

手書きスケッチ操作による三次元モデルの入力と 立体形状検索への応用

堀 越 力[†] 鈴木 智[†] 中 根 一 成[†]

3D モデリング技術の発展は著しく、形状を表現する技術は著しく進展した。しかし、形状の入力には、正確な図面や確定された3次元モデルのアイデアなどが必要とされ、現存のシステムは設計支援のツールに留まっている。一般にデザイナーが3次元モデラーに対して欲しているのは、手書きスケッチプロセスに対応する機能である。本論文では、まず、基本モデルとして超二次関数を用い、複数方向から見た3次元形状をスケッチすることで、容易に三次元モデリングを行うシステムについて述べる。超二次関数は2つの曲線を3次元空間で直交した形で定義される。そこで、これら曲線を物体の輪郭と考え、輪郭データを手書きスケッチで入力する。対象が複雑な形状である場合は、複数の部品に分割し、個々の部品の輪郭をスケッチすることになる。そして、各輪郭に2次元の超二次関数を近似し、超二次関数を定義する曲線を求める。2面または3面図から得られた複数の2次元の超二次関数を3次元の超二次関数として定義し直すことで、スケッチから三次元物体を容易に作成することができる。次に、関数パラメータの比較による立体形状検索システムを作成し、その概要について述べる。このシステムでは、スケッチにより作成された三次元モデルに類似した対象物の検索が可能である。類似性の判断には、関数パラメータのユークリッド距離を用いており、この妥当性について評価した。

3D Modeling Using Rough Sketches and 3D Shape Retrieval System

Tsutomu Horikoshi,[†] Satoshi Suzuki[†] and Kazunari Nakane[†]

This paper proposes a method that create a 3D model from rough sketches, and presents a 3D shape database system designed to be accessed by the resulting model. First, by using superquadrics, we prove that abstracting the shape of a 3D object is possible to given two rough sketches showing orthogonal cross sections of the object. Superquadrics are defined by the spherical product of two curves. These curves are obtained from the rough sketches drawn, and they are easily approximated to 2D superquadrics by using a non-linear least squares method. The 3D models of very complex objects can be reconstructed very easily by separately drawing outline sketches of each part. An experimental 3D shape database system shows that 3D objects can be compared using shape similarity by using superquadrics parameters. A model created from rough sketches is regarded as the initial 3D key. Stored objects are sorted according to the Euclidian distance of the superquadrics parameters between the stored objects and the key. Finally, the applicability of using the distance is confirmed, and it is found that the distance can be applied as a measure of similarity or difference between the shapes of objects.

1. はじめに

3次元モデリング技術の発展は著しく、現在、正確に形状を表現する能力はほぼ備わったといえる。しかし、現状では、これら3次元モデルを入力するためには、かなり正確な3次元データあるいはイメージ（アイデア）が必要であった。

一般に人間は、曖昧なイメージから、思考錯誤を繰り返す、そのイメージの具体化を行う。これを考える

と、モデリングツールの手順として、まずは簡単な形状のパーツを組み合わせて、意図した形状に変形していく手順が、人の思考過程に合っていると考えられる。つまり、最初はかなりラフな情報で三次元モデルを作成する。そして、このモデルの変形操作を繰り返すことで、イメージを確定していく。このような手順を疑似的に行うことにより、人間の思考段階に近い設計過程が可能になると考えられる。

これまで、三次元形状入力に関する研究では、図面をもとにして、正確に形状を入力することが主眼におかれ^{1),2)}、主に CAD のためのモデル入力支援として

[†] NTT ヒューマンインタフェース研究所
NTT Human Interface Laboratories

検討がなされてきた。そして、より容易な形状入力手段として手書き図面を利用する研究が多く検討されてきた^{3),4)}。これらの研究は、設計支援のツールとしての検討が主であり、先に述べた試行錯誤の段階（概念設計）を支援するような手段はほとんど検討されていない⁵⁾。筆者らは、これまで設計支援の前段階に相当する概念設計における支援を目的として3次元モデルを曖昧なイメージの段階でも容易に作成できるツールを検討してきた。

様々な形状をラフな情報で記述しようとする場合、まず描こうとする対象を簡単な形状のプリミティブ（基本要素）の組合せで記述することになる。つまり、ここで部品に分割するという事は、対象物をいくつかの限定されたパターンに分解するという事と言い換えられる⁶⁾。従来のモデリングツールは、このパターンとして、直方体、円筒、球など、複数のプリミティブの集合として形状表現を行っている。しかし、この場合、異なる形状に対して、それぞれ異なるパターンを当てはめる必要があり、異なる操作をユーザに要求することになる。そこで、容易なモデリングを行うには、このパターンすなわち、プリミティブを一種類に絞り、形状に左右されずに同様の操作ができることが重要である。

3次元モデルとして様々なモデルが提案されているが、本論文で扱うような曖昧な形状を表現するためには、形状表現に柔軟性があり、しかも、図面などから容易にそのモデルが獲得できることが重要である。

従来のモデリングツールは、用意された3次元モデル（直方体、円筒、球など）を組み合わせたか、回転軸と断面形状を定義することで回転体などの作成が可能であるが、それぞれの形状は正確に定義されている必要があり、曖昧な形状表現の条件にはそぐわない。形状表現の柔軟性のあるモデルとして、メタボールなどの分布関数モデルがある。これは、様々な形状を表現できるという特徴はあるが、作成にかなりの労力を要する。村木は、自動的にこの記述を求める手法を提案したが、現状ではかなりの計算時間を必要としている⁷⁾。

一方、超二次関数は、少ないパラメータで多様な形状が表現でき、パラメータの変化と関数形状との関係が人間の感覚と合っているため、モデリングとしては有効である⁸⁾。本論文の目的とする概念設計支援、すなわちあいまい形状の記述においては、超二次関数が有用であると考えられる。筆者らは、プリミティブモデルとして超二次関数を用い、ユーザがイメージしたもの

を複数方向から見た形（断面形状）としてスケッチすることで、容易に三次元モデリングを行うシステムを作成し、超二次関数によるモデリングの有効性を検討した。

形状を関数という数式化パターンにより形状を記述することの特長は、モデリングの容易さに加えて、形状の比較が容易であることもあげられる。本論文では、形状が超二次関数のパラメータという形でパターン化されることで、形状による分類が容易になり、類似形状の照合、比較を行うことができることを示す。つまり、超二次関数を使うことにより、既存のモデルと、新たに作成したモデルの類似形状モデルを容易に照合することが可能であり、設計段階での思考支援システムとして有用と考えられる。

本論文では、既存の3面図のデータから3Dモデルを容易に作成できることを示し、さらに、実際に作成された三次元モデルを構成する関数パラメータ値の比較により、立体形状の類似性の比較が可能であることを示す。

2. 3次元形状の記述

2.1 超二次関数

半径 r の球は極座標 (η, ω) を用いると次式で表現できる。

$$\mathbf{x}(\eta, \omega) = \begin{pmatrix} r \cos(\eta) \cos(\omega) \\ r \cos(\eta) \sin(\omega) \\ r \sin(\eta) \end{pmatrix}. \quad (1)$$

$$-\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2}, \quad -\pi \leq \omega \leq \pi.$$

ここで、この三角関数のべき乗 $(\epsilon_1, \epsilon_2 \in \mathbf{R})$ を考えることで超二次関数 (Superquadrics) を定義することができる。

$$\mathbf{S}(\eta, \omega) = \begin{pmatrix} a_1 \cos^{\epsilon_1}(\eta) \cos^{\epsilon_2}(\omega) \\ a_2 \cos^{\epsilon_1}(\eta) \sin^{\epsilon_2}(\omega) \\ a_3 \sin^{\epsilon_1}(\eta) \end{pmatrix}. \quad (2)$$

$$-\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2}, \quad -\pi \leq \omega \leq \pi.$$

ここで、各 xyz 軸方向の半径をそれぞれスカラー変数 a_1, a_2, a_3 に置き換えている。超二次関数の形状は、形状パラメータ (ϵ_1, ϵ_2) 、スケールパラメータ (a_1, a_2, a_3) のみで、その形状が決定される。

ところで、(2)式は、スケールパラメータを別にすれば、次のように分解できる⁹⁾。

$$C_1: f_1(\omega) = (\cos^{\epsilon_1}(\omega), \sin^{\epsilon_1}(\omega)), \quad 0 < \omega \leq 2\pi \quad (3)$$

$$C_2: f_2(\eta) = (\cos^{\epsilon_2}(\eta), \sin^{\epsilon_2}(\eta)), \quad 0 < \eta \leq 2\pi \quad (4)$$

$$f_1(\omega) \otimes f_2(\eta) = S(\eta, \omega) \quad (5)$$

ここで、 \otimes は2つの曲線を3次元空間で直交させる演算を表す。

すなわち、超二次関数は、2つの閉曲線 C_1 , C_2 に対して演算 \otimes を施すことであり、言い換えれば、2つの曲線が決定されると、(5)式により、立体形状が表現できることになる(図1参照)。2つの曲線は、それぞれ独立であるから2つの形状パラメータに注目すると、 ϵ_1 は超二次関数を真横から見た形状の角張り具合を制御し、 ϵ_2 は関数を真上から見た形状の角張り具合を制御していることがわかる。

従って、ある物体の真上から見た形状から、 ϵ_2 , a_1 , a_2 が求まり、真横から見た形状から ϵ_1 , a_3 , a_1 あるいは a_2 をそれぞれ求めることができる。つまりは、物体の断面形状の輪郭データに、超二次関数を近似させることで、輪郭を表す閉曲線 C が求まることになる。少なくとも2つの直交する断面の輪郭データが得られれば、それに近似した閉曲線から、容易に3次元としての復元が可能であることを意味している。

3. 手書きスケッチプロセスへの応用

超二次関数は、 z 軸対称、 x - y 平面对称な形状に限定された形状しかできない。超二次関数で多様な形状を表現するには、対象物を複数の部品に分割する必要があり、手書きスケッチとして、各部品の輪郭形状を描くことになる。

以下、スケッチによるモデリングシステムの処理手順に従って各処理内容を述べる。

3.1 輪郭形状をスケッチする

まず始めに、ある物体を真上から見たときの各部品の輪郭および、真横から見たときの輪郭図形をマウスを動かすことにより描く。スケッチのデータは、 x , y 座標の時系列データとして記憶されていく。

一般に、スケッチされた複数の輪郭データから、個々の部分の輪郭を自動的に求めることはかなり困難で

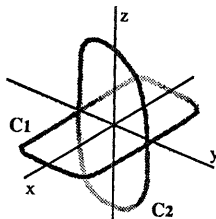


図1 超二次関数の定義

Fig. 1 Definition of superquadrics.

ある。そこで、スケッチを時系列データとして扱い、各部品ごとに閉曲線を描くという操作のもとでスケッチを入力する。

意図した対象物をより具体的に表現するためには、あらかじめユーザは関数で表現しやすい部品に分割し、各部品の輪郭形状を描くことになる。描かれるスケッチは結果的には、対象物の各部品の輪郭が透視されて1つの平面に投影されたようなスケッチが得られる。

また、平面図、側面図のスケッチにおいて、各部品をスケッチする順番を同じにすることで、部品間の対応付けの問題は回避できる。例えば、ある椅子のスケッチを描こうとした場合、平面図で背もたれ-座部-足の順番で描いたならば、側面図を描くときにも、背もたれ-座部-足の順番でスケッチする。

3.2 輪郭データの変換

マウスの移動速度により輪郭データのサンプリング間隔は変化してしまうため、密度が偏ったデータができ、主軸の決定や最小二乗法による関数近似には好ましくない。そこで等間隔に輪郭データを再サンプリングするため、まず、スケッチ画面上を x 軸を実軸、 y 軸を虚軸と考え、複素数 z

$$z(j) = x(j) + iy(j), j=1, \dots, n, \quad (6)$$

を点 (x, y) と同一視する。すると、スケッチされた曲線 C は、複素数値関数 z と考えることができ、次のように表される。

$$\ell(j) = |z(j) - z(j-1)|, \quad (7)$$

$$L = \ell(1) + \ell(2) + \dots + \ell(n). \quad (8)$$

L がスケッチされた曲線 C の全長となる。ここで、 $\ell(j)$ の値は、マウスを動かす速度により変化する。しかし、複素数列 L を十分に小さい幅の ℓ' で再分割することで、 ℓ' の値が一定、すなわち L が等間隔に区分されるような複素数列に変換することができる¹⁰⁾。これにより、スケッチ曲線は $z(0)$ を始点として等間隔 ℓ' でサンプリングしたデータに変換することができる。

3.3 各部品の関数近似

2次元の超二次関数は、

$$S(x_i; \epsilon, a_1, a_2) = \left(\frac{x}{a_1}\right)^{2/\epsilon} + \left(\frac{y}{a_2}\right)^{2/\epsilon} = 1, \quad (9)$$

と表され、スケッチされた各部品の輪郭データに対してこの超二次関数を、非線形最小二乗法を用いて当てはめる¹¹⁾。平面図ならびに側面図として入力された時系列データを、それぞれ D_{top} , D_{side} と表すこととし、

各輪郭データに超二次関数を当てはめていく。このとき、各輪郭データの重心 (x_c, y_c) および、モーメント主軸法により求めたデータの主軸の傾き θ をもとにデータのワールド座標系 (w) を超二次関数の座標系 (s) に変換する。

$$D^s = (\text{Rotation}(\text{Translation}D^w)). \quad (10)$$

ここで、*Translation* は重心への平行移動、*Rotation* は主軸への座標軸の回転を表す。

パラメータの初期値として、形状パラメータ ϵ は 1.0、スケールパラメータ a_1, a_2, a_3 は、各座標列 $D_{i_{op}}, D_{i_{ide}}$ の x, y 座標の最大値 x^{max}, y^{max} 、最小値 x^{min}, y^{min} とするとき、

$$a_1 = |x^{max} - x^{min}|, \quad a_2 = |y^{max} - y^{min}|, \quad (11)$$

とする。そして、各データ列に関数近似を行い、 $D_{i_{op}}$ ならびに $D_{i_{ide}}$ から、それぞれ曲線 $C_{i_{op}}, C_{i_{ide}}$ が求められる。

3.4 非線形最小二乗法による形状近似

超二次関数は非線形であるから、最小二乗近似として、非線形最小二乗法 (Levenberg-Marquadt 法) を用いる。輪郭データ (x, y) が近似された関数 S 上に存在するならば、 $S=1$ になることから、輪郭データ $D\{x_i | i=1, \dots, n\}$ に対する最小二乗誤差 χ^2 は、次式で表される。

$$\chi^2 = \sum_{i=1}^n \{1 - S(x_i; \epsilon, a_1, a_2, \theta, x_c, y_c)\}^2. \quad (12)$$

回転パラメータ θ と平行移動パラメータ (x_c, y_c) も考慮すると、全部で 6 個のパラメータについて、二乗誤差が最小になる各パラメータの組合せを求める。

また、実際には超二次関数の表現能力をあげるため、(13)式に示すように、スケールパラメータ a_1, a_2 を、スカラーではなく先細り関数(14)式に置き換え、三角形や台形も 1 つの超二次関数で表現できるようにしている。

$$\mathbf{x}'(\eta, \omega) = \begin{pmatrix} f_1(x) \cdot a_1 \cos^{\epsilon_1}(\eta) \cos^{\epsilon_2}(\omega), \\ f_2(x) \cdot a_2 \cos^{\epsilon_1}(\eta) \sin^{\epsilon_2}(\omega), \\ a_3 \sin^{\epsilon_1}(\eta). \end{pmatrix} \quad (13)$$

$$f_i(x) = 1 - k_i \cdot \frac{x}{a_3}, \quad i=1, 2. \quad (14)$$

この場合、先細りのパラメータ k_i も可変パラメータとして、(12)式に組み込まれる。

対象となる輪郭データが関数で表現し得る形状であれば、この最小二乗法を用いることで、ほぼ安定に収束させることができる。図 2 は、スケッチの近似過程を示している。黒点が等間隔にサンプリングされたスケッチのデータであり、曲線が超二次関数である。初

期形状は楕円であるが、徐々にスケッチデータに近似されていく様子がわかる。

3.4.1 パラメータの正規化と 3D レイアウトの計算

スケッチで描かれた形状は、平面図と側面図とでは、相似形であるとしても、絶対的な大きさが同じとはならない。そこで、両方のスケッチから得られるパラメータを用いて、平面図を基準に各部品の正規化を行わなければならない。

側面図として y 軸方向からの形状が描かれたと仮定すると、 a_1 が両方のスケッチから求められることになる。すなわち、平面図からスケールパラメータ $a_{1_{op}}$ 、側面図から $a_{1_{ide}}$ が得られる。そこで、側面図から得られたパラメータを、 $\alpha = a_{2_{op}}/a_{2_{ide}}$ 倍にすることで大きさの補正を行う。対象物の 3 次元形状 P は次式により作成することができる。

$$P = C_{i_{op}} \otimes \alpha C_{i_{ide}} \quad (15)$$

ここで、部品の主軸が θ だけ回転していた場合、

$$\alpha = \frac{a_{2_{op}}}{a_{2_{ide}}} \cos(\theta), \quad (16)$$

となる。

しかし、複数の部品 (m 個) からなる物体を作成する場合、各部品の相対的な位置関係も重心の位置関係から求める必要がある。まず、平面図から (x_i^{op}, y_i^{op}) 、側面図から (x_i^{ide}, y_i^{ide}) が重心として求められたとする。平面図は xy 平面、側面図は xz 平面を表すとき、両図面から求められる x_i^{op}, x_i^{ide} 座標は、共に三次元空間上で y 座標を表しているの、側面図の位置補正のため $\beta (= x_{i_{op}}/x_{i_{ide}})$ 倍する。つまり、各部品 P_m の重心 g は $(x_i^{op}, y_i^{op}, \beta \cdot y_i^{ide})$ となる。

最終的には、各部品ごとに再構成された超二次関数 (P_m) を、求められた各重心の位置に配置することで、スケッチにより描かれた対象物の三次元モデルを表現することができる。図 3 は、椅子のスケッチを基に 3 次元モデルを作成した例であり、10 個の部品で構成さ

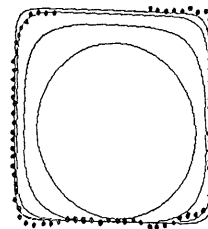


図 2 近似過程
Fig. 2 Fitting procedure.

れている。2つの図面のスケールが異なり、かなり粗いスケッチであっても簡単に3次元モデルが作成できることがわかる。

一般に、2面図では、3次元形状を正確に表現することはできない。この椅子のスケッチの例で言うならば、正面図と平面図の2つのスケッチのみであり、側面から見た形状に関する情報が欠けているため、各製品の側面形状は、すべて楕円形状になっている。一度入力された3次元モデルを変形することでこの欠点を解除することは可能であるが、入力段階で3次元形状を正確に表現しようとするならば、少なくとも3面図が必要になる。3面図の場合、形状に関するパラメータおよび、回転パラメータが増えるが、同様に、先に述べた方法を適用することができる。

例えば、3面図が必要な例として自動車の形状を考えてみる。自動車を屋根および車体部分に分けて表現するならば、それぞれの図面において、2つの断面形状を図7に示すように四角い輪郭をスケッチしていく。

ここで、 x 軸、 y 軸、 z 軸に対して、図4に示すような先細り関数 $f_{x,y,z}(x)$

$$a_1: f_x(z) = 1 - k_x \cdot \frac{z}{a_3} \tag{17}$$

$$a_2: f_y(z) = 1 - k_y \cdot \frac{z}{a_3} \tag{18}$$

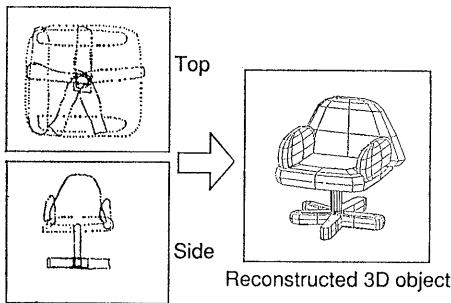


図3 椅子のスケッチから得た3次元モデル
Fig. 3 Result of the 3D model using the sketches (chair).

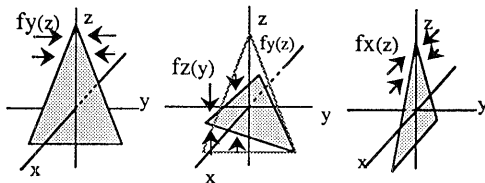


図4 自動車モデルに対する先細り関数
Fig. 4 Tapering function for a car model.

$$a_3: f_z(y) = 1 - k_y \cdot \frac{y}{f_y(z)} \tag{19}$$

を組み込んでいる。

図5は各種の自動車をスケッチにより作成したいくつかの例である。対象形状に合わせて、変形関数を適当に選ぶことにより、超二次関数を2個用いるだけでも、かなりの種類の形状を表現することができるがわかる。

4. 立体形状検索への適用

つぎに、超二次関数により記述された3次元モデルを用いて、その類似形状の比較が可能であることを、自動車の形状の例について示す。

4.1 システム構成

筆者らは、スケッチによる形状をもとにした類似形状検索システムを作成した。このシステムでは、マウスを用いて描かれたスケッチデータをもとに3次元モデルを作成する。そして、このモデルを基本参照モデルとして、類似形状の3次元モデルを探すことができる。システム概要を図6に示す。まず、マウスを用いて3次元形状のスケッチが描かれると、提案した方法により、その超二次関数のパラメータが計算される。そして、超二次関数の組合せとして表現された3次元画像が生成される。この3次元画像とその超二次関数のパラメータがデータ蓄積部に記憶される。そして、既に蓄積されている形状との比較が行われ、類似形状順にディスプレイに表示されるようになっている。こ

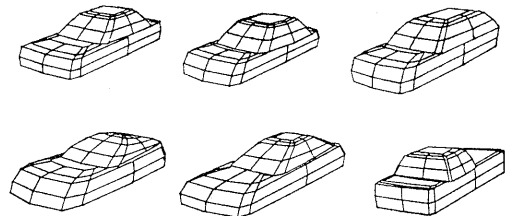


図5 手書きスケッチとその3次元モデル
Fig. 5 3D models created by sketch drawings.

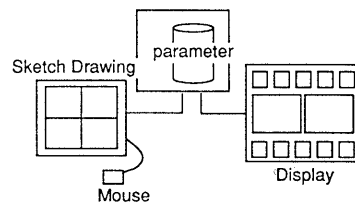


図6 3次元モデリング & 検索システム
Fig. 6 3D modeling & retrieval system.

ここで、比較とはモデルどうしの各関数のパラメータのユークリッド距離の大きさを比較することである¹²⁾。実際の検索画面としては、図6のディスプレイ上部の小ウインドウに類似形状順に候補画像が表示され、中央の左に最も類似した蓄積画像、右側にその3次元モデルが表示され、下のウインドウには、検索された画像の3面図など、属性データが表示される。

4.2 検索のパラメータ

ここで、パラメータとしては、屋根の部分、車体部分それぞれについて、形状パラメータ (ϵ_1, ϵ_2)、スケールパラメータ (a_1, a_2, a_3)、先細りパラメータ (k_x, k_y, k_z) があり、2つの部品の関連パラメータとして、車体の重心座標を中心としたときの、屋根の重心の位置ベクトル (d_x, d_y, d_z) がある。すなわち、19個のパラメータを用いて、各パラメータの値の差を比較する。

使用したモデルは、自動車のモデル、セダン6種、ハッチバック2種、ワゴン3種、トラック1種の図5に示したような形状のモデル合計12種類であり、3面図の輪郭をスケッチによりトレースすることで2つの部品で構成される3次元モデルを作成したものである。

4.3 検索過程

最初に、検索しようとする自動車の形状を図7に示すようにスケッチとして入力する。左上が平面図、右上が側面図、左下が正面図のスケッチを表している。そして、右下が、このスケッチから作成された3次元モデルである。

スケッチにより作成された3次元モデルのパラメータ

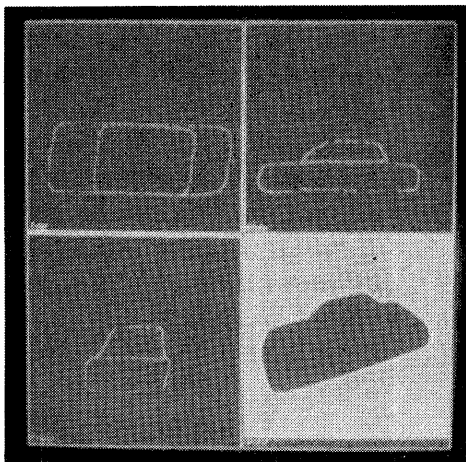


図7 車のスケッチ

Fig. 7 Sketch drawings for a car model.

タが基本参照モデルとなる。図8は、このモデルをもとに検索された車である。スケッチがセダンであり、検索された形状もそれに類似した形状の車が検索されていることがわかる。

また、関数パラメータを変化させることで容易に他の形状の車を作成できる。図9は、セダンのキャビン部分を移動させ、ハッチバックの形状に変形させた様子であり、図10は、その形状による検索結果である。

4.4 形状の類似性比較の妥当性

ここで形状の類似性比較には、各パラメータのユークリッド距離を用いているが、この距離の妥当性について検討する。検索に用いた13個のモデルをユークリッド距離をもとに、クラスタ分析により分類を行っ

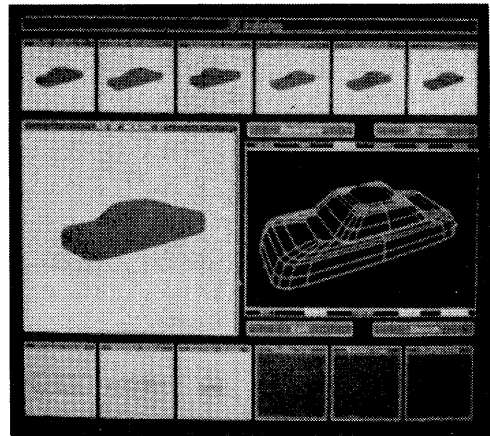


図8 スケッチモデルによる検索結果

Fig. 8 Result of retrieval using the model created by sketch.

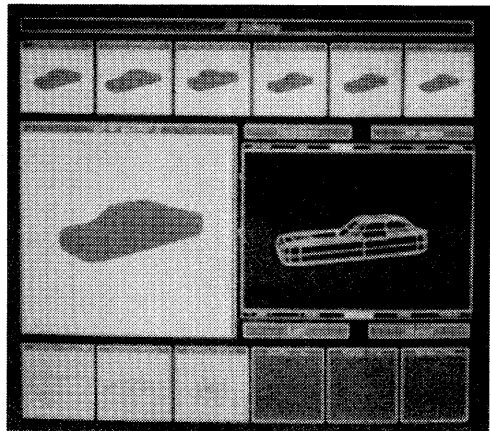


図9 3次元モデルの変形

Fig. 9 An example of modifying 3D model.

た結果を図 11 に示す。クラスタ間の非類似度は、ウォード法を用いた¹³⁾。つまり、クラスタ (p) 内の偏差平方和の合計を

$$S_p = \sum_{j=1}^m \sum_{i=1}^{n_p} (x_i^{(p)} - \bar{x}^{(p)})^2, \quad (20)$$

と表すとき、クラスタ p と q を融合してクラスタ t をつくる時クラスタ内の平方和の合計の増分 ΔS_{pq} は

表 1 スケッチモデルと蓄積モデルとの距離
Table 1 The distance between the sketch model and stored models.

	car #2	car #7	car #9	car #3
car# 13	1.811	1.833	1.858	1.893

$$\Delta S_{pq} = \frac{n_p n_q}{n_p + n_q} \sum_{j=1}^m (\bar{x}^{(p)} - \bar{x}^{(q)})^2, \quad (21)$$

となる。クラスタ pq 間での非類似度を ΔS_{pq} として定義するとき、クラスタ内平方和ができるだけ小さくなるように融合を行うこととして、クラスタ形成を行っていくと、図 11 に示すような木が求められる。ここで、適当な閾値を設けることで、複数のクラスタに分類できる。例えば、図 11 で示した閾値 (=0.5) を用いると、

クラスタ 1: car #1, car #2, car #3, car #4, car #5, car #6 (セダン)

クラスタ 2: car #7 (角張ったハッチバック)

クラスタ 3: car #8, car #9, car #10 (ワゴン)

クラスタ 4: car #11 (トラック)

クラスタ 5: car #12 (ハッチバック, スポーティタイプ)

といった大きく 5 つのクラスタに分類できる。このように、単にユークリッド距離を用いるだけでも比較的妥当な分類が可能であるといえる*。

スケッチにより作成したモデル car #13 のパラメータと、各モデルとのユークリッド距離を見ると、表 1 に示すように、car #2 との距離が最も近い。しかし、クラスタ 2 に属する car #7 が、2 番めに近いモデルとなっている。これは、屋根と車体の個々の形状が類似しているためであり、屋根と車体の位置関係を表すパラメータの比重を大きくすることで、セダンの車に分類することができる。

以上のことから、単純なユークリッド距離では、パラメータをうまく選定すること、パラメータにうまく重みづけすることが妥当な分類を得るために必要であるといえる。また、あらかじめモデルを複数のクラスタに分類しておき、最隣接モデルを探し出すといった手順が妥当であることがわかる。

5. あとがき

超二次関数を用い、手書きスケッチから 3 次元モデ

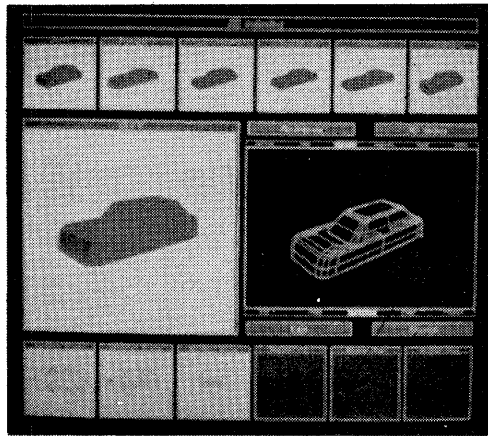


図 10 検索結果
Fig. 10 Result of retrieval.

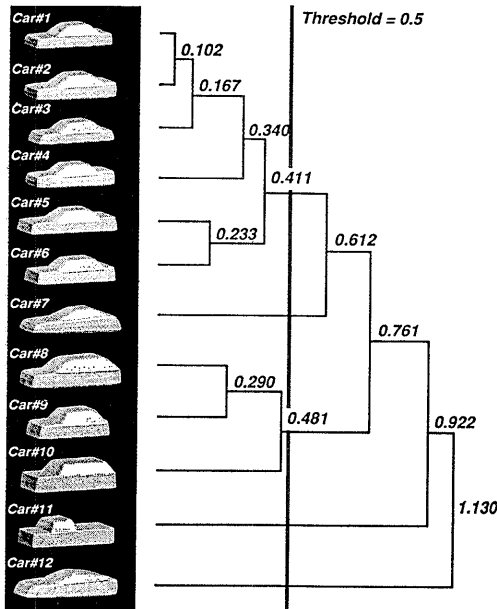


図 11 クラスタリング結果
Fig. 11 Result of clustering.

* 距離ならびにクラスタの統合の定義の仕方により、図 11 の形は変化してしまうため、実際は目的に合った定義を考える必要がある。

ルを作成し、この作成されたモデルをもとに類似形状のモデルを検索するシステムについて述べた。本システムでは、3次元物体を複数の部品に分割して記述しているため、異なる分割がなされている対象に対して、類似形状の検索ができないといった課題が残っている。そのため、任意のユーザを対象とした検索システムにするためには検索対象の分割方法をアシストする手段が必要である。

また、類似形状の判断として、ユークリッド距離による比較の妥当性について検討した。この場合、対象物に関わらず、丸まり具合いや大きさの比率といった形状の比較になってしまうが、対象分野によっては、類似形状といっても特定の見方がある。そのような場合、心理評価実験をもとに、正準相関分析などを行う必要がある。

本論文で示した3次元のモデリングは、従来のシステムのような、正確なモデルを作成するためのツールではなく、試行錯誤を行う概念設計段階でのツールとして検討を進めてきたが、このシステムは、プリミティブが楕円体に近い形状を有しているため、人間など動物の3次元モデルの簡易入力ツールとしても有用と考えられる。

謝辞 本研究を進めるにあたり、ご指導頂いたNTT ヒューマンインタフェース研究所マルチメディア処理研究部栗原定見部長、マルチメディア処理研究部視覚情報処理研究グループ末永康仁主幹研究員に深く感謝いたします。また、本研究に対し有益なご助言ならびにご協力を頂いた第6プロジェクト大和淳司氏をはじめ、マルチメディア処理研究部の皆様にも感謝いたします。

参 考 文 献

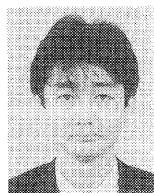
- 1) 千田豊満: 三面図からのもとの立体の自動復元, 情報処理学会論文誌, Vol. 32, No. 9, pp. 1122-1128 (1991).
- 2) 高橋正充, 伊藤 潔: 擬似ブール代数解法による三面図からの曲面物体の自動合成, 情報処理学会論文誌, Vol. 32, No. 6, pp. 740-748 (1991).
- 3) 棟上昭男, 美濃導彦, 田中 聡ほか: 手書き設計図面自動入力技術ワークショップ報告書, 情報処理振興事業協会技術センター (1985).
- 4) Hosaka, M. and Kimura, F.: Using Handwriting Action to Construct Models of Engineering Objects, *Computer*, Vol. 15, No. 11, pp. 35-47 (1982).
- 5) 安居院猛, 奥山健一, 中嶋正之: スケッチ画を用いた立体形状の入力 第3報, 信学会秋季全国大

会予稿集, D-424 (1990).

- 6) 沖野教郎: コンピュータによる自動デザイン, 日刊工業新聞社 (1967).
- 7) Muraki, S.: Volumetric Shape Description of Range Data Using Blobby Model, *Computer Graphics (Proc. SIGGRAPH '91)*, Vol. 25, No. 4, pp. 227-235 (1991).
- 8) Pentland, A.: Recognition by Parts, *Proc. IEEE 1st Int. Conf. on Comp. Vision*, pp. 612-620 (1987).
- 9) Barr, A.H.: Superquadrics and Angle-Preserving Transformations, *IEEE Comput. Graphics and Applications*, pp. 11-23 (Jan. 1981).
- 10) 上坂吉則: 開曲線にも適用できるフーリエ記述子, 信学論, Vol. J 67-A, No. 3, pp. 166-173 (1984).
- 11) Solina, F. and Bajcsy, R.: Recovery of Parametric Models from Range Images: The Case for Superquadrics with Global Deformations, *IEEE Pattern Anal. Machine Intell.*, Vol. PAMI-12, No. 12, pp. 131-147 (1990).
- 12) 堀越 力, 笠原久嗣: 超二次関数による三次元インデクシング, 信学論, Vol. J 73-D-II, No. 10, pp. 1716-1724 (1990).
- 13) 田中 豊, 垂水共之, 脇本和昌: パソコン統計解析ハンドブック II, 共立出版 (1971).

(平成5年12月24日受付)

(平成6年6月20日採録)



堀越 力 (正会員)

1985年慶應義塾大学工学部電気工学科卒業。1987年同大学大学院修士課程修了。同年、日本電信電話(株)に入社。以来、画像データベース技術、画像処理技術、3次元形状モデリングに関するヒューマンインタフェースの研究に従事。現在、画像通信事業本部映像通信サービス部主任技師。電子情報通信学会会員。



鈴木 智 (正会員)

1979年静岡大学工学部情報工学科卒業。1984年同大学院博士課程修了。同年、日本電信電話公社に入社。以来、線図形認識、コンピュータビジョンの研究開発に従事。1991年から1992年、カリフォルニア大学アーバイン校にて客員研究員。現在、NTTヒューマンインタフェース研究所企画部担当課長。工学博士。コンピュータビジョン、画像処理、並列処理に興味を持つ。電子情報通信学会、IEEE各会員。

**中根 一成**

1970年慶應義塾大学工学部電気工学科卒業。1972年同大学大学院修士課程修了。同年、日本電信電話公社（現 NTT）に入社し、武蔵野および横須賀の研究所にて画像通信システム、画像処理システムの研究実用化に従事。1988年より ATR 視覚情報機構研究所に出向。1992年より NTT ヒューマンインタフェース研究所主幹研究員。電子情報通信学会、TV 学会、人工知能学会、IEEE 各会員。
