

## OBBTree 表現を形状特徴とする類似検索システム\*

加来啓太郎<sup>1</sup> 岡田義広<sup>2,3</sup> 新島耕一<sup>2</sup>

<sup>1</sup>九州大学大学院 システム情報科学府

<sup>2</sup>九州大学大学院 システム情報科学研究所

<sup>3</sup>独立行政法人 科学技術振興機構 さきがけ研究「協調と制御」領域

### 1. はじめに

本研究の目的は大規模な形状データ集から必要とする形状を正確かつ高速に検索するための形状特徴をよく表す特徴量を提案することである。著者らは、OBBTree[1]を形状モデルの特徴量として用い、2つの形状モデルのOBBTreeについて、対応するノードにあるOBBの中心位置座標の誤差を類似度の尺度として用いる形状検索システム[2]を提案している。本稿では、D2と呼ばれる手法[3]との比較を行うことによりその有用性を述べる。

### 2. OBBTree

OBBTreeとは、ポリゴンモデルの衝突検出を高速に行うために提案されたポリゴンモデルの近似表現形式である。OBB(Oriented Bounding Box)とは向きを持った bounding box のことであり、OBBに含まれる部分形状を再帰的に2分割していくことにより、OBBの二分木であるOBBTreeが得られる。図1にOBBTreeの例を示す。

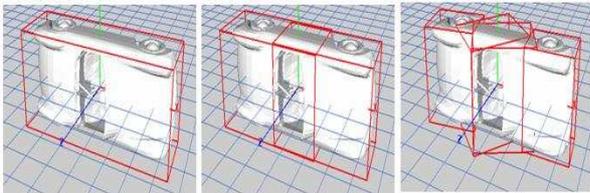


図1. OBBTree

### 3. 類似検索手法

#### 3.1 準備

OBBTreeのノードであるOBBの中心位置の誤差を類似度の尺度として用いるため、スケールを合わせる必要がある。そこで、立方体の箱に収まるように元形状を正規化する。この際、アスペクト比の情報が失われる。アスペクト比は形状モデルの類似検索の重要な特徴量のひとつであり、後で利用するために、正規化を行う前に各形状モデルについてアスペクト比を計算し保存しておく。

次に形状モデルからOBBTreeを生成する。しかし形状モデル表面の各面の大きさにばらつきがあ

る場合にはOBBの分割がうまく行えないという問題が発生した。これは、表面上の点を一様にサンプリングし、その点の周りに同じ大きさの極小の三角形を面として生成させたものを形状モデルとして扱うことにより解決した。

#### 3.2 類似度の計算

##### 3.2.1 アスペクト比に関する類似度

2つの形状モデル $A, B$ についてBB(bounding box)のアスペクト比を特徴量として類似度を計算する。 $A$ のBBの第1, 第2, 第3主軸の長さを $A_1, A_2, A_3$ ,  $B$ のBBの第1, 第2, 第3主軸の長さを $B_1, B_2, B_3$ とし、アスペクト比に関する誤差 $E_a$ を以下の式により計算する。値が小さいものを類似度が高いとする。

$$E_a = \sqrt{\left(\frac{A_2}{A_1} - \frac{B_2}{B_1}\right)^2 - \left(\frac{A_3}{A_1} - \frac{B_3}{B_1}\right)^2} \quad (1)$$

##### 3.2.2 OBBの中心位置座標に関する類似度

比較する形状モデル $A, B$ のOBBTreeについて、深さ $l$ を指定し類似度を計算する。深さ $l$ におけるOBBTreeのノードの数 $n$ は $2^l$ 個であり左から $i$ 番目のノードの座標をそれぞれ $(x_i^A, y_i^A, z_i^A)$ 、 $(x_i^B, y_i^B, z_i^B)$ としてOBBの中心位置座標に関する誤差 $E_c$ を以下の式により計算する。値が小さいものを類似度が高いとする。

$$E_c = \sum_{i=1}^n \sqrt{(x_i^A - x_i^B)^2 + (y_i^A - y_i^B)^2 + (z_i^A - z_i^B)^2} \quad (2)$$

しかし、2つの類似した形状モデルから得られるOBBTreeの同じ深さ $l$ における各ノードの左からの位置が2つのOBBTreeにおいて対応しているという保証はない。すなわち、誤差が最小になるという保証はない。そこで次のアルゴリズムを用いて深さ $l$ におけるノードを並べ替えてから(2)式により誤差を計算する。

まず、比較する2つのOBBTreeにおいて対応するノード $A, B$ に着目する。 $A$ の左右の子ノードを $L_A, R_A$ とし、 $B$ については $L_B, R_B$ とする。

a) 以下の2つの異なる組み合わせについて距離の

\*Polygonal Model Search System Using OBBTree-based Similarity Measure

Keitarou Kaku<sup>1</sup>, Yoshihiro Okada<sup>1,2</sup>, Koichi Nijijima<sup>1</sup>

<sup>1</sup>Graduate School of Information Science and Electrical Engineering, Kyushu University

<sup>2</sup>Intelligent Cooperation and Control, PRESTO, JST

和を計算する.

$$d_1 = |L_A - L_B| + |R_A - R_B| \quad (3)$$

$$d_2 = |L_A - R_B| + |R_A - L_B| \quad (4)$$

b)  $d_1 \geq d_2$ であれば部分木  $L_B$  と  $R_B$  を入れ替える.

指定された深さ  $l$  まで, a) b) のアルゴリズムを再帰的に適用して,  $B$  の子孫ノードをソートする.

### 3.2.3 類似度

OBB の中心位置座標に関する誤差  $E_c$  とアスペクト比に関する誤差  $E_a$  にそれぞれ重み  $w_c, w_a$  を掛けて2つの形状に関する誤差  $Error$  を以下の式により計算する. 値が小さいものを類似度が高いとする.

$$Error = \frac{w_c E_c + w_a E_a}{w_c + w_a} \quad (5)$$

同じクラスの形状データであっても3つの主軸の順番が一致しないことがある. そのため比較する形状を3つの主軸について回転させた全ての場合について  $Error$  を計算し最小のものを2つの形状モデルの類似度とする.

## 4. 評価実験

### 4.1 D2 について

D2 は形状の表面上のランダムな2点間の距離の分布を形状の特徴量として用いる方法であり, 類似検索の手法としてよい結果を与えているもののひとつである. この方法との比較実験を行うことにより提案する類似検索手法の有用性を検証した.

### 4.2 形状データベースと評価指標

図2に示すように, 形状データベースとして25クラスからなる計145個の形状モデルを用意した. 評価指標として以下の2つの指標を用いた. 質問形状モデルの属するクラスの総数を  $k$  とすると,

- ・第1指標は, 上位  $k-1$  位に, 残りの  $k-1$  個のいくつの形状が含まれているかの割合とする.
- ・第2指標は, 上位  $2(k-1)$  位に, 残りの  $k-1$  個のいくつの形状が含まれているかの割合とする.

### 4.3 実験結果と考察

表1は第1指標, 第2指標の平均をまとめたものである. 表1から分かるとおり,  $l=2,3$  において第1,2指標, 検索時間ともにD2よりも良い結果が得られた. 当然, これらの結果は形状データベースに依存する. 複雑な形状については  $l$  の値を上げることにより精度が向上するが, 検索時間も大幅に必要となる. ボール, びんなど比較的単純な形状だとD2の方がよい.  $w_c : w_a$  については5:5~6:4くらいが適当であることが分かった.

## 5. おわりに

本稿では形状モデルからOBBTreeを構築しノードであるOBBの中心位置座標から2つの形状モデルの類似度を計算する手法を提案し, D2法との比較実験を行うことによりその有用性を述べた. 複雑な形状の場合には, OBBTreeの深さを大きくすることにより精度を向上させることができる. 本稿では説明できなかったが, GUIをもつ類似検索システムを開発したことにより, より直感的に検索を行うことができる. 今後は, Watershed法[4]による形状の分割を行い, 分割された部品を質問形状とする類似形状検索手法の検討と, 検索システムの開発を行なう予定である.

### 参考文献

- [1] S. Gottschalk, M.C. Lin, F. Manocha, OBBTree: A Hierarchical Structure for Rapid Interference Detection, Proc. Of ACM SIGGRAPH'96, 171-180, 1996
- [2] K. Kaku, Y. Okada, K. Nijima, SIMILARITY MEASURE BASED ON OBBTREE FOR 3D MODEL SEARCH, in International Conference Computer Graphics, Imaging and Visualization (CGIV04), July, 2004.
- [3] R. Osada, et al, Matching 3D Models with Shape Distributions, Int. Conf. on Shape Modeling & Applications, 154-165, 2001
- [4] Alan P. Mangan, Ross T. Whitaker, Partitioning 3D Surface Meshes Using Watershed Segmentation, IEEE Transactions on Visualization and Computer Graphics, Volume 5, Issue 4, 308-321 October, 1999

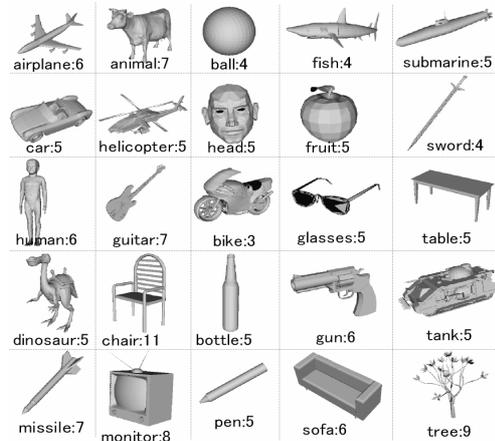


図2. 形状データベース

$l, w_c : w_a$	第1指標	第2指標	時間(sec)
2, 5:5	0.29	0.43	16
3, 5:5	0.36	0.47	22
5, 6:4	0.38	0.52	63
6, 2:8	0.31	0.44	116
6, 6:4	0.39	0.52	116
6, 8:2	0.36	0.48	116
D2	0.28	0.40	28

表1. D2, パラメータを変化させた時の実験結果