

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¹, 小田切 智², 大淵 竜太郎³

¹Masatoshi Ibato, ²Tomo Otagiri, ³Ryutarou Ohbuchi

zaq54940@cat.zero.ad.jp, k6052@kki.yamanashi.ac.jp, ohbuchi@acm.org

¹茨城計算センター株式会社

Ibaraki Computing Center, Incorporated

² 山梨大学大学院工学研究科 (山梨県甲府市武田 4-3-11)

Graduate School of Engineering, Yamanashi University (4-3-11 Kofu, Yamanashi, Japan)

³ 山梨大学コンピュータ・メディア工学科 (山梨県甲府市武田 4-3-11)

Department of Computer Science and Media Engineering, Yamanashi University (4-3-11 Kofu, Yamanashi, Japan)

Abstract: As popularity of three-dimensional (3D) geometric models increase, interests in methods to search 3D models based on their shape similarity have increased. Two of the most difficult issues in the shape similarity search of 3D geometric models are a feature vector that succinctly describes the shape of the model and a method to compute distance among a pair of feature vectors that reflects the user's preference. In this paper, we propose a human-directed 3D shape similarity search method that reflects, to some extent, the user's preference by using a learning classifier. Given an example 3D mesh model, the system first retrieves, as the initial guess, a set of models that are similar to the query based only on the unbiased mechanical measure of distance (*i.e.*, Manhattan distance) among feature vectors. The user then iteratively refines the query by tagging a subset of the retrieved models as either similar or dissimilar. The system learns the user's preference using a learning classifier *Support Vector Machine* (SVM) so that the distance values among feature vectors are altered to reflect the user's preference. Experimental results show that our method is capable of retrieving 3D models that better reflects the preferences of the user than the simple method using Manhattan distance only.

要旨: 3次元(3D)モデルが普及するにつれ、これらのモデルの形状に基づく類似検索手法への関心が高まりつつある。3次元モデルの形状類似検索においておそらく最も困難な2つの課題は、形状特徴を簡潔でかつ、曖昧性無く表現した特徴ベクトルと、場合によってはユーザごとに異なる形状類似性の判断を反映した特徴ベクトル間距離の尺度である。本論文では、ユーザの形状類似性の判断を反映した類似検索を目指し、学習型の類別アルゴリズムを用いたインタラクティブ教示型の3次元モデル類似検索手法を提案する。ユーザがシステムに対し、検索を行いたい3次元モデルを1つ例示すると、システムは、特徴ベクトル間のManhattan距離をそのまま用いて類似検索した3次元モデル群を初回候補群としてユーザに提示する。ユーザは、これらシステムが検索し提示したモデル(の一部)それぞれに、そのモデルが求めるモデルに類似すると判断するか否かを示すタグを付ける。システムは、これらのタグを元に、SVM (Support Vector Machine) という学習型類別アルゴリズムを用いてユーザの判断を学習し、学習後の類別アルゴリズムの類別結果を用いて距離値を調整して、得た新たな候補群を検索し提示する。この検索・教示・学習の組を数回繰り返すことで、そのユーザの欲しいモデルが検索結果の上位に現れるようになる。実験の結果、我々の試作システムを用いて得られた検索結果はManhattan距離のみを用いた結果に比べユーザの類似度判定を反映した結果となることがわかった。

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

衣嶋 昌俊¹, 小田切 智², 大淵 竜太郎³

zaq54940@cat.zero.ad.jp, k6052@kki.yamanashi.ac.jp, ohbuchi@acm.org

¹茨城計算センター株式会社, ²山梨大学大学院工学研究科, ³山梨大学コンピュータ・メディア工学科

山梨県甲府市武田 4-3-11

1 はじめに

近年, コンピュータ技術の発展とともに 3 次元グラフィクスや 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ともに, 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, 3, 4, 5, 7, 9, 10, 11, 15, 18, 20, 23, 24, 25, 26, 27, 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有り, 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似アルゴリズムの類似検索結果で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. 学習型類似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, 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, 次回の検索において学習した SVM の類似検索結果を元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, 学習無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, 次節でこれまでの 3 次元形状モデルの類似検索手法を概観する. ついで, 第 3 節で我々の検索手法について述べ, 第 4 節で評価実験とその結果を, それぞれ述べる. さらに第 5 節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 5 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) 索引と取り出し: 形状相違度に基づき, 求めるモデルまたはモデル群を効率良く取り出すか.

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(の組み合わせ)としてスケッチするもの[18]、さらには Igarashi らの Teddy[19]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殆ど不可能であったりする。検索の為に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無い[2, 3, 4, 7, 9, 10, 11, 15, 18, 20, 23, 24, 25, 26, 27, 28]。その難しさの1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無い為である。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「ポリゴンスープ」である。ポリゴンスープとは、(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」のように集まってきている。ポリゴンスープ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、幾何学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ー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]。

ポリゴンスープで定義された、「性質の悪い」3次元形状の場合、幾何要素の接続性が十分でなかったり、複数の単連結要素があったり、立体で無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2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]、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, 10, 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)を取り出すことになる。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1つである鈴木らの研究[9, 10, 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[23]の研究では、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[2]の研究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。

索引と取り出し: 3次元モデルの検索と検索された形状モデルの取り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

本論文中で提案する 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・取り出す。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、ポリゴンスラブ等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 D2[4]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 3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 3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、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。まず、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、(機械的に計算された)相違度が低い幾つかのモデルをデータベースから取り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を教示する。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。次回の検索では、その相違度計

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元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。

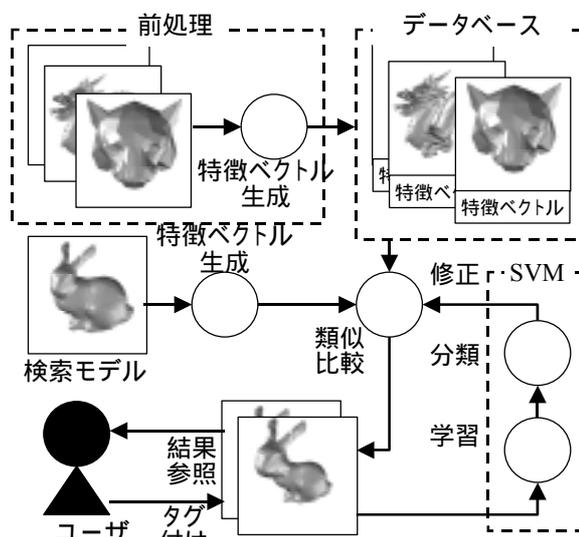


図1. 検索手法の概要

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 3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に点を生成する。その点の中からランダムに選ばれた 2 点対の間の距離を全ての 2 点対について計算する。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 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 A, B, 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(1)式の 2 つの数 r_1, r_2 には疑似乱数

を使用しているが、我々の mD2 では準乱数のひとつである Sobol [30]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、擬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)。

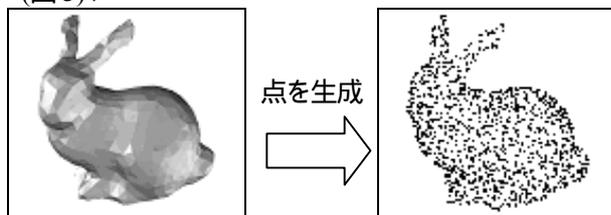


図2. モデルの表面上に点を生成

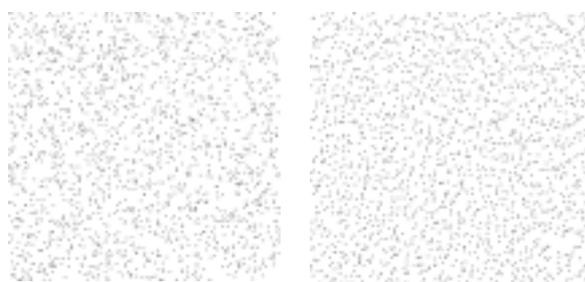


図3. 擬似乱数(左)とSobolの準乱数(右)により点を発生した結果。

生成された点の中から任意に2点 P_1, 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、これら全てに対して距離 D_p を求める。

$$D_p(P_1, P_2) = \sqrt{|p_1 - p_2|^2} \quad (2)$$

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、及び平均値を求める。次にヒストグラムの区間数 N を元に、平均値から最小値、平均値から最大値の2つの区間をそれぞれ $N/2$ 区間に分割し、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示す。

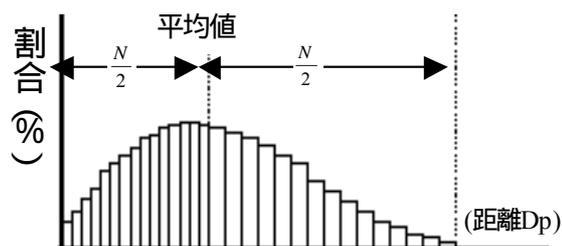


図4 距離ヒストグラム生成

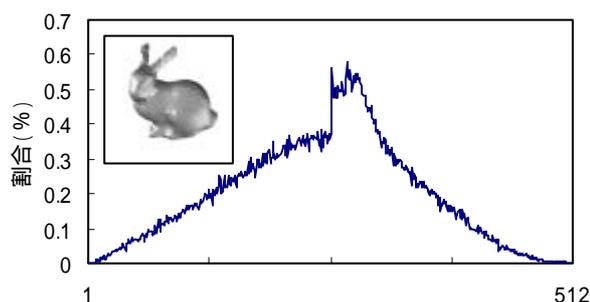


図5 BunnyのmD2ヒストグラム(区間数512)

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, 13, 14, 16, 17]. 学習パターンとして n 個のベクトル $x_i (i = 1, \dots, n)$ 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、正負例間の距離が最大になる識別関数は式(3)を用いて表せる。

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

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

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

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。

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

$$\forall i, y_i \cdot (\omega \cdot x_i + b) - 1 \geq 0, i = 1, \dots, n \quad (6)$$

この式(6)を満たす 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様々な解法(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 $\phi_i(x) (i = 1, \dots, d)$ を

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

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

しかし x はベクトルであるため、 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。

$$K(x_i, x_j) = (x_i \cdot x_j + 1)^p \quad (9)$$

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[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、その空間に浮かぶ点(特徴ベクトル)を SVM で類別する。教示は、人間が検索モデルと取り出されたモデル(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]。

x, 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 $D(x, y)$ は式(10)により求められる。

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

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 の分類結果が似ている場合は $\alpha = 0.5$ を用い Manhattan 距離よりも D を小さく、似ていない場合は $\alpha = 2.5$ を用い Manhattan 距離よりも D を大きくした。 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^{Light} [6] を Ruping らが C++ で実装した mySVM [16] 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 gmttools 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教示を行う。従って第 1 候補群にある程度の数の正例(欲しい形状モデ

ル)が含まれないと教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, その性能が検索性能を左右する. 従って, 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 2 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, 1024 の 3 種類, 述べ 9 通りの組み合わせで実験した. 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)個のモデルを抜き出す. ここで k とは検索対象が属するカテゴリ内のモデル数をあらわす.
- 4) 検索対象と同じカテゴリに属するモデルが 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, 1024 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.

表 1. mD2 特徴ベクトルの識別能力

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

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, 及び 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), 3), 4)の作業を 2 度繰り返す.

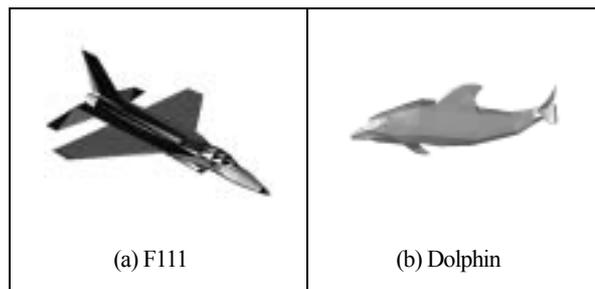


図6. 類似検索で使用したモデル

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 9 である. 図中でモデル左下側の数字は各順位を示し, 各順位の右の「()」はそのモデルにあて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 にあてえる際に付与されたタグを, また「()」の付いているものは既に SVM の学習データとして選出済みであることを示す.

図 9 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(+1)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(-1)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 10 に示す. さらに, 図 10 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 11 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 11 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である.

実験で使用したデータベースには, 検索モデル 1 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 11 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 2 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 による学習及び、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 7 に示す 2 つは、2 度の学習後も 15 位以内表れなかった。図 7(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かは議論の分かれるところである。しかし、図 7(b)のモデルは検索できても良さそうである。図 7(b)のモデルがうまく検索できなかった 1 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 12 に示す。図 12 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 7(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 13, 図 14, 図 15 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 13 は Manhattan 距離のみにより検索を行った結果である。さらに、図 14 と図 15 は、いずれも上位 15 位までをタグ付けし、それぞれ 1 度及び 2 度の学習を行って検索をした結果である。図 14 ではモデル{27, 25, 8, 26}を似ているモデル(+1)として、また{209, 285, 300, 90, 179, 299, 215, 21, 270, 130}を似ていないモデル(-1)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 14 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 15 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全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)を対称とした高々 2 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、測定実験の組み立てを工夫する必要がある。

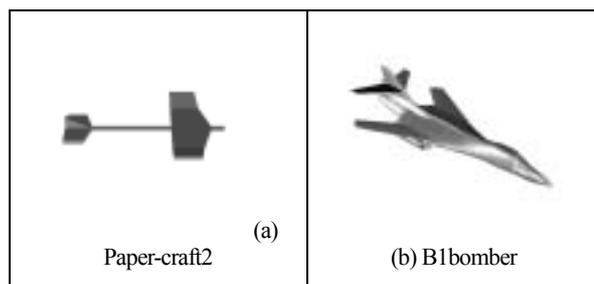


図7. 上位15位モデルを対象としたタグ付けでは検索結果上位15位に無かった「飛行機らしい」モデル

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無い。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り出す「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 16)。教示データには 4 つの正例と 11 個の負例が含まれている。次に図 8 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。

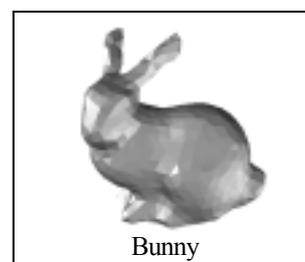


図8. 検索対象モデル

実験で使用したデータベース中(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 1 つのモデルは 15 位以内に表れなかった。これには主に 2 つの原因が考えられる。原因の 1 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み合わせる、あるいは全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2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次元形状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より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次元形状モデルが取り出せることである。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)と呼ばれる学習型分類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、学習無しの場合に比べ、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も含めた複数の計上特徴ベクトルとSVMとを組み合わせ、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。従って、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。

6 謝辞

本論文の実験を行う上で、類似検索システムの基盤となった gmttools 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。

7 参考文献

- [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), August, 2000.
- [3] M. Novotni, R. Klein. A Geometric Approach to 3D Object Comparison. Proc. *Int'l Conf. on Shape Modeling and Applications 2001*, pp. 167-175, Genova, Italy, May, 2001.
- [4] R. Osada, T. Funkhouser, Bernard Chazelle, and David Dobkin Shape Distributions, *ACM TOGS*, 21(4), pp. 807-832, (October 2002).
- [5] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. Kunii. Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes. Proc. *SIGGRAPH 2001*, pp. 203-212, Los Angeles, USA. 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*, pp. 345-352, Ottawa, Canada, May 12-15, 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*, 16:103-122, 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)*, pp. 850-856, IEEE-PR08353, Vienna, Austria, Aug. 1998.
- [10] M. T. Suzuki, T. Kato, N. Otsu. A similarity retrieval of 3D polygonal models using rotation invariant shape descriptors. *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, Dec. 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. Vranić, 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, October 2001.
- [16] S. Ruping. mySVM-Manual. <http://www-ai.cs.uni->

- dortmund.de/SOFTWARE/MYSVM/. October 2000.
- [17] 前田英作, 痛快! サポートベクトルマシン, 情報処理 2001 年 7 月号 pp.676-683, 2001 年 7 月.
 - [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, (January, 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 SICMOD Int. Conf. On Management on Data*, pp. 419-430, Philadelphia, PA., 1999.
 - [21] 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.
 - [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.
 - [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 September 9-12, 2001, Pittsburgh, Pennsylvania.
 - [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, 衣鳩 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, 映像情報メディア学会誌, Vol. 56, No. 10, pp. 1589-1597, 2002 年 10 月. (映像情報メディア学会).
 - [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, January 2001, pp. 133-145.
 - [28] T. Zaharia, F. Prêteux, Shape-based retrieval of 3D mesh models, Proc. IEEE ICME 2002, Lausanne, Switzerland, August 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, 1992, pp. 309-315.

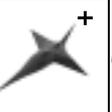
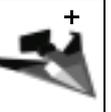
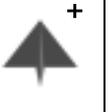
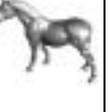
 +		 +	 +	 +
検索対象	1:(200)	2:(54)	3:(197)	4:(198)
 +				 +
5:(241)	6:(265)	7:(175)	8:(46)	9:(189)
 +				
10:(254)	11:(220)	12:(131)	13:(177)	14:(178)

図9. 「飛行機」のモデルをManhattan距離のみで検索

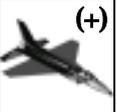
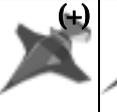
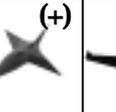
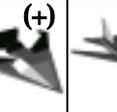
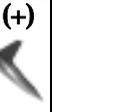
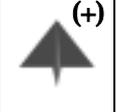
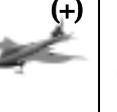
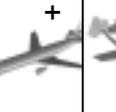
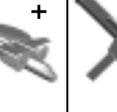
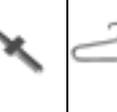
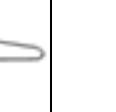
 (+)	 (+)	 (+)	 (+)	 (+)
検索対象	1:(54)	2:(197)	3:(198)	4:(241)
 (+)	 (+)	 +	 +	
5:(189)	6:(254)	7:(52)	8:(224)	9:(69)
				
10:(150)	11:(95)	12:(204)	13:(85)	14:(97)

図10. 上位15位をタグ付けし1回学習させた検索結果

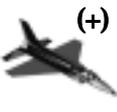
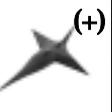
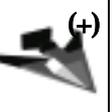
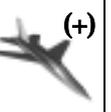
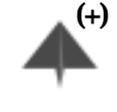
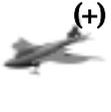
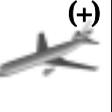
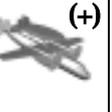
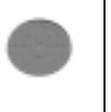
 (+)	 (+)	 (+)	 (+)	 (+)
検索対象	1:(54)	2:(197)	3:(198)	4:(241)
 (+)	 (+)	 (+)	 (+)	
5:(189)	6:(254)	7:(52)	8:(224)	9:(69)
				
10:(98)	11:(122)	12:(25)	13:(200)	14:(127)

図11. 上位15位をタグ付けし2回学習させた検索結果

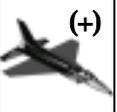
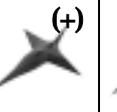
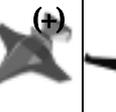
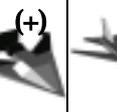
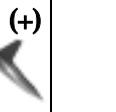
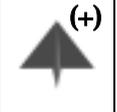
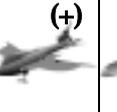
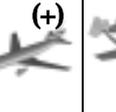
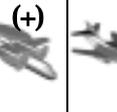
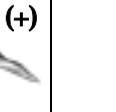
 (+)	 (+)	 (+)	 (+)	 (+)
検索対象	1:(197)	2:(54)	3:(198)	4:(241)
 (+)	 (+)	 (+)	 (+)	 (+)
5:(189)	6:(254)	7:(52)	8:(224)	9:(45)
				
10:(207)	11:(229)	12:(262)	13:(251)	14:(150)

図12. 上位30位をタグ付けし2回学習させた検索結果

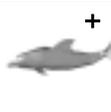
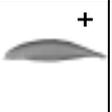
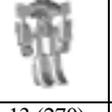
 +	 +	 +	 +	
検索対象	1:(27)	2:(25)	3:(8)	4:(209)
				 +
5:(285)	6:(300)	7:(90)	8:(179)	9:(26)
				
10:(299)	11:(215)	12:(21)	13:(270)	14:(130)

図13. 「イルカ」のモデルをManhattan距離のみで検索

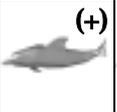
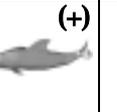
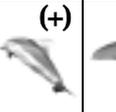
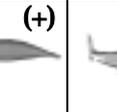
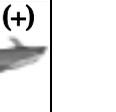
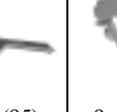
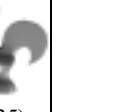
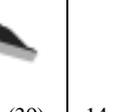
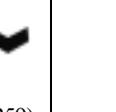
 (+)	 (+)	 (+)	 (+)	 (+)
検索対象	1:(27)	2:(25)	3:(8)	4:(26)
				
5:(168)	6:(183)	7:(42)	8:(95)	9:(35)
				
10:(74)	11:(160)	12:(173)	13:(39)	14:(259)

図14. 上位15位をタグ付けし1回学習させた検索結果

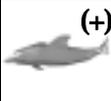
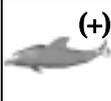
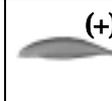
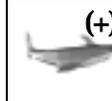
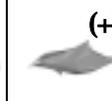
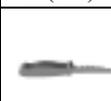
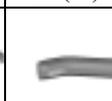
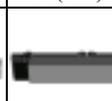
 (+)	 (+)	 (+)	 (+)	 (+)
検索対象	1:(27)	2:(25)	3:(8)	4:(26)
 5:(159)	 6:(106)	 7:(47)	 8:(72)	 9:(160) (+)
 10:(48)	 11:(232)	 12:(36)	 13:(20)	 14:(272)

図15. 上位15位をタグ付けし2回学習させた検索結果

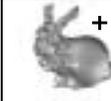
 + (4)	 + (5)	 + (8)	 + (9)	 (70)
 (116)	 (92)	 (62)	 (82)	 (47)
 (128)	 (119)	 (142)	 (45)	 (134)

図16. あらかじめSVMに与えられた学習データ

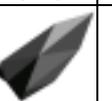
 検索対象	 1:(6)	 2:(7)	 3:(10)	 4:(79)
 5:(37)	 6:(3)	 7:(83)	 8:(204)	 9:(182)
 10:(112)	 11:(144)	 12:(206)	 13:(188)	 14:(12)

図17. 「いわゆる」ウサギの検索結果