幾何拘束に基づく形状モデルの構成操作†

日高康雄†† 安藤英俊†††
鈴木宏正‡ 木村文彦‡‡

最近の機械系CADの研究の流れの一つに，設計対象の満たすべき幾何拘束を計算機に入力し，計算機が形状の幾何属性を幾何推論によって求めることで，形状の直接表現よりも設計仕様に近い段階から設計を支援しようという試みがある．しかし，設計者が幾何拘束を計算機に入力する方法は，あまり研究が行われていない 本論文では，形状要素と位相構造の入力を行う形状構成操作，幾何拘束を設定する場所の指定，幾何拘束のパラメタ値の入力の三種類の操作を使って，幾何拘束と現行の形状モデルとを組み合わせた形状モデルを計算機内に構成する方法を提案する．形状構成操作には，集合演算と局所変形操作を可能とし，形状構成の大きな自由度を許す．また設計者によって幾何拘束がすべて入力されない場合にも，幾何属性の決定を可能とするために，デフォルトの幾何拘束を導入する．使用者がさらに幾何拘束を入力し，デフォルトの幾何拘束が不要になった時にそれを削除するために，幾何拘束の書き換え処理を導入する．幾何拘束の書き換えは，幾何拘束と幾何属性の依存関係に基づいて行い，書き換え後の幾何推論を保証する．これらの手法を用いた二次元形状モデリングシステムを試作し，形状を段階的に詳細化する過程に沿って，形状モデルを構成できること，形状モデルを構成する自由度が大きいこと，興味のない部分を詳細化せずに残し，興味のある部分から詳細化できることなどの有効性を確認した．

1. はじめに

機械製品の機能の多くは，形状によって発現しているため，その設計においては，与えられた設計仕様を満たすような製品の形状を定める作業が重要である．しかし，現行の機械系CADシステムでは，形状を直接表現しているために設計仕様を扱うことができない．それゆえ，設計者は，設計がほぼ終了して形状が大部分決まってからでなければ，CADシステムを使うことができない．これは，設計生産活動において頻繁に起こる仕様の変更に応じた製品形状の変更に対し，CADシステムが有効でないことを意味する．このような製品設計を計算機によって支援するためには，結果として得られた形状だけではなく，なぜ，その形状になっているのかを計算機の中に表現しなければならない．

製品に対する設計仕様をそのまま計算機内にモデル化するのは，極めて困難な問題である．しかし，設計仕様とは，製品の機能に要求される条件であることと，機械製品の機能の多くは，形状によって発現されていることを考えると，設計仕様の多くは，形状に対する拘束条件に置き換えることができるといえる．設計仕様から形状に対する拘束条件への変換は，人間の設計者が行い，その拘束条件からそれを満たす形状を導く作業を計算機化するだけでも，大きな利益が得られる．このような立場で，形状に対する拘束条件によって，形状を扱うための研究が盛んに行われてきており，その成果も上がってきている[1)~8)]．

だが，これまでの研究の多くは，拘束条件の表現方法と，拘束条件から形状を求める方法に関する研究であった．そこでの拘束条件の表現方法は，形状を求める処理の都合で決められたもので，人間が記述するための表現方法ではなかった．人間が拘束条件を記述するためには，グラフィックス表示を用いた入力方法を開発する必要がある．

寺沢らは，初期形状を用いた拘束条件の入力方法を示した[9)]が，初期形状を保存しなければならないため，初期形状と大きく異なった形状に変更するのは困難であった．山口らは，形状を構成する操作の持つ意味に基づいて，拘束条件を入力する方法を示した[10)]．しかし，拘束条件を陽に扱っていないために，小さな入力の自由度しか得られなかった．両者とも，集合演算のような形状構成操作は実現できなかった．

本研究の目的は，設計者の立場に立った拘束条件の

† Construction and Manipulation of a Geometric Model with Geometric Constraints by YASUO HIDAKA (Department of Electric Engineering, Faculty of Engineering, The University of Tokyo), HIDETOSHI ANDO (Department of Precision Machinery, Faculty of Engineering, The University of Tokyo), HIROMASA SUZUKI (Department of Computer and Graphic Science, College of Arts and Sciences, The University of Tokyo) and FUMIHIKO KIMURA (Factory Automation, Research Center of Advanced Science and Technology, The University of Tokyo).
†† 東京大学工学部電気工学科
††† 東京大学工学部精密機械工学科
‡ 東京大学教養学部情報・図形科学教室
‡‡ 東京大学先端科学技術研究センターファクトリーオートメーション分野

入力方法を確立することである．そのために，設計者が決定した拘束条件から順番に入力でき，集合演算などの強力な手段が使えるような方法をとる．

2. 幾何拘束に基づく形状モデルの構成操作

一般に，境界表現の形状モデルは，頂点，稜線，面などの形状要素，形状要素の幾何的なパラメタを表す座標などの幾何属性，形状要素同士の接続関係を表す位相構造からなっている．一方，形状に対する幾何学的な拘束条件のことを幾何拘束と言い，幾何拘束から幾何形状を求めることを幾何推論と言う．幾何拘束には，寸法や形状特徴のように形状要素を直接規定するものと，部品同士の干渉の回避のように明確に規定しないものとがあるが，本研究では前者を扱う．つまり，幾何推論は，形状要素に関する幾何拘束から，形状要素の幾何属性を推論することとし，幾何推論によって形状要素を生成することは考えない．このような立場では，形状要素と幾何拘束の間には，

形状要素の存在→幾何拘束の存在，

という順序関係が考えられる．すなわち，モデルを構成する際には，形状要素を先に作り，幾何拘束の追加はその後になる．

さて，本研究では，設計者が形状の詳細設計をする際の過程を，図1のように仮定する．すなわち，

a．設計対象の存在が決まる．

b．一部の形状要素（稜線や頂点など）の存在と位相構造（形状要素の接続関係）が決まる．

c．既に存在する形状要素について，どの要素と要素の間に幾何拘束を与えるかが決まる．

d．寸法のようにパラメタを必要とする幾何拘束に，取りあえず適当な値を与えておく．

e．他の形状要素の存在や位相構造，幾何拘束を与える場所が決まっていく（b，c，dを繰り返す）．

f．すべての形状要素の存在，位相構造，幾何拘束の場所が決まり，最終的な拘束のパラメタ値が決まる．

ここでいうパラメタとは，寸法，位置，角度，半径，直径，曲率などである．

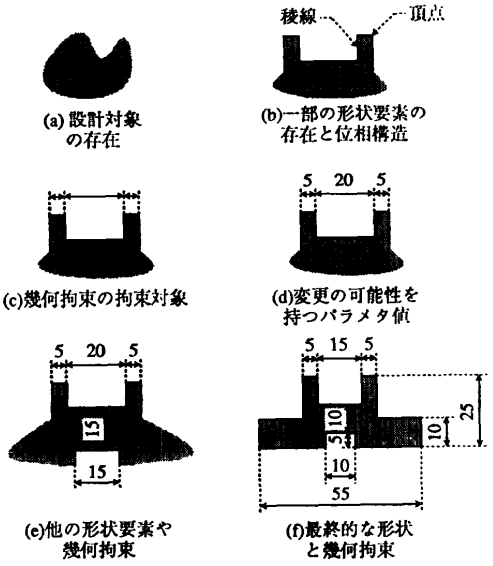


図 1 形状の詳細設計の過程

Fig. 1 The process of shape design.

この仮定の主張は，形状の設計が局所的には，形状要素の存在と位相構造，幾何拘束の場所，パラメタ値の順番で進み，大局的には，それらが前後して進んでいくと考えることである．

そこで，本研究では，幾何拘束に基づく形状モデルを，次の三種類の操作によって構成する．

1．形状要素と位相構造の入力

2．幾何拘束の拘束対象の指定

3．幾何拘束のパラメタ値の入力

ただしこのままでは，形状要素や位相構造が入力されても，パラメタ値が入力されるまで形状が決まらず，グラフィック表示ができない．これを解決するために，デフォルトの幾何拘束を導入する．デフォルトの幾何拘束は，形状要素と位相構造を入力する時に，システムによって自動的に生成される幾何拘束である．その目的は，人間の意図する幾何拘束が入力されるまでの間，仮の形状を規定しておくことと，通常人間の意識に上らないような幾何拘束を，使用者がわざわざ指定する必要をなくすことである．

次に，本研究では，形状の位相構造も，幾何拘束とは独立に決定されるもので，幾何推論によって求められるものではないと考える．そこで，形状要素を，位相構造を表すための位相要素と，幾何属性を表し，幾何拘束を記述するための幾何要素とに分ける．幾何要素は，点，曲線，曲面などである．幾何拘束は，幾何要素の間の拘束関係で，幾何拘束からの幾何推論によって，幾何要素の幾何属性（座標，方程式の係数など）が求められる．位相要素は，頂点，稜線，面などで，境界を持っており，境界には他の位相要素が存在する．位相要素の接続関係が，位相構造である．そして，位相要素は幾何属性を直接表さずに，その幾何属性を表している幾何要素の名前を持っている．幾何要素の中には，位相要素から参照されない補助的な幾何

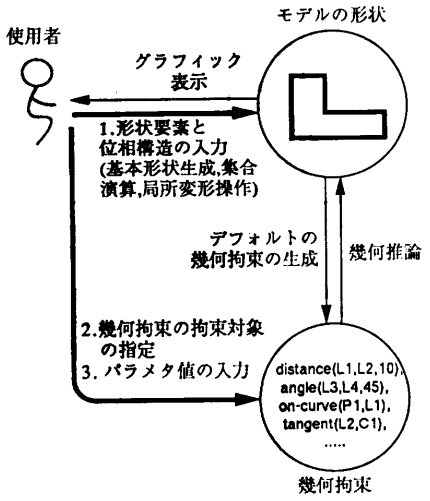


図 2 モデルの構成操作と使用者，幾何拘束，形状の関係

Fig. 2 Modelling operations and the relationship between a user, geometric constraints and the shape of the model.

要素もある．**図 2** は，これらの構成操作と使用者，幾何拘束，モデルの形状の関係を示している．

1．形状要素と位相構造の入力

これは，図1の(b)に相当する操作である．具体的には，現行の形状モデラと同様に，基本形状の生成，形状同士の集合演算，形状の局所変形操作を使って行う．これらの操作では，使用者には幾何拘束を意識させない．ユーザインタフェースには，グラフィックス表示を用いた対話的な処理を使い，形状の位置や大きさはすべて画面上で指定する．

システムは，入力された形状が幾何推論によって導出されるような，デフォルトの幾何拘束を自動的に生成する．形状に対するデフォルトの幾何拘束は，唯一には定まらない．しかし，デフォルトの幾何拘束は，使用者が意識したものではないので，使用者に表示をするのは控える．

2．幾何拘束の拘束対象の指定

これは，図1の(c)に相当する操作である．使用者が指定した場所に幾何拘束を追加し，その代わりに取り除くことのできる幾何拘束を削除する．この処理を，“幾何拘束の書き換え”という．この時，形状の持つ意味は変化するが，形状モデル自身には変化が起きない．

使用者の入力した幾何拘束を追加すると，拘束は過剰になる．一般に過剰な拘束は形状の幾何学的不整合を起こすと考えられるが，入力する幾何拘束をパラメタを持つものに限定し，パラメタの値を適切に計算することによって幾何学的不整合を避けることができる．

過剰な幾何拘束は，一時的には不整合を起こさなくても，後に一部のパラメタの変更によって不整合を起こす可能性を持っている．不整合が起きた時点でその不整合を避ける方法も考えられるが，本研究では，不整合が起きないうちに過剰な拘束を削除する方法をとる．この削除する拘束の求め方は後述する．

3．幾何拘束のパラメタ値の入力

これは，図1の(d)に相当する操作である．2.で入力された幾何拘束を選択し，そのパラメタ値を変更する．モデルの持つ幾何拘束は，過剰ではなくなっているため，パラメタ値の変更が，拘束の過剰による矛盾を起こすことはない．

パラメタ値の変更は，形状の各部の変化をもたらす．デフォルトの幾何拘束がどこにあるのか，使用者にはわからないため，形状の変化を正確には予測できない．しかし，形状の中で，デフォルトの幾何拘束が残っている部分は，使用者の意図する幾何拘束がまだ与えられていない部分である．使用者は，その部分の位相構造は決めたかもしれないが，幾何拘束の場所を決めていないのであるから，位相構造が変化しない限り，形状のどの部分が変化しても構わない．逆に，設計者によって幾何拘束を与えられた部分は，必ずその拘束が保証される．

3. ATMS による幾何推論の依存関係管理

幾何拘束の書き換えは，幾何推論における依存関係に基づいて行う．この依存関係は，ATMS (Assumption-Based Truth Maintenance System)[11] によって管理される．ATMS は，本来非単調論理における整合性管理システムであり，本研究でも幾何拘束の変更時に推論事実の更新を的確に行うために導入されたものである[1]．推論のメカニズムには，Rete アルゴリズム[12]を用いたプロダクションルールベースの高速な前向き推論を使う．ここでは，ATMS の基本的な用語と，幾何推論における依存関係の説明をする．

3.1 ATMS のデータ構造

● ノード (node)

ノードは，ATMS における基本的なデータ構造であり，各事実に対応して一つのノードが作られる．例えば，“直線 L1 と L2 は並行で距離がDである”といった幾何拘束や，“点 P1 の座標は (x1, y1) であ

る”といった幾何推論によって求められた幾何属性が，ノードの表す事実である．ノードには，その事実がどの仮説に依存しているかを表すラベルが付けられる．

- 仮説 (assumption)

推論する前に仮定された事実を表すノードを，特に仮説という．幾何拘束は仮説として表現される．

- 環境 (environment)

仮説の集合．環境は，その中に含まれる仮説をすべて同時に仮定することを意味する．空の環境は，何の仮定もしないこと，つまり，「常に」ということを意味する．

- ラベル (label)

各ノードに付加される環境の集合．ラベルは，その中に含まれる環境のいずれによっても，そのノードが導出されることを意味する．

3.2 幾何推論における依存関係

図3に幾何要素と幾何拘束の例を示す．幾何要素は，L1からL6の6本の直線と，P1からP4の四つの点であり，幾何拘束は，A1: ref-x(L1) から A6: angle (L3, L4, 45) の六つである．ref-x(L1) や ref-y(L2) は，L1 や L2 が基準要素であることを，distance (L1, L3, 10) は，L1 と L3 の距離が 10 であることを，angle (L3, L4, 45) は，L3 と L4 のなす角度が 45 度であることをそれぞれ意味している．この形状の位相要素を図4に示す．L5 と L6 の二つが補助的な幾何要素であり，他の幾何要素は位相要素から参照されている．この例について幾何推論を行った時の各事実の依存関係を図5に示す．幾何推論は，それまでの事実に形状決定規則を適用して別の事実を導く，ということの繰り返しで行われる．最初は，幾何拘束のみが事実であるが，次第に幾何属性が事実として追加されていく．形状決定規則は，“二直線間の距離と一方の直線の幾何属性→もう一方の直線の幾何属性”とか“二直線間の幾何属性→交点の幾何属性”のような形状決定のパタンをあらかじめ用意したものである．ただし，この例題は，説明のために簡略化してある．実際の幾何推論では，“点 P1 が直線 L1 にのっている”などの幾何拘束も使わなければ，幾何推論はできない．

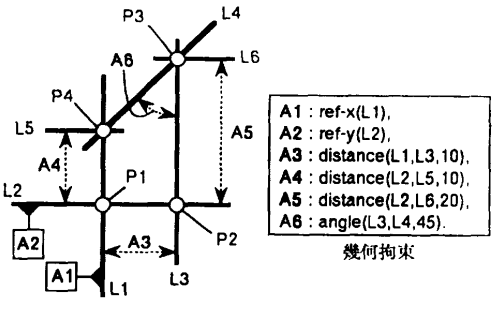


図3 幾何要素と幾何拘束の例
Fig. 3 An example of the geometric elements and geometric constraints.

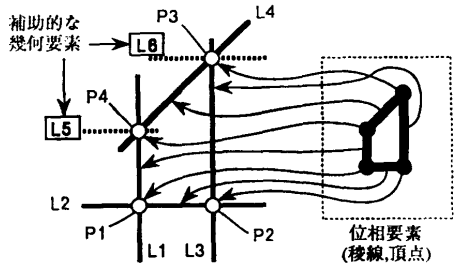
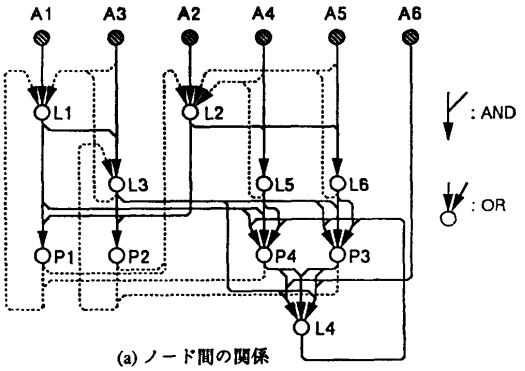


図4 位相要素と補助的な幾何要素の例
Fig. 4 An example of the topological elements and auxiliary geometric elements.



(a) ノード間の関係

L1:{{A1}},
L2:{{A2}},
L3:{{A1,A3}},
L4:{{A1,A2,A3,A4,A5},
{A1,A2,A3,A5,A6},
{A1,A2,A3,A4,A6}},
L5:{{A2,A4}},
L6:{{A2,A5}},
P1:{{A1,A2}},
P2:{{A1,A2,A3}},
P3:{{A1,A2,A3,A5},
{A1,A2,A3,A4,A6}},
P4:{{A1,A2,A4},
{A1,A2,A3,A5,A6}}.

(b) 各ノードの持つラベル

図5 幾何推論における依存関係の例
Fig. 5 An example of dependencies in geometric reasoning.

図5の(a)は，どの事実からどの事実が推論されるかを表している．上部の斜線の入った円は ATMS の仮説を示し，図3で示した A1 から A6 の幾何拘束を表している．他の白い円は，幾何推論によって求められた幾何要素の幾何属性を表すノードである．円の間を結んでいる矢印は，実線矢印，点線矢印ともに，推論の1ステップを表している．矢印の合流は AND

条件である．一つのノードを複数の矢印が指しているのは，OR 条件である．実線の矢印は，その先のノードのラベルが更新されたもので，点線の矢印は，ラベルに変化がなかったものである．図5の(b)は，各ノードの持つラベルを表している．

A1 から L1 への実線の矢印は，A1: ref-x(L1) という仮説から L1 の幾何属性が直接求められることを表している．L1 は，A1 のみから求めることができ，そのラベルは {{A1}} となる．L1 と L2 から P1 への実線の矢印は，L1 と L2 の幾何属性から P1 の幾何属性が決まることを表している．そして，L1 と L2 のラベルの直積から P1 のラベルは，{{A1, A2}} となる．P1 と P2 から L2 への点線の矢印は，P1 の幾何属性と P2 の幾何属性から，L2 の幾何属性が決まることを表している．そして，P1 と P2 のラベルの直積をとって，{{A1, A2, A3}} が加えられて，L2 のラベルは {{A2}, {A1, A2, A3}} となるのであるが，{A2} は，{A1, A2, A3} を一般化したものであるため，L2 のラベルは {{A2}} のままで変わらない．

P3 のラベルの {{A1, A2, A3, A5}, {A1, A2, A3, A4, A6}} は，P3 の幾何属性が，A1, A2, A3, A5 からでも，A1, A2, A3, A4, A6 からでも求められることを表している．

点線の矢印は，一見，無駄な推論を行っているように見える．しかし，以下で述べる幾何拘束の書き換えにおいては，これらが有効に生かされる．そして，幾何拘束の述語や推論規則の対称性によって，可能なかぎりの推論を行うことが，幾何拘束の書き換えの自由度を向上させることにつながる．

4. 幾何拘束の書き換え

幾何拘束の書き換えで，使用者が入力する幾何拘束とその代わりに削除する幾何拘束は，パラメタを持つ幾何拘束のみに限定する．これは，形状の設計者が意識する幾何拘束の多くがパラメタに関する幾何拘束であること，パラメタを持たない幾何拘束は位相構造と密接な関連があるため，位相構造の生成時に一緒に作られるデフォルトの幾何拘束としての表現が自然であること，パラメタを持たない幾何拘束を不用意に書き換えると矛盾を起こしたり，後の幾何拘束の書き換えで不都合が生じる恐れがあること，わかりやすいグラフィック表示が困難であることなどの理由による．

4.1 幾何拘束の書き換え処理の概要

幾何拘束の書き換え処理の概要は，以下のとおりである．

1．もし，新しい幾何拘束を追加するために，新しい補助的な幾何要素が必要であれば，それを生成する．例えば，距離を平行な二直線間の距離としてしか表せない場合に，直線と点の間の距離を幾何拘束として入力するには，与えられた直線に平行で，与えられた点を通る直線を新たに生成する．

2．新しい幾何拘束（パラメタを持つ幾何拘束）をモデルの中に追加し，一時的に幾何拘束が過剰な状態にする．ただし，矛盾を導くことのないようにするため，現在の形状の幾何属性を参照して，整合性の取れた矛盾を導かないパラメタ値を持った幾何拘束を追加する．

3．過剰となったパラメタを持つ幾何拘束の中から，削除可能な幾何拘束の組合わせを選び出す．

4．削除可能な幾何拘束の組合わせの中で，デフォルトの幾何拘束だけの組合わせがあれば，その中の適当なものを選んで削除する．

5．デフォルトの幾何拘束だけの組合わせがない場合は，システムは，削除可能な幾何拘束の組合わせを使用者に提示し，使用者との対話的処理によって削除する幾何拘束の組合わせを決定する．使用者が取り除こうと思う幾何拘束がない場合でも，別の幾何拘束を追加することによって，選択の余地が広がることがある．そこで，使用者には，さらに幾何拘束を追加することも許す．

6．不要になった補助的な幾何要素を削除する．また，その要素に関連するパラメタを持たない幾何拘束も削除する．不要な幾何要素とは，その幾何要素に関する幾何拘束が，パラメタを持たない幾何拘束も含めてすべて，削除可能となっている幾何要素である．

4.2 幾何推論の依存関係に基づいた，削除可能な幾何拘束の検出

図3～図5の例では，図5(a)を見ると，A6 はなくても，すべての幾何要素の幾何属性は求められることがわかる．また，A4 がなければ幾何属性を求められない幾何要素は L5 だけであり，A5 がなければ幾何属性を求められないのは L6 だけである．図4でわかるように L5 と L6 は補助的な幾何要素であるので，A4, A5, A6 は削除可能である．これは，特定の幾何拘束が削除可能であるかどうかを判定する方法であるが，これを拡張すると，同時に削除可能な幾何拘

束の組合わせ（環境）をすべて求めることができる．以下に，そのアルゴリズムを述べる．

1．パラメタを持つ幾何拘束の仮説をすべて含む環境を作り，それだけを唯一の要素とするリストを candlist とする．candlist は，削除できそうな環境のリストを保持する．

2．必須の幾何要素（位相要素から参照される幾何要素と，中心線などの削除してはならない幾何要素）の幾何属性のノードの各々に対して，以下の処理を行う．

a．candlist 中の環境(A)とこれらのノードのラベル中の環境(B)のすべての組合わせに対して，AとBの差 $(A \cap \bar{B})$ を求めて環境のリストを作る．

b．a．で求めた環境のリストから，重複している環境と，他の環境の真部分集合となっている環境を除いたものを candlist の新しい値とする．

3．candlist 中の各環境は，削除可能な幾何拘束の組合わせを表している．

以上が幾何拘束の書き換えのアルゴリズムであるが，本手法の利点は，幾何拘束の書き換えを健全に行える，つまり，書き換えによって必要な幾何拘束が欠落し，幾何推論が不可能になる恐れがないことである．

幾何推論では，推論の方法や用意されている推論規則によっては幾何拘束が十分である時でも幾何推論に失敗する場合がある．幾何拘束の書き換え処理においても不用意に幾何拘束を書き換えると，幾何推論ができないようになってしまう危険がある．しかし，本手法では，幾何推論における依存関係を根拠として幾何拘束の書き換えを行うため，書き換えられた幾何拘束からの幾何推論も保証されている．数式処理で連立方程式を解く手法を使わず，プロダクションルールだけで幾何推論をする場合，十分な幾何拘束が与えられれば必ず幾何推論に成功するように，推論規則を完全にするのは，非常に困難である．本手法では，幾何推論処理の能力に合わせて幾何拘束を書き換えるので，システムを頑健にすることができる．

5. 位相構造を変化させる形状構成操作

位相構造を変化させる形状構成操作には，通常の形状モデラと同様に，基本形状生成，集合演算，局所変形操作がある．**図6**にそれらの例を示す．これらは，形状要素と位相構造の入力を志向する構成操作で，幾何拘束を使用者に意識させない．

これらの処理における通常の形状モデラとの違いは，幾何属性を直接生成せずに，目的の幾何属性を導くようなデフォルトの幾何拘束を生成する点である．ただし，過渡的には，集合演算の交点，交線計算などで，幾何属性を直接使った計算も行われる．しかし，これは交点を持つかどうかの判断などに用いるものであり，操作終了後は，生成された交点や交線の幾何属性も，幾何推論によって求められるように，デフォルトの幾何拘束が生成される．

これらの処理の概要は，次のようになる．

1．この操作によって，どのような形状要素が新たに作り出され，既に存在する形状要素がどのように変更，削除されるかを，従来の形状モデラと同様の計算によって求める．

2．新たに生成される幾何要素を，補助的な幾何要素としてモデルに追加する．

3．生成した幾何要素の幾何属性を導けるようなデフォルトの幾何拘束を生成し，モデルに追加する．

4．位相要素，位相構造の変更を行う．また，位相要素の参照する幾何要素の変更も行う．

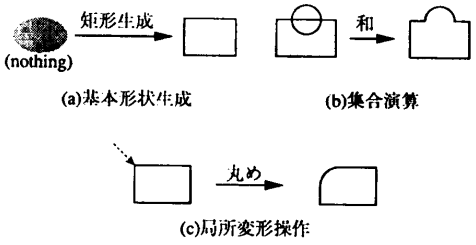


図 6 位相構造を変化させる形状構成操作
Fig. 6 Shape construction operations for the change of topological structures.

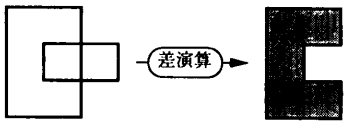


図 7 二つの矩形の差の集合演算
Fig. 7 A difference operation of two rectangles.

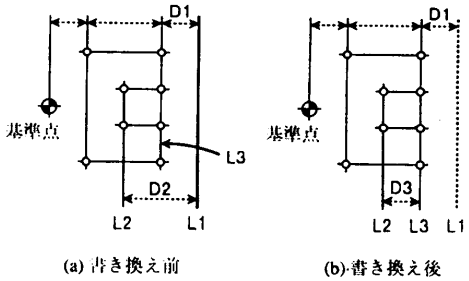
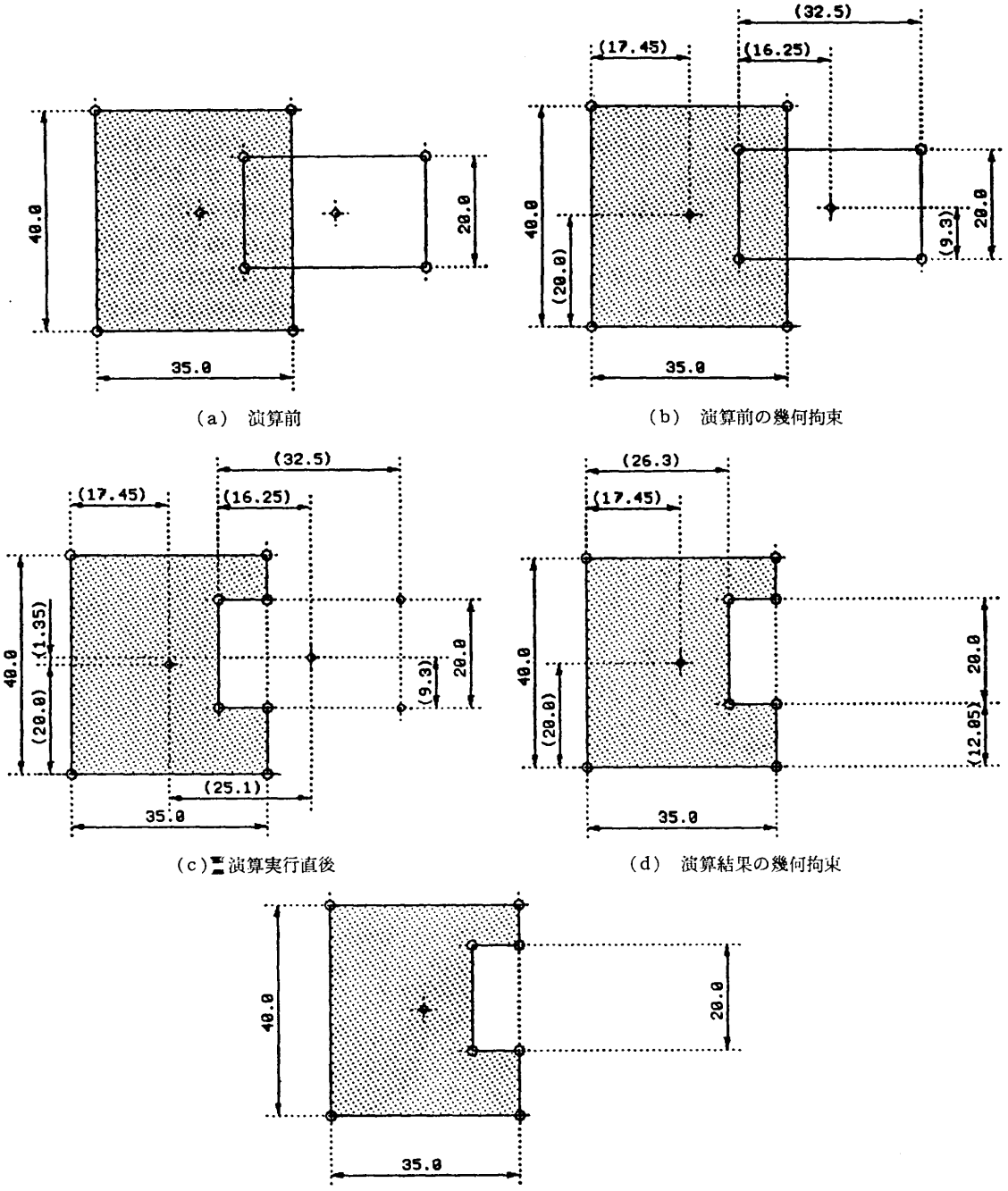


図 8 幾何拘束の書き換えによって不要になる幾何要素
Fig. 8 The rewriting of geometric constraints makes a geometric element useless.

5．後処理として，不要な幾何要素，幾何拘束の削除を行う．幾何拘束の書き換えによって削除が可能になる場合もある．

これらの中で従来の形状モデラにはなかった処理は，デフォルトの幾何拘束の生成と，不要な幾何要素，幾何拘束の削除である．デフォルトの幾何拘束の生成は，幾何要素が生成された状況に応じて行う．また，その生成のパタンで正しく機能するような幾何推論の規則を用意する．例えば，集合演算で直線と円の交点が生成された場合，デフォルトの幾何拘束として



(a) 演算前　(b) 演算前の幾何拘束

(c) 演算実行直後　(d) 演算結果の幾何拘束

(e) 演算結果の形状

図 9 差の集合演算の実行例

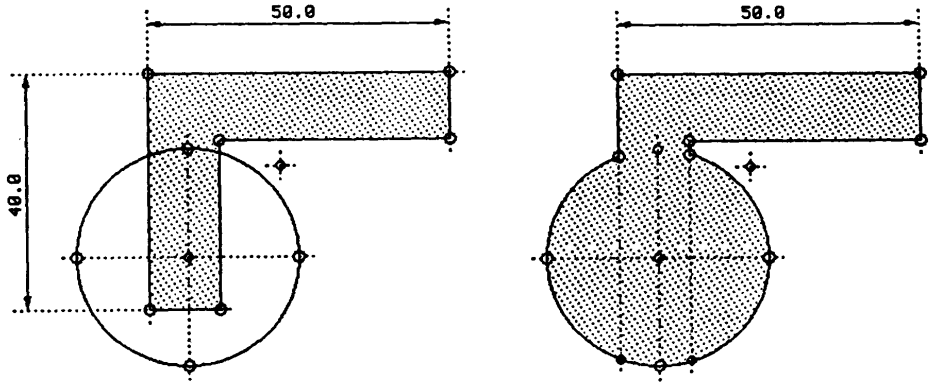
Fig. 9 An example of the difference set operation.

は，生成された点がその直線に乗っているという拘束，同じく円に乗っているという拘束，直線と円の二交点の片方を特定するための拘束の三つの幾何拘束を生成し，直線と円とその交点が存在する時に，直線と円の幾何属性から交点の幾何属性を求める推論規則を用意する．

5.の後処理を行う前の段階では，モデルとしての整合性はとれているが，不要になった幾何要素，幾何拘束が残っているので，これらを削除し，モデルを簡潔にする必要がある．不要になった補助的な幾何要素の検出の仕方は，幾何拘束の書き換えの最後の処理と同じで，その幾何要素に関する幾何拘束が，パラメタを持たない幾何拘束も含めてすべて削除可能となっている幾何要素である．

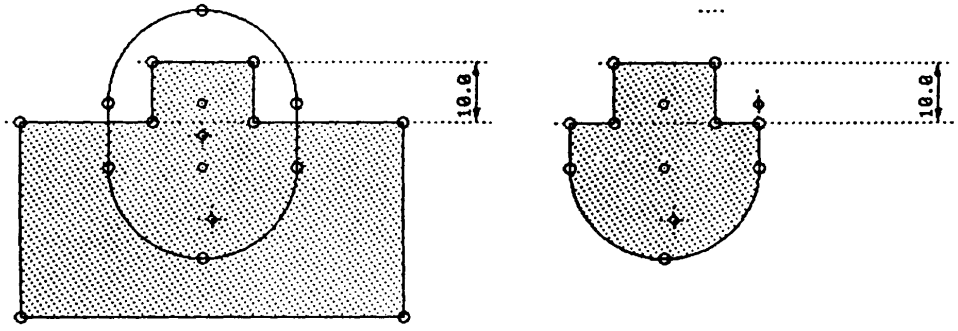
また，補助的な幾何要素の中には，その幾何要素に関するデフォルトの幾何拘束を書き換えることによって不要になるものがある．例えば図7に示す二つの矩形の差の演算において，演算結果のモデルが図8(a)のようになったとする．この直線L1は，それを参照していた稜線がなくなったため，補助的な幾何要素に変化したが，この状態では稜線から参照されている直線L2の幾何属性を求めるために，D1とD2の幾何拘束が必要なため，L1を削除することができない．しかし，同図の(b)のように，D2の代わりにL2とL3の間の幾何拘束D3を加えてD2を削除すれば，L2の幾何属性を求めるのにD1を使う必要がなくなるので，L1を削除できるようになる．(D2の代わりにD1を書き換えてもよい．)

モデルを簡潔に保つためには，このように幾何拘束の書き換えによって不要になる場合もなるべく検出した方が好ましいが，この処理では，明らかに無駄な幾何要素が削除できれば良いのであり，削除できる幾何要素をすべて検出する必要はない．なぜならば，この形状モデルでは，無駄な幾何要素が含まれていても，モデルの整合性は保たれるからである．



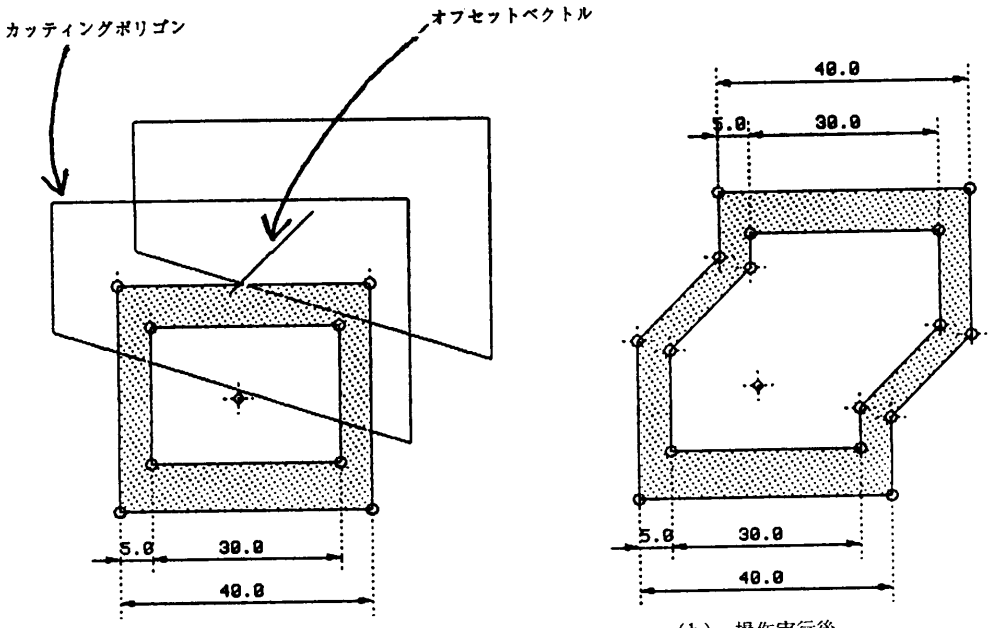
(a) 演算前　　(b) 演算結果

図10 和の集合演算の実行例
Fig. 10 An example of the union set operation.



(a) 演算前　　(b) 演算結果

図11 積の集合演算の実行例
Fig. 11 An example of the intersection set operation.



(a) 操作実行前　　(b) 操作実行後

図 12 スイープ操作の実行例
Fig. 12 An example of the sweep operation.

6. 試作システム

本研究の手法の有効性を確認するために，幾何拘束を扱う二次元形状モデリングシステムを試作した．試作システムは，Symbolics 3640 上の Common Lisp で書かれている．試作システムでは，「局所的には，形状要素の存在と位相構造，幾何拘束の場所，パラメタ値の順に決まり，大局的には，それらが前後する」という形状の詳細設計の過程に沿って形状を構築できることが確認できた．

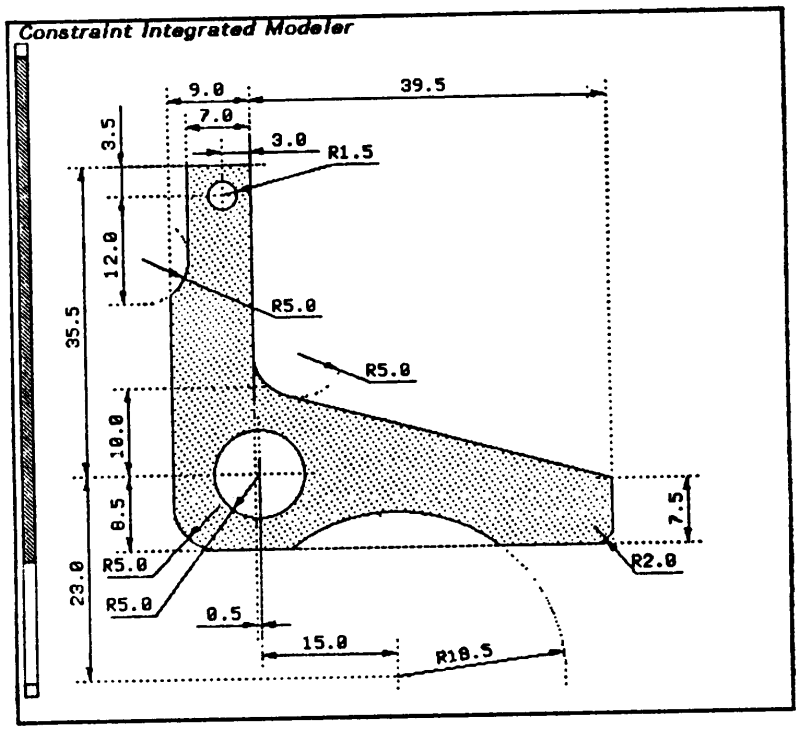


図 13 形状の構成の例—ラッチ
Fig. 13 An example of the construction of shapes—a latch.

差の集合演算の様子を図 9 に示す．図 9 (a)は，集合演算の前の形状を表している．形状は二つとも矩形であり，数箇所に使用者の指定した幾何拘束が入っている．矩形の中央付近にあるのは，その形状の原点と基準線である．これを，デフォルトの幾何拘束も含めて表示したものが，図 9 (b) である．寸法値が括弧で囲まれているものが，デフォルトの幾何拘束であり，それ以外が使用者の指定した幾何拘束である．ただし，パラメタを持たない幾何拘束は表示していない．図 9 (c)は，差演算を実行した直後の形状である．後処理は何も行ってい

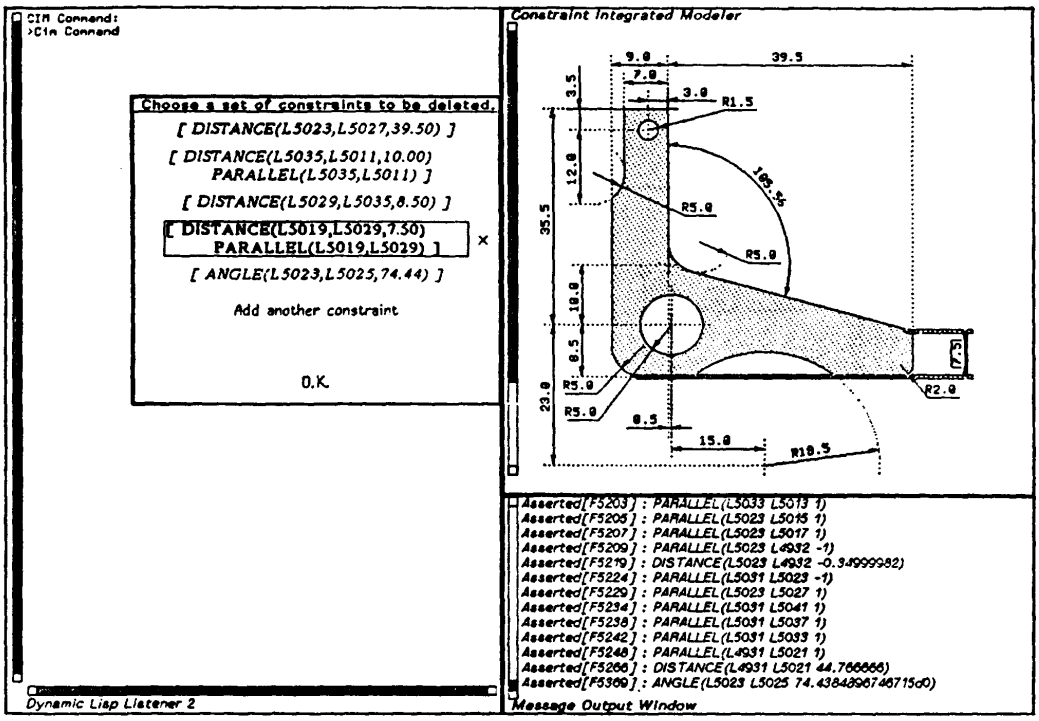


図 14 削除する幾何拘束の選択
Fig. 14 Selection of geometric constraints to be deleted.

ない．引く側の矩形の右側の辺が参照していた直線などがまだ残っている．後処理を行って不要な幾何要素や幾何拘束を削除すると，幾何拘束は，図9（d）のようになり，使用者に対する表示は最終的に図9（e）のようになる．

和，積の集合演算も差の演算と同様に行われる．**図 10** に和演算の実行例を，**図 11** に積演算の実行例を示す．**図 12**（a）は，スイープ操作の実行前の形状と，その入力パラメタを示している．形状には，部分的に幾何拘束が与えられている．入力パラメタは，平行移動させる部分を表すカッティングポリゴンと移動量を表すオフセットベクトルである．カッティングポリゴンの右上の台形は，カッティングポリゴンをオフセットベクトルの分だけ移動させたものである．図12(b)に，このスイープ操作の実行結果の形状を示す．初めに指定されていた幾何拘束が複製されているのがわかる．局所変形操作としては，スイープ操作のほかに，丸め操作，面取り操作なども実現されている．

試作システムを使って構成したラッチの形状と幾何拘束を**図 13** に示す．**図 14** は，図 13 の形状に angle の幾何拘束を追加し，過剰を避けるために削除する幾何拘束の選択を行っているところである．

7. 終 り に

本研究では，幾何拘束を扱う形状モデリングシステムに，デフォルトの幾何拘束と，幾何拘束の書き換え処理を導入し，形状要素と位相構造の入力を行う形状構成操作，幾何拘束の場所の指定，パラメタ値の入力の三種類の操作によって，形状モデルを計算機内に構成することを提案した．また，試作システムを通じて，次の利点が得られることを確認した．

1．形状を局所的に詳細化することが可能であり，未確定な幾何拘束の入力を遅らせたり，興味のある幾何拘束だけを意識して，形状を設計することができる．

2．グラフィックス表示を使った，対話的なユーザインタフェースが可能である．

3．集合演算や局所変形操作が可能であり，大きな形状構成の自由度が得られる．

今後の課題は，三次元形状への拡張や，組立品のモデリングシステムや形状生成システムとの融合である．

謝辞 本研究の一部は，精密工学会産学協同研究協議会“高度生産自動化のためのプロダクトモデリングシステム開発研究協力分科会”によるものである．

参考文献

1) 安藤英俊，鈴木宏正，木村文彦：機械設計のための幾何拘束処理システム，情報処理学会研究会報告，88-CAD-31-4 (1988).
2) 鈴木宏正，木村文彦，佐田登志夫：プロダクトモデルに基づく幾何学的拘束関係の記述と寸法処理への応用，精密工学会誌，Vol. 52, No. 6, pp. 105-110 (1986).
3) Aldefeld, B.: Variation of Geometries Based on a Geometric-Reasoning Method, *Comput. Aided Des.*, Vol. 20, No. 3, pp. 117-126 (1988).
4) Arbab, F. and Wing, J. M.: Geometric Reasoning: A New Paradigm for Processing Geometric Information, *Preprints of IFIP WG 5.2 Working Conference on Design Theory for CAD*, pp. 107-121 (1985).
5) Gossard, D. C., Zuffante, R. P. and Sakurai, H.: Representing Dimensions, Tolerances, and Features in MCAE Systems, *IEEE Comput. Gr. Appl.*, Vol. 8, No. 2, pp. 51-59 (1988).
6) Hillyard, R. C. and Braid, I. C.: Analysis of Dimensions and Tolerances in Computer-Aided Mechanical Design, *Comput. Aided Des.*, Vol. 10, No. 3, pp. 161-166 (1978).
7) Hillyard, R. C. and Braid, I. C.: Characterizing Non-Ideal Shapes in Terms of Dimension and Tolerances, *Proc. of SIGGRAPH '78*, pp. 234-238 (1978).
8) Light, R. and Gossard, D.: Modification of Geometric Models through Variational Geometry, *Comput. Aided Des.*, Vol. 14, No. 4, pp. 209-214 (1982).
9) 寺沢幹雄，原田毅士，木村文彦，佐田登志夫：機械部品形状モデリングのための対話処理用モデラの開発，第4回設計自動化講演会論文集，pp. 75-77，精密工学会，日本機械学会 (1986).
10) 山口 泰，木村文彦：CADのための拘束条件モデリング環境，情報処理学会論文誌，Vol. 30, No. 11, pp. 1512-1521 (1989).
11) de Kleer, J.: An Assumption-Based TMS, *Artif. Intell.*, Vol. 28, pp. 127-162 (1986).
12) Forgy, C. L.: Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem, *Artif. Intell.*, Vol. 19, pp. 17-37 (1982).

(平成2年2月8日受付)
(平成2年9月11日採録)

日高 康雄（学生会員）

昭和38年生．平成元年東京大学工学部精密機械工学科卒業．現在，同大学大学院情報工学専攻修士課程在学中．昭和59年(株)創夢設立．現在，同社非常勤取締役．機械系CAD，形状モデリング，計算機アーキテクチャ，並列処理に関する研究に従事．電子情報通信学会，精密工学会各会員．

安藤 英俊（正会員）

昭和38年生．昭和61年東京大学工学部精密機械工学科卒業．昭和63年同大学大学院工学系研究科情報工学専攻修士課程修了．現在，同大学大学院工学系研究科情報工学専攻博士課程に在学中．平成2年度より日本学術振興会特別研究員．平成元年度情報処理学会研究賞受賞．CADにおける制約処理の応用に興味を持つ．精密工学会，ACM 各会員．

鈴木 宏正（正会員）

1957年5月7日生．1981年東京大学工学部精密機械工学科卒業．1987年東京大学大学院工学系研究科博士課程修了．工学博士．東京大学教養学部情報・図形科学教室，助教授．専門は，形状モデリング，機械系CAD/CAM，インテリジェントCAD．精密工学会，機械学会，人工知能学会，ソフトウェア科学会，日本図学会，ACM，IEEE CS，AAAI 各会員．

木村 文彦（正会員）

昭和20年生．昭和49年東京大学大学院博士課程修了．同年電子技術総合研究所パターン情報部入所．昭和54年より東京大学工学部精密機械工学科助教授．昭和62年より同教授．マン・マシン・システム，コンピュータ・グラフィックス，形状モデリング，CAD/CAM などの研究に従事．工学博士．IFIP-WG 5.2-5.2-5.3 委員．精密工学会，日本機械学会などの各会員．