

仮想触空間における物体形状の実時間提示法

星野 洋* 平田 亮吉* 前田 太郎* 舘 暲*

A Method for Representing the Shape of a Virtual Object in Real Time

Hiroshi Hoshino*, Ryokichi Hirata*, Taro Maeda* and Susumu Tachi*

We have proposed a system which enables an operator to feel himself/herself touching an arbitrary surface with edges in virtual environment. This system represents the shape of an object in virtual environment by locally constructing its shape around his/her finger tip with Shape Approximation Device(SAD), of which each surface has convex and concave edges.

In this paper we propose a shape model of an object in virtual environment for this local construction of its shape and an algorithm of shape representation based on this model. We design the shape model so that it is possible to construct the represented shape including curved surfaces and curved edges in real time. In this model an object in virtual environment is divided into quadrangles called patches, and the shape on each patch is described using Bézier surface, which is suitable to describe a complicated shape. We realized a hardware system which could represent a shape which included curved surfaces and curved edges in real time.

Key Words: Shape Representation, Real Time, Virtual Object, Curved Edge, Bézier Surface

1. はじめに

人工現実感において、操作者が仮想環境内の仮想の物体を触ったときに触ったと感じられるよう、感覚提示することは、仮想環境の臨場感を高めるうえで重要である。

特に、仮想環境内で仮想物体をなぞる、仮想物体を操作して位置決めする、というように、仮想環境との接触が伴う作業においては、操作者に視覚のみならず、触覚、特に形状を提示する必要がある。

本システムは、仮想物体をなぞるような作業において、面の傾きや稜の向き・角度といった形状を提示することで、より臨場感を高めることを目的とするシステムである。

形状提示を目的とするシステムのなかでも、従来の研究において開発されてきたシステムは、操作者に何らかの形で力を発生する装置（アクチュエータ）を取り付け、操作者の運動を計測しつつ、仮想環境内の物体と接触に応じて発生する力を計算し、操作者に取り付けたアクチュエータを用いて力を発生させることで操作者に力を提示し、結果として形状を提示しようという、いわば力提示型のシステムであった[1][2]。そのため

- (1) 操作者は、装着したアクチュエータのダイナミクスの影響を常に受けるので、仮想環境内で物体と接触していない自

由運動の状況でも運動が拘束されてしまう。

- (2) 仮想環境内で物体と接触した瞬間に、急激にアクチュエータの制御を切替える必要があるため、高い制御性能が要求され、壁のように押しても動かない物体の形状の提示が困難である。

- (3) 稜や頂点などの微分不可能な形状は提示できない。

という欠点があった。これに対し著者らは、操作者にアクチュエータを取り付けて、これに力を発生させて形状を提示するのではなく、「操作者が環境と接触しているときに感じることでできる形状は、接触点の近傍のみであるから、接触点のまわりの形状を操作者の運動に応じて実物体を用いて構成し、操作者の運動に応じて制御することで、形状提示を行う」という方法を提案し、実機を試作してその妥当性を確認した[3]~[5]。

この方法の場合、操作者に、アクチュエータを取り付ける必要がないので、従来法の操作者の運動を拘束してしまうという欠点が解消される。また、この方法では、実物体を用いて形状を構成しているから、操作者が仮想環境と接触した瞬間に急激に制御を切替える必要がなく、したがって従来法の第二の問題も容易に解消される。

McNeelyらも、同様の原理に基づく形状提示システムを提案したが[6][7]、一般的な形状をどのようにして提示するかについては言及しておらず、また実際に開発したシステムにおいて提示している形状も、特定のものに限定されてしまっている[7]。

また、広田らは、やはり同様の原理に基づいて形状提示シス

原稿受付 1996年5月27日

*東京大学大学院工学系研究科

*Faculty of Engineering, the University of Tokyo

テムを開発し、曲面の形状の提示は成功しているが[8]~[11]、稜を有する形状の提示に成功していない。

これに対して本システムは、曲面と稜が組み合わさった形状を提示が可能であり[3][5]、さらに仮想環境内の物体に運動を付加することも可能である[4][5]。しかし、前論文[5]では、提示すべき形状を円筒や球といった特定の形状に限定しており、一般的な形状をどのように提示するかに関する考察はなされていなかった。本論文では、曲面と曲稜が任意に組み合わさった形状を表現するのに適したモデルとこのモデルに基づく提示アルゴリズムを提案する。

2. システムの原理と概要

2.1 本システムの原理

本システムの原理は、以下の通りである[5]。まず操作者が仮想環境内の物体と指先一点で接触しているとしたうえで、操作者に触らせるための、面と稜を有する実物体を用意しておく。そして操作者の指先の位置に応じて、この実物体の位置と姿勢を次のように制御する。

- (a) 操作者が仮想物体の面を触っているとき：実物体の面が、接触点において、提示すべき面の接平面となるように位置・姿勢を制御する (Fig. 1 (a))。
- (b) 操作者が仮想物体の稜を触っているとき：実物体の稜が、接触点において、提示すべき稜の接線となるように位置・姿勢を制御する (Fig. 1 (b))。

これによって、曲面、曲稜がそれぞれ提示可能である。

次に、面に対して任意の稜を提示する場合であるが、これは、Fig. 2 (a), (b) のように、実物体を提示している面内で並進・回転させることにより、実物体の提示面に対して、実物体の稜を任意の位置、姿勢に配置できるので、提示可能である。また、

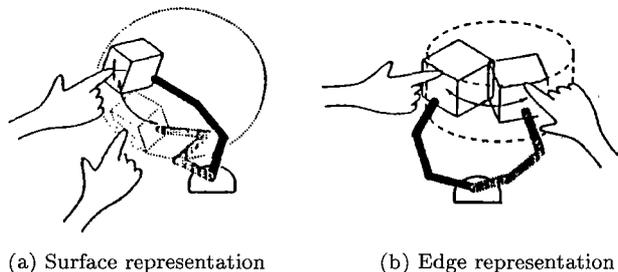


Fig. 1 Representation of surface and edge

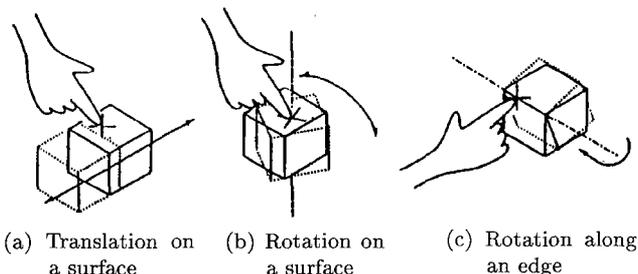


Fig. 2 Motion for representing surface and edge

稜に対して任意の面を提示する場合も Fig. 2 (c) のように、実物体を提示している稜を軸として回転させることにより、実物体の提示稜に対して、実物体の面を任意の角度に設定できるので、やはり可能である。

そしてさらに、これらの運動を Fig. 3 のように組み合わせることにより、任意の面と稜が組み合わさった形状提示が可能となる。

つまり、形状提示に用いる実物体として面と稜を有する物体を用いることにより、面と稜が組み合わさった仮想物体の形状が提示が可能になる。

なお、この際に提示すべき形状に対して、実物体のどの部分を用いるかが問題になってくるが、これについては 4.1 節で述べる。

2.2 本システムの概要

本システムの概要を Fig. 4 に示す。すなわち、操作者の上肢の運動を Passive Master Arm で測定し、これに応じて操作者に触らせる実物体、Shape Approximation Device (SAD) の位置・姿勢を 6 自由度を有するマニピュレータ Active Environment Display (AED) でもって制御する[3][5]。Passive Master Arm は極めて軽量であり、これによって 1 章で述べた従来の方式の第一の欠点は解消されている。

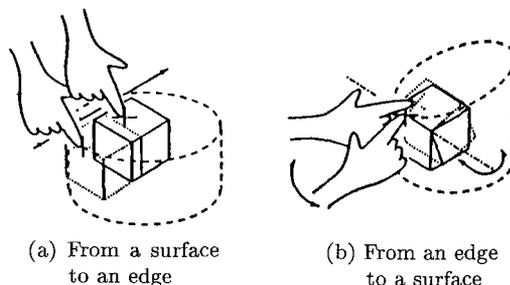


Fig. 3 Combination of motions for representing surfaces and edges

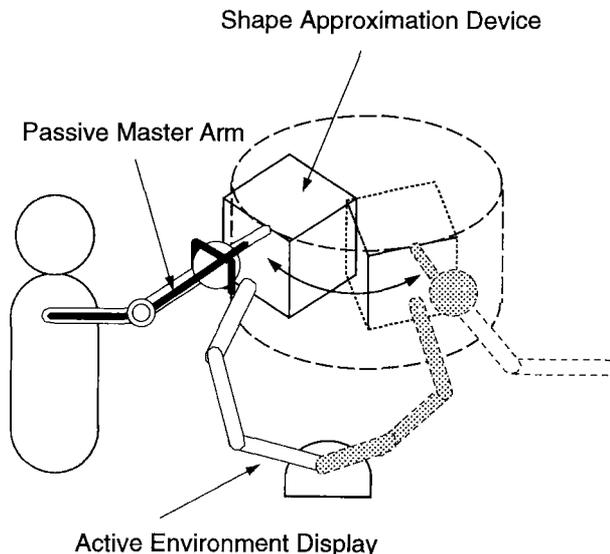


Fig. 4 Conceptual diagram of the proposed system

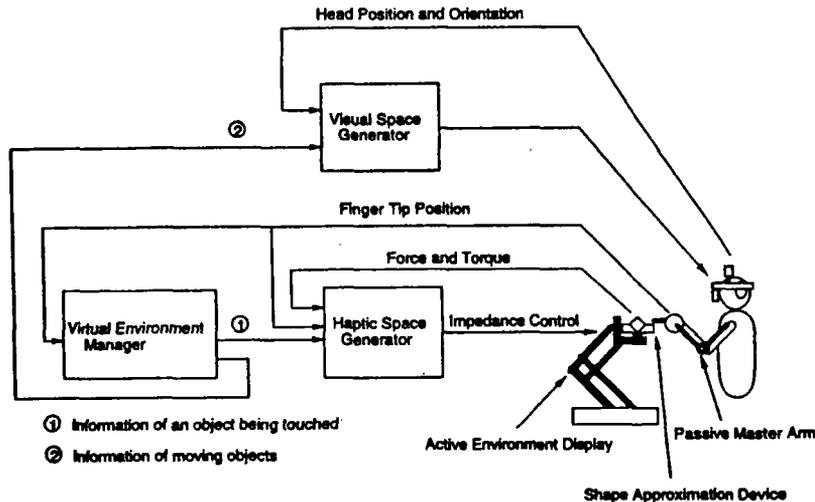


Fig. 5 Block diagram of virtual visual and haptic space representation system

また Passive Master Arm の先端は、ボールペン状になっている。これは、実物体との点接触を実現し、かつ、先端において生じる実物体との間の滑りの運動が操作者に伝わらないようにするためである。特に後者は、2.1 節で述べた実物体の面内における並進・回転の運動が、操作者に伝わらないための機構である。

SAD は Fig. 16 に示すものを用いる。この SAD は凹凸の稜を有しているので、凹と凸の稜を含む形状の提示が可能であり [5]、従来の方式の第三の欠点が解消される。

なお、この SAD の形状の根拠については、5 章で詳しく述べる。

また、本システムのダイアグラムは Fig. 5 に示す通りである [3] [5] [12] [13]。Virtual Environment Manager は、仮想環境の形状モデルを管理する計算機で、Passive Master Arm より操作者の指先の位置を取り込み、これと仮想環境のモデルから Haptic Space Generator に対し、形状提示に必要な情報を送信する。

Haptic Space Generator は、AED の制御を行う計算機で、Passive Master Arm から取り込んだ操作者の指先の位置と、Virtual Environment Manager より受信した形状情報をもとに、SAD が提示すべき形状を構成するように制御を行う。

Visual Space Generator は、操作者に画像提示を行う計算機で、磁気センサより操作者の頭部の位置・姿勢を取り込み、これに応じて HMD に画像を出力する。

このように計算機を 3 台用いているのは、一般に、仮想環境を更新するタスクや、仮想環境を描画するタスクのループ速度に比べて、マニピュレータを制御するタスクのループ速度は、かなりの高速性が要求されるので、各タスクを別々の計算機で実行し、必要な場合のみ通信をすることで各タスクが最高速に実行されるようにするためである。

なお、本システムでは、計算機間の通信には共有メモリを用いることで、高速かつ非同期に通信を行い、仮想環境のモデルの計算を行いながら AED の安定な制御を行えるようにして

いる。

3. 仮想物体の形状モデル

本章では、仮想環境内の仮想物体の形状を表現するための、形状モデルについて述べる。

3.1 曲面のパラメトリックな表現

まず、面と稜とが任意に組み合わさった形状を表現するのに適した記述方法について考えてみる。

一般に曲面の記述方法としては、以下の方法が考えられる。

- (1) 世界座標系において、陽関数で表す表現。
- (2) 世界座標系において、陰関数で表す表現。
- (3) 局所座標系において、パラメトリックに表す表現。

(1) は、例えば、形状を $z = f(x, y)$ というように表現する方法である。この方法の場合、計算負荷は軽いという長所はあるが、表現が世界座標系に依存してしまうため、球などの閉じた形状や一般的な曲稜の記述が記述が困難である、という問題が生じる。

(2) は、形状を $f(x, y, z) = 0$ というように表現する方法である。この場合 (1) の閉曲面への対応が困難、という欠点は解消されるが、曲稜の記述が困難になる。

なぜなら、形状をこのように陰関数で表現した場合、曲稜は、二曲面 $f(x, y, z) = 0$ と $g(x, y, z) = 0$ との交線として記述されるが、一般に、この交線を求める問題は計算負荷が大きく、実時間で求めるのが困難であるからである。

(3) は、形状を曲面上に適当に設定した二次元の局所座標系 (u, v) に対して、 $p(u, v)$ というように記述する方法である。

この方法の場合、(1) ほどではないが、計算負荷は軽く、また、パラメータ (u, v) のうちの一方を固定すれば、この式は、そのまま空間曲線を表す式になるので、曲稜の記述も容易である。

以上の理由より、一般的な曲面の形状を記述する場合、(3) のパラメトリックな表現を用いるのが適切である。

3.2 ベジエ曲面による曲面の表現

本システムは、3.1 節で述べた理由から、曲面をパラメトリックに表現する。

つまり、仮想環境内の物体を CG でいうところのパッチに分割し、各パッチにおける形状をパラメトリック曲面で表すことにする。

パラメトリックな曲面としては、以下の曲面が一般的であるが、

- (1) ベジエ曲面
- (2) B スプライン曲面
- (3) NURBS 曲面

このうち、(3) の NURBS 曲面は、(1) のベジエ曲面や (2) の B スプライン曲面に比べて、より複雑な形状を表現できるが、計算負荷が大きいという問題が生じる。

また、(1) のベジエ曲面と (2) の B スプライン曲面は、表現できる形状の複雑さは、同じ程度であるが、4.3 節のパッチの端点を求める手続きにおいて、(2) の B スプライン曲面が、パッチの端点を通過しないのに対し、(1) のベジエ曲面は、パッチの端点を通過するので手続きがより容易になるという、利点がある。

以上の点を考慮して、本形状モデルでは (1) のベジエ曲面を用いる。

また、本システムではベジエ曲面の階数は、法線の連続性より 3 以上でなくてはならないが、今回は計算負荷との兼ね合いから、式 (1) で表される、4 階のベジエ曲面を用いる。

$$p(u, v) = B(u)^T Q B(v) \tag{1}$$

$$B(t) = \begin{bmatrix} (1-t)^3 \\ 3(1-t)^2 t \\ 3(1-t)t^2 \\ t^3 \end{bmatrix} \tag{2}$$

$$Q = \begin{bmatrix} q_{00} & q_{01} & q_{02} & q_{03} \\ q_{10} & q_{11} & q_{12} & q_{13} \\ q_{20} & q_{21} & q_{22} & q_{23} \\ q_{30} & q_{31} & q_{32} & q_{33} \end{bmatrix} \tag{3}$$

ここで、 $\{q_{ij}\}$ は制御点と呼ばれ、ベジエ曲面の形状を規定するパラメータである。

このように形状モデルを表すことで、次の利点が生じる。

- ベジエ曲面は、パラメトリックな曲面であるから、パッチの境界においてベジエ曲線と一致する。したがって曲稜が容易に計算できる。
- 最終的に、操作者に対して提示する形状は、接触点の近傍部分の形状のみであるから、2.2 節で述べた Virtual Environment Manager と Haptic Space Generator との間の通信の際には、接触点近傍のパッチにおける形状情報のみを送信すればよく、これによって通信量を大幅に削減できる。

本モデルでは、上で述べたように各パッチに制御点 $\{q_{ij}\}$ の情報を持たせて、形状を表現することにしたが、さらに、

- 隣接するパッチを示すポインタ

- 稜の有無・凹凸を示すフラグ
 - 自分の属する面を示すポインタ
- の情報も持たせるようにした。ただし、‘面’とは、滑らかに接続している一群のパッチを指すものとする。
- これらの情報をどのように用いるかについては 4.2 節、4.3 節で述べる。

4. 形状提示アルゴリズム

形状提示は、3 章で導入したモデルに基づいて、以下の各処理を、Fig. 6 の順で行う。

- (1) 局所形状情報の送信
- (2) SAD の提示部の指定
- (3) 仮想物体の形状提示箇所の位置・姿勢の決定
- (4) SAD の位置・姿勢の制御

以下、各処理について詳しく述べる。

4.1 SAD の提示部の指定

まず初めに、形状提示を行う際に SAD のどの部分を用いるかを以下のようにして決定する。

- (1) 提示すべき形状が面で稜を含まない、あるいは稜が指から十分遠い間は面の提示を行えばよいから、SAD の面のうちの任意の一面を指定する。
- (2) もし提示すべき形状に稜が含まれていて指から十分近くなった場合には、提示すべき稜が凹か凸かに応じて、(1) で指定した SAD の提示面に接続する、凹または凸の稜を指定する。
- (3) そしてさらに、もし操作者が、(2) で提示した稜から、(1) で提示した面と異なる側の面を触る場合には、SAD の提示面として、(1) で指定していた面と異なる側の面を指定する (Fig. 7)。

なお、(3) において、提示すべき面が異なるか否かを調べるのは、3.2 節で述べた各パッチの有する、属する面を示すポインタを調べることで行っている。

4.2 局所形状情報の送信

Virtual Environment Manager は、操作者の指先の位置に応じて以下のようにして、Haptic Space Generator に提示すべき点付近の形状情報 (局所形状情報) を送信する。

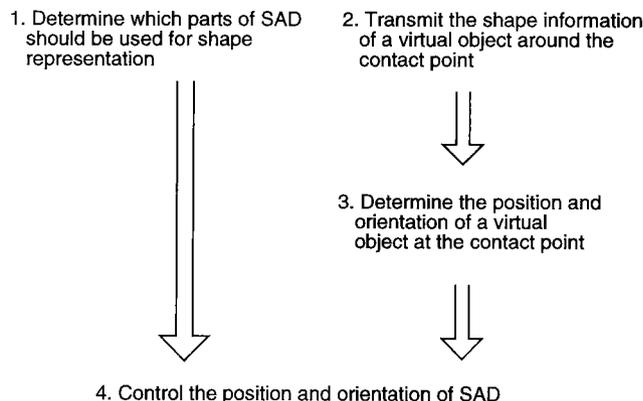


Fig. 6 Flowchart of the proposed algorithm

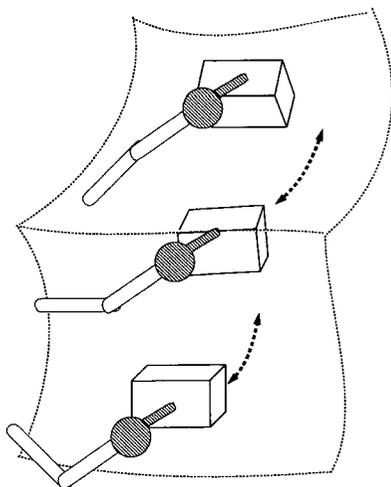


Fig. 7 Changing parts of SAD for representation of edged surface

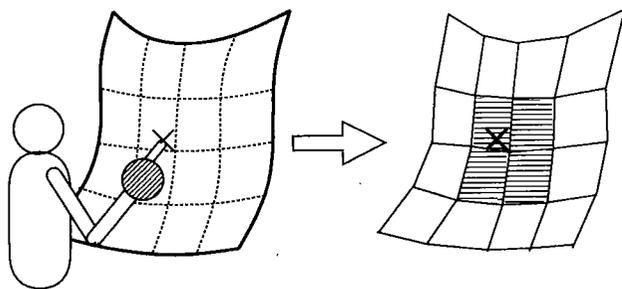


Fig. 8 Adjacent patches

すなわち Virtual Environment Manager は、まず Passive Master Arm より、操作者の指先の位置を取り込み、指先から、最も近いパッチを求める。次に、3.2 節で述べた隣接するパッチを示すポイントの情報を用いて、Fig. 8 に示すように、これに隣接するパッチを求める。そして、これらのパッチの有する制御点の情報と稜の有無・凹凸を示すフラグ、属する面を示すポイントの情報を Haptic Space Generator に送信する。

ここで、形状情報として、最も近いパッチにおける情報のみでなく、隣接するパッチにおける情報も送信するのは、もし送信する時点で、指がパッチの境界のすぐそばにあった場合、Virtual Environment Manager が次に送信するまでの間に、指がこのパッチから外れてしまい、Haptic Space Generator が形状を構成できなくなってしまうからである。

4.3 仮想物体の形状提示箇所的位置・姿勢の決定

Haptic Space Generator は Virtual Environment Manager から受信したパッチの制御点の情報と、稜のフラグに基づいて以下の手順で仮想物体の、提示すべき部分における位置と姿勢を決定する。

- (1) まず Virtual Environment Manager から送信されてきた k 枚のパッチに対して、指から最も近いパッチを求める。
- (2) (1) で求めたパッチに対して稜の有無を調べる。
- (3) (1) で求めたパッチに対して、(2) の結果をもとにして、指先の位置から、式 (1) におけるパラメータ (u, v) を求

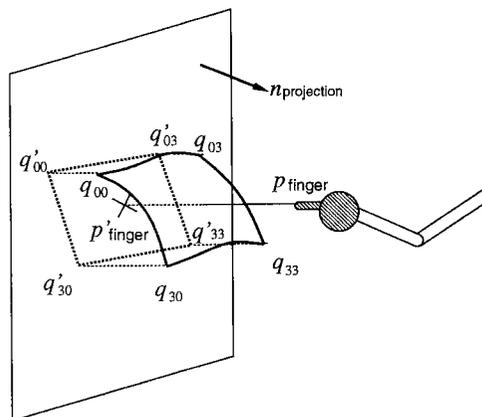


Fig. 9 Projection of the patch

める。

- (4) (3) で求めた (u, v) から、形状提示箇所における位置・姿勢を決定する。

ここでまず、(1) の手続きであるが、これは単なる距離計算であるから容易であり、また (2) の手続きもパッチの持つ稜に関するフラグを用いれば、やはり容易に行える。以下、(3) と (4) の手続きについて詳しく述べるが簡単のために、パッチは平行四辺形であることを仮定しておく。

まず (3) の手続きであるが、はじめに稜を含まない場合について説明する。

いま、その法線ベクトル n が、

$$n = \sum_{i=1}^k n_i \quad (4)$$

を満たす、すなわち Virtual Environment Manager から受信した k 枚のパッチの法線ベクトルの平均となるような平面を考え、指の位置 p_{finger} と (1) で求めたパッチの端点をこの面に投影する (Fig. 9)。このとき、 p_{finger} が p'_{finger} に、パッチの 4 隅の点 $\{r_{00}, r_{01}, r_{10}, r_{11}\}$ が、 $\{r'_{00}, r'_{01}, r'_{10}, r'_{11}\}$ に投影されたとすると、これらの間には明らかに

$$p'_{\text{finger}} = \alpha(r'_{10} - r'_{00}) + \beta(r'_{01} - r'_{00}) + r'_{00} \quad (5)$$

という線形関係が成り立つ。ところで、3.2 節で述べたように、ペジエ曲面はパッチの端点を通るから、 $\{r_{ij}\}_{i,j=0,1}$ は $\{q_{ij}\}_{i,j=0,3}$ に相当し、式 (5) は

$$p'_{\text{finger}} = \alpha(q'_{30} - q'_{00}) + \beta(q'_{03} - q'_{00}) + q'_{00} \quad (6)$$

となる。そこで、この場合にはこの式 (6) を満たす (α, β) を求め、そのまま (u, v) として用いる。

次に稜を含む場合であるが、この場合には受信したパッチのうち、(1) で求めた最も近いパッチの属する面と、同じ面に属するパッチを 3.2 節で導入した各パッチの有する面ポイントを用いて探索し、これら k' 枚のパッチに対して、稜を含まない場合と同じ要領で平均の法線ベクトルを計算して投影面を設定し、いったん (α, β) を求める。

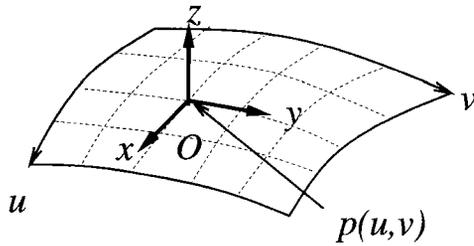


Fig. 10 Coordinate for rerepresenting a virtual surface

そしてさらに、もし指が稜に十分近い場合には、前に述べた、ベジエ曲面はその境界でベジエ曲線になる、という性質を利用する。すなわち、稜の有無を示すフラグから、上で求めた (α, β) のうち、一方を 0 または 1 に固定し、これを最終的に (u, v) の値とする。

なお、指が稜に十分近い、という状況とその判定法については付録 A 節で詳しく述べる。

次に、(4) の形状提示箇所における位置・姿勢の決定の手続きであるが、まず提示すべき形状が面である場合、上で求めた (u, v) に対して、

$$n(u, v) = \frac{\partial p}{\partial u}(u, v) \times \frac{\partial p}{\partial v}(u, v) \quad (7)$$

$$l(u, v) = \frac{\partial p}{\partial u}(u, v) \quad (8)$$

$$m(u, v) = n(u, v) \times l(u, v) \quad (9)$$

を計算した上で、

- $p(u, v)$ を原点 O
- $n(u, v)$ を z 軸
- $l(u, v)$ を x 軸
- $m(u, v)$ を y 軸

と設定する (Fig. 10)。

また提示すべき形状が稜である場合には、上で求めた (u, v) に対して、

$$n(u, v) = \frac{\partial p}{\partial u}(u, v) \times \frac{\partial p}{\partial v}(u, v) \quad (10)$$

$$l(u, v) = \pm \frac{\partial p}{\partial w}(u, v) \quad (11)$$

w は、 u 軸、 v 軸のうち稜と一致するほう

$$m(u, v) = n(u, v) \times l(u, v) \quad (12)$$

を計算したうえで、

- $p(u, v)$ を原点 O
- $n(u, v)$ を z 軸
- $l(u, v)$ を x 軸
- $m(u, v)$ を y 軸

とする。ただし、 x 軸に関しては、向きに不定性が生じるため、 y 軸が、面の内部から隣接する面に向かう向きと一致するように設定する (Fig. 11)。

4.4 SAD の位置と姿勢の制御

操作者の指先の位置に対して、4.1 節までの手順で SAD の提示に用いるべき部分が、4.3 節までの手順で、仮想物体の提

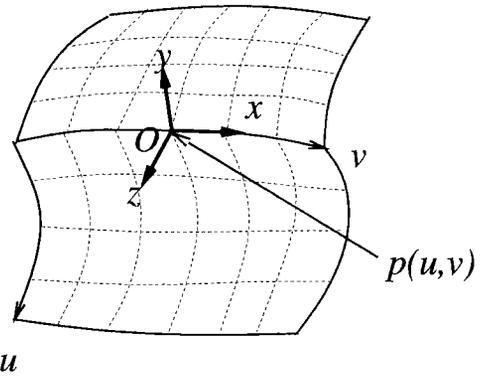


Fig. 11 Coordinate for rerepresenting a virtual edge

示箇所における位置・姿勢が求まったので、両者が一致するように制御をすればよい。

すなわち、世界座標系における仮想物体の形状提示箇所を表す同次変換行列を T_{VO}^U 、世界座標系における SAD の提示部の位置・姿勢を表す同次変換行列を $T_{SAD-Rep}^U$ としたときに、

$$T_{VO}^U = T_{SAD-Rep}^U \quad (13)$$

となるように、SAD の位置と姿勢を制御すればよい。ところで、世界座標系における SAD の座標系を表す同次変換行列を T_{SAD}^U 、SAD の座標系における SAD の提示部の位置・姿勢を表す同次変換行列を $T_{SAD-Rep}^{SAD}$ とすると、

$$T_{SAD-Rep}^U = T_{SAD}^U \cdot T_{SAD-Rep}^{SAD} \quad (14)$$

であるから、結局、

$$T_{SAD}^U = T_{VO}^U \cdot (T_{SAD-Rep}^{SAD})^{-1} \quad (15)$$

となるようにすればよい。

T_{VO}^U は、4.3 節で設定したものにほかならない。そこで、 $T_{SAD-Rep}^{SAD}$ は、 T_{VO}^U のこの設定に基づいて次のように設定する。

まず、SAD の各面の、面提示に用いる部分に対しては次のように設定する。

- SAD の面の中央を原点 O
- SAD の面の法線外向きの方向を z 軸
- 原点 O から稜に向かう方向を y 軸
- y 軸、 z 軸と直交し、かつ右手系になる方向に x 軸と設定する (Fig. 12)。

また、SAD の各面の、稜提示に用いる部分に対しては次のように設定する。

- SAD の稜の中央を原点 O
- SAD の提示面の法線外向きの方向を z 軸
- SAD の稜の接線方向を x 軸

- z 軸、 x 軸と直交し、かつ右手系になる方向に y 軸

なお、 x 軸は、向きに関して不定性が生じるが、 y 軸が、面の中央から稜に向かう向きになるように、設定する (Fig. 13)。

$T_{SAD-Rep}^{SAD}$ をこのように設定したとき、式 (15) を満たすように AED を制御すれば、SAD の提示部分は、仮想物体の形状提示箇所と一致する。

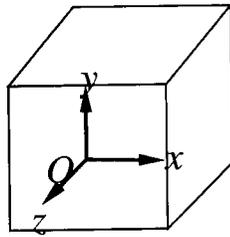


Fig. 12 Coordinate of the surface of SAD

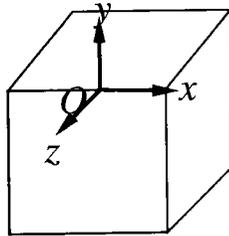


Fig. 13 Coordinate of the edge of SAD

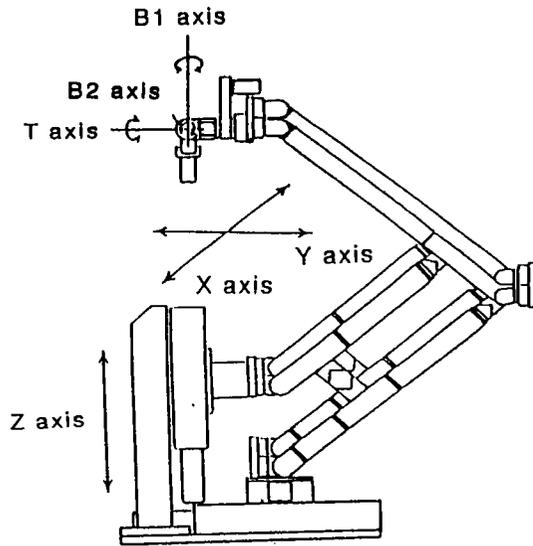


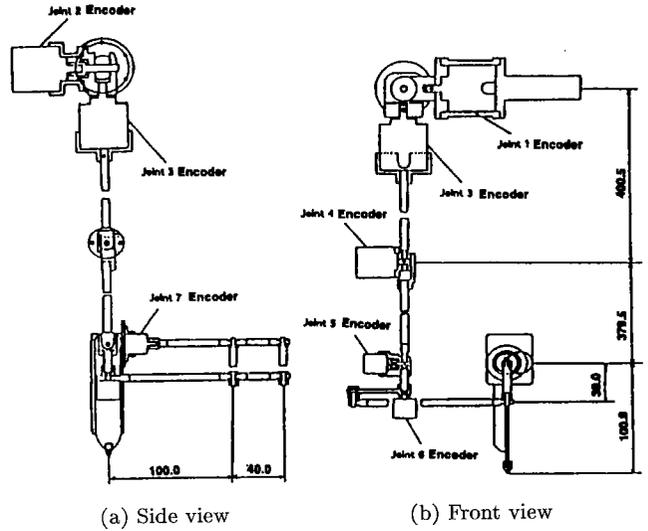
Fig. 14 Active Environment Display (AED)

5. 実 験

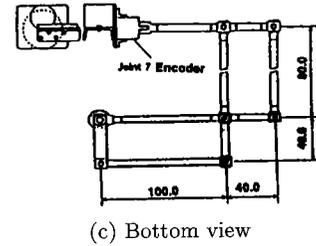
実験に用いた AED, Passive Master Arm, SAD をそれぞれ Fig. 14, Fig. 15, Fig. 16 に示す。

AED の仕様は以下の通りであり、広い面の再現と十分な接触力提示を可能としている。

- x 軸 : ± 300 [mm]
- y 軸 : ± 300 [mm]
- z 軸 : ± 300 [mm]
- T 軸回転角 : $\pm 180^\circ$
- B_1 軸回転角 : $\pm 90^\circ$
- B_2 軸回転角 : $\pm 90^\circ$
- x, y, z 軸方向推力 : 30 [N] (max)



(a) Side view (b) Front view



(c) Bottom view

Fig. 15 Passive Master Arm

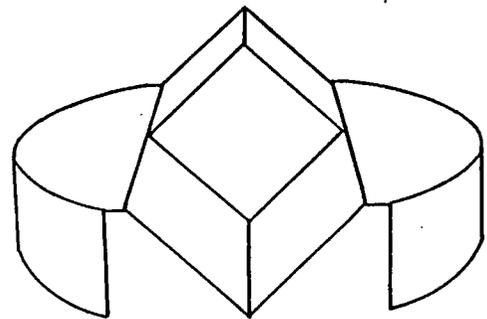


Fig. 16 Shape Approximation Device (SAD)

- T, B_1, B_2 軸トルク : 1.3 [Nm] (max)

Passive Master Arm の各関節の分解能は、肩の 3 自由度がすべて $2'10''$ 、肘が $3'36''$ 、手首の 3 自由度がすべて $10'48''$ である。

また SAD は、各面が凹と凸の稜を有する構造になっている。これは、2.1 節で述べた実物体の運動を用いて任意の面から、凹あるいは凸の稜を提示するためには、必ず実物体の提示に用いている面から、凹と凸の稜がたどれなくてはならないからであり、このためには、実物体の各面が凹と凸の稜を持つような形状でなくてはならないからである。

次に、各計算機の仕様であるが、

- Virtual Environment Manager

IBM-PC/AT (Pentium 90 [MHz]) を使用。ループ速度は $5 \sim 10$ [msec]。

● Haptic Space Generator

PC-9801 (Cyrix586 120 [MHz]) を使用. ループ速度は 5 [msec].

● Visual Space Generator

IBM-PC/AT (Pentium 90[MHz]) を使用. 画像出力は FIRE-Board (i860 搭載) を 2 枚用いて 2 画面出力. ルー

プ速度は 100 [msec] 程度である. 各計算機間は, 共有メモリを用いて 700 [Kbyte/sec] 程度の速度で通信を行った.

なお今回は, Virtual Environment Manager から Haptic Space Generator に送信するパッチの枚数は, 指先から最も近いパッチ 1 枚とこれに隣接するパッチの 3 枚の計 4 枚とした.

Fig. 17 に実験システムの全景を, Fig. 18, 19 に形状提示を行った例を示す. Fig. 18 が, HMD より操作者に視覚提示さ



Fig. 17 General view of the proposed system

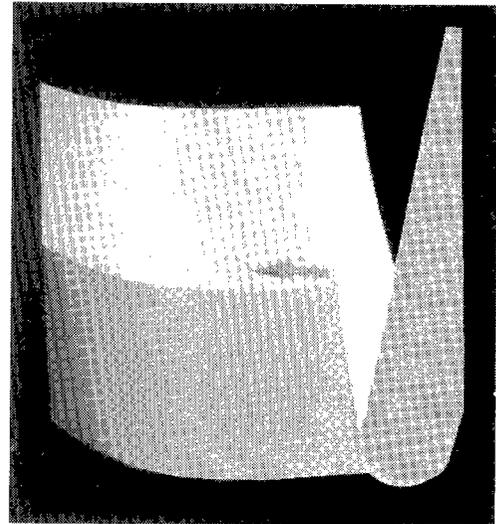


Fig. 18 Virtual environment

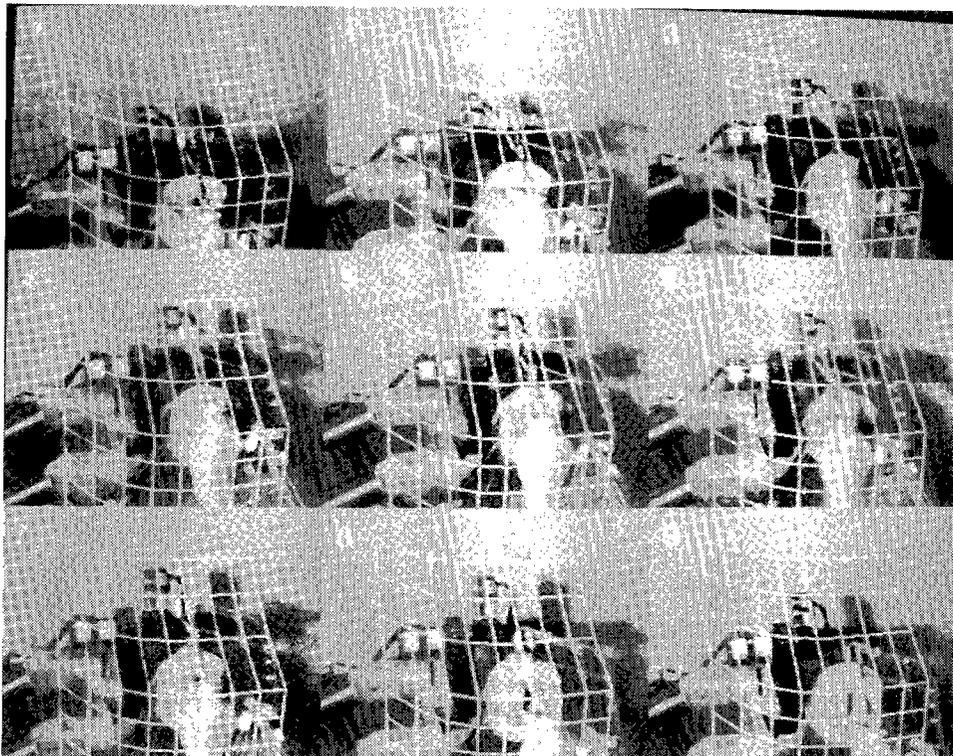
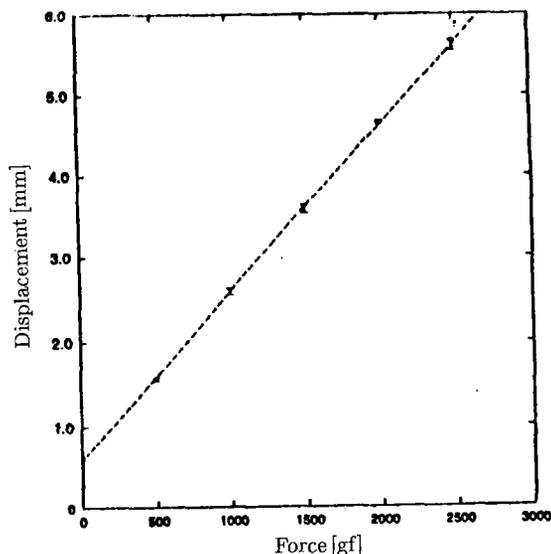


Fig. 19 Controlled movement of SAD to represent the edged surface

Table 1 Experimental result of force vs. displacement

Force [gf]	Displacement [mm]	Standard deviation [mm]
500	1.57	0.02
1,000	2.59	0.04
1,500	3.57	0.05
2,000	4.65	0.02
2,500	5.62	0.06

**Fig. 20** Measured stiffness of a virtual surface**Table 2** Result of linear regression

a [mm/gf]	b [mm]	Regression coefficient
0.00203	0.552	0.9998

れている仮想環境の画像であり、Fig. 19 は、これを SAD を用いて形状提示している一連の状況を示したものである。

すなわち、Fig. 19 中の 1 では仮想物体の稜の下側の面を触っており、2 において稜を触り、3~6 では稜の上側の面を触っている。そして、7 において再び稜を触り、8、9 でまた稜の下側の面を触っている。なお、Fig. 19 中の網目上の図形は、Fig. 18 中の形状に対する一連のタッチを示したものである。

なお、この形状は 64 枚のタッチから構成されており、各タッチの大きさは 50 [mm]×50 [mm] 程度である。

次に、形状提示を行っている際に、操作者が SAD を、一定の力で押したときの力とそれに対する変位を測定した結果を **Table 1**, **Fig. 20** に示す。この結果に対して直線回帰 ($y = ax + b$) を行った結果は **Table 2** の通りであり、この値より本システムが形状提示を行っている際の剛性を計算すると、4,830 [N/m] となる。

6. おわりに

本論文では、本システムの

- 形状を接触点の周りで局所的に構成することで表現する。
- 稜を含む形状も表現可能である。

という特徴を考慮して、仮想環境内の面と稜が組み合わさった形状を実時間で提示するためのモデリングと、これを用いた提示アルゴリズムを提案し、実機を用いて妥当性を確認した。この形状モデルとアルゴリズムは、以下の特長を有する。

- (1) 曲面の形状を、パラメトリック曲面の一種であるベジエ曲面を用いて表現することにより、曲稜も容易に表現できる。
- (2) 仮想物体をあらかじめ、パッチに分割してあるので、操作者に提示すべき形状を計算する際に、計算する範囲を接触点近傍に限定でき、高速に計算できる。
- (3) 形状モデルおよびそれに基づくアルゴリズムが、特定の座標系に依存しないようになっているので、閉じた形状の表現も容易である。

なお、形状モデルにおいてパッチが平行四辺形であると仮定しているために、本システムの表現できる形状が限定されてしまっている点については、今後改良する所存である。

参考文献

- [1] 平田幸広, 猿渡基裕, 佐藤誠: “モータ駆動による力覚制御能力をもつ空間インタフェース装置の提案”, ヒューマンインタフェース pp.65-71, 1992.
- [2] 岩田洋夫, 中川隆志, 矢野博明: “広稼働範囲フォースディスプレイの開発”, ヒューマンインタフェース pp.54-64, 1992.
- [3] Susumu Tachi, Taro Maeda, Ryokichi Hirata and Hiroshi Hoshino: “A Construction Method of Virtual Haptic Space,” International Conference on Artificial Reality and Tele-Existence (ICAT '94), pp.131-138, 1994.
- [4] Susumu Tachi, Taro Maeda, Ryokichi Hirata, and Hiroshi Hoshino: “A Machine which Generates a Virtual Haptic Space”, Virtual Reality Annual International Symposium (VRAIS '95), Video Proceeding, 1995.
- [5] 平田亮吉, 星野洋, 前田太郎, 館暉: “人工現実感システムにおける物体形状を提示する力触覚ディスプレイ”, 日本バーチャルリアリティ学会論文集, vol.1, no.1, pp.23-32, 1996.
- [6] William A. McNeely: “Robotic Graphics: A New Approach to Force Feedback for Virtual Reality,” Virtual Reality Annual International Symposium (VRAIS '93), pp.336-341, 1993.
- [7] Boeing Defense and Space Group: “Implementation of Robotic Graphics for A Virtual Control Panel,” Virtual Reality Annual International Symposium (VRAIS '95), Video Proceeding, 1995.
- [8] 広田光一, 齊藤淳, 廣瀬通孝: “局面面の提示の考え方にもとづく触覚ディスプレイ”, 日本ロボット学会第 11 回学術講演会, pp.355-356, 1993.
- [9] Koichi Hirota and Michitaka Hirose: “Development of Surface Display,” Virtual Reality Annual International Symposium (VRAIS '93), pp.256-262, 1993.
- [10] 広田光一, 廣瀬通孝: “面提示型触覚ディスプレイのための曲面表現デバイス”, 第 10 回ヒューマン・インタフェース・シンポジウム論文集, pp.193-196, 1994.
- [11] Koichi Hirota and Michitaka Hirose: “Simulation and Presentation of Curved Surface in Virtual Reality Environment through Surface Display,” Virtual Reality Annual International Symposium (VRAIS '95), pp.211-216, 1995.
- [12] 星野洋, 平田亮吉, 前田太郎, 館暉: “バーチャルハプティックベースの研究 (第 6 報)”, 計測自動制御学会第 34 回学術講演会, pp.1103-1104, 1995.
- [13] Hiroshi Hoshino, Ryokichi Hirata, Taro Maeda, and Susumu Tachi: “A Construction Method of Virtual Haptic Space (II),” International Conference on Artificial Reality and Tele-Existence (ICAT '95), pp.211-220, 1995.

付録 A. 指と稜が十分に近い状況について

仮想物体の形状を表現する際に面を提示する場合には、SADの面の中心を用いるが、稜を提示する場合には、SADの稜を用いており、両者でSADの提示に用いる部分が異なる。

したがって、もし仮想物体の稜の付近の形状を提示する場合には、SADを面提示の条件で配置してしまうと、SADの稜が、仮想物体の稜よりもハミ出てしまう場合が生じる (Fig. A)。つまり、この場合はSADを稜提示の条件で配置しなくてはならない。このような状況を稜に十分近い場合と呼ぶことにする。

これは、面提示の条件で提示しているときのSADの稜の位置が稜を提示しているときのSADの稜よりも、外側にきてしまう状況に相当している。

したがって、稜提示時の世界座標系におけるSADの提示面の中央の位置を $P_{Surface}^U$ 、稜提示時の世界座標系におけるSADの提示稜の中央の位置を P_{Edge}^U 、面提示時の世界座標系におけるSADの提示面の中央の位置を $Q_{Surface}^U$ としたときに下式の

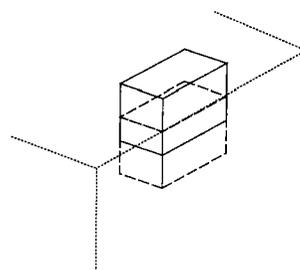


Fig. A The situation of representing a virtual edge incoorrectly

内積を計算し、

$$(P_{Edge}^U - Q_{Surface}^U, P_{Surface}^U - Q_{Surface}^U) \quad (16)$$

もし、この値が負なら面提示の条件でSADを配置し、正なら稜提示の条件で配置することで対応できる。



星野 洋 (Hiroshi Hoshino)

1971年2月14日生。1993年東京大学工学部計数工学科卒業。1995年東京大学大学院工学系研究科修士課程修了。現在、東京大学大学院工学系研究科博士課程在学中。人工現実感の研究に従事。

(日本ロボット学会学生会員)



前田太郎 (Taro Maeda)

1965年2月3日生。1987年東京大学工学部計数工学科卒業。同年通産省機械技術研究所に入所。1992年東京大学先端科学技術研究センター助手。1994年東京大学工学部助手。工学博士。

(日本ロボット学会正会員)



平田亮吉 (Ryokichi Hirata)

1967年10月3日生。1992年、九州大学工学研究科電気工学専攻卒業。同年、株式会社安川電機入社。1993年9月から1996年3月まで東京大学先端科学技術研究センター協力研究員としてテレグジスタンスの研究を行う。(日本ロボット学会正会員)



館 暲 (Susumu Tachi)

1946年1月1日生。1968年東京大学工学部計数工学科卒業。1973年東京大学大学院工学系研究科博士課程修了。工学博士。東京大学助手。1975年通産省機械技術研究所研究員、その後主任研究官、遠隔制御課長、バイオロボティクス課長。1979年から1980年マサチューセッツ工科大学客員研究員。

1989年東京大学助教授、1992年東京大学教授。

(日本ロボット学会正会員)