人の教示に基づく3次元メッシュモデルの形状類似検索

Human-Directed Search of Three-Dimensional Mesh Models Based on Shape Similarity

衣 鳩 昌 俊[†], 小田切 智^{††}, 大 瀬 竜 太 郎^{†††}

Masatoshi Ibato[†], Tomo Otagiri^{††} and Ryutarou Ohbuchi^{†††}

Abstract As the popularity of three-dimensional (3D) geometric models has increased, so too has interest in methods to search 3D models based on their shape similarity. Two of the most difficult problems in finding shape similarity between 3D geometric models are a feature vector that succinctly describes the shape of the model and a method to compute the distance between a pair of feature vectors that reflects user preferences. We describe a human-directed 3D shape similarity search method that reflects, to some extent, the user preferences by using a learning classifier. Given an example 3D mesh model, the system first retrieves, as an initial guess, a set of models that are similar to the query based only on an unbiased mechanical measure of the distance (i.e., the Manhattan distance) between a set of feature vectors. The user then iteratively refines the query by tagging a subset of the retrieved models as being either similar or dissimilar. The system learns the user's preference using a learning classifier support vector machine (SVM), so that the distance values between the set of feature vectors are altered to reflect these preferences. Experimental results show that our method is capable of retrieving 3D models that better reflect the preferences of the user than the simple method using only the Manhattan distance.

キーワード：3次元グラフィックス，形状モデリング，ポリゴンメッシュ，類似検索，学習アルゴリズム，サポートベクタマシン

1. ま え が き

近年，コンピュータ技術の発展とともに，3次元グラフィックスや3次元形状CADが広く普及し，3次元モデルの数は増加している．ゲームコンソールやパーソナルコンピュータ上でのいわゆるポリゴンゲームはもとより，テレビや映画における特殊効果や広告画像，自動車を初めとする機械部品や機械製品の形状設計などで数多くの3次元モデルが作り出されているからである．3次元モデルの普及にともない，これらのモデルを効率よく管理し，再利用する目的で3次元モデルの本質的属性である3次元形状による類似検索が注目されつつある^{2)～5) 7)～11) 15) 18) 20) 23)～28)}．しかしこの

分野の研究は，まだ始まったばかりであり，数多くの問題が残されている．まず，検索システムの入力としてどのような形状表現を仮定するのが難しい．また，簡潔で曖昧性が低く，計算が容易な形状特徴はいまだ見つかっていない．さらに，特徴ベクトルを比較して相違度を計算する手法にもいろいろあり，いずれも一長一短である²¹⁾．特に問題となるのが，機械的に計算された相違度に基づく検索結果とユーザの期待する検索結果が異なる点である．人と機械の差のみならず，ユーザごとに期待する類似性に差があることも困難さを増す原因である．

本論文では，あるユーザが期待する3次元モデルの形状類似検索結果を得るための一手法を提案する．本論文で提案する手法では，検索とその結果に対するユーザによる教示を1サイクルとし，このサイクルを数回反復することで，ユーザの意図をより反映した3次元モデル群を検索し，提示する．本手法ではこれを，教示を学習した学習型類別アルゴリズムの類別結果を基に，特徴ベクトル間の相違度計算結果を修正することで実現する．学習型類別アルゴリズムとしてはSVM (Support Vector Machine) を用いる．

システムはまず，初回の検索結果として，特徴ベクトル間のManhattan距離のみによる結果をユーザに提示する．ユーザは，その結果得られたモデル (の1部) に対し，求めるモデルと似ている・似ていないを示すタグを付け，シス

2002年3月4日受付，2003年3月17日最終受付，2003年4月11日採録

[†] 茨城計算センター株式会社

(〒316-0025 日立市森山町4-8-2, TEL 0294-54-1311)

^{††} 山梨大学 大学院工学研究科

(〒400-8510 甲府市武田4-3-11, TEL 055-220-8570)

^{†††} 山梨大学 医学工学総合研究部

(〒400-8510 甲府市武田4-3-11, TEL 055-220-8570)

[†] Ibaraki Computing Center, Incorporated

(4-8-2, Moriyama-cho, Hitachi-shi, Ibaraki, 316-0025, Japan)

^{††} Graduate School of Engineering, Yamanashi University

(4-3-11, Takeda, Koufu-shi, Yamanashi, 400-8510, Japan)

^{†††} Interdisciplinary Graduate School of Medicine and Engineering, University of Yamanashi

(4-3-11, Takeda, Koufu-shi, Yamanashi, 400-8510, Japan)

テムに教示する。システムのSVMは、そのタグからユーザの判断を学習し、次の検索において学習したSVMの分類結果を基に特徴ベクトル間のManhattan距離に修正を加える。この検索・教示サイクルを数回繰り返すと、ある程度人間の意図を反映した検索結果が得られるようになる。特徴ベクトルとしては、OsadaらがD2と呼ぶ形状特徴⁴⁾に少々の変更を加えたものを用いる。D2は3次元モデルの向きや位置に依存せず、計算が容易で、かつ形状モデルの位相的あるいは幾何的縮退にも強い。実験の結果、学習を用いた場合には、学習なしの場合に比べて、よりユーザの求めるモデルが検索できることがわかった。

以後、次章でこれまでの3次元モデルの類似検索手法を概観する。ついで、第3章で我々の検索手法について述べ、第4章で評価実験とその結果を、それぞれ述べる。さらに第5章でまとめと今後の課題を述べる。

2. 関連研究

形状そのものの類似性に基づく3次元モデルの検索では、以下の4点を考慮する必要がある。

- (1) 検索要求の提示：いかにして3次元モデルの形状を提示し、類似検索を行うか
- (2) 形状特徴ベクトル：どのような形状特徴ベクトルを用いるか。形状特徴ベクトルとその抽出法は、対象となる形状の表現形式に依存する
- (3) 形状相違度計算：どのように形状特徴ベクトル間の相違度(あるいは類似度)を計算するか
- (4) 索引と取出し：いかにして効率良く求めるモデルまたはモデル群を取出すか

検索要求の提示

3次元モデルによる類似検索は、探したい3次元モデルをシステムに提示することで始まるが、これは容易ではない。

現在、多くの3次元モデルの類似検索システムは、求める形状に似た3次元モデルが手元にあると仮定し、これを検索モデルとして提示する。本論文で述べる手法もこの方法を前提としている。しかし、この仮定が成り立たない場合も多い。他に、求める形状を3面図のような2次元画像(の組合せ)としてスケッチするもの¹⁸⁾、さらにはIgarashiらのTeddy¹⁹⁾のような3次元のスケッチ手法を用いて、3次元モデルを直接入力するもの¹⁸⁾、等が考えられる。2次元スケッチでも、3次元スケッチでも、貫通穴が開いていたり、紐が絡んでいたりするような複雑な形状を入力するのは困難であったり、ほとんど不可能であったりする。検索のために3次元モデルを指定する手法は、今後の研究が必要な部分である。

形状特徴ベクトル

形状を表す特徴ベクトルは、多数提案されているが、いまだ決定的なものがない^{2)~5) 7)~11) 15) 18) 20) 23)~28)}。その難しさの一つは、3次元モデルの形状を表現する手法には数多くの種類が存在し²²⁾、その多くに相互交換性がないため

である。真に「3次元」の、体積が計算でき、物体の内外が判定できる立体形状の表現に限っても、境界表現(曲面で囲まれたものを含む)、陰関数によるレベルセット表現とその離散化であるボクセル表現、など何種類もある。これら立体は性質が良く、近似的な相互変換も可能である。これに対し、インターネット上で最も多く普及しているWeb3D(VRML)モデルは、いわゆる「ポリゴンスूप」である。ポリゴンスूपとは、(互いに独立な)多角形、多角形とその隣接関係で構成されたポリゴンメッシュ、あるいは折れ線や点等が「ごった煮」のように集まってできている。ポリゴンスूपによる形状表現は立体を定義せず、面等の集合により見かけだけ3次元の形を表現する。また、ポリゴンメッシュや立体でも、幾何的あるいは位相的なノイズや縮退があると処理が困難となる。例えば、同一座標の頂点が複数あったり、面積0の面があったり、大きさ0の貫通穴があったりすると扱いが厄介になる。

立体を定義した性質のよいモデルの場合には、理論的に裏付けのある手法を適用しやすい。主に幾何学的特徴を用いた手法としては、元の形状とその凸包との体積比などを用いた手法²³⁾、凸包の性質を用いて与えられた形状を構成要素に分解し、構成要素ごとに形状特徴を求める手法²⁵⁾、ボクセル形式に変換して特徴を求める手法^{3) 15) 18)}、曲面の曲率の分布のヒストグラムを用いる手法²⁷⁾、等がある。また、3次元医療画像のボクセル形式データを入力とし、一種のモルフォロジによる多重解像度分解により、特徴抽出を行う手法もある²⁰⁾。位相的特徴を用いた手法では、単連結な多様体曲面で定義された形状に対し、その曲面の位相構造を解析して特徴とする手法が注目される⁵⁾。また、立体の境界表現における面分の接続関係などの情報をグラフとして表し、そのグラフのメッシュベクトルを特徴ベクトルとして用いる手法もある²⁴⁾。

ポリゴンスूपで定義された、「性質の悪い」3次元形状の場合、幾何要素の接続性が充分でなかったり、複数の単連結要素があったり、立体でなかったりすることから、曲面の位相構造や立体の体積を計算したりすることができない。こうした「性質の悪い」形状モデルを対象とした特徴には、①モデルの姿勢、つまり位置・向き・大きさの正規化が必要なものと、②姿勢の正規化が不要なもの、の二つに分類することができる。姿勢の正規化が必要な特徴には、面の法線ベクトルの分布^{7) 8)}、モデル全体から計算した慣性モーメント²⁾、モデルのbounding boxを一様分割した3次元格子内の頂点の分布^{9)~11)}、モデルの慣性主軸沿いに離散的にパラメータ化したモーメント等の包絡線²⁶⁾、面分集合のHough変換²⁸⁾、などがある。これに対し、モデルの向きに依存しない特徴には、面上の点の分布の統計量、例えば、任意の2点間の距離のヒストグラム(D2と呼ばれる)などを用いたOsadaらの手法⁴⁾、面分をボクセル表現に変換し、この分布を球面調和関数で近似する手法^{15) 18)}等がある。

形状相違度計算

特徴ベクトルが求まると、これを比較し、特徴ベクトル間の距離、あるいは相違度を求め、これをもとに相違度の低いモデル（またはモデル群）を取出すことになる。特徴ベクトル間の類似度を計算するのに用いる距離尺度としてはEuclid距離、Manhattan距離、Hausdorff距離、等数多くの距離尺度が考えられる²¹⁾。

ここで考慮する必要があるのが、人による類似度の判断と、「機械的に」計算したモデル間の類似度が一致しない場合が多いことである。形状類似検索において先駆的な研究の一つである鈴木らの研究^{9)~11)}では、多次元尺度法を用い、人の主観と検索される形状の一致を試みた。Corneyらの研究²³⁾では、データベースの部分集合のモデルを対象に人が求めた類似行列と、機械が求めた類似行列が近くなるように特徴ベクトルの重みを調整した。また、Eladらの研究²⁾のように、学習型の類別アルゴリズムを用い、反復して正しい検索例と正しくない検索例とを教示することで、漸進的にユーザの欲するモデル（群）を検索しようとするものもある。

本論文で述べる我々の手法は、Eladらの手法と同様、検索システムがユーザの教示から学習することで、そのユーザが求める形状を検索する。

索引と取出し

3次元モデルの検索と検索された形状モデルの取出しを効率化するためには、何らかのインデックス構造を考える必要がある。例えば、ボクセルデータを入力として検索を行ったKeimらの手法²⁰⁾では、木構造の検索インデックスを用いて検索を効率化した。しかし、3次元モデルの検索に特化した索引手法の研究は数少ない。3次元モデルの形状類似検索の研究分野そのものが新しいため、現在のところ、処理の効率化よりは、むしろ形状特徴ベクトルやその比較手法に研究の力点がおかれているためである。

3. 検索システム

本論文で提案する3次元モデルの形状類似検索手法では、ユーザが検索システムに対して、インタラクティブに繰り返し検索と教示を行い、教示の内容をシステムが学習することで、徐々にユーザの欲する3次元モデルを検索・取出す。システムの処理の概要を図1に示す。

本論文の類似検索手法は、検索対象として、頂点座標と頂点の接続性から形状が定義される、ポリゴンスプと呼ばれる広い意味での3次元ポリゴンメッシュモデルを対象とする。これは、本手法で用いた特徴ベクトルであるD2⁴⁾が、形状表現に対して寛容なためである。対象とするモデルがソリッドでなくとも良いのはもちろん、複数の単連結要素の集合や、面積ゼロの多角形も許容する。ポリゴンとしては三角形を仮定しているが、多角形が与えられた場合にはあらかじめ三角形分割をして処理すればよい。

本手法では、3次元モデルを検索モデルとして提示し、

図1 検索手法の概要

A shape similarity search framework.

インタラクティブに検索を行う。ユーザはまず、検索キーである3次元モデルをシステムに提示する。システムは提示された検索モデルから特徴ベクトルを抽出し、データベース内のモデルの特徴ベクトル（あらかじめ計算され、モデルと共に格納されているとする）との相違度比較を行う。初回の検索は、検索モデルとデータベース中の全モデルの特徴ベクトル間のManhattan距離で行う。システムは、（機械的に計算した）相違度が低い幾つかのモデルをデータベースから取出して、検索結果としてユーザに提示する。ユーザは、これをもとに、システムに対し教示を行う。具体的には、システムが提示したモデル群に対し、求めるモデルであるなら“+”のタグを、そうでないモデルなら“-”のタグ付けを行う。タグ付けされたモデル群は学習データとしてSVM (Support Vector Machine) に与えられる。SVMは、与えられた学習データを基に、各モデルの特徴ベクトルに対するユーザの判断を学習する。次の検索では、相違度計算の際、SVMの類別結果に基づき、相違度計算結果を修正し、修正して得られた相違度値を基に検索した結果をユーザに提示する。この手順を繰り返し、SVMに幾度か学習させることで、人の意図をある程度反映した形状類似検索が実現される。

3.1 特徴ベクトルの作成

本手法で我々が使用した特徴ベクトルは、Osadaらが提案した複数の統計的手法⁴⁾のなかで、性能・処理時間の両面で最も良い結果を与えたD2である。D2の最大の特徴は、3次元モデルに加えられた相似変換に対し不変なことである。さらに、モデルがソリッドを表現していなくとも、非連結な複数のメッシュから構成されていても、またメッシュですらなく、独立な三角形の集合であっても、D2は計算できる。我々の実装したD2は、その詳細が多少OsadaらのD2と異なるので、以後、我々のそれをmD2と呼ぶ。

D2を計算するにはまず、3次元モデルの面上にランダム

図2 モデルの表面上に点を生成
Generating a set of points on the model's surface.

図3 疑似乱数列 (drand48 ()) を使用 (左) と Sobol の準乱数列 (右) により点を生成した結果

Points generated by using the pseudo-random number sequence (drand48 ()) (left) and the Sobol's quasi-random number sequence.

に点を生成し、そのすべての2点対間の距離を計算する。この距離のヒストグラムがD2特徴ベクトルである⁴⁾。独立三角形と3角形メッシュより構成されるモデルの面上に、一様に分布する点の座標 $\mathbf{P}$ を生成するには、モデルの頂点とその接続性より構成される3角形の頂点座標 $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ から式 (1) を用いて計算する (図2)。

$$\mathbf{P} = (1 - \sqrt{r_1})\mathbf{A} + \sqrt{r_1}(1 - r_2)\mathbf{B} + \sqrt{r_1}r_2\mathbf{C} \quad (1)$$

Osadaらは式 (1) の二つの数, r_1 , r_2 (0から1の間) には疑似乱数を使用しているが、我々のmD2では準乱数の一つであるSobol²⁹⁾を用いることにした。準乱数は、モンテカルロ積分等の用途で収束を早めるために用いられている。準乱数には規則性があるが、疑似乱数と比較してより一様な密度で分布した点座標群を生成でき、ヒストグラムの計算結果がより安定する (図3)。

生成された点の中から、任意に2点 $\mathbf{P}_1$, $\mathbf{P}_2$ を選び、式 (2) により距離を求める。点の総数が n のとき、点対の総数は $n(n-1)/2$ となるが、これらすべてに対して距離 D_p を求める。

$$D_p(\mathbf{P}_1, \mathbf{P}_2) = \sqrt{\|\mathbf{P}_1 - \mathbf{P}_2\|^2} \quad (2)$$

生成された数値群より、距離ヒストグラムを生成するにあたり、モデルの大きさなどによる差異を吸収するため、ヒストグラムの正規化を行う必要がある。Osadaらは、いくつかの正規化手法を実験しているが、その中でコストと性能の両面で、最も良い結果を与えている平均値による正規化を用いた。平均値によるヒストグラムの正規化を行うには、まず、数値群の最小値、最大値、および平均値を求める。次にヒストグラムの区間数 N を基に、平均値から最小値、平均値から最大値の二つの区間をそれぞれ $N/2$ 区間

図4 距離ヒストグラム生成
Creating a histogram of the point-pair distance.

図5 BunnyのmD2ヒストグラム (区間数512)
An example of the mD2 histogram (512 intervals) for the bunny model.

に分割し、それぞれの区間における点対数の頻度ヒストグラムを計算する (図4)。区間数512のヒストグラムの例を図5に示す。

3.2 SVMによる分類

3.2.1 SVM (Support Vector Machine)

SVM (Support Vector Machine) は、パターン認識のための学習型類別アルゴリズムである^{12)~14) 16) 17)}。学習パターンとして、 n 個のベクトル $\mathbf{x}_i$ ($i=1, \dots, n$) が与えられたとき、それらを正例と負例に分け、正例と負例の間の距離 $\|\omega\|$ が、最大になるような超平面を最適解として求める。この距離 $\|\omega\|$ の最大化がSVMの大きな特徴である。学習パターンが線形分離可能な場合、正負例間の距離 $\|\omega\|$ が最大になる識別関数は式 (3) を用いて表せる。

$$f(\mathbf{x}) \equiv \text{sign}(g(\mathbf{x})) = \text{sign}(\omega \cdot \mathbf{x} + b) \quad (3)$$

ここで n 個の学習パターン $\mathbf{x}_i$ が満たすべき条件を式 (4) とする。これは平面 $g(\mathbf{x}) = \pm 1$ の間に学習パターンが存在しないことを示している。

$$\forall i, g(\mathbf{x}) = \omega \mathbf{x}_i + b \begin{cases} \geq +1 & \mathbf{x}_i \in \text{正例} \\ \leq -1 & \mathbf{x}_i \in \text{負例} \end{cases} \quad (4)$$

$\mathbf{x}_i$ が正例と負例のどちらに属するかを変数 y_i 式 (5) を用いて示すと、式 (4) は、式 (6) のように表せる。

$$y_i = \begin{cases} +1 & \mathbf{x}_i \in \text{正例} \\ -1 & \mathbf{x}_i \in \text{負例} \end{cases} \quad (5)$$

$$\forall i, y_i \cdot (\omega \cdot \mathbf{x}_i + b) - 1 \geq 0, i = 1, \dots, n \quad (6)$$

この式 (6) を満たす ω, b のうちで、

$$L(\omega) = \frac{1}{2 \cdot \|\omega\|^2} \quad (7)$$

を最小にするものが分離超平面と呼ばれる最適解である。この最小化問題には、さまざまな解法 (Lagrange 未定乗数法, 共役勾配法など) が存在する。

学習パターンの線形分離が不可能な場合には、スカラーを出力する非線形関数 $\phi_i(\mathbf{x})$ ($i=1, \dots, d$) を,

$$\phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \dots, \phi_d(\mathbf{x}))^t \quad (8)$$

と定義する。この $\phi(\mathbf{x})$ を用いて入力ベクトル $\mathbf{x}$ を変換し、変換後の空間で線形識別をする。 $\phi(\mathbf{x})$ 空間で得られた分離超平面は、変換前の $\mathbf{x}$ 空間において非線形な分離面をなす。そのため、 $\mathbf{x}$ 空間では分離不可能な学習パターンも、 $\phi(\mathbf{x})$ 空間では分離することが可能となる。

しかし ϕ によって表される空間の次数が大きくなり、計算が複雑になりがちである。この問題は、カーネル関数を用いてベクトルの内積を置換することで解決することができる。カーネル関数には多くの種類があるが、本手法では $p=2$ の多項式型カーネル (式 (9)) を使用した。

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^p \quad (9)$$

カーネルにはこの他、Gaussianカーネルなどもあるが、予備実験の結果、どのカーネルを用いても識別能力には大きな差が見受けられなかったため、計算時間が短い多項式型カーネルを使用した。

SVMは、soft margin hyper planeを用いて汎化能力を得ているが、その性質 (大まかに言って、超平面の滑らかさ) を決めるパラメータ C を操作することで、その汎化能力を調節することができる (詳しくは文献13) を参照)。

3.2.2 学習結果による距離の修正

本手法では、SVMに与える N 次元特徴ベクトルとして、3.1節で作成した距離ヒストグラム $mD2$ を使用し、 N 次元空間に浮かぶ点 (特徴ベクトル) をSVMで類別する。指示は、人間が検索モデルと取出されたモデル (複数) との相違度を2値で判定した結果のタグ (+1: 似ている, -1: 似ていない) を付与することで行う。

SVMを形状相違度の順序付けに使用する場合、分離超平面からの距離により分類をする方法が考えられる。分離超平面からの距離による分類はEladらが用いた²⁾が、本手法ではモデル比較の際、SVMでモデルを分類した結果を用いてManhattan距離を修正し、修正後の値が小さいものを類似なモデルであるとした。

モデル対の形状の比較は、モデルごとに作成した特徴ベクトルである距離ヒストグラム間のManhattan距離を計算して行う。ここでManhattan距離を用いたのは、Osadaらの実験結果では、Euclid距離よりもManhattan距離のほうが良い検索結果が得られたためである⁴⁾。

$\mathbf{x}, \mathbf{y}$ を任意の2モデルの特徴ベクトルとすると、これら2モデルの間の距離にSVMによる修正を加えた後の距離

$D(\mathbf{x}, \mathbf{y})$ は式 (10) により求められる。

$$D(\mathbf{x}, \mathbf{y}) = \alpha \cdot \sum_{i=1}^N |x_i - y_i| \quad (10)$$

ここで、前述のように、 N はヒストグラムの区間数である。インタラクティブな反復の第1回目は、SVMによる類別はできないので $\alpha=1.0$ として、Manhattan距離のみで得られた相違度で検索を行い、これを第1候補群としてユーザに対し提示する。反復の2回目以後は、SVMの類別結果により、 α の値を切り替えた。今回の実験では、SVMの分類結果に基づき、もし似ているなら $\alpha=0.5$ を、似ていないなら $\alpha=2.5$ を用いた。つまり、SVMの判定が「似ている」ならManhattan距離の和の0.5倍が、「似ていない」ならManhattan距離の2.5倍が、それぞれ相違度 $D(\mathbf{x}, \mathbf{y})$ の値となる。 α の値は、数個の値の中から最も良い検索結果が得られた値を選んだ。真に最適な α の値の選択は困難である。「最適」な α の値は、SVMのパラメータ (用いるカーネルにもよるが数個ある) や特徴ベクトルの次数 N 等、多くのパラメータに依存するためである。

4. 実験と評価

SVMは、Joachimsらのアルゴリズム $SVM^{Light6)}$ をRupingらがC++で実装した $mySVM^{6)}$ を用いた。検索システムは3次元モデリングと表示のツールを集めた $gmtools$ を元にC++で実装した。実験には、VRML (Virtual Reality Modeling Language) モデル269個からなるデータベースを用いて行った。

4.1 特徴ベクトルのパラメータの選択

本手法では、SVMに対し、インタラクティブな反復検索セッションの初回の検索結果 (第1候補群) に対しユーザがタグ付けした結果で指示を行う。したがって第1候補群にある程度の数の正例 (欲しい形状モデル) が含まれないと指示がうまく行かない。第2回以後の反復においても、検索に用いる距離 D はManhattan距離に基づいており、特徴ベクトルとその「機械的」な比較法の性能が検索性能を左右する。したがって、できる限り良い性能の良い特徴ベクトルを選択することが重要である。

そこでまず、本手法で用いた特徴ベクトルである $mD2$ の二つパラメータ、すなわち、① $mD2$ のヒストグラムの区間数 (つまり特徴ベクトルの次元数)、② $mD2$ を計算するために配置する点の数、を決定する実験を行った。この実験の結果、最良の成績を収めたパラメータの組を以後の実験に用いた。

実験する条件としては、区間数が128, 256, 512の3種類、また表面上に生成する点の総数が256, 512, 1,024の3種類、合計9通りの組合せで実験した。各条件での識別能力は以下のようにして求めた⁴⁾。

- (1) 検索対象となるモデルを決定
- (2) 検索対象と269個のモデルそれぞれを比較
- (3) 検索結果から相違度が低い順に (k-1) 個のモデルを抜

表1 mD2特徴ベクトルの識別能力
Classification performance of the mD2 feature vector..

点 区間	256	512	1024
128	40.6%	51.1%	45.6%
256	39.4%	51.7%	45.6%
512	42.2%	52.8%	46.7%

図6 類似検索で使したモデル
Examples of models queried.

き出す．ここでkとは検索対象が属するカテゴリー内のモデル数を表す

- (4) 検索対象と同じカテゴリーに属するモデルが(3)で抜き出したモデル中に存在する割合を測定．この割合を識別能力とする

検索対象として，ウサギ(10モデル)，飛行機(11モデル)の計21モデルを使用し，それぞれのモデルに対する識別能力の平均値を総合的な識別能力とした．各条件下の識別能力を表1に示す．

表1結果から，区間数512，生成する点の総数512の場合に識別能力が最も高い．これは生成する点の数が，256ではモデルの幾何形状を捉えるためには不足しており，1,024では点の数に比べ区間数が多すぎてノイズの影響が出てくるためと考えられる．

これ以後の実験では区間数512，生成する点の総数512でヒストグラムを作成する．

4.2 形状類似検索

この実験では，図6に示したモデルを検索する．SVMを使用した場合と使用しなかった場合，およびSVMの学習過程における検索結果の相違を見た．実験の手順は以下に示す．

- (1) 検索対象となるモデルを決定
- (2) 検索対象と269個のモデルをそれぞれ比較
- (3) 相違度が低い上位15モデル(検索対象モデルを含む)を選出
- (4) (3)のモデルからSVMの学習データを作成
- (5) (2)～(4)の作業を2度繰り返す

まず，SVMを使用せずに図6(a)の検索モデルとデータベース中の各モデルとの相違度を求めて検索を行った．その結果が図7である．図中でモデル左下側の数字は各順位

図7 「飛行機」のモデルをManhattan距離のみで検索
Querying "airplane" models; the initial result using the Manhattan distance only (i.e., no relevance feedback.).

図8 「飛行機」のモデルの検索
上位15位をタグ付けし1回学習させた検索結果
Querying "airplane" models; after one relevance feedback step.
Top 15 results are tagged.

を示し，各順位の右の「()」はそのモデルにあたえられているインデックス番号を示す．以後，各モデルはインデックスを用いて表す．また，各モデル右上の「+」「-」はそのモデルを教示データとしてSVMにあたえる際に付与されたタグを，また「()」の付いているものはすでにSVMの学習データとして選出済みであることを示す．

図7では，検索対象自身(順位は1となった)とモデル {54, 197, 198, 241, 189, 254} を似ているモデルとして「+」のタグを付け，モデル {200, 265, 175, 46, 220, 131, 177, 178} を似ていないモデルとして「-」のタグを付けた．このタグ付けしたモデル群を学習データとしてSVMに与え，Manhattan距離に修正を加えた後に検索した結果を図8に示す．さらに，図8において未だタグ付けされていないモデルを対象として，2度目のタグ付けと学習を行い，その後に検索した結果が図9である．2度目の教示では，似ているモデルが {52, 224}，似ていないモデルが {69, 150, 95, 204, 85, 97} である．図9は，1回目，2度目を合わせて，正例負例合計で23個のモデルにタグ付けをした結果に基づく結果

図9 「飛行機」のモデルの検索
上位15位をタグ付けし2回学習させた検索結果
Querying "airplane" models; after two relevance feedback steps.
Top 15 results are tagged.

図11 「飛行機」のモデルの検索
上位30位をタグ付けし2回学習させた検索結果
Querying "airplane" models by tagging top 30 (instead of 15) results.
After two relevance feedback steps.

図10 上位15モデルを対象としたタグ付けでは検索結果上位15位に無かった「飛行機らしい」モデル

These two "airplane"-like models did not appear in the top 15 retrievals of the final result, when the relevance feedback is done using the top 15 retrievals.

である。

実験で使ったデータベースには、検索モデル一つを含めて「飛行機」のモデルが11個存在した。2回の教示を経た図9では、検索モデルを除いた全10個のうち8個が類似モデルとして結果の1～8位に選出された。初回の検索結果では上位15個に含まれていなかった {52, 224} の二つのモデルが、1回の教示の後に15位以内にまで上昇したことが注目される。

この例では、SVMによる学習および、SVMの分類結果による距離の修正がある程度有効に働いており、ユーザの意図がある程度反映できた。しかし「いわゆる」飛行機のカテゴリに属しているモデルの内、図10に示す二つは、2度の学習後も15位以内表れなかった。図10 (a) のモデルの形状が検索モデルやタグ付けされたモデルに近いことは議論の分かれるところである。しかし、図10 (b) のモデルは検索できても良さそうである。図10 (b) のモデルがうまく検索できなかった一つの原因は、上位15モデルのみをタグ付けしたことにあると考えられる。

より広い範囲のモデルを対象にタグ付けを行えば、ある程度の検索結果の改善が期待されるため、今度は上位30モ

デルをタグ付けして、1度だけ学習した後に検索を行った結果を図11に示す。図11に示す結果では、上位15モデルのタグ付けでは、2度の学習後も15位以内には現れなかったモデル {45} が9位に現れている (図10 (b))。タグ付けの対象となるモデル数が多い場合の方が、効果的に学習できることがわかる。

図12～図14は、イルカのモデルを検索例として提示して検索を行った例である。図12はManhattan距離のみにより検索を行った結果である。さらに、図13と図14は、いずれも上位15位までをタグ付けし、それぞれ1度および2度の学習を行って検索をした結果である。図12ではモデル {27, 25, 8, 26} を似ているモデル“+”として、また {209, 285, 300, 90, 179, 299, 215, 21, 270, 130} を似ていないモデル“-”としてタグ付けした。その結果、次の検索セッションの結果である図13では新たなイルカのモデル {160} が11位に表れた。さらに、モデル {160} も似ているモデルとしてタグ付けし、検索を行ったところ、2度目の学習後の検索結果である図14では新たにモデル {159} が5位に表れた。これでデータベース中のすべてのイルカのモデルの検索に成功した。

ここで示したのは、比較的小さなデータベース (モデル数269) を対称としたたかだか二つの実験例であるが、提案手法が一定の効果を上げていることがわかる。

今後、多くの検索例を用いた数値的な性能評価が必要である。しかし、人間が介在したインタラクティブな検索手法であり、また個人ごとに求める形状モデルが変わるため、測定実験の組立てを工夫する必要がある。

4.3 検索システムの汎化能力

学習に基づく類似検索システムでは、教示例だけを正しく検索しても意味がない。教示例には含まれないにもかかわらず、検索対象モデルと似ているモデルを取出す「汎化」の能力が求められる。そこで、汎化能力を確認するための実験を行った。

ここでは、SVMにあらかじめ教示データを与えて学習さ

+	+	+	+	-
検索対象	1:(27)	2:(25)	3:(8)	4:(209)
-	-	-	-	+
5:(285)	6:(300)	7:(90)	8:(179)	9:(26)
-	-	-	-	-
10:(299)	11:(215)	12:(21)	13:(270)	14:(130)

図12 「イルカ」のモデルをManhattan距離のみで検索
Querying "dolphin" models; the initial result using the Manhattan distance only (i.e., no relevance feedback.)

+	+	+	+	-
(4)	(5)	(8)	(9)	(70)
-	-	-	-	-
(116)	(92)	(62)	(82)	(47)
-	-	-	-	-
(128)	(119)	(142)	(45)	(134)

図15 「ウサギ」を学ぶため、あらかじめシステムに与えられた学習データ

The training data given to the system to learn "bunny" models.

(+)	(+)	(+)	(+)	(+)
検索対象	1:(27)	2:(25)	3:(8)	4:(26)
-	-	-	-	-
5:(168)	6:(183)	7:(42)	8:(95)	9:(35)
-	-	-	-	-
10:(74)	11:(160)	12:(173)	13:(39)	14:(259)

図13 「イルカ」モデルの検索
上位15位をタグ付けし1回学習させた検索結果
Querying "dolphin" models; after one relevance feedback step.
Top 15 results are tagged.

(+)	(+)	(+)	(+)	(+)
検索対象	1:(27)	2:(25)	3:(8)	4:(26)
-	-	-	-	(+)
5:(159)	6:(106)	7:(47)	8:(72)	9:(160)
-	-	-	-	-
10:(48)	11:(232)	12:(36)	13:(20)	14:(272)

図14 「イルカ」モデルの検索
上位15位をタグ付けし2回学習させた検索結果
Querying "dolphin" models; after two relevance feedback steps.
Top 15 results are tagged.

図16 検索対象モデル
Queried model.

検索対象	1:(6)	2:(7)	3:(10)	4:(79)

5:(37)	6:(3)	7:(83)	8:(204)	9:(182)

10:(112)	11:(144)	12:(206)	13:(188)	14:(12)

図17 「ウサギ」の検索結果
(学習データはデータベースから除外した)
Querying "bunny" after the training.
(Training data are excluded from the queried database.)

せる。ついで、学習に用いたモデルを含まないデータベースを対象に検索を行い、汎化能力を見た。まず、「いわゆる」ウサギに似た形状のモデルを検索するための教示データをあらかじめ15個作成した(図15)。教示データには四つの正例と11個の負例が含まれている。次に図16に示すモデルを検索対象モデルとして検索を行った。その結果が図17である。

実験で使ったデータベース中(教示に使ったモデルを除いた後)には、検索モデル以外にウサギのモデルが5個存在する。図17を見ると、そのうちの4個が上位15位以内に選出されており、本手法は一定の汎化能力を持つことがわかった。

しかし、残る一つのモデルは15位以内に表れなかった。これには主に二つの原因が考えられる。原因の一つ目は、mD2特徴ベクトルの性質と性能の限界である。D2特徴ベクトルは、Osadaらも論文で述べているように⁴⁾、3次元モデルを素早く大まかに分類することを目的としている。逆に言うと、形状の微細な相違に対する感受性が低く、また異なるモデルにもかかわらず、類似した特徴ベクトルを与えることもある。この問題に対する解決法としては、mD2とそれ以外の特徴ベクトルを組合せる、あるいはまったく新たなより性能の高い特徴ベクトルを用いる、などが考えられる。原因の二つ目は、SVMのカーネルやパラメータ値が最適でない可能性である。旧来のニューラルネットワークでは、その構成により識別能力が大きく影響される。SVMは旧来のニューラルネットワークに比べ、ネットワークの構成を意識しなくて良いなど、はるかに使いやすい。しかし、カーネルの種類やそのパラメータ(例えば、多項式カーネルの次元)、誤認識と一般化能力のトレードオフをするパラメータ、など、いくつか調整すべき点がある。最適なカーネルやパラメータ値を見つけると能力が向上する可能性がある。

5. まとめと今後の課題

本論文では、ユーザごとに異なる形状類似性の判断を反映して検索を行う、3次元モデルの形状類似検索手法を提案した。本検索手法の特徴は、検索と検索結果に基づくユーザからの教示を反復して行うことで、数回の反復の後に、ユーザの形状類似性判断を反映した3次元モデルが取出せることである。ユーザの好みを学習するために提案したシステムでは、SVM(Support Vector Machine)と呼ばれる学習型類別アルゴリズムを用いた。形状特徴ベクトルとしては、3次元モデルの向きや位置に依存せず、計算が容易で、かつ形状モデルの位相的あるいは幾何的縮退にも強いOsadaらがD2と呼ぶ形状特徴⁴⁾を用いた。実験の結果、学習を用いた場合、学習なしの場合に比べ、ユーザの求める形状モデルがより多く含まれた検索結果が得られることがわかった。

今後の課題は数多い。まず、SVMの分離超平面から特徴ベクトルへの距離を、候補の順位付けに用いる手法を実装し評価したい。また、D2とは異なる性質を持つ形状特徴ベクトル単体や、D2も含めた複数の形状特徴ベクトルとSVMとを組合せ、検索性能を高めることも実験したい。さらに、より大掛かりな実験を行い、数値的な性能指標を求め、提案手法の有効性をより客観的に評価する必要がある。

また、本システムは基本的にインタラクティブなシステ

ムである。したがって、類似検索アルゴリズムそのものの性能を上げると共に、検索のユーザインタフェースを工夫するなどしてシステム全体の性能を上げることも必要と考えられる。

最後に、本論文の実験を行った類似検索システムの基盤となったgmtoolsを提供していただいた東京大学の高橋成雄助教授に感謝する。

〔文 献〕

- 1) C. Burges: "A Tutorial on Support Vector Machines for Pattern Recognition", Data Mining and Knowledge Discovery, 2, pp.1-47 (1998)
- 2) M. Elad, A. Tal, S. Ar: "Directed Search in A 3D Objects Database Using SVM", HP Laboratories Israel Technical Report, HPL-2000-20 (R.1) (Aug. 2000)
- 3) M. Novotni, R. Klein: "A Geometric Approach to 3D Object Comparison", Proc. Int'l Conf. on Shape Modeling and Applications 2001, Genova, Italy, pp.167-175 (May 2001)
- 4) R. Osada, T. Funkhouser, Bernard Chazelle, David Dobkin: "Shape Distributions", ACM TOGS, 21, 4, pp.807-832 (Oct. 2002)
- 5) M. Hilaga, Y. Shinagawa, T. Kohmura, and T. Kunii: "Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes", Proc. SIGGRAPH 2001, Los Angeles, USA, pp.203-212 (2001)
- 6) T. Joachims: "Making Large Scale SVM Learning Practical, in Advances in Kernel Methods", - Support Vector Learning, B. Schölkopf, et al. Eds., MIT Press (1999)
- 7) E. Paquet and M. Rioux: "Nefertiti: a Query by Content Software for Three-Dimensional Databases Management", Proc. Int'l Conf. on Recent Advances in 3-D Digital Imaging and Modeling, Ottawa, Canada, 12-15, pp.345-352 (May 1997)
- 8) E. Paquet, A. Murching, T. Naveen, A. Tabatabai, M. Roux: "Description of Shape Information for 2-D and 3-D Objects", Signal Processing : Image Communication, 16, pp.103-122 (2000)
- 9) M. T. Suzuki, T. Kato, H. Tsukune: "3D Object Retrieval based on Subject Measures, Proc. 9th Int'l Conf. and Workshop on Database and Expert Systems Applications (DEXA98), IEEE-PR08353, Vienna, Austria, pp.850-856 (Aug. 1998)
- 10) M. T. Suzuki, T. Kato, N. Otsu: "A Similarity Retrieval of 3D Polygonal Models Using Rotation Invariant Shape Descriptors", Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC2000), Nashville, Tennessee, pp.2946-2952 (2000)
- 11) 鈴木一史, 加藤俊一, 築根秀男: "主観的類似度を反映した3次元多面体の検索", 第3回知能情報メディアシンポジウム, pp.9-16 (December 1997)
- 12) V. N. Vapnik: "Statistical Learning Theory", Wiley (1998)
- 13) V. N. Vapnik: "The Nature of Statistical Learning Theory", Second Edition. Springer (1999)
- 14) R. C. Veltkamp: "Shape Matching : Similarity Measures and Algorithms, Invited talk", Proc. Int'l Conf. on Shape Modeling and Applications 2001, pp.188-197, Genova, Italy (May 2001)
- 15) D. V. Vrani, D. Saupe, and J. Richter: "Tools for 3D-object Retrieval : Karhunen-Loeve Transform and Spherical Harmonics", Proc. of the IEEE 2001 Workshop Multimedia Signal Processing, Cannes, France, pp.293-298 (Oct. 2001)
- 16) S. Rüping: "mySVM-Manual", <http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM/> (Oct 2000)
- 17) 前田英作: "痛快! サポートベクトルマシン", 情処学誌, 7, pp.676-683 (July 2001)
- 18) T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dokin, D. Jacobs: "A Search Engine for 3D Models, ACM TOGS, 22, 1, pp.83-105 (Jan. 2003)
- 19) Takeo Igarashi, Hidehiko Tanaka, Satoshi Matsuoka: "Teddy ; A Sketching Interface for 3D Freeform Design, Proc. SIGGRAPH '99, pp.409-416 (1999)
- 20) D. Keim: "Efficient Geometry-based Similarity Search of 3D Spatial Databases", Proc. ACM SIGMOD Int. Conf. On Management on Data, Philadelphia, PA, pp.419-430 (1999)

- 21) R. C. Veltkamp: "Shape Matching ; Similarity Measures and Algorithms, Invited Talk", Proc. Shape Modeling and Applications 2001, Genova, Italy, pp.188-197 (May 2001)
- 22) M. E. Mortensen: "Geometric Modeling", Second Edition, John Wiley & Sons, Inc. (1997)
- 23) J. Corney, H. Rea, D. Clark, J. Pritchard, M. Breaks, R. MacLeod: "Coarse Filters for Shape Matching", IEEE Computer Graphics and Applications 2002, pp.65-74 (2002)
- 24) D. McWherter, M. Peabody, W.C. Regli, A. Shokoufandeh: "Transformation Invariant Shape Similarity Comparison of Solid", 2001 ASME Design Engineering Technical Conferences & Computers and Information in Engineering Conference 2001, Pittsburgh, Pennsylvania, pp.9-12 (September 2001)
- 25) S. Mukai, S. Furukawa, M. Kuroda: "An Algorithm for Deciding Similarities of 3-D Objects", Proc. ACM Symposium on Solid Modeling and Applications 2002, Saarbrücken, Germany (June 2002)
- 26) 小田切智, 衣嶋昌俊, 武井 毅, 大淵竜太郎: "慣性モーメントの包絡線を用いた3次元モデルの形状類似検索", 映情学誌, 56, 10, pp.1589-1597 (Oct. 2000)
- 27) T. Zaharia, F. Prêteux: "Three-Dimensional Shape-Based Retrieval within the MPEG-7 Framework", Proc. SPIE Conference 4304 on Nonlinear Image Processing and Pattern Analysis XII, San Jose, CA, pp.133-145 (Jan. 2001)
- 28) T. Zaharia, F. Prêteux: "Shape-Based Retrieval of 3D Mesh Models", Proc. IEEE ICME 2002, Lausanne, Switzerland (Aug. 2002)
- 29) William H. Press et al.: "Numerical Recipes in C-The Art of Scientific Computing", Second Edition, Chapter 7.7 : Quasi- (that is, Sub-) Random Sequence, Cambridge Universe Press, Cambridge, UK, pp.309-315 (1992)

いばと 昌俊 2002年, 山梨大学工学部電子情報工学科卒業. 同年, 茨城計算センター(株)入社. 現在に至る.

おだぎり 智 2000年, 山梨大学工学部電子情報工学科卒業. 同年, 同大学工学部研究生として在籍. 2001年, 同大学大学院工学研究科博士前期課程入学. 3次元モデルの形状類似検索に関する研究に従事. 2003年, 山梨大学大学院工学研究科博士前期課程修了. 同年, (株) コンピュータマインド入社.

おおぶりゅうたろう 大淵竜太郎 1981年, 上智大学理工学部電気電子工学科卒業. 1983年, 電気通信大学大学院計算機科学科修士課程修了. 同年, 日本アイ・ビー・エム(株)入社. 1994年, University of North Carolina at Chapel Hill Computer Science DepartmentよりPh. D. 取得. 同年, 日本アイ・ビー・エム(株)東京基礎研究所勤務. 1999年より, 山梨大学工学部コンピュータ・メディア工学科助教授. 3次元グラフィクス, 形状モデリング, 拡張現実システム等に興味を持つ.