

実物体を利用した 3 次元形状モデル検索

市田 浩 靖[†] 伊藤 雄 一[†]
北村 喜 文[†] 岸野 文 郎[†]

多くの 3 次元形状モデルが格納されたデータベースから、ユーザが必要とする物体を検索する作業において、ユーザが実際に手に持って操作できるような実物体を用い、積み木を組み立てる感覚で自分の意図する形状を構築し、それを用いてデータベースの検索を行い 3 次元形状モデルを得ることができれば、その検索インターフェースはユーザにとって非常に簡単で、直感的なものとなる。さらに実物体で形状を構築する作業は、言語や文化にそれほど依存せず、また専門的な知識も必要としない。このような実物体を用いた直感的な 3 次元形状モデル検索を実現するためのインターフェースとして ActiveCube を利用する。ユーザは ActiveCube のブロックを組み立てることで、リアルタイムに 3 次元形状をコンピュータに入力することができる。ユーザがブロックを組み立て、その形状がコンピュータに入力されるとシステムはその形状のボクセルデータと、データベース内の各モデルに対応するボクセルデータを比較し、類似度を算出する。その結果を基に、システムはユーザに候補を提示し、その中から所望のモデルを選択する。これによりどんなユーザでも直感的に 3 次元形状モデルの検索を行うことができる。

Retrieval of 3D Shape Models Using Real Objects

HIROYASU ICHIDA,[†] YUICHI ITOH,[†] YOSHIFUMI KITAMURA[†]
and FUMIO KISHINO[†]

The topic of this paper is retrieval of 3D shape models using real objects. The user constructs a 3D structure by connecting blocks as she/he wishes, and then the system automatically recognizes the shape of the constructed structure in real-time and picks up 3D shape models that are well similar to constructed structure. Afterwards, she/he chooses one on the 3D shape models that she/he intends. Thus, it is not dependent on generations, cultures, and languages to input the query shape by using real objects, as the same time it does not require special knowledge. In order to implement proposed system, we use ActiveCube system.

1. はじめに

近年、多くの 3 次元形状モデルがインターネットなどで公開・販売されるようになっており、コンピュータグラフィックスの専門家のみならず、だれもが入手して利用することが可能となり、3 次元形状モデルを生成する技術に加えて、多くの 3 次元形状モデルに対する検索技術も重要になってきている^{1)~10)}。

しかし、テキストや画像を検索する場合に比べて、3 次元形状モデルを検索する作業は、形状の情報を言い表すのが困難であるため、だれもが直感的に容易に行えるものとはいえない^{11),12)}。

そこで我々は、多くの 3 次元形状モデルの中から、ユーザが所望する 3 次元形状モデルを検索する際に、

クエリを作成するための新たなユーザインターフェースとして、実物体を用いるシステムを提案する。提案システムでは、ユーザが実物体を用いて手で 3 次元形状を構築すると、自動的にその 3 次元形状がクエリとしてシステムに入力される。システムは、3 次元形状モデルデータベース内の各モデルの形状と入力形状を比較し、外観形状が似ているものを即座にユーザに候補モデルとして提示する。本論文では、提案手法を実装したシステムと、そのシステムに関して検討した内容を報告する。

2. 関連研究

大量の 3 次元形状モデルからユーザが所望するモデルを探すためには、いくつかのアプローチが考えられる。まず、各モデルにキーワードを付加する手法が考えられる。実際、3 次元形状モデルを配布するウェブサイトの中には、モデルを探す際にキーワードを利用

[†] 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology,
Osaka University

したテキスト検索を採用しているものも多い。しかし、この手法では、記述されたキーワードが限定的であったり、曖昧な表現となったり、正確な内容を表現しきれていなかったりする。さらに、付加されるキーワードが言語・文化・世代などに依存する可能性がある。特に、3次元形状モデルを画面上に表示するためには、視点位置や角度によって見え方が変化する場合もあるため、さらに多くのキーワードで表現しなければならず、単純なキーワードだけでは表現しにくい。

また、市販のモデリングソフトなどを利用して作成した類似モデルや、既存の3次元形状モデルをクエリとして用い、その形状から抽出される特徴量に注目した検索手法も提案されている。Alexandria というオンラインデモにおいては、ポリゴンモデルの各面の法線ベクトルや色のヒストグラムを特徴量として使用し、形状・色といった個々のモデルの属性に関して重み付けをして検索することが可能である^{1),2)}。鈴木らは3次元形状モデルを格子状のセルに分割し、各セル内のモデルの頂点密度を利用することで検索時間の短縮を実現した⁴⁾。Eladらのシステムでは、3次元形状モデルの各面から計算されるn次元モーメントを特徴量として採用し、検索結果として提示された候補モデルに、ユーザが「適」「不適」のマークを施すことで類似判定に使用する関数に影響を与える重み値を変更し、再帰的に検索するシステムを実現した⁶⁾。Hilagaらはポリゴンモデルの位相構造から骨格構造を導き、ポリゴンモデル間の類似度を算出する技術を提案した⁷⁾。Chenらは、Hilagaらの技術の前処理として、ポリゴンモデルの各面の再構成を行い、検索の正確性と即応性を追求し、実用化した⁸⁾。Osadaらは3次元形状モデル全体の幾何属性を求める関数を利用して、形状分布を算出した¹⁰⁾。しかし、これらの研究では、主に3次元形状の厳密な検索を目的としたアルゴリズムに研究の中心があり、ユーザにとって使い勝手を良くする、クエリ入力のためのユーザインタフェースに関してはほとんど考慮されてこなかった。

最近では、新たなユーザインタフェースを用いた検索手法も提案されるようになってきている。ChenらやMinらのシステムにおいては、3次元形状を入力するために、上面・側面・前面から見た2次元の輪郭線を入力することができる^{9),11)}。さらに、クエリとしての3次元形状モデルを作成する際に、2次元でスケッチするだけで3次元形状を作成できるTeddy¹³⁾のようなシステムを利用することも考えられている¹²⁾。

本論文では、実物体であるブロックを組み立てることで、ユーザが思い描く3次元形状を具体的に作成し、

これと類似の形状を、あらかじめ計算機に蓄えられた3次元形状モデルのデータベースの中から検索するシステムを提案する。

3. 実物体を利用した 3 次元形状モデル検索

3.1 実物体の利用

3次元形状モデルの検索システムにおいて、実際に手に持って扱えるような実物体を用い、積み木を組み立てる感覚で自分の頭の中でイメージする形状を構築し、それをクエリとして用いて3次元形状モデルの検索を行い、その結果を即座に得ることができれば、検索インタフェースはユーザにとって非常に簡単で、直感的なものとなる。実物体で形状を構築する作業は、言語・文化・世代などにそれほど依存せず、3次元形状を2次元に投影する必要もなく、また専門的な知識も必要としない。さらに、検索に使用した実物体を、仮想空間内のモデルに対応する実空間の存在としてとらえることにより、実空間と仮想空間の境界を越えて、両空間を結ぶユーザインタフェースとして利用価値の高いものが構築できる可能性もある。そういったシステムを実現するために、実物体の3次元構造をリアルタイムに認識し、実空間と仮想空間のモデルとのインタラクションが可能なインタフェースが必要となる。

実物体を使った3次元形状認識の研究は、建築の分野で古くから行われており^{14)~17)}、最近では、組み立てられたブロックの構造をコンピュータにリアルタイムに入力するシステム¹⁸⁾やLEGO型ブロックを組み立てた後に、コンピュータに接続して電源を入れると、その形状がコンピュータに入力されるものがある¹⁹⁾。本研究では、図1に示すActiveCube²⁰⁾を利用する。ActiveCubeは、ブロックを組み立てるだけでリアルタイムにその3次元形状をコンピュータ内に入力することができる機能や、ブロック自体に装備された各種のセンサやアクチュエータなどによってユーザの操作



図1 ActiveCubeシステム
Fig. 1 ActiveCube system.

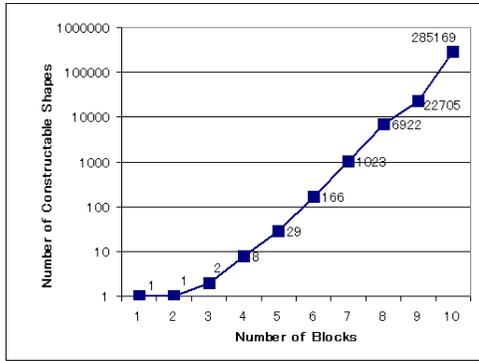


図2 ブロックの個数と構築可能な形状数の関係

Fig. 2 Relationship between the number of used blocks and the number of constructed shapes.

意図を入力し、計算機内でのシミュレーション結果を表示するといった入出力機能を有する双方向ユーザインタフェースである。ActiveCubeの各ブロックは一辺5cmの立方体からなり、接続可能なすべての面は接続に関してまったく等価である。

3.2 ブロック組み立てによる形状表現の自由度

ActiveCubeのブロックでクエリ形状を作成するため、6面が接続に等価な立方体のブロック n 個を用いることでどの程度の形状表現の自由度があるかを考える。ブロック n 個を用いて構築可能な形状の数は以下の手順で求める。まず、ActiveCubeのようなブロックでは表現できないような形状(たとえば、ブロックどうしがつながっていない形状)が求まるのを避けるため、 $n-1$ 個のブロックを用いて構築可能な全形状の空いている各面にブロックを1つずつ接続し、 n 個のブロックで構成される新形状の候補をすべて求める。次に、これらの形状をボクセルデータで表現する。その後、5章で述べる類似度算出式を用いて同じ形状をすべて取り除く。これによって得られた形状は一意であり、ブロック n 個を用いて構築可能な形状を求めることができる。構築可能形状の算出は、複数のPCをLANで接続し、類似度の計算を分散処理することで行った。

6面が接続に関して等価な立方体のブロックの個数と、その個数のブロックで構築可能な形状の数の関係を図2に示す。ただし、回転と平行移動は考慮してある。ブロックの個数が1つ増えるごとに構築可能な形状の数は指数関数的に増大していく。このグラフより、10個程度のブロックで構築可能な形状は約30万個ほどもあることが分かり、100~10,000個程度のモデルを対象とするデータベース検索システムにおいては十分であると考えられる。また、数十、数百個のブ

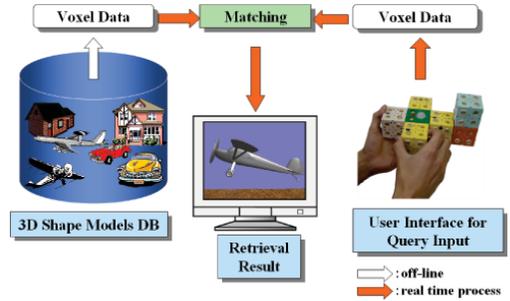


図3 提案システムでの流れ

Fig. 3 System flow.

ロックを使うことで精度の高い詳細な形状を組み立てられると考えられるが、そのような多くのブロックを用いて形状を組み立てる作業は非常に時間と労力を要するうえ、このような多数のブロックを使って詳細な形状を構築しなければ、所望の形状が得られないのであれば、提案システムは現実的・有効的であるとはいえない。

3.3 提案手法

ユーザがクエリ形状を作成する際には、立方体のActiveCubeのブロックを組み立てることで、所望のモデルの3次元的なボリュームを表現し、クエリとして入力する。ほかにもActiveCubeで構築された形状を、3次元的な骨格構造としてクエリとする手法も考えられるが、ActiveCubeは5cm角の立方体のブロックであり、骨格構造よりも、その大きさを利用したボリュームを表現する手法の方が適していると考えられる。このボリュームを表現するために、ユーザがActiveCubeを使って構築する3次元形状と、データベースに格納される多面体から構成される3次元形状モデルの双方に、ボクセルデータ表現を用いる。

提案システムは、図3に示すように、ユーザがActiveCubeのブロックを10個程度使い、頭の中でイメージする形状を組み立てクエリ形状として入力すると、4章で述べる手法を用いて、そのクエリ形状を即座にボクセルデータに変換する。次に、あらかじめボクセルデータで表現されたデータベース内の各モデルと類似度を算出する。5章で述べる類似度算出の際には、比較する両者のボクセルデータのスケールが問題となる。2つのボクセル数が異なる場合には、両方のボクセルデータを重ね合わせた共通部分の割合を算出するだけでは問題が生じる。一方のボクセルデータが、他方のボクセルデータよりも多くのボクセルからなっている場合、前者に後者が完全に含まれてしまうことがあり、正確な類似度を算出できない。そこで、あらかじめ各3次元形状モデルを $n(n=1, 2, \dots, N)$ のボク

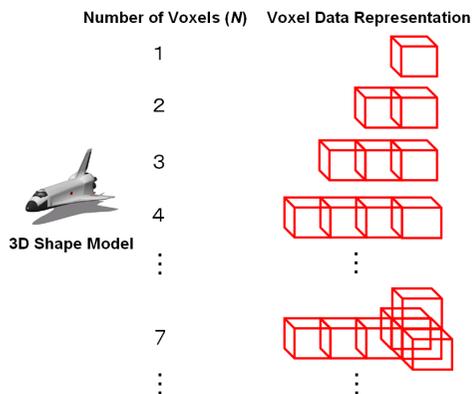


図 4 3次元形状モデルに対するボクセル定義の例
Fig. 4 An example of definition of voxel data for a 3D shape model.

セルデータで表現したものをデータベースに格納しておき、クエリ形状が入力された際には、その形状のボクセル数と同数で表現された 3 次元形状モデルのボクセルデータを利用して、類似度を算出する．図 4 にスペースシャトルのモデルに対する $n(n = 1, 2, \dots, N)$ 個のボクセルデータの定義の例を示す．定義の詳細に関しては 4.2 節で述べる．このようにクエリ形状を構成するブロック数に応じて、対応する各 3 次元形状モデルのボクセルデータを使うことでスケールの問題は解消できる．さらに、単純な計算で類似度を算出できるため、高速に処理することができる．システムは、その結果を基にユーザに外観形状の近い候補モデルを提示する．

4. ボクセルデータ表現

ユーザによって構築された形状は、システムによってリアルタイムにボクセルデータに変換される．また 3 次元形状モデルデータベースには、3 次元形状モデルのポリゴンモデルと、そのモデルに対応するボクセルデータを格納する．

4.1 クエリ形状のボクセルデータ表現

ユーザが ActiveCube を組み立て、クエリ形状を構築すると、システムはツリー状の接続状況マップ²⁰⁾を用いることによって、図 5 に示すように 1 ブロックを 1 ボクセルとするボクセルデータにリアルタイムで変換する．その際には、接続された ActiveCube の中でホスト PC とつながるコネクタが 1 つの面に接続されているベースブロックと呼ばれるブロックを原点とする座標軸を定義する．図 5 (a) に入力形状の例を、図 5 (b) に入力形状の最も左端にあるブロックがベースブロックである場合のボクセルデータ表現の例

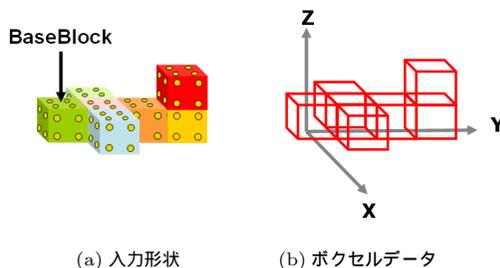


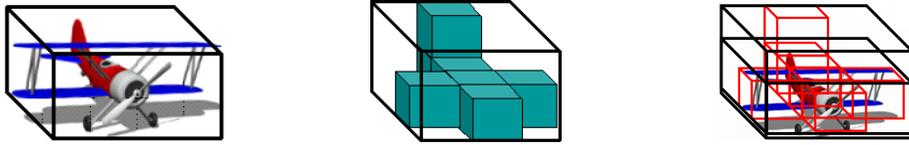
図 5 ActiveCube 形状に対するボクセルの定義の例
Fig. 5 An example of definition of voxel data for a shape of ActiveCube.

を示す．

4.2 3次元形状モデルのボクセルデータ表現

データベースに格納する 3 次元形状モデルをあらかじめ $n(n = 1, 2, \dots, N)$ 個のボクセルで表現する．ある 3 次元形状モデルを n 個のボクセルデータで表現するためには、まず最初に n 個のボクセルで表現可能な形状をすべて算出し、得られた形状すべてに対して 3 次元形状モデルとの類似度を算出し、類似度が最も高いものをその 3 次元形状モデルを n 個のボクセルで表現したボクセルデータと定義する．以下に、ある 3 次元形状モデルを、 n 個のボクセルデータで定義する手順について述べる．

- (i) 3.2 節で述べた手法を用いて、 $n(n = 1, 2, \dots, N)$ 個のボクセルを使って構築されるすべての形状を算出する．
- (ii) ボクセルデータを定義する 3 次元形状モデルを包含する最小直方体を求める (図 6 (a)). ここでは、モデルがあらかじめ正面向きに定義されていることを前提とし、求める最小直方体は、その座標軸の各軸に対して垂直な、6 平面から構成される直方体の中で、モデルを包含する最小直方体のことである．
- (iii) (i) で求めた形状から 1 つの形状を選び出し、その形状を包含する最小直方体を求める (図 6 (b)).
- (iv) (ii) で求めた最小直方体を (iii) で求めた最小直方体に正規化し、重ね合わせる (図 6 (c)). その際に、直方体内のボクセルデータが占める領域にある 3 次元形状モデルのポリゴンの頂点数が最大となるように回転する．
- (v) 直方体内のボクセルデータが占める領域に存在する 3 次元形状モデルのポリゴンの頂点数 (V) を求め、3 次元形状モデルを構成する全ポリゴンの頂点数 (V_{all}) から、式 (1) を用いて全頂点のうちボクセルデータ内に含まれる頂点の割合



(a) 3次元形状モデルに対する最小直方体 (b) ボクセルデータに対する最小直方体 (c) (a)と(b)の直方体の重ね合わせ

図6 3次元形状モデルに対する7個のボクセルの定義の手順の例

Fig. 6 An example of the procedure for the definition of 7 voxels for a 3D shape model.

を算出し、ボクセルデータと3次元形状モデルの類似度 (*Model-Voxel Similarity*) とする。

$$MV Sim. = \left(\frac{V}{V_{all}} \right) \cdot 100 \quad [\%] \quad (1)$$

- (vi) (iii) から (v) までは (i) で得られた形状の数だけ繰り返し、類似度が最大となるものをその3次元形状モデルを n 個のボクセルで表現したものとして定義する。
- (vii) 手順 (ii) から手順 (vi) までは各3次元形状モデルに対して行うことでデータベース内の全3次元形状モデルを n 個のボクセルデータで表現し、データベース内に格納する。

このようにあらかじめボクセルデータを定義しておくことで、システムはユーザが入力した形状を構成するブロック数 (n) に応じて、各モデルを n 個のボクセルで表現したデータと類似度を算出することが可能になる。このデータベースへの登録作業はシステムがオフラインで自動的にを行う。

5. 類似度算出アルゴリズム

本章では、 n 個のブロックからなる入力クエリ形状のボクセルデータと、3次元形状モデルを n 個のボクセルで表現したボクセルデータとの、類似度の算出方法に関して述べる。

ボクセルデータどうしの類似度を判定する式は4章の手法を用いることで非常に単純なものとなる。まず、平行移動・回転を繰り返して双方のボクセルの重なる共通部分が最大となるように重ね合わせ、その共通部分に含まれるボクセル数 (i) を求め、全ボクセル数 (n) に対する割合を求める。つまり、類似度 (*Voxel-Voxel Similarity*) は式 (2) から算出する。

$$VV Sim. = \left(\frac{i}{n} \right) \cdot 100 \quad [\%] \quad (2)$$

クエリ形状から得られるボクセルデータと3次元形状モデルのボクセルデータの類似度算出が完了すると、システムはすべての3次元形状モデルのうち、類似度が、ある閾値以上のものを検索結果としてユーザに提示する。

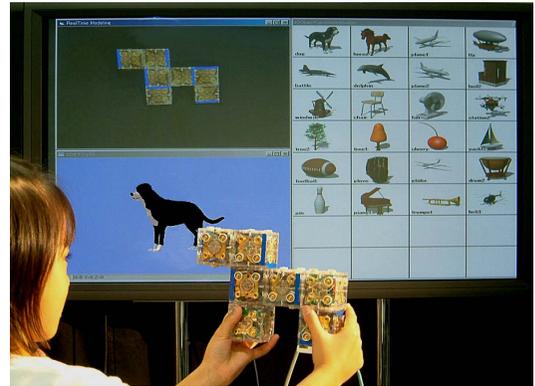


図7 提案システムでの検索の様子

Fig. 7 A snapshot of our system.

6. 実装システム

様々なカテゴリから選んだ96個の3次元形状モデルをActiveCubeのブロックを使って検索するシステムを実装した。検索対象となるモデルの数が100個程度なので、3.2節で述べたように10個程度のブロックで検索可能と考えられるので、 $N = 10$ とした。実装環境はノートPC (Let's Note CF-L1, OS: Microsoft Windows98 Second Edition, CPU: Intel Pentium III 600 MHz, RAM: 192 MB) を使用し、開発言語として検索部分はBorland C++ Builder 5, 表示部分はVisual Basic 6.0, ActiveCubeの制御はActiveCubeライブラリ²¹⁾を利用した。類似度算出の結果、ユーザに検索結果として提示する候補モデルは類似度が80%以上のものとした。図7に、低学年児童のユーザがActiveCubeを使ってクエリとなる3次元形状を構築し、この形状を利用して3次元物体のシステムに入力して検索を行っている様子を示す。図8に示すように定義された犬のモデルに対するボクセルデータを、図9のような手順で作成したときの、すべてのモデルとの類似度の変化を図10に示す。各グラフの縦軸は類似度(%)で、横軸はクエリ形状を構成するブロックの数(この場合、1個から8個まで)である。

クエリ形状が1個のブロックのときは、すべての3

次元形状モデルは 1 個のボクセルで表現されているため、類似度は 100%となる。ここにブロックを 1 個接続し、クエリ形状が 2 個のときもすべての 3 次元形状モデルは 2 個のボクセルで表現され、類似度は 100%となる。クエリ形状が 3 個になると、3つのボクセルでは 2 通りの表現を持っているため、候補モデルは半分くらいに絞り込まれる。さらにブロックを接続していくと、各モデルごとにボクセルデータ表現が変わってくるため、提示されるモデルが絞り込まれていく。図 9 の手順で 8 個のブロックを次々に接続すると、システムは類似度が 80%以上のものとして、図 10 左上の枠で囲まれた 3 つのモデルを提示する。検索にかかる平均の時間のグラフを図 11 に示す。図より 8 個

の場合でも約 3 秒であり、十分実用に供しうるシステムであるといえる。

また、実際に子供が提案システムを利用している様子を観察した。結果として、被験者の子供は、積み木で遊んでいるかのような感覚でブロックを次々に組み立て、提示される検索結果と見比べながらモデルを探した。このことより、コンピュータに不慣れなユーザでも検索することができるシステムが構築できたといえる。

7. 検 討

提案システムでは、実物体の 3 次元形状をクエリとして入力し、検索結果として候補となるモデルが即座に提示される。実物体を利用することによって、専門的な知識を必要とせず、3 次元形状を扱う作業を容易に行うことが可能となり、だれにでも直感的に容易に

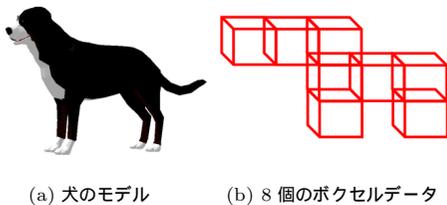


図 8 犬のモデルに対して定義された 8 個のボクセルデータ
Fig. 8 8 voxel data for a 3D shape model of a dog.

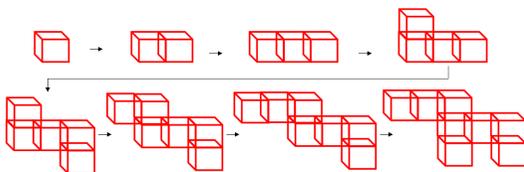


図 9 犬のモデルを検索した際のブロック形状の作成手順
Fig. 9 A procedure of connecting blocks for a dog.

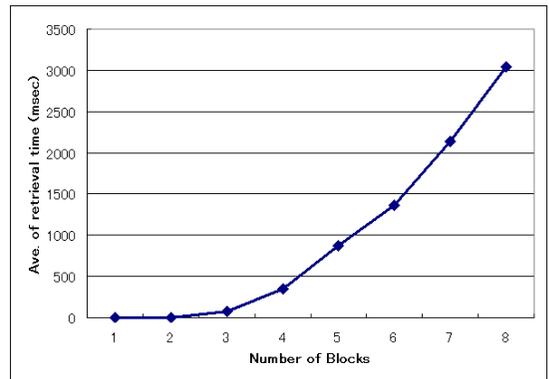


図 11 検索にかかる時間とクエリ形状の個数の関係
Fig. 11 Relationship between the number of blocks and average of retrieval time.

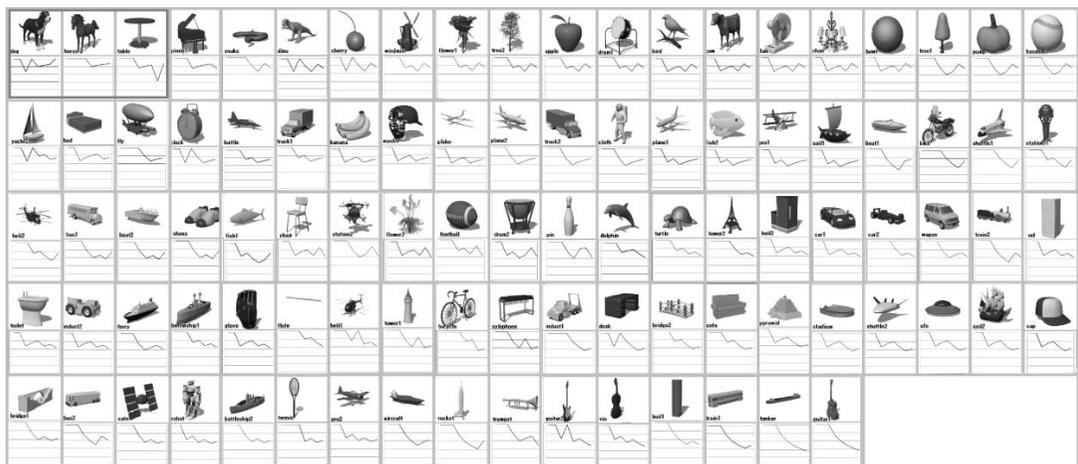


図 10 全 3 次元形状モデルに対する類似度変化
Fig. 10 Transition of similarity of all 3D shape models.

使用できるものとなった．実際に，初心者や子供でも積み木を組み立てる感覚で，ブロックを組み立てるだけで検索できることを確認した．

しかし，ユーザが頭の中でイメージしている3次元形状を検索するために，ブロックを次々に接続してクエリ形状を作成しても，必ずしも頭の中でイメージするモデルが提示されないことがある．たとえば，「蛇」を検索する際には，「とぐろをまいた蛇」か，「地面を這っている蛇」か，どちらをユーザがイメージしたかで検索結果は大きく変わってしまう．また，提案したポリゴンモデルに対するボクセルデータ表現では， n 個のボクセルを定義する際に， $n-1$ 個のボクセル定義を考慮していない．そのため，徐々に絞り込まれずに， n 個のクエリ形状を作成すると $n-1$ 個の形状の際には候補の上位にはなかったモデルがいきなり提示されることがある．そこで， n 個のボクセルデータ定義アルゴリズムに， $n-1$ 個までのボクセルデータ表現を考慮したアルゴリズムを使用することで，多くのモデルから候補モデルを絞り込むイメージで検索できると考えられる．しかしながら，ユーザの意図しないモデルがいきなり提示されることでユーザは思いがけない発見をし，イメージを膨らませることができる可能性があると考えられる．

次に，提案したアルゴリズムでは，3次元形状モデルがあらかじめ正面向きに定義されたモデルを対象としている．つまり，モデルが定義されている座標軸とクエリ形状に設定する座標軸を各軸回りに90度ずつ回転させれば，クエリとモデルの向きが一致することが前提となっている．モデルが座標軸に対して傾いた状態で定義されている場合には，モデルの向きとクエリ形状の向きを一致させる手順を導入する必要がある．

また，検索時間に関しては，クエリ形状を構成するブロックの数が増えるに従って指数関数的に増大していく．これは，ボクセルデータ間の類似度を判定する際に，90度ずつの回転と平行移動を繰り返し，ユーザがベースブロックに対してあらゆる方向にクエリ形状を作成することと，ベースブロックがクエリ形状のどの位置にあってもよいことを想定しているからである．検索時間を短縮するためには，あらかじめベースブロックの向きと位置を決めておくことで改善できると考えられる．

さらに，今回提案したボクセルデータの比較による手法だけでは，形状の似ているものどうしの識別は困難であると考えられる．そこで，ボクセルデータの比較だけでなく，ActiveCubeが装備する入出力機能や各ブロック表面の色などのあらゆる情報をクエリと

して利用する．それにより図2に示す形状数に加えて，さらに多くの情報をクエリとして扱うことができる．たとえば，同じ飛行機の形状を構築した際にも，先頭部分を入出力機能を装備していないプレーンブロックで構築するか，ファンなどの回るという機能を持つ出力装置を装備したブロックで構築したかによって，プロペラ機とジェット機との区別をするようなシステムも考えられる．また，実際にシステムを使用した被験者の中には「具体的な3次元形状を考えずに」ブロックを組み立てて形状をモデルへ変換するのを楽しむ者もいた．これは，本システムが具体的なイメージを持たない検索スタイルの可能性を示していると考えられる．この面を利用し，クエリ形状とその検索結果との関連を発見する学習ツールとしての可能性も調査していきたい．

8. おわりに

本論文では，実物体を検索インタフェースに用いた3次元形状モデル検索に関して述べた．実物体のユーザインタフェースとして，ActiveCubeを利用し，3次元形状モデルに関する専門的な知識がないユーザでもActiveCubeのブロックを利用して3次元形状を構築するだけで多くの3次元形状モデルに対するクエリを生成し，検索することができることを確認した．

謝辞 本研究の一部は，文部科学省21世紀COEプログラム(研究拠点形成費補助金)の研究助成，科学研究費補助金基盤研究(B)(2)12480096の助成を受けた．

参考文献

- 1) Paquet, E. and Rioux, M.: Nefertiti: a query by content system for three-dimensional model and image databases management, *Proc. Image and Vision Computing* 17, pp.157-166 (1999).
- 2) Paquet, E. and Rioux, M.: Nefertiti: A Tool for 3-D Shape Databases Management, *SAE Trans.: Journal of Aerospace* 108, pp.387-393 (2000).
- 3) 鈴木一史，加藤俊一，築根秀男：主観的類似度に適応した3次元多面体の検索，電子情報通信学会論文誌 D-I, Vol.J82-D-I, pp.185-193 (1999).
- 4) Suzuki, M.T.: A Web-based Retrieval System for 3D Polygonal Models, *Proc. Joint 9th IFSA World Congress and 20th NAFIPS International Conference (IFSA/NAFIP'01)*, pp.2271-2276 (2001).
- 5) 西村 剛，寺本淳司，長田秀信，紺谷精一：3次元物体データベースにおける類似物体検索の検討，*IEICE database system*, 122-62, pp.479-485

- (2000).
- 6) Elad, M., Tal, S. Ar. A.: Content based retrieval of VRML objects — an Iterative and Interactive approach, *Proc. Eurographics Workshop, Multimedia 2001*, pp.107–118 (2001).
 - 7) Hilaga, M., Shinagawa, Y., Kohmura, T. and Kunii, T.L.: Topology matching for fully automatic similarity estimation of 3D shapes, *Proc. SIGGRAPH 2001*, pp.203–212 (2001).
 - 8) Chen, D.Y. and Ouhyoung, M.: A 3D object retrieval system based on multi-resolitional reeb graph, *Proc. Computer Graphics Workshop*, pp.16–22 (2002).
 - 9) Chen, D.Y. and Ouhyoung, M.: A 3D Model Alignment and Retrieval System, *Proc. International Computer Symposium*, Vol.2, pp.1436–1443 (2002).
 - 10) Osada, R., Funkhouser, T., Chazelle, B. and Dobkin, D.: Shape Distributions, *ACM Trans. Graphics*, Vol.21, No.4, pp.807–832 (2002).
 - 11) Min, P., Chen, J. and Funkhouser, T.: A 2D Sketch Interface for a 3D Model Search Engine, *SIGGRAPH CAA (Technical Sketch)*, p.138 (2002).
 - 12) Min, P., Halderman, J.A., Kazhdan, M. and Funkhouser, T.A.: Early Experiences with a 3D Model Search Engine, *Proc. Web3D Symposium*, pp.7–18 (2003).
 - 13) Igarashi, T., Matsuoka, S. and Tanaka, H.: Teddy: A Sketching Interface for 3D Freeform Design, *Proc. SIGGRAPH 1999*, pp.409–416 (1999).
 - 14) Aishi, R. and Noakes, P.: Architecture without numbers — CAAD based on a 3D modeling system, *Computer-Aided Design*, Vol.16, No.6, pp.321–328 (1984).
 - 15) Frazer, J.H., Frazer, J.M. and Frazer, P.A.: Three-dimensional data input devices, *Proc. Conference on Computers/Graphics in the Building Process* (1982).
 - 16) Frazer, J.H.: *An evolutionary architecture*, Architectural Association, London (1995).
 - 17) Anagnostou, G., Dewey, D. and Patera, A.T.: Geometry-defining processors for engineering design and analysis, *The Visual Computer*, 5, pp.304–315 (1989).
 - 18) Gorbet, M.G., Orth, M. and Ishii, H.: Triangles: tangible interface for manipulation and exploration of digital information topography, *Proc. Conference on Human Factors in Computing Systems (CHI'98)*, pp.49–56 (1998).
 - 19) Anderson, D., Frankel, J., Marks, J., Agarwala, A., Beardsley, P., Hodgins, J., Leigh, D., Ryall, K., Sullivan, E. and Yedidia, J.:

Tangible interaction + graphical interpretation: a new approach to 3D modeling, *Proc. SIGGRAPH 2000*, pp.393–402 (2000).

- 20) 伊藤雄一, 北村喜文, 河合道広, 岸野文郎: リアルタイム 3 次元形状モデリングとインタラクションのための双方向ユーザインタフェース ActiveCube, *情報処理学会論文誌*, Vol.42, No.6, pp.1338–1347 (2001).
- 21) 市田浩靖, 伊藤雄一, 河合道広, 浅井政嗣, 北村喜文, 岸野文郎: 柔軟なアプリケーション開発のための ActiveCube ライブラリ化に関する一検討, *電子情報通信学会 2001 年総合大会講演論文集*, A-16 (2001).

(平成 15 年 4 月 21 日受付)

(平成 15 年 9 月 5 日採録)



市田 浩靖

昭和 52 年生。平成 14 年大阪大学大学院工学研究科電子情報エネルギー工学専攻博士前期課程修了。同年同大学院情報科学研究科マルチメディア工学専攻博士後期課程入学。ヒューマンインタフェースの研究に従事。日本バーチャルリアリティ学会, 電子情報通信学会各会員。



伊藤 雄一

昭和 50 年生。平成 12 年大阪大学大学院工学研究科電子情報エネルギー工学専攻博士前期課程修了。同年同専攻博士後期課程入学。平成 14 年より同大学院情報科学研究科助手, 現在に至る。ヒューマンインタフェースの研究に従事。ACM, 日本バーチャルリアリティ学会, 電子情報通信学会各会員。



北村 喜文(正会員)

昭和 37 年生。昭和 62 年大阪大学大学院基礎工学研究科博士前期課程修了。同年キヤノン株式会社情報システム研究所, 平成 4 年 ATR 通信システム研究所, 平成 9 年大阪大学大学院工学研究科助教授, 平成 14 年より同大学院情報科学研究科助教授, 現在に至る。ヒューマンインタフェース等の研究に従事。博士(工学)。ACM, IEEE, ヒューマンインタフェース学会, 日本バーチャルリアリティ学会等会員。

**岸野 文郎**

昭和 21 年生．昭和 46 年名古屋工業大学大学院電子工学専攻修士課程修了．同年日本電信電話公社（現 NTT）電気通信研究所入所．同ヒューマンインタフェース研究所において、

高速・広帯域通信システムの研究・実用化，画像処理の研究に従事．平成元年 ATR 通信システム研究所知能処理研究室室長．画像処理，知能処理，臨場感通信の研究に従事．平成 8 年大阪大学大学院工学研究科教授，平成 14 年より同大学院情報科学研究科教授，現在に至る．博士（工学）．ACM，IEEE，電子情報通信学会，ヒューマンインタフェース学会，日本バーチャルリアリティ学会等会員．
