

EusLispの幾何モデリング機能を用いた組立状態推論

Assembly Goal Inference using Geometric Modeling Facilities of EusLisp

松井 俊浩 坂根 茂幸 比留川 博久

T. MATSUI S. SAKANE H. HIRUKAWA

In order for an assembly planner to generate assembling sequences, the goal state which describes locations and orientations of each component in the assembly must be given. However, specifying 6 translational and rotational coordinates for each of them is a delicate and error-prone task for mechanical assembly designers. What we propose in this paper is a way to automatically infer those parameters in a plausible goal state. The planner generates candidates of combination of two parts based on fastener-matching and cylindrical matching and evaluates their utilities by interference, area of contact, and degrees of freedom constrained. A realistic demonstration of this method is shown using an assembly of a model aircraft engine. Geometric modeling facilities of the robot-oriented programming language, EusLisp, which enabled above application, is also discussed.

§1 はじめに

ロボットが原子力施設で点検、保守、組立などの知的な作業を行うためには、ロボットや環境の形状を計算機の内部で表現し、このモデルに基づいてセンサーの配置、障害物回避軌道、組立作業、協調行動などの動作を計画する必要がある。この内部表現のことを幾何モデル、それを取り扱うコンピュータソフトウェアを幾何モデラあるいはソリッドモデラと呼ぶ。

幾何モデリングは、いわゆる CADシステムとして長い研究が行われてきており、すでにいくつかの商用版がある。しかし、そのほとんどが形状の設計や表示、有限要素法による解析などを主眼とした閉じたシステムであるため、作業計画のような高次の機能や、視覚処理のように部分情報からモデルを動的に構成していく手法との結合には困難が大きい。

いくつかの既存のソリッドモデラを検討すると、その実現言語としてはオブジェクト指向型 Lispが最適であるとの結論に至った。本論文では、幾何モデリング機能を記述し知能的な計画をプログラミングするためのベースとなる言語として EusLispを導入し、その特徴、設計、機能について述べる。さらに、EusLispの原子力施設用ロボット技術への応用として、視点と光源の配置計画、動作拘束の導出、組立状態推論について論ずる。

§2 EusLispの特徴

2.1 オブジェクト指向と Lisp

EusLispは、オブジェクト指向に基づき知能ロボットのプログラミングを目的とした Lisp言語である^{1,2)}。オブジェクト指向は、文字どおり3次元のオブジェクトを直感的に理解するのに適しているだけでなく、プログラムをモジュール化し、継承によって大規模なロボットシステムに向けて拡張を加えるのに向いている。Lispは、記号・リスト処理言語でありモデル要素間の位相構造を表現するのに適している他、新たな言語のインプリメントや簡便な対話インタフェースの構築にも向いている。

Lispとオブジェクト指向の結合としては、古くは Flavor³⁾が、また Common Lispの標準として CLOS³⁾がある。これらが、Lispの上でオブジェクト指向プログラミング機能を提供するものであるのに対し、EusLispは、オブジェクト指向を基盤として Lispの機能を積層して提供するものである。これによって次のような特徴が生まれる。

- (1) symbol, cons, vectorなどの Lispの組込データ型もクラスで定義されるのでユーザーが定義するサブクラスによって拡張できる (Fig. 1)。
- (2) すべてのデータオブジェクトが同一のメモリ管理手法で管理できる。
- (3) Lispのほとんどの基本関数が、instantiate, slot, send, defclassなどの僅かな数のオブジェクト指向プリミティブで統一的に記述できる。

オブジェクト指向では、メッセージ送信時にメソッドを探索する手続きが必要になる。これが、オブジェクト指向言語の柔軟性を生み出すと同時に効率を下げる原

因にもなる。EusLispでは、メソッドキャッシュによる高速なメソッド探索と継承構造を単一スーパークラスに制限することで効率の劣化を最小限に抑えている。また、単一継承の整然とした階層構造を利用して、クラスの継承の深さに依存せず定数時間で型の判別を行う手法を実装している。さらに、フィボナッチバディ法によるメモリ管理によってさまざまな種類の大きさのメモリ要求に応えつつ高いメモリ効率を発揮することに成功している。

2.2 幾何モデラの基本設計

幾何モデラーは、ベクターやマトリクスの演算を多く含むので、Fortranのような数値演算向きの言語が適していると誤解されやすいが、実際にアルゴリズムが複雑になるのは、面やエッジ、頂点などの位相情報の管理と特異点の扱いであり、アルゴリズムとモデルの記述に重要なのはリスト操作である。したがって、リスト処理機能を備えたオブジェクト指向言語は、幾何モデラの記述にとって最適な言語となる。

一方、集合演算や干渉検査などの操作の実行には、

数値演算の効率が影響する。したがって、これらの幾何演算プリミティブは組み込み関数として効率の向上を図るのが望ましい。EusLispでは、浮動小数点数だけを要素とするベクタや配列をフLOATベクタ、マトリクスとして組み込み型として定義し、それらの基本演算を数十の組み込み関数として用意している。

このようなアプローチにより、従来のFortranによる実装に比べて、EusLispは、約十分の1のコード量、より短い開発期間、同等の実行効率を実現している。

形状表現の手法には、CSG、Brep、掃引法、ボクセルなど多数あるが、利用の目的に応じて使い分けられなければならない。ロボットへの応用では、視覚や触覚などのセンサーにとっては、表面の状況に興味がある。作業計画、軌道計画にとっては、移動・回転などの変換と干渉検査が重要な計算となる。これらを総合すると、Brepを内部表現とするのがもっとも適当である。他の表現法からBrepに変換することも可能であるので、Brepは最も汎用性の高い表現であると言える。

そこで、EusLispでは、物体の形状をBrep(Boundary Representation)によって表現する。Brepとは、方向付けられたface、edgeなどの要素で物体の境界面を表現し、

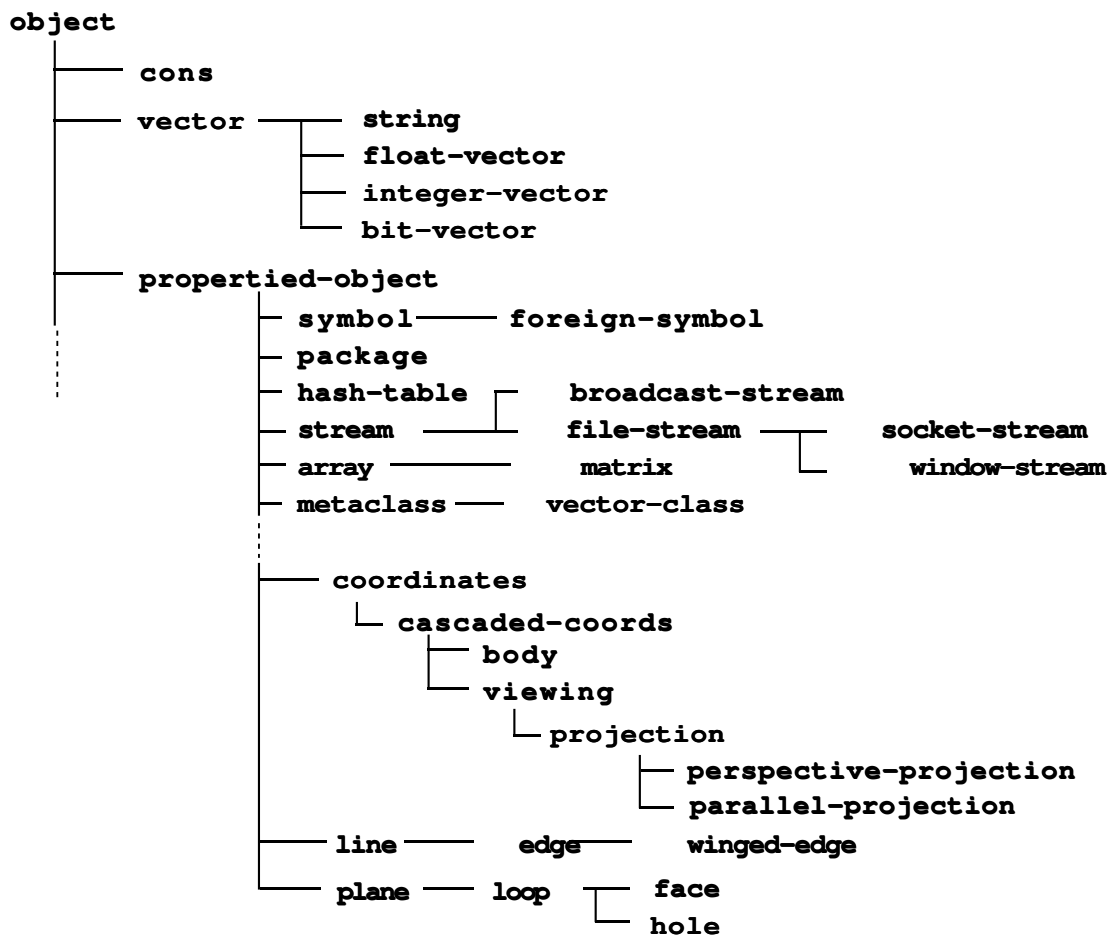


Fig 1. EusLisp's Class Hierarchy

その向きによって物体の内と外を区別する手法である。また、モデルの定義には、人間に直感的にわかりやすい CSGで行う。CSGとは、円柱、直方体などのプリミティブ形状を集合演算によって合成することで複雑な形状を表現する方法である。

2.3 幾何モデリング機能

Fig. 1に、EusLispが定義するデータ型の階層構造の一部を示す。Coordinatesまでが Common Lispの実現に使われているクラス、それ以下が幾何モデルの表現のために定義されたクラスである。

また、Fig. 2は、物体 (body)、面 (face)、稜線 (edge)と頂点(vertex)のオブジェクトの内容と接続関係を表している。頂点は、3次元のベクタで表現される。EusLispの幾何モデルは、すべて多面体表現であり、2次局面や自由局面を持つことはできない。円柱プリミティブは、多角柱で近似される。

座標系は、coordinatesオブジェクトの回転マトリクスと並進ベクタによって表現する。任意の座標系における回転、並進をメッセージによって指定できる。cascaded-coordsは、座標系の連結を表現する。ロボットの腕のような階層的に接続されたリンク機構を表

現するのに必要となる。これらの座標系オブジェクトは、物体だけでなく、視点や投影の位置、方向を示すのにも使用される。

形状は、cascaded-coordsをスーパークラスとするbodyで表現する。bodyには、スーパークラスで定義される座標系に加えて、面、エッジ、頂点のリストが記録される。面は、方向付けられた外周や内部の穴のエッジのリストをループとして保持している。エッジは、両頂点と両側の面を記録する。面とエッジの関係を表す構造として winged-edgeが有名であるが、winged-edgeは冗長な情報を含んでおり、管理のコストがかかるだけでなくバグを生じやすいので EusLispはより単純な構造を採用している。冗長な情報は、メソッドによって計算法を定義しておき、必要になったときに一貫した値を求めることができる。

形状の入力は、CSG(Constructive Solid Geometry)によって行う。用意されている形状プリミティブには、直方体、円(角)柱、円(角)錐、回転体、測地ドーム(Gdome)などがある。また、多数の点から凸包を形成することで物体を作成できる。

これらの単純な形状を元に、和、差、積、平面による切断などの集合演算によって複雑な形状を定義で

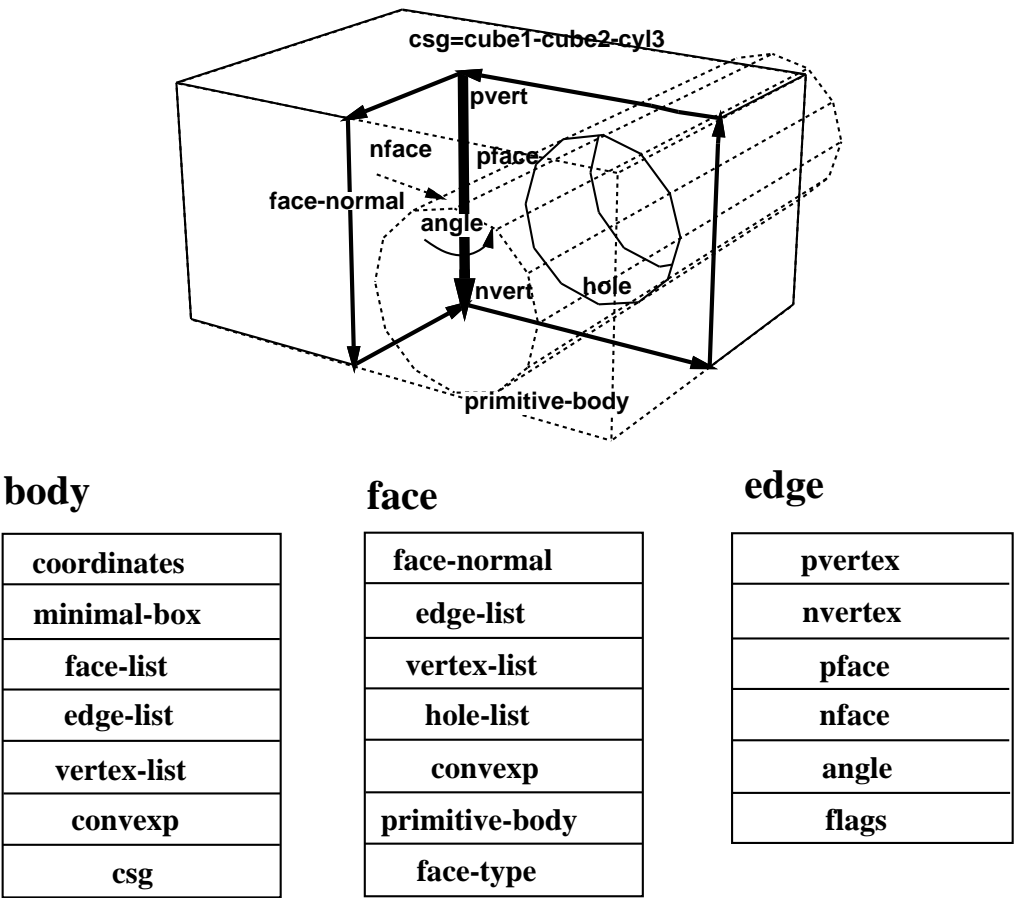


Fig. 2 Representation of body, face, and edge

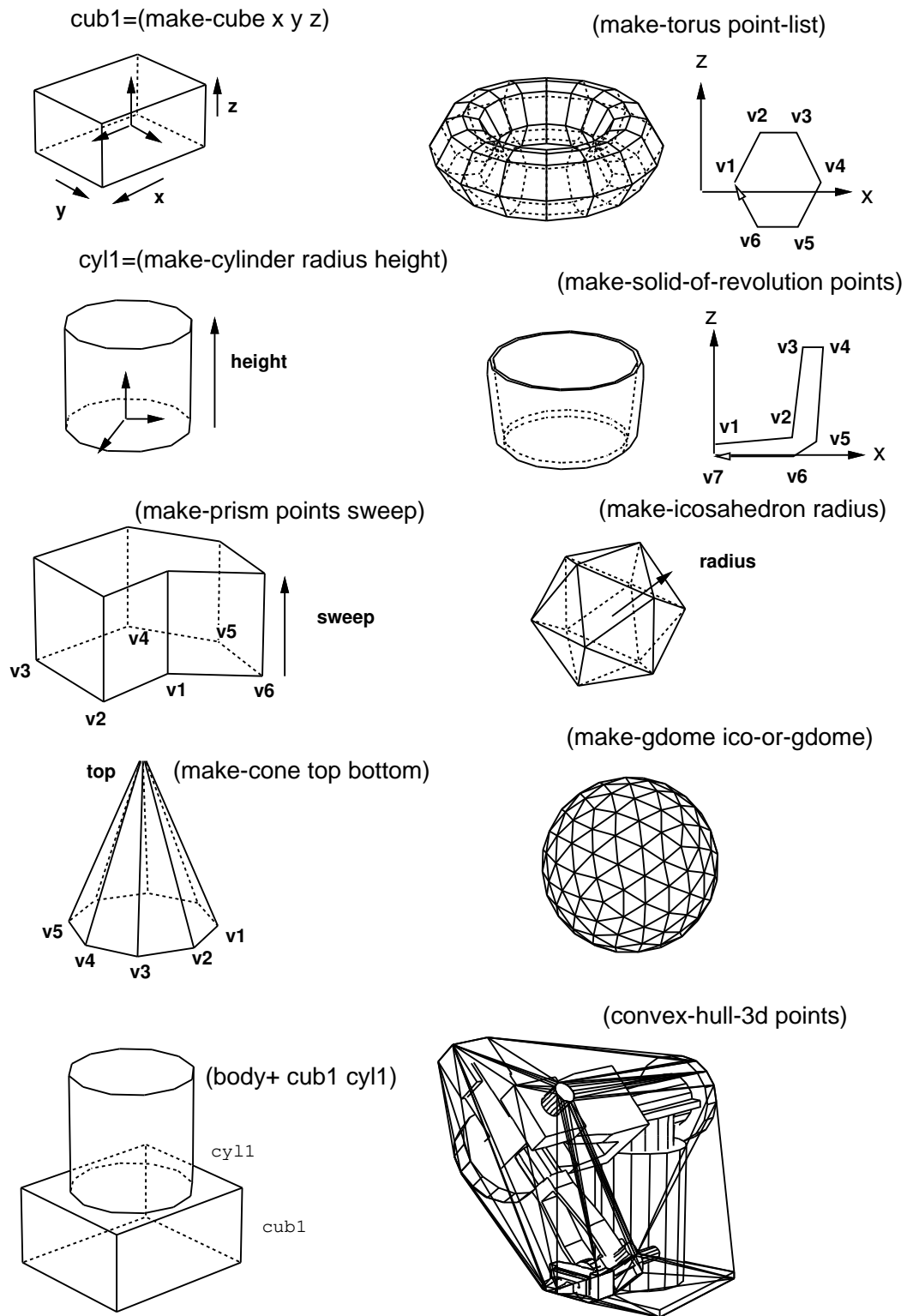


Fig. 3 Shape Primitives and Set Operations

きる。bodyとbody, face, edge, pointとの干渉を検査できる。Fig. 3に、一連のプリミティブと合成された形状の例を示す。

2.4 その他のロボット向き機能

EusLispは、上記の他にもロボットのプログラミング

に適した特性をいくつか備えている。

(1) 他言語プログラムインタフェース

EusLispのカーネルはCで記述されており、EusLispのコンパイラは中間言語としてCにコンパイルするので、EusLispプログラムとCとの結合は容易である。load-foreign、defforeign、defun-c-callableなど

の関数で、他言語のオブジェクトモジュール、ライブラリをロードし、それらの関数を Lisp から呼んだり、逆に C のプログラムから呼ばれる call-back 関数を EusLisp で定義することが可能である。この方法で Fortran の数値計算ライブラリ、C の画像処理プログラム、window ライブラリなどが利用できるようになっている。

(2) グラフィックス環境

すぐれたプログラミング環境とマンマシンインタフェースのために、ウィンドウ環境は必須である。上記の他言語インタフェースを用いて Xlib の全機能が利用できる。Xview などのツールキットの他、EusLisp 自身で記述したツールキットがある。隠線消去やレイトレーシングされた幾何モデルが Xwindow に表示できるほか、PostScript や PICT 形式での出力も可能である。

(3) 通信機能

EusLisp は、Unix のほとんどのシステムコールと標準のライブラリ関数を組み込み関数として定義している。プロセス間通信のために、共有メモリ、FIFO、ソケットなどのインタフェースがある。defasync マクロを用いれば、非同期入出力を実現することができる。この機能を用いて VxWorks のような実時間 OS の RPC (Remote Procedure Call) インタフェースが実現されている。

EusLisp 同士が通信する場合は、幾何モデルのような複雑な構造を通信する必要が生じる。EusLisp は、*print-object* を T にすることで、再帰的な参照関係を持ったオブジェクトをコピーしたり、再読み込み可能な形式で print することができる。上記のプロセス間通信と組み合わせることで、任意の構造を送りあうことができる。

§3 EusLisp のロボット応用

3.1 光源・視点配置計画

ロボットが、配管が交錯した環境で対象部位に適当な光を照射し、物体を認識するためには、その環境にあった光源とカメラの配置を計画する必要がある。対象部位の環境を幾何モデルによって記述し、障害物による隠れのない光源位置と視点の関係を計算する⁵⁾。

Fig. 4 は、環境モデルを EusLisp で作成し、隠線消去した表示である。ここで、対象物を中心として測地ドームを設定する。測地ドームは、立体角をほぼ均等に分割するために用いる。測地ドームの各々の窓の中心と対象物を結ぶ直線を考え、光源と視点を配置しようとする球殻との間で障害物との干渉がないかを検査する。

干渉がない方向は、光源や視点を配置する点の候補となる。視点については、そこから対象物が見えるだけでなく、実行しようとしている作業に対して対象物を監視するのに適した方向がある。たとえば、物体が回転しないように把持したいのであれば、その回転が最もよく検出できるような方向が存在する。上記の候補となる方向に対して、観察されるエッジの直交性を特異値分解を用いて評価する。この評価値の高い視点の方向が決まれば、それに対応した光源の位置を定めることができる。

3.2 動作拘束の導出

組立作業とは、物体の接触状態を変えることである。組立状態を接触によって記述することもできる

(a) shadow-free regions on a geodesic dome

(b) candidate directions for light source placement

Fig. 4 Viewpoint and Light-Source Location Planning using Geodesic domes

し、接触状態の遷移によって組立作業を記述することも可能である。接触状態および接触によって生ずる動作拘束は、幾何モデルから導出できる⁹⁾。

例として Fig. 5に示すように body-Aがbody-Bと接触している場合を考えると、知りたいのは body-Aは接触状態を破らずにどのように動けるか、どのような力を及ぼしあえるかということである。接触状態にある全エッジの二つずつの端点につき、その微小変位が可能な領域を不等式の集合によって表現することができる。特異値分解によって不等式の係数行列から独立な動作が抽出できる。この解は、加えられた力に対する各々の並進方向と回転方向での動きの感度に相当する。全ての可能な動きは、12の基底の線形和で表現される。Fig. 5の例では、接触状態を変えずに矢印の方向に回転できることが解析できた。

§ 4 組立作業の計画

組立作業計画とは、部品の形状と組立状態を記述しておき、どのような順序、経路で部品を組み合わせていけば最終状態に到達できるかを計画する問題である。ここでは、バルブやエンジンなどの機械の組立作業を扱う。この場合、最終状態は与えられているので、組上がった状態から分解する手順が得られれば、その順序を逆転することで組立手順が得られると考える。

ここで提案するのは、最終状態が与えられない場合に、部品の属性や形状から組立状態を幾何学的に推論す

る問題である。二つの単純な部品について考えると、適当な位置関係というのは多数、場合によっては無限に考えられるが、制約を及ぼしあう部品が多数あれば、妥当な解は有限の数におさまる可能性がある。重力に対する安定性や接触面積、凸包の体積などから組立状態の評価を得ることも可能である。完全自動化が不可能であるとしても、局面に応じて設計者が複数の候補姿勢の中から意図したものを指定してやれば、短時間で組立ゴールを得ることができるであろう。実際、部品のモデルが与えられたとして、個々の部品の3次元座標と3つの回転角を指定することで間違いのない最終組立状態を達成するのは人間の設計者にとっても困難な作業である。期待している組立状態が得られるかどうかを見ることは、部品設計の助けにもなる。

4.1 幾何モデルの拡張

このような応用を可能にするため、EusLispの幾何モデルに拡張を加えた。一つは、接触状態を検出できるようにすることである。これは、干渉検査において特異な状態として除外していたケースをより詳細に解析し、面接触、辺接触、点接触などを見つけるものである。面接触の場合は、さらに共有している面の部分の形状や面積を計算する。

もうひとつは、面やエッジオブジェクトに、円柱の底面である、回転体の側面であるなどの情報を付加することである。これは、モデル作成の際に設計者が指定するCSGの情報を蓄積することで可能となる。

4.2 組立状態推論

すべての部品について、モデルの作成に用いたCSG情報とBrepモデルが与えられていることを仮定する。部品は、変形を受けない固い部品であるとする。ここでは、対象物として、模型飛行機用エンジンを考える。部品の組み合わせとして、面と面の接触、円柱と円柱状の穴の接触だけを考慮し、稜や頂点が関与する接触は対象外とする（そのような接触状態は、機械部品にとって有害である）。

プラナーは、仮説生成とその評価を繰り返す。組み合わせの候補は、(1)ネジ整合と、(2)円柱整合の二つの方法で生成される。(3)形の部分的合同による方法が将来の拡張として考えられる。整合は、まず、部品同士の干渉がないかを検査し、その後、接触面積による順位付けを行う。この評価法は極めて単純であるが、ペントミノのような詰め込みパズルでも容器の形を与えることなく解けることが期待できる。それぞれの組み合わせは、

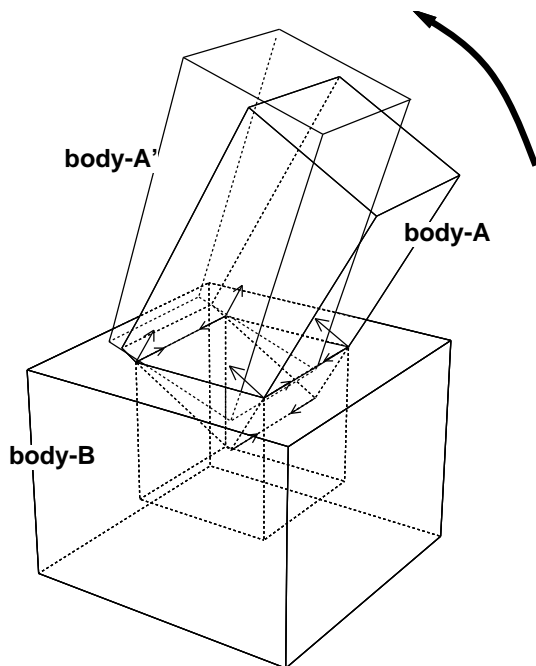


Fig. 5 Motion Constraint in Peg-in-Hole

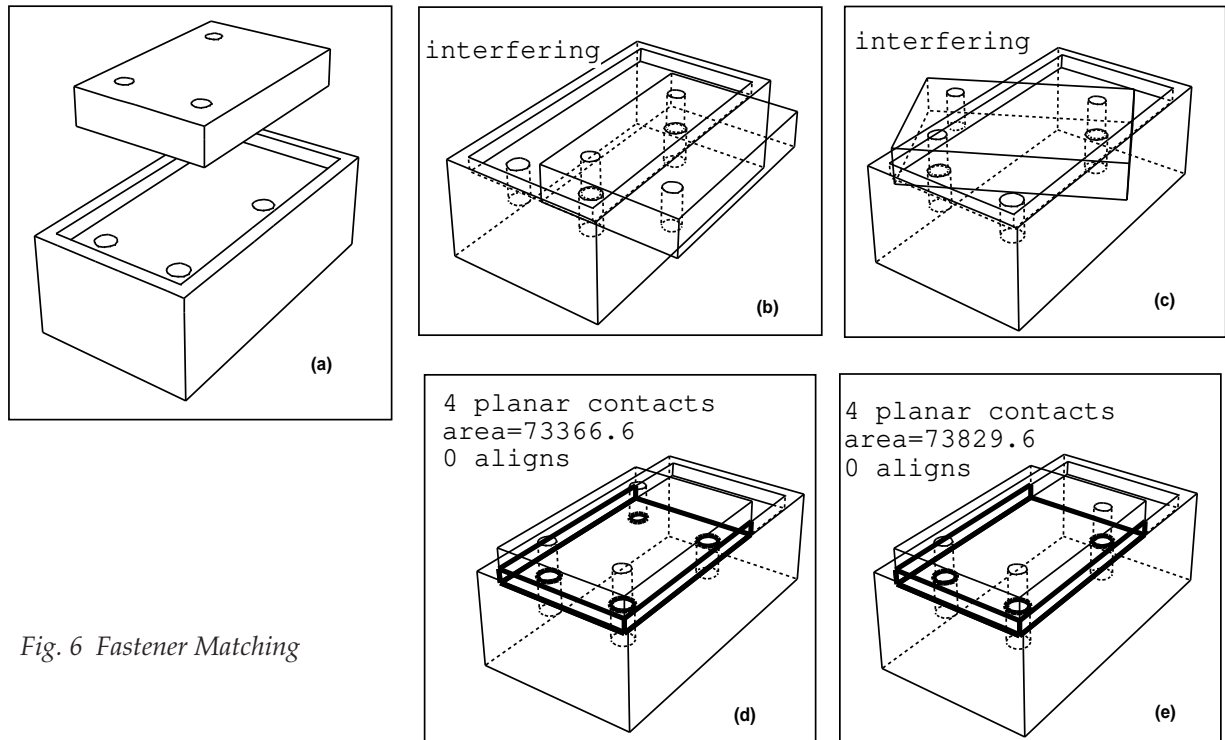


Fig. 6 Fastener Matching

より複雑な構造を形成するように組み合わせられる。部品の動きの自由度は組み合わせによって減少することが予想されるので、この自由度の減り具合も評価の尺度となる。この処理は、数式解析によって行われる。

4.3 組み合わせ候補の生成

(1) ネジ整合

CSG情報に基づき、全部品から concavity(凹)部と protrusion(凸)部を持った部品を取り出して凹部品リストと凸部品リストに記録する。凹部品については、ねじ穴とそうでないものに分別する。このうち、ねじ穴については、穴を空けられた面を見つけ、その面について、ねじ穴の位置、半径、深さなどの情報を記述する。ねじ穴の位置は、二つの物体で正確に一致する必要があり、通常は最終的にすべてがフィットする相手がみつかるはずであるから、図形の相似よりも強い制約を与える。

二つの物体のネジ穴のある面の組について、ネジ穴の直径とネジ穴間の距離を照合する。 Fig. 6は、3つのネジ穴を持つ箱とその蓋の組み合わせである。通常、ネジ組み付けでは2つ以上のネジを用いるので、組み合わせ姿勢は6自由度すべてが定まる。

Fig. 6の例では、二つ以上の穴がフィットする姿勢は8つ求まる。各々についてモデルを用いた干渉検査を行うと、このうちの6つはフランジと干渉があることが計算される。これらの候補は棄却される。干渉のない二つのケースについては接触部分の面積を計算する。 Fig. 6

で濃い実線で囲まれた部分がそれである。両者は同じに見えるが、蓋が裏返っており、一方は三つの穴のうち二つだけが一致している。このため接触面積は小さくなり、正しい姿勢が高い評価を得る。接触面の他に、エッジとエッジが接触したり、二つの面がそろって共面を構成する場合は高い評価値を与えることもできる。

(2) 円柱整合

ねじ穴がない物体については、ある円柱とそれとほぼ等しい半径を持つ円柱状の凹部の組合せを列挙する。エンジンモデルでは、回転する部分が多いので、円柱同士のマッチングが重要である。ほぼ同じ直径を持った凸

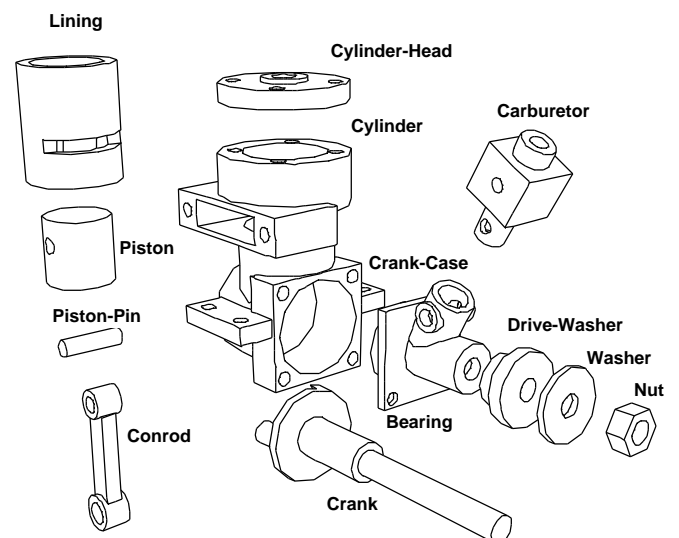


Fig. 7 Components in Model Aircraft Engine Assembly

円柱と凹円柱を基本形状に持つモデルを組み合わせる。

この組合せでは、円柱の軸回りの回転と軸方向の並進の自由度が残る。この自由度の範囲を限定するため、二つのモデルを円柱の軸に垂直な平面に直行投影し、二つの影の共通部分を求める。この共通部分に投影されるモデルの面の最短距離が、並進できる距離である。この距離を円柱の切り欠きなどから決まるいくつかの離散化された回転角度に対して計算すれば、2次元の姿勢空間における許容動作範囲を求めることもできる。回転角を固定し、他の部品と衝突せずに並進できる領域を見つけるために並進方向に2つの部品を投影する。その影の共通領域の中に、並進によって衝突する頂点、エッジあるいは面がある。共通領域の中のすべての頂点について、もう一つの部品との距離を計算し、そのなかの最小値を見つければ、それが並進できる最大距離である。

これらの円柱状の部品は、さらに別の部品と組み合わせられるので、その許容動作範囲はさらに限定され、最終的に非常に狭い許容動作範囲が得られる。たとえば、エンジンでは、crankとpistonが円柱物体であるが、各々はbearingおよびliningと組み合わせられ、さらにconrodとpiston-pinを介してリンクを形成する。

このような振舞いを解析するため、数式処理を用いる。残された回転、並進自由度を変数で表し、固定された自由度を定数座標系で定義する。それらの座標系の結合が両端での境界条件を満たすようにして解く。pistonとcrankの例では、pistonの回転が固定され、その並進領域も制限される解が得られる。

4.4 模型エンジンの組み立て計画

Fig. 7に示されるエンジンには12個の部品がある。Fig. 9にはシステムが発見したネジ整合が太い実線

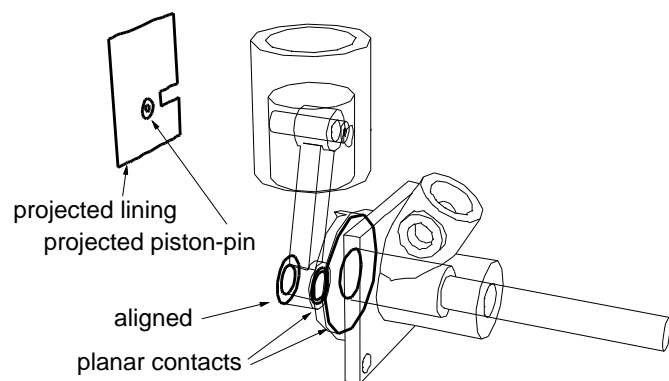


Fig. 8 Propagation of Cylinder Matching

で、円柱整合が細い実線で示されている。ネジ整合により、cylinder-headはcylinderに対して4つの可能な姿勢を持つことが発見される。cylinderとbearingも同じくネジ整合により4つの等しい評価の姿勢が求まる。そのうちの正しい姿勢は、これまでの議論だけでは定められない。実際、経験のない作業者に組立を行わせた場合、正しい組立状態を決定することはできなかった。

他の部品について円柱整合が10個発見された。円柱の大きさによる制約から、このほとんどが実際に有効な組合せである。接触面積を最大にする組み合わせを選ぶことにより、bearing、drive-washer、washerとnutをこの順で正しく組み合わせることができる。liningからpiston、piston-pin、conrod、crankを経てbearingに至るリンクには、互いに制約を及ぼしあう4つの円柱整合が見つかる。crankに対するwasher、bearingに対するcarburetorは、制約が弱過ぎてそれ以上の推論は不可能かつ興味の対象とならない。

完全には制約されない部品については許容範囲の中央値を取ることにすると、最終的にFig. 10に示すような組上がり状態を推論することができた。

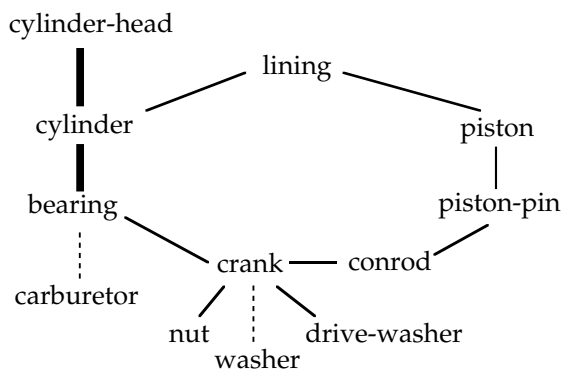


Fig. 9 Combinations and Links Found among components

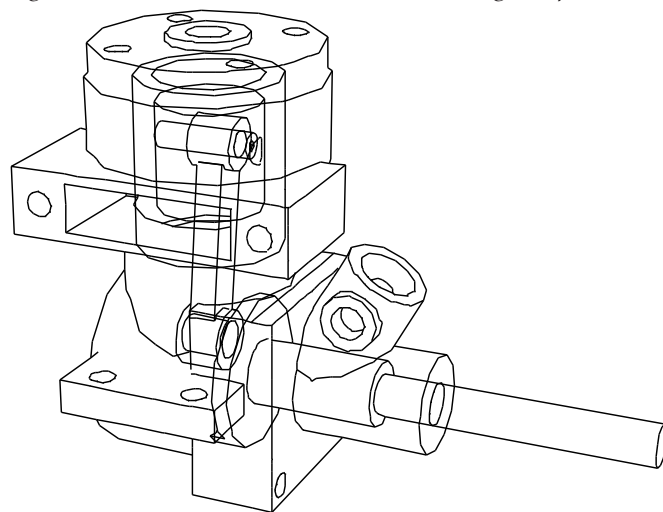


Fig. 10 Result of Assembly Goal Planning

§5 結論

EusLispは、すでに数十の研究機関に配布され、ここまで述べた応用の他にも、6自由度マニピュレータの障害物回避軌道計画^{8,9)}、多指ハンドの把握動作計画¹⁰⁾、幾何モデルを共有しながら協調した作業を行う多数台ロボットの動作計画¹¹⁾、視覚を用いた人間の動作認識¹²⁾、遠隔作業シミュレータ^{13,14)}、など、多くのロボット応用で利用されている。オブジェクト指向に基づく拡張性の高い幾何モデリング機能が、ロボットに高次の計画機能を効率よく実装するために重要な役割を演じている。さらに EusLispは、クロスオーバー研究「能動協調センシング技術」に見られるような各種の機能の統合において、共通の基盤言語としても機能している。今後、並列・非同期プログラミング機能を充実させ、自律移動ロボットへ応用することを計画中である。

謝辞

本研究は、科学技術庁原子力平和利用研究クロスオーバー研究として実施され、一部はスタンフォード大学ロボット工学研究所における在外研究として実施した。研究の機会を与えて下さった電総研知能システム部高瀬部長、スタンフォード大学 Jean-Claude Latombe教授、ならびに両研究所での支援と熱心な議論に感謝いたします。

参考文献

- 1) T. Matsui, *EusLisp version 7.27 reference manual*, ETL-TR-92-36, 1992.
- 2) T. Matsui and M. Inaba, *EusLisp: an object-based implementation of Lisp*, Journal of Information Processing, vol. 13, no. 3, 1990.
- 3) S. E. Keene, *Object-oriented programming in Common Lisp*, Addison-Wesley, 1989.
- 4) T. Matsui, T. Omata, and Y. Kuniyoshi, *Multi-agent architecture for controlling a multi-fingered robot*, Proc of IROS'92, 1992.
- 5) S. Sakane and T. Sato, *Automatic planning of light source and camera placement for an active photometric stereo system*, Proc. of the 1991 IEEE International Conference on Robotics and Automation, pp.1080-1087, 1991.
- 6) H. Hirukawa, T. Matsui and K. Takase, *A general algo-*

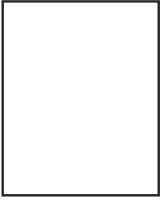
rithm for derivation and analysis of constraint for motion of polyhedra in contact, Proc. of IROS'91, pp. 38-43, 1991.

- 7) R. H. Wilson and T. Matsui, *Partitioning an assembly for infinitesimal motions in translation and rotation*, Proc. of IROS'92, Raleigh, 1992.
- 8) H. Onda, T. Hasegawa, and T. Matsui, *Collision avoidance for a 6-DOF manipulator based on empty space analysis of the 3-D real world*, Proc. of IROS'90, Tsuchiura, 1990.
- 9) T. Hasegawa, T. Suehiro, T. Ogasawara, T. Matsui, K. Kitagaki, and K. Takase, *An integrated tele-robot system with a geometric environment model and manipulation skills*, Proc. of IROS'90, pp. 335-341, 1991.
- 10) T. Omata, *Finger position computation for 3-dimensional equilibrium grasps*, Proc. of IEEE Int. Conf. on Robotics and Automation, Atlanta, pp. 216-222, 1993.
- 11) S. Sakane, T. Sato, H. Okoshi, and M. Kakikura, *A distributed sensing system with 3D model-based agents*, Proc of IROS'93, pp. 1157-1163, 1993.
- 12) Y. Kuniyoshi and H. Inoue, *Qualitative recognition of ongoing human action sequences*, Proc. of IJCAI'93, pp. 1600-1609, Cahmbery, France, 1993.
- 13) T. Matsui and M. Tsukamoto, *An integrated robot teleoperation method using Multi-Media Display*, Proc. of 5th ISRR, Tokyo, pp. 156-163, 1989.
- 14) S. Hirai, T. Sato, T. Matsui, and M. Kakikura, *Integration of a task knowledge-base and cooperative maneuvering system for the telerobot MEISTER*, Proc of IROS'90, 1990.

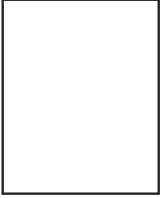
著者紹介



松井俊浩
Toshihiro MATSUI
知能システム部通信知能研究室
ロボット用プログラミングシステム
の研究に従事



坂根 茂幸
Shigeyuki SAKANE
知能システム部自律システム研究室長
ロボットのビジョン、ハンドアイシステムの研究に従事



比留川 博久
Hirohisa HIRUKAWA
知能システム部行動知能研究室
幾何モデルを用いたロボットの動作プログラミングの研究に従事

