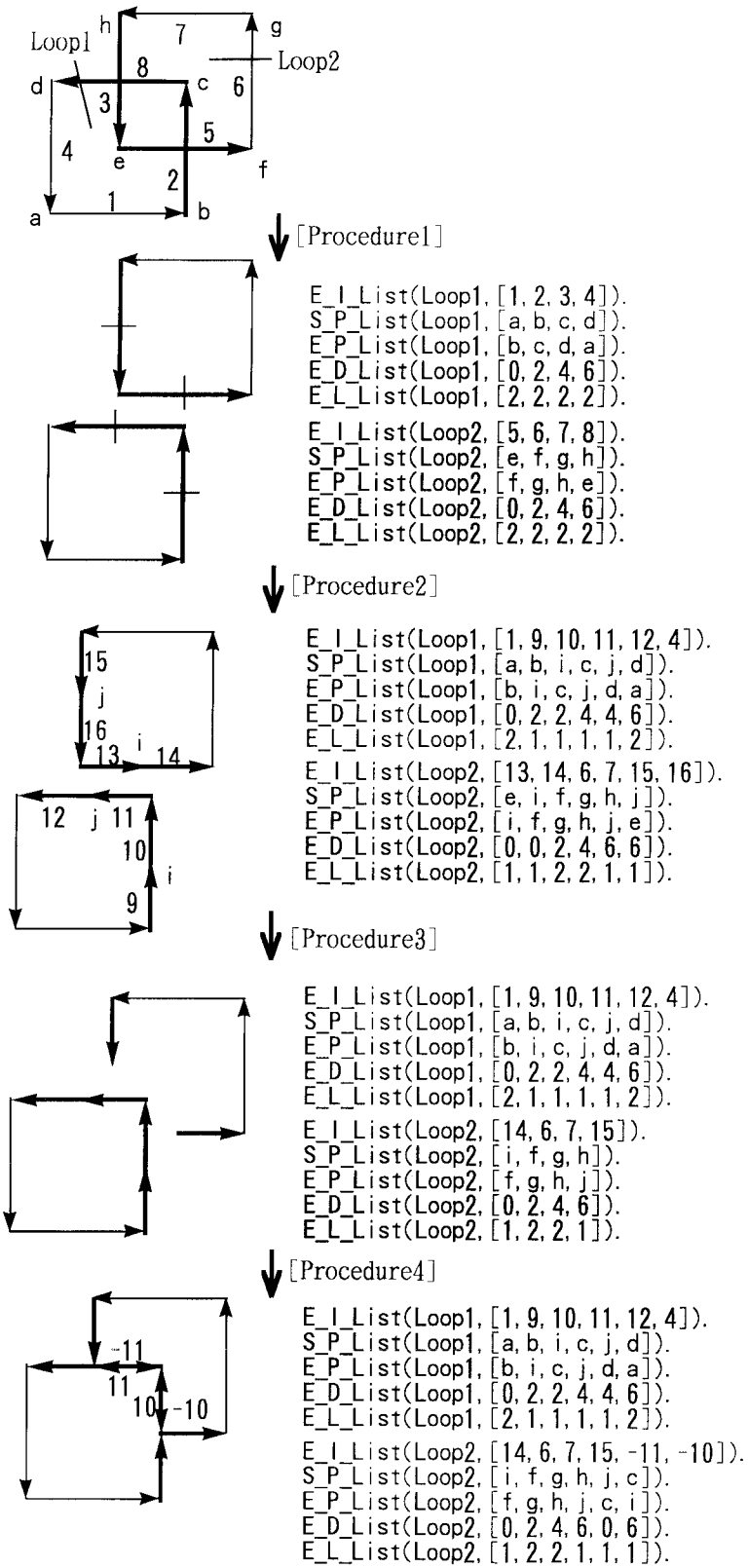
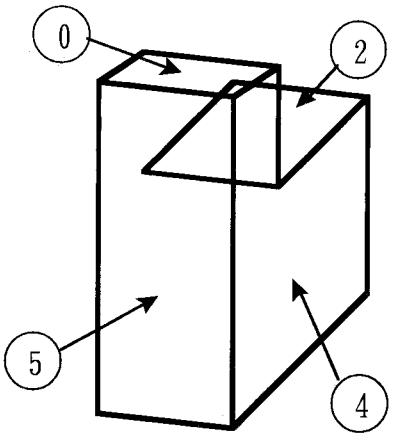


図4_7 MCBR法を用いた隠面処理.



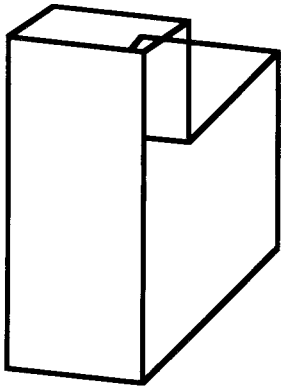

```
E_I_list(0,[-1,-2,-3,-4 ]).
S_P_list(0,[0,1,3,2 ]).
E_P_list(0,[1,3,2,0 ]).
E_L_list(0,[40.1,14.4,36.0,14.4]).
E_D_list(0,[7.8,1.2,4.0,4.7]).

E_I_list(2,[7,9,-10,-8 ]).
S_P_list(2,[4,5,7,6 ]).
E_P_list(2,[5,7,6,4 ]).
E_L_list(2,[36.2,45.2,40.0,39.5]).
E_D_list(2,[7.8,1.0,4.0,5.0]).

E_I_list(4,[-12,-9,-5,2,15,-14 ]).
S_P_list(4,[9,7,5,3,1,10 ]).
E_P_list(4,[7,5,3,1,10,9 ]).
E_L_list(4,[80.0,45.2,40.0,14.4,116.0,56.5]).
E_D_list(4,[2.0,5.0,2.0,5.2,6.0,1.0]).

E_I_list(5,[-15,1,18,17 ]).
S_P_list(5,[10,1,0,11 ]).
E_P_list(5,[1,0,11,10 ]).
E_L_list(5,[116.0,40.1,116.0,40.1]).
E_D_list(5,[2.0,3.8,6.0,7.8]).
```

(a) 処理前



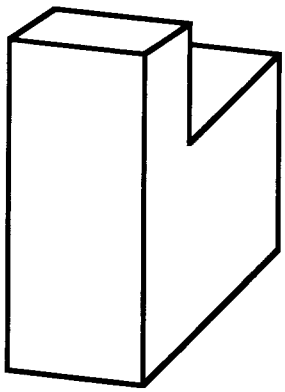
```
E_I_list(0,[-101,-100,-2,-3,-4]).
S_P_list(0,[0,12,1,3,2]).
E_P_list(0,[12,1,3,2,0]).
E_L_list(0,[33.4,6.7,14.4,36.0,14.4]).
E_D_list(0,[7.8,7.8,1.2,4.0,4.7]).

E_I_list(2,[701,9,-10,-801,-100,1500).
S_P_list(2,[13,5,7,6,12,1]).
E_P_list(2,[5,7,6,12,1,13]).
E_L_list(2,[10.1,45.2,40.0,7.0,6.7,29.0]).
E_D_list(2,[7.8,1.0,4.0,5.0,7.8,6.0]).

E_I_list(4,[-12,-9,-5,2,1500,1501,-14]).
S_P_list(4,[9,7,5,3,1,13,10]).
E_P_list(4,[7,5,3,1,13,10,9]).
E_L_list(4,[80.0,45.2,40.0,14.4,29.0,87.0,56.5]).
E_D_list(4,[2.0,5.0,2.0,5.2,6.0,6.0,1.0]).

E_I_list(5,[-1501,-1500,100,101,18,17]).
S_P_list(5,[10,13,1,12,0,11]).
E_P_list(5,[13,1,12,0,11,10]).
E_L_list(5,[87.0,29.0,6.7,33.4,116.0,40.1]).
E_D_list(5,[6.0,6.0,7.8,7.8,2.0,3.8]).
```

(b) 処理途中



```
E_I_list(0,[-1,-2,-3,-4 ]).
S_P_list(0,[0,1,3,2 ]).
E_P_list(0,[1,3,2,0 ]).
E_L_list(0,[40.1,14.4,36.0,14.4]).
E_D_list(0,[7.8,1.2,4.0,4.7]).

E_I_list(2,[9,-100101,501]).
S_P_list(2,[5,7,15]).
E_P_list(2,[7,15,5]).
E_L_list(2,[45.2,24.0,32.0]).
E_D_list(2,[1.0,4.0,6.0]).

E_I_list(4,[-12,-9,-501,-500,2,15,-14 ]).
S_P_list(4,[9,7,5,15,3,1,10 ]).
E_P_list(4,[7,5,15,3,1,10,9 ]).
E_L_list(4,[80.0,45.2,32.0,8.0,14.4,116.0,56.5]).
E_D_list(4,[2.0,5.0,2.0,2.0,5.2,6.0,1.0]).

E_I_list(5,[-15,1,18,17 ]).
S_P_list(5,[10,1,0,11 ]).
E_P_list(5,[1,0,11,10 ]).
E_L_list(5,[116.0,40.1,116.0,40.1]).
E_D_list(5,[2.0,3.8,6.0,7.8]).
```

(c) 処理結果

図4_8 MCBR法を用いた隠面処理の例.

4.4　結論

本章では画像認識を目的としたアスペクト生成アルゴリズムの開発とその有効性について検討した．MCBR法を用いた隠面処理には，隠面処理操作を簡便なリスト操作で行え，面単位の隠面処理であるため，対象物体の視認可能面が一目瞭然である等の利点があり，その結果，

(1)　2次元データが面ループの位相データとして保存される．

(2)　得られた2次元図形のトポロジカルな特徴が容易に把握可能である．

(3)　定性的な情報と定量的な情報が明示されているため，データの分類が容易に行える．

ことが明らかとなった．

第5章　リスト表現形状モデルの画像の認識への応用

5.1　緒論

視覚情報から3次元の空間を認識しようとする研究はこれまでに多数行われている[5_1]．両眼立体視[5_2]や移動立体視[5_3]は，その代表的手法としてこれまで数多くの研究，報告がなされている．また一方で前章で述べたように，3次元モデルからアスペクトを作成し，視覚から得られる画像とのマッチングにより，認識率の向上，物体とカメラの相対的な位置関係の決定等の研究もなされている．

これらの手法の核となる問題の一つに，複数の画像間でどのような特徴量を抽出し，マッチングを如何に行うかがある．これまで報告されている手法は，図5_1に示すように与えられた画像のある領域をもとにマッチングを行う手法と，画像より抽出したある特徴量をもとにマッチングを行う手法に大別でき，さらに後者は，用いる特徴量で，点，線に分けられる．

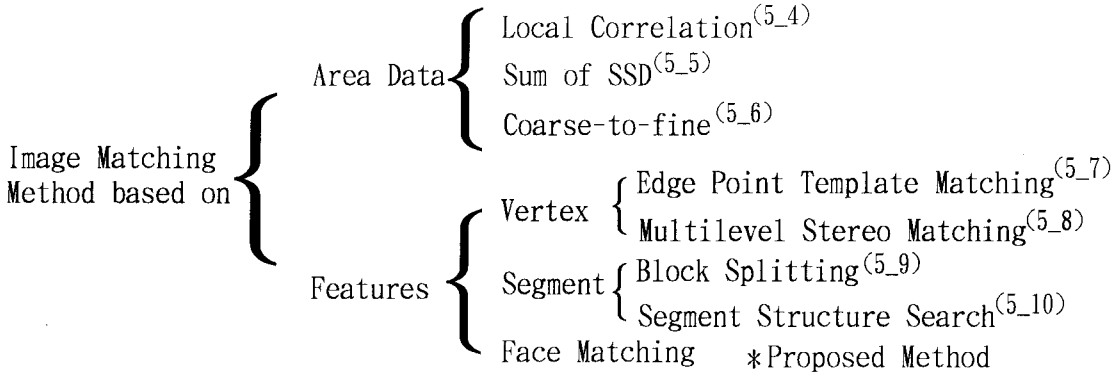


図5_1　画像マッチング手法の特徴量による分類.

このようなマッチングを行う場合，一般に対応検索する単位（領域，特徴量）が小さいものほど偽対応に陥りやすい．逆に，対応検索の単位を大きくするほど一般性を持たせるために表現モデルやアルゴリズムが複雑となり，計算コストも大きくなる．従って，対応検索手法は，偽対応に陥りにくくかつその表現モデルが平易であることが望まれる．

本章では，このような観点から対応検索の単位として面を用いることを提案する．さらに，対応検索は，対象物体の画像上での輪郭の図心（以下，輪郭中心という）と対象物体を構成する各面の図心（以下，面心という）との位置関係，面心同士の位相関係を用いる．また，対象物体を表現するモデルとしてMCBR法を用いる．MCBR法は，面を構成する線分をもとに形状を簡潔に表現可能で，様々な形状操作を簡便に行えるという特徴を持つ．これらの特徴と上記の対応検索手法を利用することで，偽対応に陥りにくく，かつ簡便なアルゴリズムを構成し対応検索を行える．

本章では，前章で議論したアスペクトと画像または画像同士のマッチングを対象と考え，まず，画像から正確な線分抽出を行う手法について述べる．さらに抽出された線分から対象を再構成し，最後に面心を用いたパターンマッチング手法について述べる．

5.2　画像認識システムの構成

図5_2にシステムの概念図を示す．構築したシステムは，大きく画像処理部と画像認識部の2つから構成されている．画像認識部では，カメラから取り込まれた画像から必

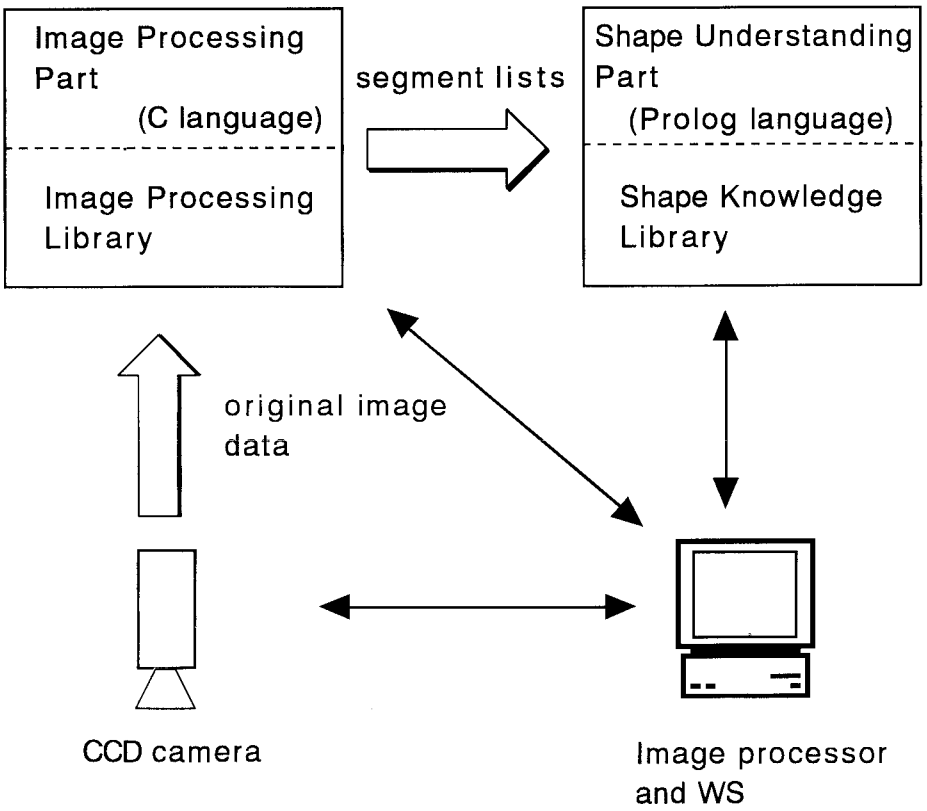


図5_2　システム概念図.

要なデータ，例えば線分群を抽出する．ここでの処理は，C言語で書かれており，画像処理装置とワークステーション（Sun Sparc Station2　以下WSという）上で実行される．得られたデータは画像認識部に送られる．画像認識部では，送られてきたデータと自身が持っている知識から画像の認識およびマッチングを行う．処理はC言語とProlog言語で記述されており，WS上で実行される．

5.3　画像からの線分の抽出（コーナー検出と直線成分抽出の協調処理）

5.3.1　Hough変換と補完アルゴリズム

Hough変換は，画像中からの直線成分の抽出に一般的に用いられている手法である[5_11]．これまでHough変換に関して様々な研究がなされ，また実用的なシステムにも用いられてきている．しかし，Hough変換には本来一本であるべき直線像から複数の直線群が得られる等の欠点が挙げられる．そこで，Hough変換の欠点を補うために画像から得られる他の情報を付加することを考えた．物体を構成する要素には，頂点，（直，曲）線，（平，曲）面が挙げられる．その中で線分の端点となる頂点は，非常に重要であり，かつ比較的抽出が容易であると考え，Hough変換と端点を抽出するコーナーポイント検出の融合を行うアルゴリズムを提案する．

5.3.2　Hough変換の問題点

ここでは，Hough変換の基本的なアルゴリズムについて述べ，Hough変換の欠点を具体的に指摘する．

Hough変換は，式（5_1）に示した変換式を用いて，x-y平面上の点をΘ－P平面上で曲線として表現し，Θ－P平面上での曲線の重なり度数からx-y平面上の直線成分を抽出する手法である．

$$\rho = x \cdot \cos\theta + y \cdot \sin\theta \qquad \cdots (5_1)$$

前処理を施しエッジを強調した画像上の濃度がしきい値を越える各画素に対して式（5_1）を適用しΘ－P平面上に曲線を描く．ここで式（5_1）を逆に考えると，Θ－P平面上の各点（θ，ρ）はx-y平面上では，原点からの距離ρ，x軸とのなす角θとなる直線を示している．つまり，x-y平面上で同一直線上の点が，Θ－P平面において，あ

る一点で交わる．このことを利用すれば曲線の交点は，x-y平面上での複数の点を通る直線を示している．このように，Θ－P平面上で曲線の重なり度数の大きい点に注目し，x-y平面に逆変換する事で，x-y平面上の直線成分を抽出することができる．

しかし，Hough変換には以下のような欠点が挙げられる．

(1)長い線分を構成する画素は当然 Θ－P平面上で非常に多くの曲線に変換される．このときΘ－P平面上でその直線を表す点（θ，ρ）の近傍の点にも多くの曲線が重なり，本来の線分を表す点を特定することが難しくなる．

(2) さらに，長い線分と短い線分が混在するような画像では，変換の結果得られたΘ－P平面上から同じしきい値で線分を抽出することが困難である．図5_3において，しきい値を長い線分に合わせて'3'とすると短い線分が抽出できない．逆に，短い線分に合わせ'2'とすると余分な直線成分，特に長い線分の近傍に余分な多くの直線成分が抽出される．

(3) 基本的には，Hough変換から得られるのは画像中の直線成分を表現する（θ，ρ）による直線の式だけである． 従って，その端点の座標までは分からない．

そこで，画像から得られる他の情報をHough変換と協調的に利用することで，上記の欠点を補完するアルゴリズムを創出すれば非常に利用価値が高いと考えられる．

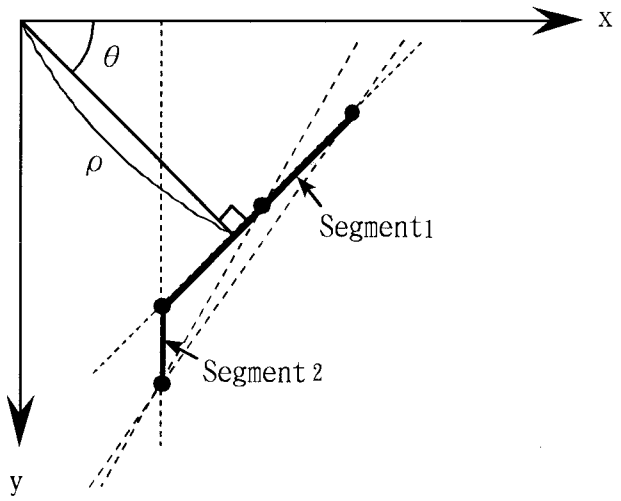


図5_3　Hough変換の問題点．

5.3.3　画像からのコーナー検出

画像に現れる直線成分のほとんどが，端点を有する線分要素である．そこでHough変換に付加する情報として端点情報を利用した．ここでは，端点を抽出するために利用したコーナー検出アルゴリズム[5_12][5_13]の概略を説明し，実際の画像からコーナー検出を行った結果を示す．

5.3.3.1　コーナー検出の基本アルゴリズム

ここで利用したコーナー検出アルゴリズムは，C_1とC_2いう2つのコーナーらしさをコーナーを定量的に扱うための指標とし，C_1，C_2の結果を合わせることにより統合的なコーナーらしさCNを求める．図5_4に処理の流れを示す．

まず，C_1を求めるために，入力画像に対してGaussian平滑化を施し，ノイズ除去を行った後，画像の各画素に対してx方向，y方向のSobelによる微分処理を行う．ここでは，ある画素pを例に挙げて処理手順を説明する．画素pにおけるx方向の微分を$grad_x$（p），y方向を$grad_y$（p）とする．いま注目している画素pがエッジ上にあると仮定すると，エッジの方向es（p），強さgd（p）は，

$$es(p) = \sqrt{grad_x(p)^2 + grad_y(p)^2} \qquad \cdots(5_2)$$

$$gd(p) = atan\frac{grad_y(p)}{grad_x(p)} \qquad \cdots(5_3)$$

と表せる．つぎに，gd（p）に対して再びx方向とy方向のSobelによる微分処理を行い，x方向とy方向の結果を$gradr_x$（p），$gradr_y$（p）とする．これらは，エッジの傾きのx方向とy方向の変化量を表している．よって，傾きの変化量の強さgm（p）は，

$$gm(p) = \sqrt{gradr_x(p)^2 + gradr_y(p)^2} \qquad \cdots(5_4)$$

と表すことができる．ここで，コーナーらしさC_1を

$$C_1(p) = es(p) \times gm(p) \qquad \cdots(5_5)$$

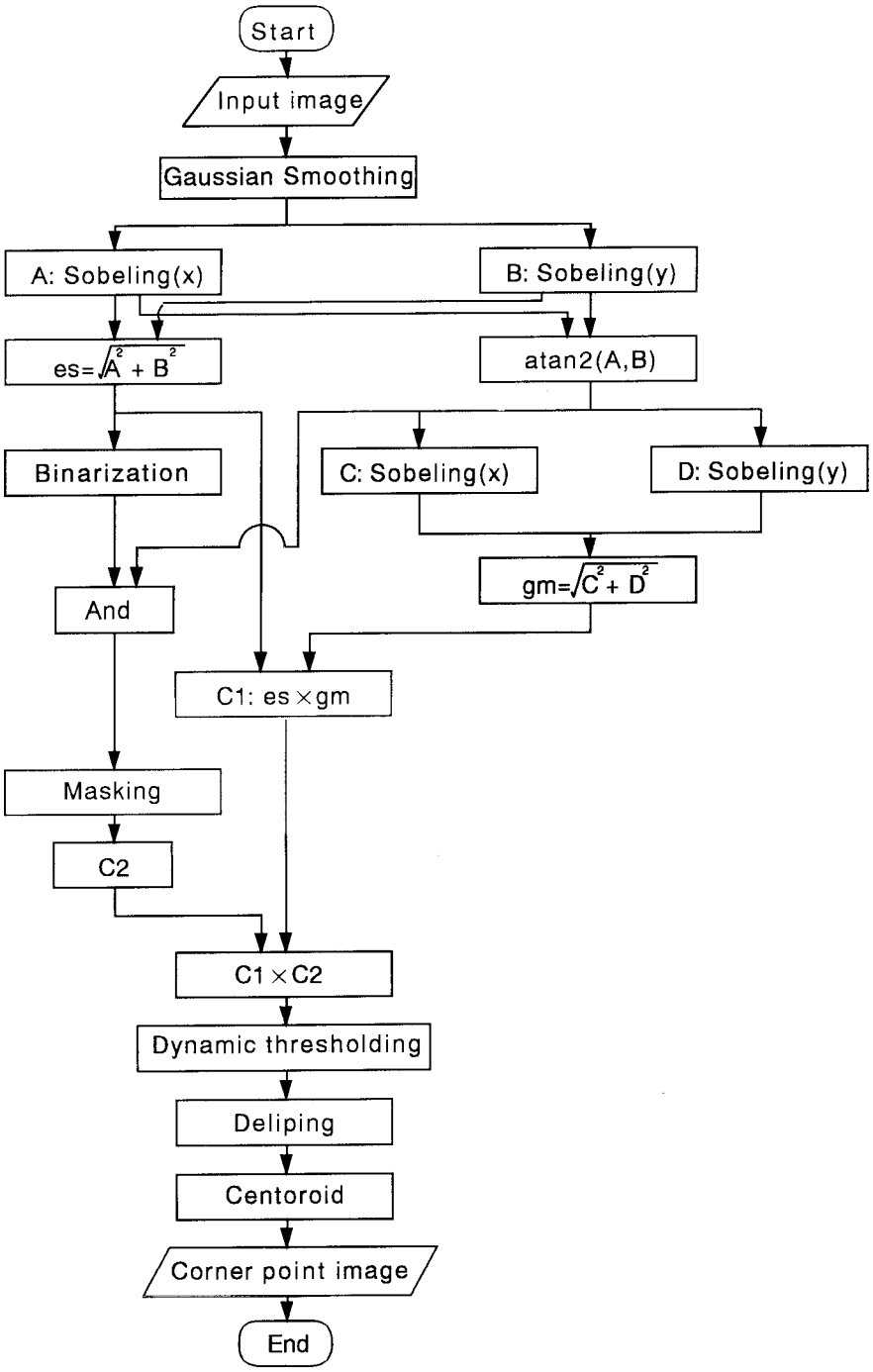


図5_4　コーナー検出アルゴリズム.

と定義する. C_1は, エッジ強度が強く画素のエッジの傾きの変化量が大きいほど大きくなる.

次に, es（p）に対して動的2値化[5_14]を行った画像をes_2（p）とすると, es_2（p）

はエッジが存在する可能性の高い画素を表す画像となる．つぎに，先で求めたgd（p）とes_2（p）とのANDをge（p）とし，es_2（p）で0でない画素pに対して，ge（p）にマスキング処理を行う．この際使用するマスクは，注目画素pには8近傍の0でない画素の数が，また8近傍には-1が入っている，さらに，得られた画像に対して動的2値化を行った結果をC_2とする．このC_2はエッジに沿ったエッジの傾きの変化量を表している．よって画素pに対してのコーナーらしさCN（p）を2つの指標C_1とC_2を使って，

$$CN(p) = C_1(p) \times C_2(p) \qquad \cdots (5_6)$$

と表す．このCN（p）に対して動的2値化を行って，コーナーを含む領域を得る．これに微小領域の除去，さらに残った領域の重心を求め，コーナーポイントを検出する．

5.3.3.2　　コーナー検出による端点抽出

図5_5にコーナーポイント検出の例を示す．図（a）が原画像を，（b）がコーナー検出結果を表している．この結果では，ほとんどのコーナーは抽出されているが，それ以外に濃度差の大きいエッジ上の点がコーナーであると誤認識されていることが分かる．

（a）　原画像　　（b）　コーナー検出結果

図5_5　コーナー検出例.

5.3.4　　Hough変換とコーナー検出との融合による線分抽出

ここでは，前節で述べたHough変換の欠点を補うために，先に述べたコーナー検出アルゴリズムからの情報をHough変換から得た情報と融合する手法について述べる．最後に，実際の画像に対してここで述べる手法を使って線分を抽出した結果を示す．

5.3.4.1　　融合アルゴリズム

図5_6に本手法の概念図を示す．まず原画像に対して，Hough変換を施しΘ－P平面

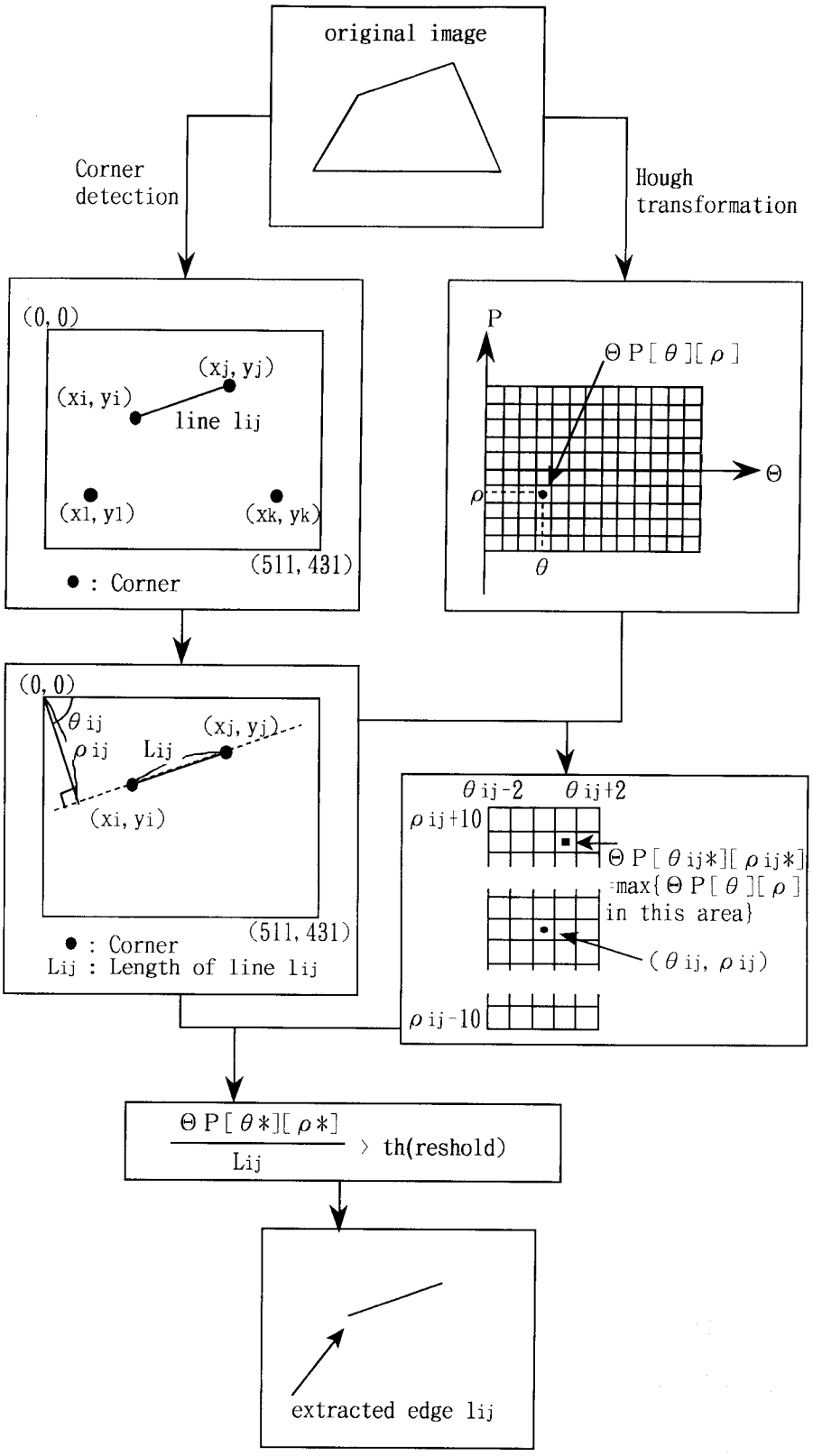


図5_6　Hough変換とコーナー検出アルゴリズムとの融合.

を得る．つぎにコーナー検出を実行し，抽出された各コーナーのxとyの座標値を得る．さらに，得られたコーナーから任意の2点CP_i（x_i, y_i），CP_j（x_j, y_j）を選び，その2点を結ぶ線分l_{ij}がエッジとなる可能性をΘ－P平面上で検証する．以下，検証手順について述べる．

手順1　原点（0,0）から線分l_{ij}までの距離ρ_{ij}とx軸とl_{ij}がなす角度θ_{ij}，さらに線分l_{ij}の長さL_{ij}を求める．

手順2　計算の誤差と画像やΘ－P平面が離散的であることを考慮して，Hough変換によって得られたΘ－P平面上で（θ_{ij}，ρ_{ij}）と次の条件で示した範囲から最大重なり度数$\Theta P[\theta_{ij}^*][\rho_{ij}^*]$を持つ点を検索する．

$$\theta_{ij}-2 \le \theta \le \theta_{ij}+2 \wedge \rho_{ij}-10 \le \rho \le \rho_{ij}+10 \qquad \cdots(5_7)$$

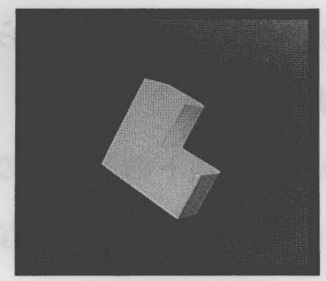
手順3　線分l_{ij}が原画像上でエッジであると仮定すると，手順1で求めたL_{ij}と手順2で求めた$\Theta P[\theta_{ij}^*][\rho_{ij}^*]$の間には線形的な関係が存在することから次式を用いることでl_{ij}が線分であるか否かの判定を行う．

$$\frac{\Theta P[\theta_{ij}*][\rho_{ij}*]}{L_{ij}} > th \qquad \cdots(5_8)$$

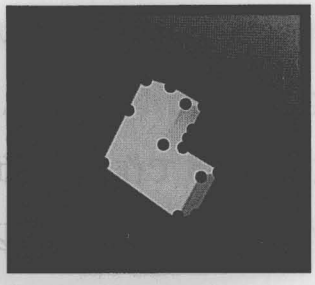
ここでthはしきい値を表す定数で，いくつかの画像を用いて検討した結果，0.4から0.6の値とした．コーナー検出によって得られたコーナーのすべての組み合わせについて上の手法を適用し，このしきい値を超える線分を抽出し，線分抽出画像を得る．

5.3.4.2　　線分抽出実験

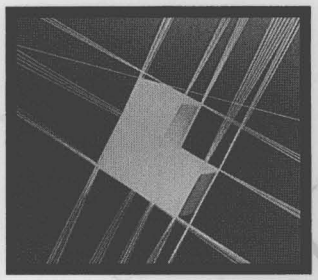
図5_7に本手法を適用して，線分を抽出した結果を示す．（a）が原画像を，（b）（c）がそれぞれコーナー検出結果とHough変換によって得られた直線群を，さらに（d）に本手法を適用した線分抽出画像を示している．この結果から，本手法が前述のHough変換の欠点をある程度補っていることが判る．



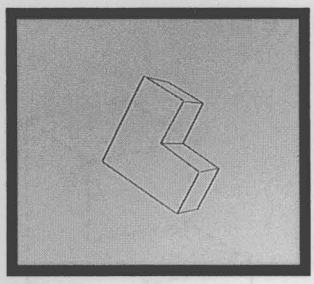
(a)　原画像



(b)　コーナー検出結果



(c)　Hough変換の結果



(d)　本手法を用いた結果

図5_7　線分抽出結果.

5.4　面心を用いた画像マッチングアルゴリズムの開発

5.4.1　面心検出

ここでは，まず線分データから対象を再構築する手法について述べ，その後，対象物体を構成する各面の面心を検出する手法について述べる.

5.4.1.1　面データ作成アルゴリズム

MCBR法を利用して，得られた線分画像から面データを作成する．作成するデータは，対象を構成する各面のデータと対象の輪郭を表すデータの2種類である．データはすべてMCBR法による形状リストで表現される．以下に手順を示す.

手順1　与えられた線分データに各データと方向が逆になる線分データを加える.

手順2　画像上でy座標が最も小さい頂点を選択し，これを検索始点とする（図5_8(a)).

手順3　検索始点から反時計回りにx軸となす角が最も小さい線分を選び，これを検索線分とする．この線分の終点を検索頂点とする（図5_8 (b)).

手順4　検索頂点を始点とした線分のうち，先の検索線分となす角が最も小さい線分を選択し，新たな検索線分とする．また，その終点をつぎの検索頂点とする．

手順5　手順4を繰り返し，検索線分の終点が検索始点と重なり，さらに形成したループが反時計回りであれば面ループリストを作成する（図5_8（c））．

手順6　検索始点を手順5で作成した面ループの始点リストの2番目の頂点に移し，手順3から繰り返す（図5_8（d））．それ以上，面ループが作成できなくなった時点でつぎの手順に移る．

手順7　これまでに使用されていない線分を時計回りにループを形成し，輪郭ループを作成する（図5_8（e））．

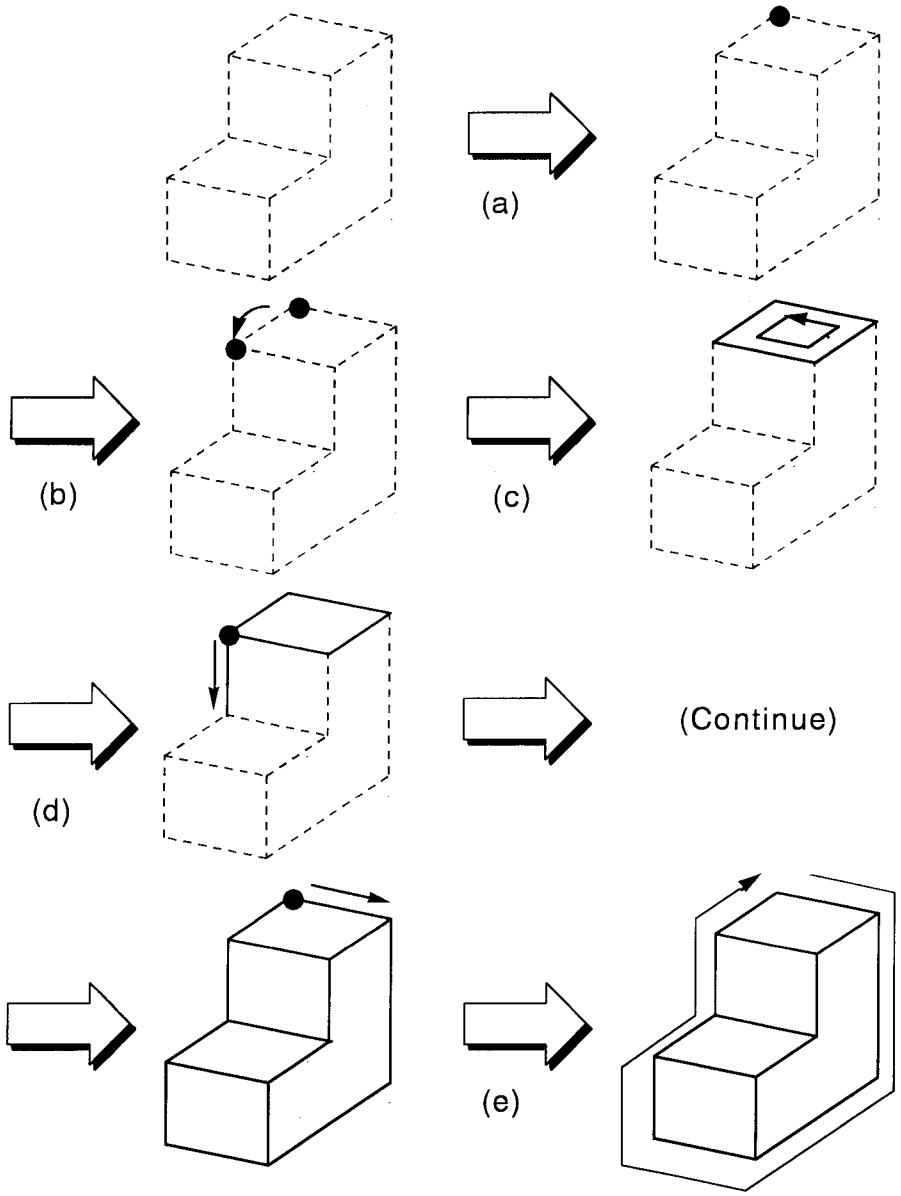
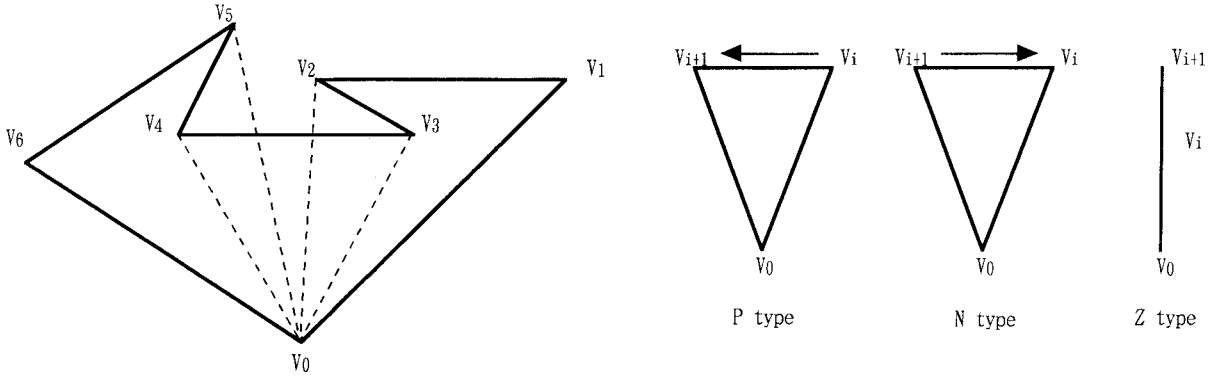


図5_8　線分データからの形状の再構築.

5.4.1.2　　面心抽出アルゴリズム

前項で述べたように，各面の情報は，その面を構成する線分のループとして与えられている．ここでは，この情報を用いて各面の面心を抽出する手法について述べる．まず，各面および輪郭の面積を以下のように求める[5_15]．



(a)　三角形分割　　(b)三角形の分類

図5_9　多角形の三角形分割とその分類.

図5_9（a）に示すように，各面をある頂点Voと他の連続する2つの頂点を結ぶ複数の3角形に分割する．さらに，これらの3角形を図5_9（b）に示すようにP型，N型，Z型の3種類に分類する．このように，ある面を3角形分割すると，面の面積S_Tは，P型のi番目の3角形の面積S_{Pi}とN型のj番目の3角形S_{Nj}を用いて，

$$S_T = \sum_i S_{Pi} - \sum_j S_{Nj} \qquad \cdots (5_9)$$

と表すことができる．

つぎに各面の面心と輪郭中心を求める．まず，先に求めた3角形の重心を利用する．ある3角形VoViVjの重心の座標（x_g, y_g）は，V_o，V_i，V_jの座標（x_o, y_o），（x_i, y_i），（x_j, y_j）を用いて，

$$x_g = \frac{x_o + x_i + x_j}{3}, y_g = \frac{y_o + y_i + y_j}{3} \qquad \cdots (5_10)$$

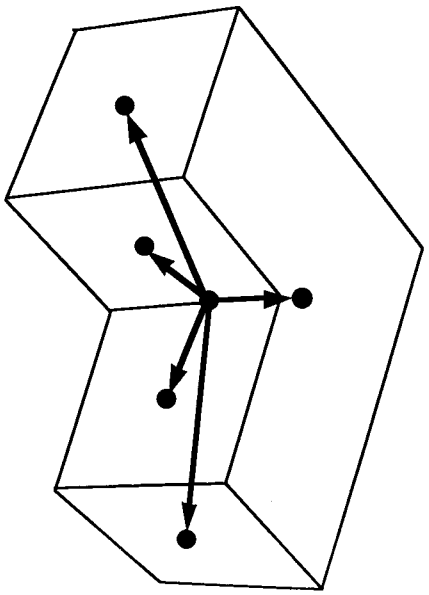
と表せる．これを利用してP型のｉ番目の3角形の重心とN型のｊ番目の3角形の重心を求め，それぞれ（x_{gPi}, y_{gPi}），（x_{gNj}, y_{gNj}）とすると，面全体の重心，すなわち面心C（x_c, y_c）は次式で求めることができる．

$$x_c = \frac{\sum_i (S_{Pi} \times x_{gPi}) - \sum_j (S_{Nj} \times x_{gNj})}{S_T}$$

$$y_c = \frac{\sum_i (S_{Pi} \times y_{gPi}) - \sum_j (S_{Nj} \times y_{gNj})}{S_T} \qquad \cdots (5_11)$$

5.4.1.3　面心抽出結果

図5_10（a）で与えられた対象の形状データと面心検出結果を図5_10（b）に示す．後で述べる面マッチングに利用するためのdirect（e_i, $length_i$, $direction_i$）は，輪郭中



（a）形状と得られた面心

```
E_I_list(11,[-4,6,7,1000000]).
S_P_list(11,[v1,v2,v3,v4]).
E_P_list(11,[v2,v3,v4,v1]).
E_L_list(11,[62,68,68,69]).
E_D_list(11,[4.18,5.45.0.11,1.58]).
・・・・・・・・・・
E_I_list(shape,[1,13,-12,-5,-8,-3,-6,4]).
S_P_list(shape,[v1,v11,v10,v9,v7,v5,v3,v2]).
E_P_list(shape,[v11,v10,v9,v7,v5,v3,v2,v1]).
E_L_list(shape,[113,132,62,53,70,60,68,62]).
E_D_list(shape,
[6.82,5.62,3.94,3.09,1.55,2.99,1.45,0.18]).

coordinate(v1,[180,66]).
coordinate(v2,[118,75]).
・・・・・・・・・・

cent_gravity(shape,R,23058.0,[169.9,174.9]).
cent_gravity(L5,L,9442.5,[202.8,172.8]).
・・・・・・・・・・

direct(15,32.96,0.08).
direct(14,88.96,5.86).
direct(13,40.41,5.42).
direct(12,32.97,3.19).
direct(11,79.44,2.55).
```

（b）形状リストと面心抽出結果

図5_10　面心抽出結果.

心から各面心（e_i）までの距離（$length_i$）と方向（$direction_i$）を表している．このデータを面心データと呼ぶ．

5.4.2　面心を用いた面のマッチングアルゴリズム

ここでは，図5_11に示すように，面心データを用いて，まず面同士の対応検索を行

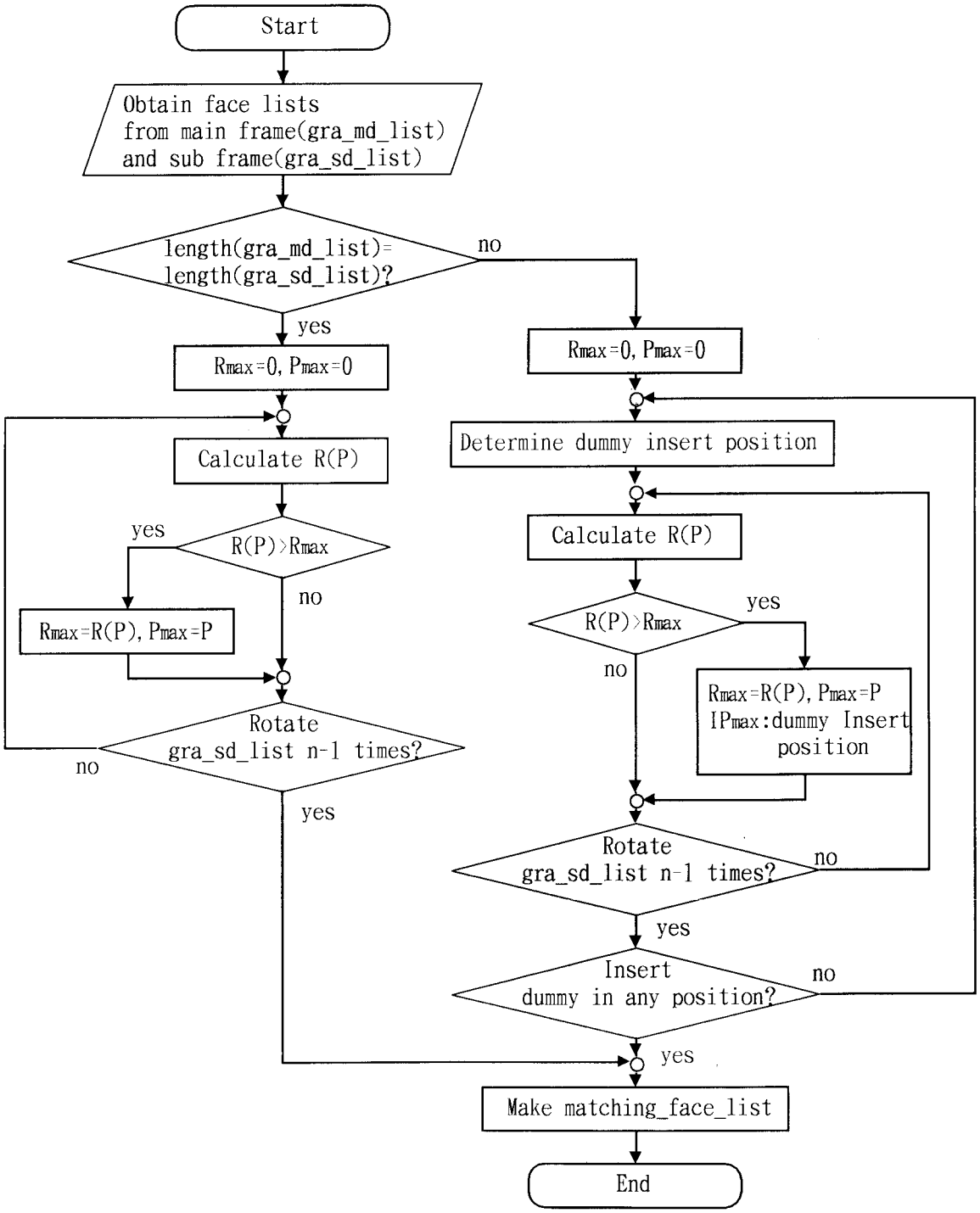


図5_11　画像のマッチングアルゴリズム．

い，さらにそれぞれの面の辺および頂点の対応検索を行うプロセスについて述べる．ただし，対象とするマッチングデータにおいては，物体を構成する面の位相関係は変わらないものと仮定する．

まず，マッチングを行う2枚の画像に対して，面心データを求める．さらに，それぞれの面心データで，方向コードの小さいものから順に並べて以下の2つのリストを生成する．

$$\begin{aligned} &\mathrm{grad_md_list}\ (me_1, me_2, \ldots\ldots) \\ &\mathrm{grad_sd_list}\ (se_1, se_2, \ldots\ldots) \end{aligned} \qquad \cdots (5_12)$$

ここで，grad_md_listとgrad_sd_listの各要素は，それぞれ画像1と画像2における輪郭中心から各面心への方向コードを示している．基本的には，上の2つのリストの相関関係を調べることで，面のマッチングを行う．

まず，画像1，画像2に同数の面が現れているとする．ここで，各面心への方向コードの差を式（5_13）のように定義する．式（5_13）では，方向コードが図2_15（a）に示したように0以上8未満であることから2つの方向の差を4.0つまりπよりも小さくなるようにしている．

$$\mathrm{sab}(me_i, se_i) = \begin{cases} |me_i - se_i| & \text{if } |me_i - se_i| \le 4.0 \\ 8.0 - |me_i - se_i| & \text{if } |me_i - se_i| > 4.0 \end{cases} \qquad \cdots (5_13)$$

また，画像2のリストを以下のように回転させる．

$$\begin{aligned} &\text{回転数 } p=0 && \mathrm{grad_sd_list}\ (se_1, se_2, \ldots\ldots se_n) \\ &\text{回転数 } p=1 && \mathrm{grad_sd_list}\ (se_2, \ldots\ldots se_n, se_1) \\ &\cdots \\ &\text{回転数 } p=n && \mathrm{grad_sd_list}\ (se_n, se_1, \ldots\ldots, se_{n-1}) \end{aligned} \qquad \cdots (5_14)$$

このときの相関量R（p）を以下のように定義する．

$$R(p) = 1.0 - \frac{\sum_{i=1}^{n} sab(me_i - se_{i+p})}{4.0 \times n}, 0 < R(p) \leq 1.0 \qquad \cdot\cdot\cdot(5_15)$$

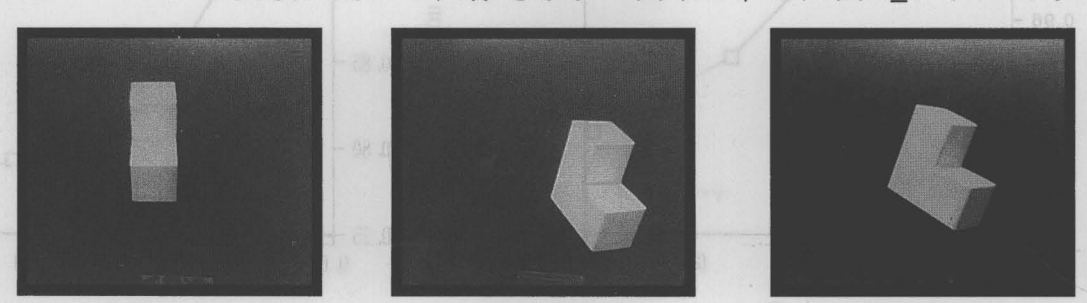
R（p） は完全にマッチングがなされたときには1.0となる．リストの回転数pを変化させて，R（p）の最大値を求め，2つの画像の相関値とする．

また，カメラと対象の位置関係によっては，一方の画像にある面が現れない場合がある．現れている面の順序が変わらないと仮定すれば，要素の数が少ない方のリストにダミーを挿入し，上に述べたマッチング手法を適用することで，対応検索が可能となる．ただしダミーを挿入する位置は特定せず，すべての可能性を検討する．

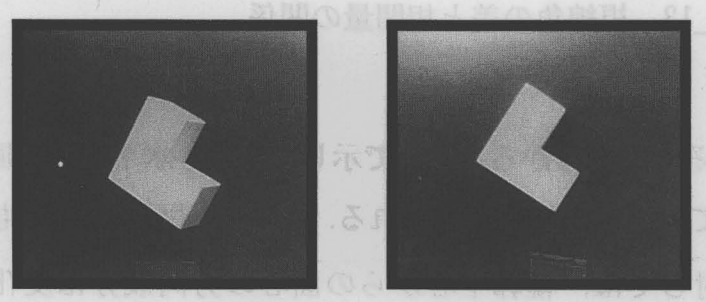
面の対応検索が終了後，それぞれの面ループから得られる情報としては，面を構成する各辺の方向，長さ，さらに各頂点の座標がある．これらの情報のうち，視差に対する定量的な影響が小さい方向コードを辺の対応検索に利用する．つまり，各面のリストのうち，その面を構成する辺の方向コードを記述したリストを用いて，面の対応と同様に方向成分の相関を計算し対応検索を行う．辺のマッチングを行った後，対応する頂点を検索する．

5.4.3　パターンマッチング実験

図5_12（a）～（e）に実験に用いた画像を示す．図中のφは，図4_1（b）で示した



（a）ϕ=0（rad）　（b）$\phi = \frac{\pi}{6}$（rad）　（c）$\phi = \frac{\pi}{3}$（rad）



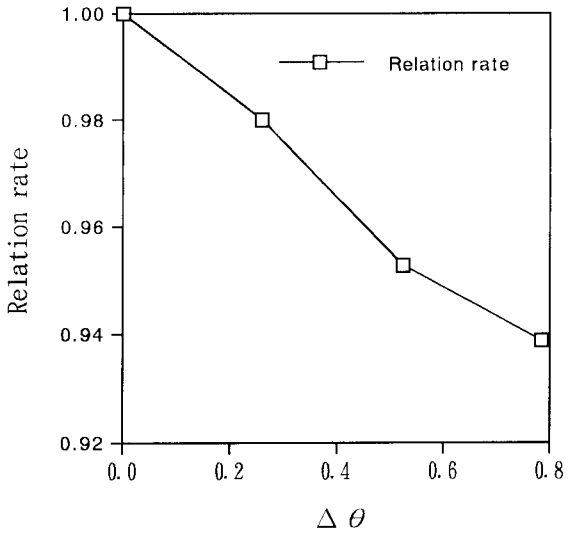
（d）$\phi = \frac{5\pi}{12}$（rad）　（e）$\phi = \frac{\pi}{2}$（rad）

図5_12　実験に用いた画像．

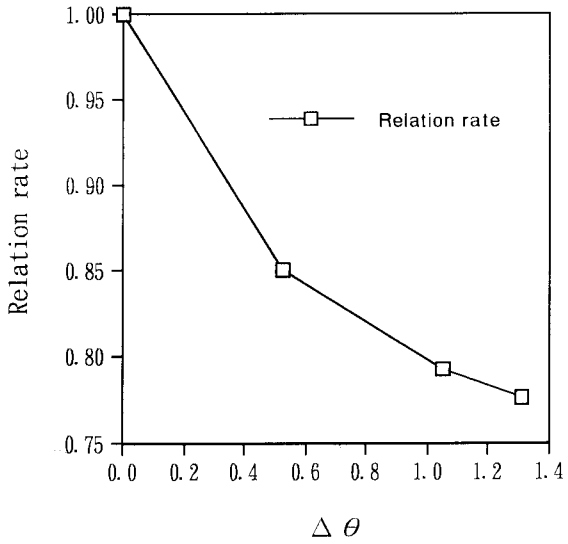
視線の角度である．これらの画像から線分を抽出し，MCBR法により形状を表現した．さらに上述のアルゴリズムに従って，すべての組み合わせに対して対応検索を行った結果を表5_1に示す．図（e）と他の4つの画像に対しては対応検索ができなかったが，それ以外の対応検索はすべて成功している．図5_13（a）に面の数が同じ場合，（b）に面の数が異なる場合の視線の角度の差による相関量の変化を示す．それぞれの図から角度によって

表5_1　実験結果．

	a	b	c	d	e
a	1.000	0.850	0.793	0.776	-
b	0.850	1.000	0.953	0.939	-
c	0.793	0.953	1.000	0.980	-
d	0.776	0.939	0.980	1.000	-
e	-	-	-	-	-



（a）　面の数が同じ場合



（b）　面の数が異なる場合

図5_13　視線角の差と相関量の関係．

相関量が連続的に変化していることが分かる．ここで示した例は，水平方向の回転角度のみであるが，他の回転に対しても同様の結果が得られる．さらに，図形の拡大もしくは縮小，画像面内の平行移動に対しては，輪郭中心からの面心の方向成分は変化しないため，本手法を用いて対応検索が問題なく行える．またここでは，3次元モデルから生成

したアスペクトと画像とのマッチングの例は示していないが，同様の結果が得られた．

5.4.4　適用限界

以上に述べたように，両眼立体視等で用いられるような画像間のマッチングに対してはほぼ信頼できる結果が得られた．しかし今後，より汎用性の高い対応認識システムを構築することを考えた場合，以下に示すような問題を解決する必要がある．

第1の問題点としては，マッチング結果でも述べたように，対象物の面が一つしか現れない場合，または図5_14のように輪郭の中央付近に面が現れる場合が考えられる．このような場合，輪郭中心と面心が重なってしまうため，その方向成分を決定することができない．

第2の問題点としては，図5_15に示すように輪郭中心に対して同一方向に複数の面が現れる場合が考えられる．この場合，方向成分の類似度だけでは対応する面を決定することができないため，誤対応を招く可能性がある．

これらの問題を解決するためには輪郭中心から面心までの距離や面の形状といった情報を付加し，総合的に判断する必要がある．たとえば，面心と輪郭中心が一致する場合，面の辺の数を対応検索の指標にすることや，同一方向に面心が現れる場合，輪郭中心からの距離を比較することで対応付けすることも考えられる．

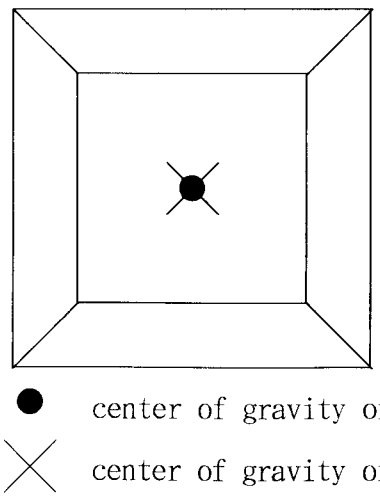


図5_14　面心と輪郭中心が一致する例．

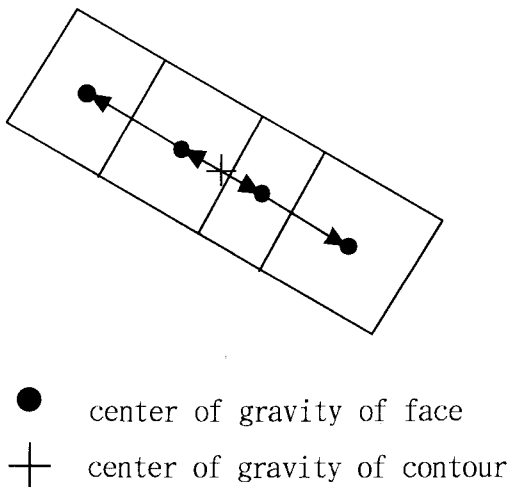


図5_15　面心への方向が他の面と一致する例．

本研究で提案した手法では，今後，総合的に判断するために必要なデータが整理あるいは容易に抽出可能である．したがって，様々な状況に対応する柔軟性を有すると考えられる．

5.5　結論

本章では，3次元物体の画像上に現れる面に着目し，それぞれの面の面心と全体の輪郭の中心との位相関係をもとに対応検索を行う手法を提案した．このとき，画像上の3次元物体の面を表す2次元モデルにMCBR法を用いた．以下，本手法により得られた結果を示す．

(1)　画像内の線分抽出の手法として従来より用いられてきたHough変換による直線成分抽出法に対して，コーナーポイント情報を与えることにより，短い線分の欠落等の問題点を解決することができる．これによって，精度よく線分図形を作成することができる．

(2) 面心および輪郭中心の位置関係よりそれぞれの面同士のマッチングを行った．本手法を用いることにより，画像内での対象の平行移動や拡大・縮小に対してロバストな照合を行うことができる．また，対応する面がすべて見えている場合においてはカメラの位置，角度の影響をあまり受けない照合が可能である．

(3)　二つの画像の面の数が同じではない場合においても同一の回転方向にみて，面の現れる順序が入れ替わらなければ，(2) の場合と同様にカメラの位置，角度に関係なく照合を行うことができる．

以上のことから，提案した面心情報を利用したマッチングの手法は生産加工システムにおけるマシンビジョンの構築において有効な手法であると考えられる．

第6章　総　括

本論文では生産システムでの要素技術である形状モデルの開発とその形状モデルを用いた'画像'データの生産システムへの適用について述べた.

まず，形状を複数のリストで表現し，様々な画像変換を簡便なリスト処理に置き換えることができる形状モデルを開発した．この形状モデルが持つ特徴として，リストの一つが形状を構成する辺の方向を示しており，形状が持つトポロジカルな性質の抽出が簡便に行える，いくつかのリストの添付・削除や，リストの内容の入れ替えが可能で，様々な状況・局面に対処できる柔軟性を持つ，等が挙げられる.

つぎに，開発した形状モデルの生産システムの3つの状況・局面での適用について検討した．まず，計画問題の画像化を目的とし，スケジューリング問題を取り上げ，形状モデルと図形処理を用いたスケジューリング・システムの開発について検討した．さらに，この形状モデルを適用した3次元形状モデルの2次元画像化と形状モデルの画像認識への適用を検討し，最後にこれからの生産システムと形状・画像情報との関わりについて述べた.

リスト表現された形状モデルとその生産システムへの適用について得られた結果を列挙する.

(1)　開発したCBR法は，形状をいくつかのリストで表現しているため，形状の操作を簡単なリスト操作に置き換えることが可能であり，また方向コードの導入により形状の定性的な取り扱いが容易である．さらに，対象とする問題によってモデル自体の改良が容易に可能であること等の特徴から，生産加工システムでの統一的なモデルとして利用される可能性がある.

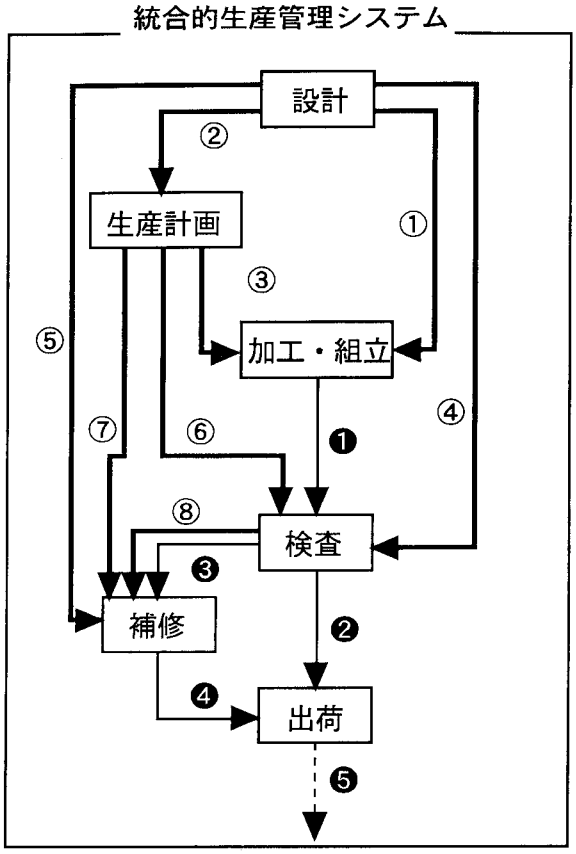
(2)　SCBR法を適用したスケジューリングシステムは，オーソドックスな問題に対しては，基本的なPERT山崩し法と同等，あるいはそれ以上の結果が得られ，PERT山崩し法では対応が難しい所要人数変更問題に対しても，本手法では対処が十分可能であり，これらの結果より，本手法が従来のスケジューリング問題のみならず，新たな問題にも十分に対応できることを示した.

(3) MCBR法を適用した画像認識を目的とした，アスペクト生成アルゴリズムの開発を行った．その結果，アスペクト生成におけるMCBR法を用いた隠面処理には，隠面処理操作を簡便なリスト操作で行える，面単位の隠面処理であるため，対象物体の視認可能面が明示される，等の特徴から得られるデータには，2次元データを面ループの位相データとして保存され，トポロジカルな特徴が容易に把握可能である，さらに定性的な情報と定量的な情報が明示されているため，得られたデータの分類が様々な指標によって容易に行える．

(4) 画像の認識において，3次元物体の画像上に現れる特徴として得られる頂点（またはそれを含む領域），線分，面のうち，対局的な照合が可能で局所的な照合への移行が簡便な面に着目し，MCBR法を用いてそれぞれの面の面心と全体の輪郭の中心との位置関係をもとに対応検索を行う手法を提案した．本手法を用いることにより，画像内での対象の位置移動や拡大・縮小に強い照合を行うことができる．また，対応する面がすべて見えている場合においてはカメラの位置，角度の影響を受けない照合が可能である．また，二つの画像の面の数が同じではない場合においても同一の回転方向にみて，面の現れる順序が入れ替わらなければ，カメラの位置，角度に関係なく照合を行うことができる．

以上の結果から，リスト表現を用いた形状モデルが，生産加工システムにおける統一的な形状モデルとして有効であると考えられる．

図6_1は，近い将来考えられる統合的生産加工システムにおける物の流れ（白抜きの数字）と情報の流れ（黒抜きの数字）を示している．様々な部品や材料が，加工・組立され，検査（さらに補修）を経て出荷される．形状・画像等の情報は，これらの工程がスムーズに行われるように流れ，利用される．本研究で実際に取り扱った対象について言えば，第3章で取り扱ったスケジューリングは，②③の生産計画工程の中で利用される．また，第4章で取り扱ったコンピュータ・グラフィックスは，設計工程において用いられ，そこから得られた結果は，①④⑤の流れを経て，第5章で取り扱った画像認識手法と共に加工・組立工程，検査工程さらに補修工程で利用される．



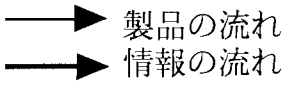


図6_1 生産システムにおける
製品と情報の流れ.

製品あるいは個々の部品の形状を始めとする情報は，この統合的生産加工システムばかりでなく，さらに製品や部品のライフサイクルさらに再生をも考えたシステムにおいても重要である．今後，このような情報を統括・管理するシステムの研究・開発が望まれる．

謝 辞

本研究は，大阪大学教授　井上勝敬博士の御指導と御教示のもとに遂行し得たものである．

本論文の作成にあたっては，大阪大学教授　仲田周次博士，荒井栄司博士により有益な御助言と御討論を賜った．

実験およびソフトウェア開発の遂行にあたり，大阪大学大学院生産加工工学専攻（溶接工学専攻）井上研究室の卒業生である山地泰久氏，鎌田隆夫氏，在学生である徐源賛氏，近畿大学理工学部原子炉工学科卒業生である宮地紀子氏，鈴木啓造氏，工藤英二氏をはじめ，多くの卒業生ならびに在学生の方々より多大な御支援，御協力を得た．

また，東京都立科学技術大学教授　福田収一博士には，本研究の遂行のきっかけを与えて頂いた．

本研究は，以上の方々をはじめ数多くの方々の御指導，御援助を得てなされたものであり，本論文を終えるにあたり，これらの方々に深甚の謝意を表す次第であります．

参考文献

1_1） CIMと経営管理，沢田，日刊工業新聞社（1994）．

1_2） コンピュータ統合生産システムと産業用ロボット，藤本，日本ロボット学会誌，12-8（1994）．

1_3） Computer-Integrated Design and Manufacturing, David D. Bedworgh, Mark R. Henderson, Philip M. Wolfe, McGraw-Hill（1991）．

1_4） システムのフロンティア，西田正吾，日本機械学会誌（機械工学年間特集），Vol. 98, No. 921（1995）．

1_5） SIGGRAPH94 Conference Proceedings, July 24-29（1994）．

1_6） 特集：「図による推論」，人工知能学会誌，Vol. 9, No. 2（1994）．

1_7） The Technique of Information Organization in Decision Support System, Wang Y. L., Proc. 16Th Int. Conf. On Computer & Industrial Eng.（1994）．

2_1） インテリジェントソリッドモデル入門，遠山茂樹，オーム社（1988）．

2_2） 3次元CAD/CAMにおける形状処理技術，塩谷景一，日刊工業新聞社（1989）．

2_3） Solid Modeling, A Historical Summary and Contemporary Assesment, A. G. Reuicha and H. B. Voelcker, IEEE Computer Graphics and Applications, Vol. 2, No. 2（1982）．

2_4） 自動設計の方法論，沖野教郎，養賢堂（1982）．

2_5） FAにおける情報処理技術の役割，木村文彦，情報処理，，Vol. 25, No. 4（1984）．

2_6） Representation for Rigid Solids ; Theory, Method and Systems, A. G. Requicha, Computing Surveys, Vol. 12, No. 4, Dec.（1980）．

2_7） Computer-Aided Design of Mechanical Components with Volume Building Bricks, in Computer Languages for Numerical Control, I. C. Braid and C. A. Lang, J. Hatvany（Eds.）, North-Holland, Amsterdam（1973）．

2_8） COMPUTER-INTEGRATED DESIGN AND MANUFACTURING, D. D. Bedworth, M. R. Henderson and P. M. Wolfe, McGraw-Hill, Inc.（1991）．

2_9） ソリッドモデリング，千代倉弘明，工業調査会（1985）．

2_10） 形状処理工学［III］，山口富士夫，日刊工業新聞社（1988）．

2_11） Geometric Modelling of Mechanical Parts and Processes, H. B. Voelcker

and A.G.Requicha, IEEE Computer, Vol.10, No.12, Dec. (1977).
2_12) Computer Processing of Line-drawing Images, Freeman H., Computer Surveys, Vol.6, No.1 (1974).

3_1) 専門家モデルに基づく生産スケジューリング・システム，関根，竹中，本田，江原，新井，美濃，人工知能学会誌，Vol.5, No.2 (1990).
3_2) 協調型スケジューリングによる鉄鋼工程スケジューリング・エキスパートシステム，森下，沼尾，戸沢，人工知能学会誌，Vol.5, No.2 (1990).
3_3) 溶接，應和，辻，朝倉書店 (1984).
3_4) PERT講座I, II, III，刀根薫，東洋経済新報社 (1977).
3_5) PERTのはなし，柳沢，日科技連 (1985).
3_6) システム工学入門，寺野寿郎，共立出版株式会社 (1989).
3_7) 情報システム開発プロジェクト管理システムにおける知識型スケジューリング方式，大場，都島，薦田，山中，情報処理学会論文誌，Vol.30, No.5 (1989).

4_1) ロボット作業のためのモデルベースドビジョン，長谷川勉，日本ロボット学会誌，Vol.10, No.2 (1992).
4_2) 3次元モデルからの物体認識用特徴探索手順の自動生成，久野義徳，岡本恭一，沼上英雄，日本ロボット学会誌，Vol.6, No.4 (1988).
4_3) Internal Representation of Solid Shape with Respect to Vision, J.H.Koenderink and A.J.Van Doorn, Biological Cybernetics, Vol.32, No.4 (1979).
4_4) 概略図を用いたアスペクトビジョン，泉正夫，川上弘起，西村崇，福永邦雄，システム制御情報学会論文誌，Vol.5, No.5 (1992).
4_5) A Survey of Object-space Hidden Surface Removal, Susan E. Dorward, International Journal of Computational Geometry and Applications, Vol.4, No.3 (1994).
4_6) Theory and Problems of Computer Graphics, Roy A. Plastock, Gordon Kalley, McGraw-Hill, Inc. (1986).
4_7) パターン理解，白井良明，オーム社 (1987).
4_8) 画像工学ハンドブック，樋渡涓二，朝倉書店 (1986).

5_1） 画像を用いた三次元計測技術の動向，太田友一，日本電気学会論文誌，Vol.107-c，No.7（1987）.

5_2） 立体視を用いた三次元計測技術，伊藤稔，日本電気学会論文誌，Vol.107-c，No.7（1987）.

5_3） ハンドアイシステムのための能動的な視覚計測の計画機能，松下俊夫，佐藤知正，Tapio Heikkila，日本ロボット学会誌，Vol.10，No.2（1992）.

5_4） 物体認識のための両眼立体視，安江利，白井良明，電子総合研究所彙報，37，1101（1973）.

5_5） 複数の基線長を利用したステレオマッチング，奥富正敏，金出武雄，電子情報通信学会論文誌D-II J75-D-II，8（1992）.

5_6） Computational Experiments with a Feature Based Stereo Algorithm，W. E. Grimson，the Institute of Electrical and Electronics Engineers Transaction，PAMI-7，1（1985）.

5_7） 輪郭点情報を用いた高速テンプレートマッチングアルゴリズム，橋本学，鷲見和彦，坂上義和，川戸慎二郎，電子情報通信学会論文誌D-II J74-D-II，10（1991）.

5_8） 信頼性の高い対応を優先した多段階ステレオ法，山口証，白井良明，中山収文，浅田稔，電子情報通信学会論文誌D-II J74-D-II，7（1991）.

5_9） 線分画像の分割とそのステレオ画像への応用，傲国利，福永邦雄，笠井保，電子情報通信学会論文誌D-II J72-D-II，6（1989）.

5_10） 特徴線分構造を用いた画対応，福永邦雄，西村崇，浅野剛，村田秀人，電子情報通信学会論文誌D-II J73-D-II，2（1990）.

5_11） Method and means for recognizing complex patterns，Hhough，P.V.C.，U.S.Patent 3069654（1962）

5_12） Gray Level Corner Detection：A Generalization and a Robust Real Time Implementation，A. Singh，M. Shneier，Computer Vision Graphic and Image Process. 51（1990）.

5_13） Gray-level corner detection，L. Kitchen，A. Rosenfeld，Pattern Recognition Letter 1（1982）.

5_14） 判別および最小2乗基準に基づく自動しきい値選定法，大津，電子通信情報学会論文集，J-63-DII，Vol.42（1980）.

5_15） 形状処理工学（III），山口富士夫，日刊工業新聞社（1988）.

発表論文

I. 本論文に関連した発表論文

1. Katsunori INOUE, Shuichi FUKUDA, Masashi OKUBO, Qing Tong, "Recognition of Objects Using Modified Coded Boundary Representation (MCBR) Method", THEORETICAL AND APPLIED MECHANICS, Vol. 41 pp. 71～76, (1992)

2. Katsunori INOUE, Shuichi FUKUDA, Masashi OKUBO, Qing Tong, "Model Based Object Recognition Using MCBR Method", IARP Workshop on Machine Vision Applications, pp. 233～238, (1992).

3. 大久保雅史, 井上勝敬, 福田収一, 山地泰久, "形状モデルCBRを用いたスケジューリング解法", 日本機械学会論文集(C編) "設計におけるフロンティア特集号", Vol. 60 No. 579, pp. 97～102, (1994).

4. 大久保雅史, 井上勝敬, 福田収一, 山地泰久, "CBR法を用いたスケジューリングモデルの開発-図形処理を用いたスケジューリング問題へのアプローチ(第1報)-, 溶接学会論文集, Vol. 13 No. 4 (1995, 11月掲載予定).

5. 大久保雅史, 井上勝敬, 山地泰久, "スケジューリング問題への適用と他の手法との比較-図形処理を用いたスケジューリング問題へのアプローチ(第2報)-, 溶接学会論文集, Vol. 13 No. 4 (1995, 11月掲載予定).

6. 大久保雅史, 井上勝敬, 鎌田隆夫, "面心を利用した3次元CADモデルと視覚情報とのパターンマッチング", 日本機械学会論文集投稿中.

II. それ以外の発表論文

1. Shuichi FUKUDA, Hideaki AIYAMA, Masashi OKUBO, "TANDIPS : AN EXPERT SYSTEM FOR NDI PROCEDURE SPECIFICATION FOR AN OIL STORAGE TANK", Proc. of ASME PV&V Conf., Vol. 177, p45～48, (1989)

2. 福田収一, 中桐 滋, 大久保雅史, "有限要素法解析支援エキスパートシスムの開発", 日本機械学会論文集(A編), Vol. 56, No. 521, pp. 168～172, (1990)

3. 福田収一, 相山英明, 大久保雅史, 長坂英明, "非破壊検査要領書作成支援システムの開発", 日本機械学会 論文集(A編), Vol. 56, No. 521, pp. 178～181, (1990)

4. Shuichi FUKUDA, Hideaki AIYAMA, Masashi OKUBO, "AN EXPERT SYSTEM FOR NDI PROCEDURE SPECIFICATION FOR A CRUDE OIL TANK", Proc. 4th Japanese-Sino Sapporo Int. Conf. Computer Applications, pp. 248～251, (1991)