

QueryBall:仮想世界データベースシステムにおける 問合せ方法の提案と実装

大杉あゆみ[†] 渡辺知恵美[†] 増永 良文^{††}

[†] お茶の水女子大学大学院人間文化研究科 〒112-8610 東京都文京区大塚 2-1-1

^{††} お茶の水女子大学理学部 〒112-8610 東京都文京区大塚 2-1-1

E-mail: [†]{ayumi,chiemi}@dbl-lab.is.ocha.ac.jp, ^{††}masunaga@is.ocha.ac.jp

あらまし 我々は、バーチャルリアリティ技術とデータベース技術を統合した「仮想世界データベースシステム」の設計・開発を行っている。本稿では、仮想世界に対する新しい問合せ手法として、QueryBallを提案する。QueryBallとは、問合せの指定範囲をボールに見立てて表示させたものである。QueryBallに対してボールの大きさと探索条件と探索条件に該当する（または該当しない）オブジェクトに対する表示方法を指定する事で、ボールの内部にあるオブジェクトにのみ問合せをかけることができる。また、QueryBallを移動させたり、大きさを変える事で、QueryBallの中での問合せ結果を見ながらインタラクティブな問合せを実行する事ができる。更に、複数のQueryBallを組み合わせる事により、問合せ範囲や探索条件を絞り込む事ができる。

キーワード 仮想世界データベースシステム, バーチャルリアリティ, 問合せ, QueryBall

QueryBall:A New Style of Query Mechanism in the Virtual World Database System

Ayumi OHSUGI[†], Chiemi WATANABE[†], and Yoshifumi MASUNAGA^{††}

[†] Graduate School of Humanities and Sciences, Ochanomizu University Ohtsuka2-1-1, Bunkyo-ku, Tokyo, 112-8610 Japan

^{††} Faculty of Science, Ochanomizu University Ohtsuka2-1-1, Bunkyo-ku, Tokyo, 112-8610 Japan

E-mail: [†]{ayumi,chiemi}@dbl-lab.is.ocha.ac.jp, ^{††}masunaga@is.ocha.ac.jp

Abstract The design and implementation of a "virtual world a database system" named VWDB has been undertaken by Ochanomizu University that intends to integrate a set of virtual reality systems and a databasesystem. In this paper, we propose a novel query method named QueryBall that is powerful and convenient for users to query data in the VWDB virtual world. The QueryBall is defined as a triple that consists of an area specification, search condition, and a presentation condition. The search condition can be applied only to the objects in the inside of a QueryBall. The query result can be shown to users according to the presentation condition. The radius of a QueryBall can be changed interactively by a user while he/she is moving a QueryBall and looking at its query result. In order to make query capability of the QueryBall more powerful, we propose a way of composing more than one QueryBall. Some implementation results are also shown in this paper.

Key words Virtual World Database System, Virtual Reality, Query, QueryBall

1. 背景と目的

近年、フライトシミュレーションや仮想美術館、建築など様々な分野でバーチャルリアリティ(Virtual Reality:VR)が実用化されている。また、ウォークスルーに伴う検索機能[1]や仮想世界オブジェクトの永続管理機能[2]、問合せによる仮想世界ナビゲーション機能[3],[4]など、VRシステムの一部の機能を

データベース機能によって管理し、機能拡張したシステムも開発されてきている。データベースシステムは、VRシステムをサポートするために必要と考えられる有用な機能を数多く備えており、これらのデータベース機能を完全に備え持つVRシステムを開発することは大変意義あることと考えられる。そこで、このバーチャルリアリティ技術とデータベース技術を統合した「仮想世界データベースシステム(Virtual World Database

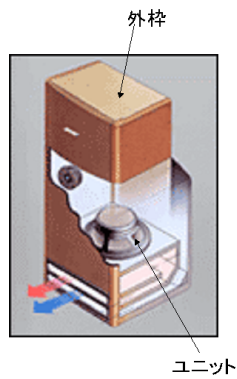


図 1 スピーカの内部構造表示

system:VWDB)」の設計・開発を行っている [5] ~ [8] .

現在、多くのバーチャルリアリティを利用したアプリケーションでは、仮想空間全体を対象にした問合せや、ある範囲を指定した問合せを行っている。これは、ユーザが仮想世界のどこに所望のオブジェクトがあるのか分からない状況や、所望のオブジェクトの位置までユーザをナビゲーションしてくれるシステムにおいては、有効であると考えられる。

しかし、例えば、スピーカを前にしたユーザが、スピーカ全体を見ながら、ユニットの部分はどの位置にあり、スピーカの上部はどのようになっているのか、下部はどのようになっているのか、というように細かい範囲指定を行いながらインタラクティブに問い合わせたい場合が考えられる(図1)。そこで、オブジェクトの部分をインタラクティブに問合せできるような新たな問合せ範囲指定方法が必要であると考えた。

新たな問合せの手法の導入のために、現実世界にある便利な道具を参考にできるのではないかと考えた時、思いついた物は虫眼鏡である。細かい字を読む時、細かい物を見る時など見たい物の上にかざす事によって、対象が大きく見やすくなる。また、人間の手で様々な角度や位置に移動する事ができる。そこで、虫眼鏡のように仮想世界にもユーザが自由に持ち運びができるような形状を持たせ、その内部に入ったオブジェクトにのみ問合せを行う手法を考えた。

問合せの範囲となる形状を球とした理由は、2つある。1つは、虫眼鏡で見る対象は2次元だが、仮想世界においては、3次元オブジェクトを形状の中に入れなければならない。そこで、3次元の球が良いと考えたからである。もう1つは、文献[9]において、3次元地図上の詳細表示させたい部分を入力デバイスを使って丸で囲む事例があり、丸で囲む範囲を3次元的に捕らえると、球状であると考えたからである。

そこで、本稿では、ユーザが仮想世界に対して細かい範囲指定を行いながらインタラクティブに問合せを行うための新しい問合せ手法として、QueryBallを提案する。QueryBallとは、問合せの範囲を球に見立てて仮想世界に表示し、問合せ領域であるこの球に対して探索条件と、探索条件に該当する(または該当しない)オブジェクトに対する表示方法を指定する事で、球の内部にあるオブジェクトにのみ問合せをかけることができる。またユーザは、仮想世界上でQueryBallを自由に移動させたり、大きさを変える事ができ、QueryBall中の問合せ結果を

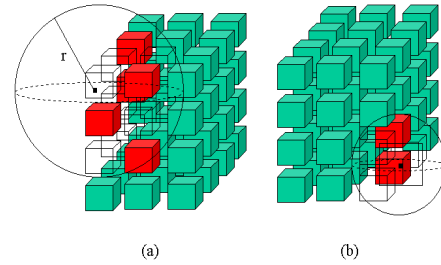


図 2 QueryBall 問合せの例

見ながらインタラクティブに問合せを行う事ができる。更に、複数のQueryBallを導入する事で、問合せの範囲や探索条件を絞り込む事ができる。

2. QueryBall の定義

QueryBall Q は、3つ組 $Q = (B, S, P)$ と定義する。ここに、 B は問合せ領域、 S は探索条件、 P は問合せ結果の表示条件とする。複数のQueryBallを区別する時は、 Q に添え字を付け、QueryBall Q_1 は、以下のように記述する。

$$Q_1 = (B_1, S_1, P_1)$$

問合せ領域 B は、QueryBallの中心と半径を持つソリッドオブジェクトの球とする。探索条件 S は、現在仮想世界データベースシステムで利用しているデータベースシステムであるObjectStoreが提供する問合せ述語とする。表示条件 P は、探索条件結果に該当するオブジェクトを強調表示または、非強調表示するかの指定を行う。また、探索条件に該当しないオブジェクトを透明表示するか、ワイヤフレーム表示にするかを指定する。

また、QueryBall Q の否定、これを \overline{Q} と書き、問合せ領域と問合せの結果の表示条件は否定されないが、探索条件のみ論理否定をする。QueryBall \overline{Q} とし、以下のように定義する。

$$\overline{Q} = (B, \neg S, P)$$

3. 単一の QueryBall による問合せ

図2(a)は、属性に "type" を持つ横3個×高さ4個×奥行き4個の緑色のボックスが積み重なった空間に対して、ユーザが以下のようにQueryBallを指定し、問合せを行う様子である。

- 問合せ領域
 - 半径 r のボール
- 探索条件
 - type == "A"
- 表示条件
 - 探索条件に該当するオブジェクト
 - 強調表示 (赤色にする)
 - 探索条件に該当しないオブジェクト
 - ワイヤフレーム

ユーザは、QueryBallを自由に動かしながら問合せを行う事

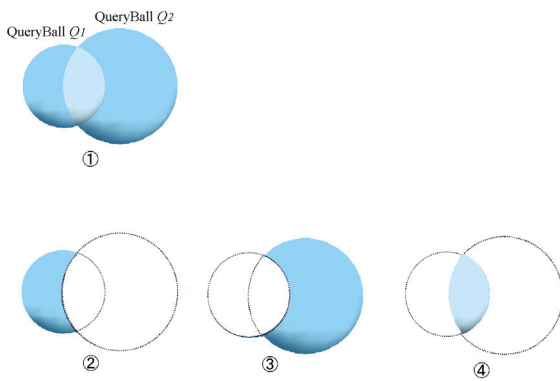


図 3 2 個の QueryBall の合体

ができる．図 2(b) は，ユーザが QueryBall の半径を小さくして，ユーザの見た位置（ここではボックスの山の右下）に移動させた様子である．

なお，システムの詳しい説明は，5 章で行う．

4. 複数の QueryBall の導入

4.1 2 個の QueryBall の合体

2 つの QueryBall を合体 (combine) させて，問合せを絞り込むことを考える．図 3 の①のような QueryBall $Q_1 = (B_1, S_1, P_1)$ と QueryBall $Q_2 = (B_2, S_2, P_2)$ の合体を次のように定義する．

$$Q_1 \cup Q_2 = (Q_1 - Q_2) \text{ UNION } (Q_2 - Q_1) \text{ UNION } (Q_1 \cap Q_2)$$

ここに， $Q_1 - Q_2$ ， $Q_2 - Q_1$ ， $Q_1 \cap Q_2$ は，図 3 の②，③，④に対応している問合せで，QueryBall Q_1 と QueryBall Q_2 の問合せ領域の重なり具合に対応して，次のように定義する．

- $Q_1 - Q_2$
 問合せ領域： $B_1 - B_2$ ここに $-$ は差集合演算子
 探索条件： S_1
 表示条件： P_1

$Q_2 - Q_1$ も同様に定義される

- $Q_1 \cap Q_2$
 問合せ領域： $B_1 \cap B_2$
 探索条件：探索条件 S_1, S_2 の 2 種類が重なるため，2 つの探索条件の関係が論理和 (\vee) または，論理積 (\wedge) のどちらをとるか指定する．

表示条件：表示条件 P_1, P_2 が 2 種類あるため，ユーザが指定しなければならない．本システムでは，QueryBall の生成順に優先度をつけ，合体した QueryBall の優先順位が高い方の結果表示条件をデフォルトとして採用する．また，それに満足しないユーザのために新たに表示条件を入力できるようにしている．今回の例では，最初に生成した QueryBall Q_1 の表示条件が優先度が高いので，表示条件は， P_1 となる．

θ は， $Q_1 \cup Q_2$ において， $Q_1 \cap Q_2$ で指定した探索条件の関係 (\vee または \wedge) を記す．なお，UNION は，各問合せの結果を並列に表示させる演算子とする．

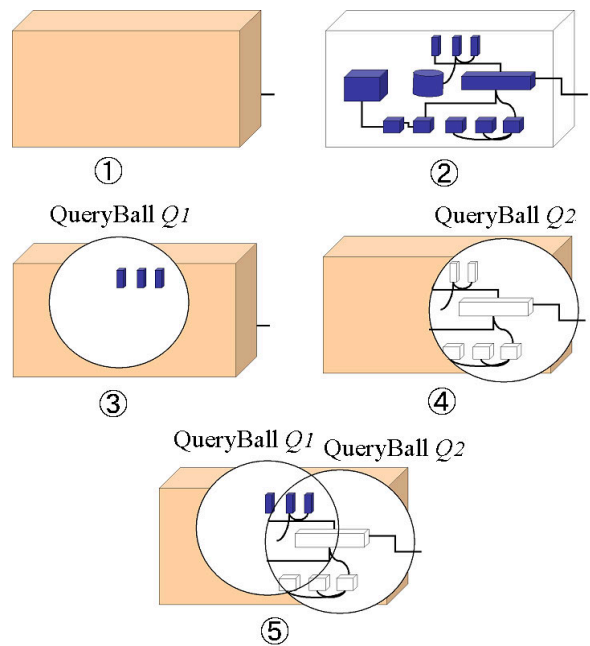


図 4 2 個の QueryBall を用いた問合せ例

まとめると， $Q_1 \theta Q_2$ は， $\theta = \vee, \theta = \wedge$ に対応して，以下のよう定義する．

論理和をとる場合 ($\theta = \vee$)

$$\begin{aligned} Q_1 \vee Q_2 &= (Q_1 - Q_2) \text{ UNION } (Q_2 - Q_1) \text{ UNION } (Q_1 \cap Q_2) \\ &= (B_1 - B_2, S_1, P_1) \text{ UNION } (B_2 - B_1, S_2, P_2) \text{ UNION } \\ &\quad (B_1 \cap B_2, S_1 \vee S_2, P_1) \end{aligned}$$

論理積を取る場合 ($\theta = \wedge$)

$$\begin{aligned} Q_1 \wedge Q_2 &= (Q_1 - Q_2) \text{ UNION } (Q_2 - Q_1) \text{ UNION } (Q_1 \cap Q_2) \\ &= (B_1 - B_2, S_1, P_1) \text{ UNION } (B_2 - B_1, S_2, P_2) \text{ UNION } \\ &\quad (B_1 \cap B_2, S_1 \wedge S_2, P_1) \end{aligned}$$

4.2 2 個の QueryBall を用いた問合せ例

前節では，2 つの QueryBall を合体させた問合せの説明を行った．本節では，どのような場合に 2 つの QueryBall を合体させた問合せを行うと有効か，例をあげて説明する．

図 4 の②のような部品が入った機械 (図 4 の①) があつたとする．ユーザは，部品 A の位置を知るために以下のように QueryBall Q_1 を指定して問合せを行った (図 4 の③)．

- 問合せ領域
 半径 r_1 のボール
- 探索条件
 部品名が A
- 表示条件
 - 探索条件に該当するオブジェクト
 非強調表示
 - 探索条件に該当しないオブジェクト
 透明度 100%

次にユーザがこの部品 A にはどのように配線が行われていたか知りたかつたとする．以前利用した QueryBall Q_2 は以下のよ

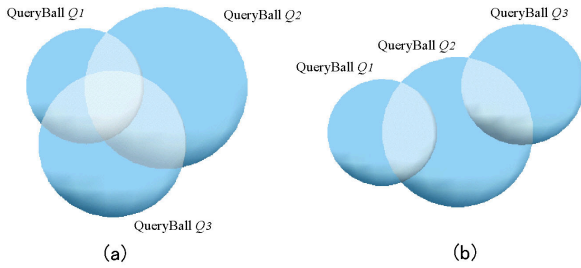


図 5 3 個の QueryBall の合体

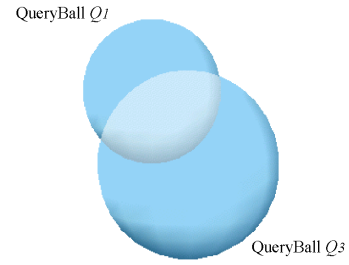


図 6 QueryBall Q_2 の削除

うに配線を見られる問合せだった(図4の④)ので, QueryBall Q_2 を再び仮想世界に生成し, QueryBall Q_1 の部品 A が表示されている部分に QueryBall Q_2 を合体させ,重なった部分の探索条件に論理和を設定すると(図4の⑤)のように部品 A からどのように配線が行われているかわかる. また, この2つの QueryBall を合体させた問合せをユーザが所望の位置に移動したり, 大きさを変更する事により, 配線がどこまでつづいているかなど見る事ができる.

- 問合せ領域
 - 半径 r_2 のボール
- 探索条件
 - 部品名が線材
- 表示条件
 - 探索条件に該当するオブジェクト
 - 非強調表示
 - 探索条件に該当しないオブジェクト
 - ワイヤフレーム

4.3 3 個以上の QueryBall の合体

QueryBall Q_1 と QueryBall Q_2 を合体させた問合せに QueryBall Q_3 を合体させる事について述べる. 図5の(b)のように, QueryBall Q_1 と QueryBall Q_2 が重なっていない部分に QueryBall Q_3 を合体させる場合も考えられるが, QueryBall Q_1 と QueryBall Q_2 の重なった部分がユーザが注目している問合せ領域であると考え, そこに QueryBall Q_3 を合体させることによって探索条件を絞り込む事が, ユーザの当面の目的であると考えている. したがって, 図5の(a)のような場合についてのみここでは考える.

QueryBall Q_1 と QueryBall Q_2 を合体させ,重なった部分の探索条件に論理和を指定した問合せに第3番目に生成した QueryBall $Q_3=(B_3, S_3, P_3)$ を合体させ,重なった部分の探索条件に論理積を指定したとすると, 以下のように記述できる.

$$\begin{aligned}
 & (Q_1 \cup Q_2) \cap Q_3 \\
 &= (Q_1 - Q_2 - Q_3) \cup (Q_2 - Q_1 - Q_3) \cup (Q_3 - Q_1 - Q_2) \cup ((Q_1 \cap Q_2) - Q_3) \cup ((Q_1 \cap Q_3) - Q_2) \cup ((Q_2 \cap Q_3) - Q_1) \cup ((Q_1 \cap Q_2) \cap Q_3) \\
 &= (B_1 - B_2 - B_3, S_1, P_1) \cup (B_2 - B_1 - B_3, S_2, P_2) \cup (B_3 - B_1 - B_2, S_3, P_3) \cup ((B_1 \cap B_2) - B_3, S_1 \vee S_2, P_1) \cup ((B_1 \cap B_3) - B_2, S_1 \wedge S_3, P_1) \cup ((B_2 \cap B_3) - B_1, S_2 \wedge S_3, P_2) \cup (B_1 \cap B_2 \cap B_3, (S_1 \vee S_2) \wedge S_3, P_1)
 \end{aligned}$$

図5の問合せは, QueryBall の重なり具合に着目して, $Q_1 - Q_2 - Q_3, Q_2 - Q_1 - Q_3, Q_3 - Q_1 - Q_2, (Q_1 \cap Q_2) - Q_3, (Q_1 \cap Q_3) - Q_2, (Q_2 \cap Q_3) - Q_1, (Q_1 \cap Q_2) \cap Q_3$ の7つの問合せに分ける. また, 3個以上の問合せを記述も2個の場合と同様に QueryBall の重なり具合に着目し, 最も細かく分けて記述するものとする.

それぞれの問合せに対して, 問合せ領域, 探索条件, 表示条件の3つ組みになるように書き下しを行った. 問合せ領域は, 空間的な集合演算を行い, 探索条件は述語演算が行える. 表示条件は, QueryBall が重なっている場合, QueryBall の優先度が高い方の表示条件を採用するのがデフォルトである.

ここで, 注意する点は, 問合せ $(Q_1 \cap Q_3) - Q_2$ と $(Q_2 \cap Q_3) - Q_1$ における探索条件である. それぞれの探索条件は, 探索条件の述語演算を展開したものを採用する. つまり, $Q_1 \cup Q_2$ の探索条件 $S_1 \vee S_2$ と QueryBall Q_3 の探索条件 S_3 の関係は論理積と指定されているので, $(S_1 \vee S_2) \wedge S_3$ となり, これを展開すると $(S_1 \wedge S_3) \vee (S_2 \wedge S_3)$ となるので, $(Q_1 \cap Q_3) - Q_2$ の探索条件は, $S_1 \wedge S_3, (Q_2 \cap Q_3) - Q_1$ の探索条件は, $S_2 \wedge S_3$ となる.

4.4 QueryBall の削除

QueryBall がなくなっただけの場合, QueryBall を削除する事も考えられる. 図6のように, 前節で利用した QueryBall $Q_1 = (B_1, S_1, P_1)$ と QueryBall $Q_2 = (B_2, S_2, P_2)$ と $Q_3 = (B_3, S_3, P_3)$ を合体させた問合せから, QueryBall Q_2 を削除した場合, 以下のように記述する. なお, DELETE, 問合せから QueryBall を削除する演算子とする.

$$\begin{aligned}
 & ((Q_1 \cup Q_2) \cap Q_3) \text{ DELETE } Q_2 \\
 &= Q_1 \cap Q_3 \\
 &= (Q_1 - Q_3) \cup (Q_3 - Q_1) \cup (Q_1 \cap Q_3) \\
 &= (B_1 - B_3, S_1, P_1) \cup (B_3 - B_1, S_3, P_3) \cup (B_1 \cap B_3, S_1 \wedge S_3, P_1)
 \end{aligned}$$

図6を見ると分かるように, QueryBall Q_2 を削除する事によって, QueryBall 問合せの形状が変化する. 問合せの重なり具合に着目して, 問合せを分けると, $Q_1 - Q_3, Q_3 - Q_1, Q_1 \cap Q_3$ となる.

これは, 正に第4.1節で述べた2個の QueryBall を合体させた問合せと同じになる.

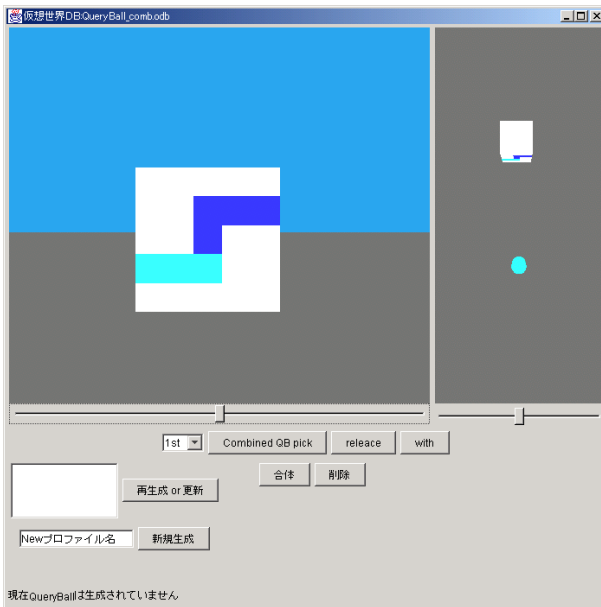


図 7 メインウィンドウ

5. QueryBallシステムの実装

5.1 実装環境

プラットフォームには、DynaBook^(注1) DB60P/4DA2(RAM: 128MB, CPU: Mobile PentiumIII^(注2) 600MHz, OS: Windows2000Professional^(注3))を用いた。また、フロントエンドを JAVA 言語用の仮想世界構築 API である JAVA3D を利用して構築し、バックエンドに商用のオブジェクト指向データベース管理システムであるエクセロン社の ObjectStore^(注4)を利用して QueryBall システムの実装を行った。

5.2 QueryBall システムの概要

ユーザが Query Ball システムを起動すると、データベースシステムに格納された仮想世界を読み出して画面上に仮想世界を作り出す(図 7)。図 7 は QueryBall システムのメインウィンドウである。上の 2 つのウィンドウは、仮想世界を表示しており、左上のウィンドウが仮想世界に没入したユーザの化身であるアバタから見た仮想世界、右上のウィンドウは、鳥瞰的に見た仮想世界をそれぞれ表示している。また、右上のライトグリーンの円柱は、仮想世界にいるアバタの現在位置を示している。

ユーザは、マウスのドラッグ操作を行う事で仮想世界をウォークスルーし、画面上に現れた仮想世界オブジェクトを視認することができる。

メインウィンドウ(図 7)の下部分は、QueryBall の生成、問合せ内容の更新、削除、選択などの操作をするための GUI である。この GUI で選択された機能別に新たに GUI が立ち上がり、所望の操作を入力することができる。また、仮想世界に生成された QueryBall をこの GUI で選択し、マウスのドラッグ操

(注 1): DynaBook は株式会社 東芝の商標である

(注 2): Pentium は Intel 社の商標である

(注 3): 米国 Microsoft Corporation の米国及び他の国における登録商標である

(注 4): ObjectStore はエクセロン社の商標である

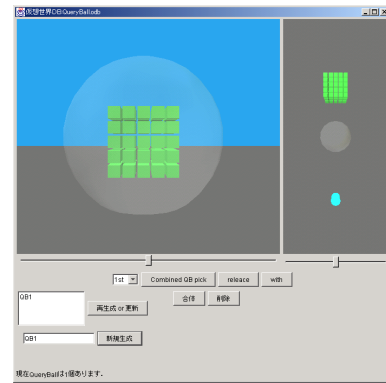


図 8 単一の QueryBall 問合せに使用した仮想世界

作を行う事で仮想世界内で QueryBall を動かす事ができる。

ユーザが仮想世界において QueryBall を移動させる事により、問合せ領域が時事刻々と変化するため問合せも逐次データベースシステムに発行していかなければならない。そこで本システムでは、JAVA のメソッドである Behavior を用いて、5 秒毎にデータベースシステムに問合せを発行している。

5.3 単一の QueryBall による問合せの実例

単一の QueryBall による問合せの実例で利用した仮想世界、QueryBall について以下に述べる。

仮想世界

非空間的属性の属性名に "type" を持ち、緑色で一辺が 50cm の立方体の仮想世界オブジェクトのインスタンスを横 5 個 × 高さ 5 個 × 奥行き 5 個に山積みにした仮想空間を用いた。積み重なった角にあたる部分と、左から 2 列目、下から 3 番目、右から 2 番目下から 3 番目の "type" の属性値を A とし、それ以外の "type" の属性値は、C とした(図 8)。

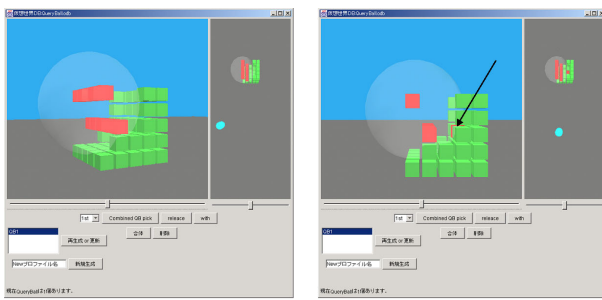
QueryBall

"type" の属性値が A のオブジェクトがどこにあるか分からないユーザが、以上で述べた仮想世界に存在するオブジェクトから "type" の属性値 A のオブジェクトを探すために以下のような QueryBall を生成して問合せを行ったとする。

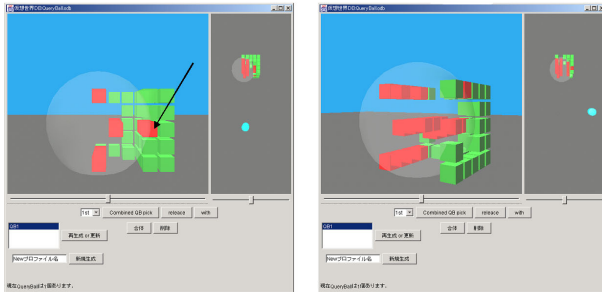
- 問合せ領域
 - 半径 2 メートルのボール
- 探索条件
 - type == "A"
- 表示条件
 - 探索条件に該当するオブジェクト
 - 強調表示
 - 探索条件に該当しないオブジェクト
 - 透明度 100%

図 9 は、問合せを行っている様子である。

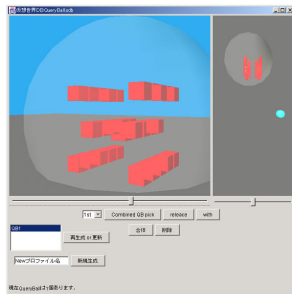
図 9(a) は、ユーザが、積み上げられたオブジェクトの左上に QueryBall を移動しているところである。また、ユーザも左に移動している。積み上げられたオブジェクトの左上角には、"type" の属性値が A であるオブジェクトが並んでおり、また、左から 2 列目、上から 3 行目にも "type" の属性値が A である



(a) (b)



(c) (d)



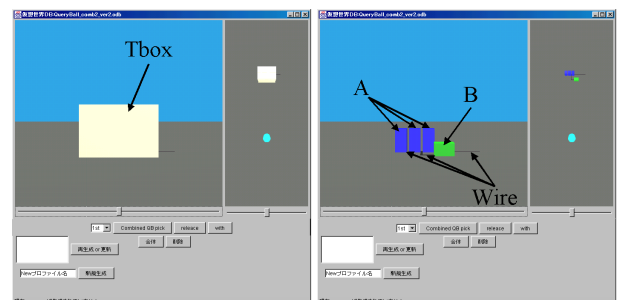
(e)

図 9 単一の QueryBall 問合せを行っている様子

オブジェクトが並んでいるので、QueryBall の問合せ領域内に入っているそれらが赤く強調表示されている事がわかる。一方、QueryBall の問合せ領域内に入っているが、“type”の属性値が A でないオブジェクトは、透明度が 100%となり、QueryBall 内に入っているそれらのオブジェクトは、消えている事が分かる。

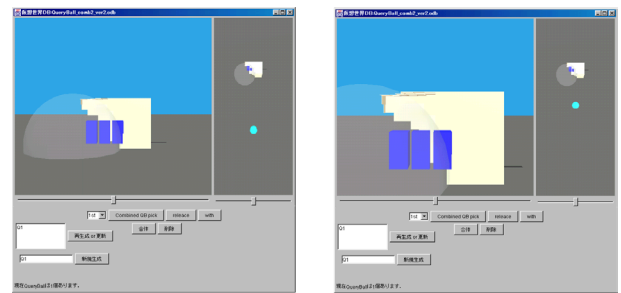
図 9(b) は、ユーザ(アバタ)が中央まで戻り、少し QueryBall を右に移動させた様子である。右から 2 列目、上から 3 行目のオブジェクトも “type”が A であるので、(a)の時に比べ QueryBall の問合せ領域内に入ったオブジェクト 1 個が赤く強調表示されている事がわかる(矢印)。

図 9(c) は、(b)と同じ位置で、QueryBall を右下に移動させた様子である。右下角のオブジェクトも “type”の属性値が A であるので、QueryBall の問合せ領域内に入ったオブジェクトが赤くなった事がわかる。また、QueryBall を右下に移動させたので、QueryBall の問合せ領域内に入る右から 2 列目、上から 3 行目のオブジェクトが増えた事も分かる。ここで、矢印の部分のオブジェクトは、QueryBall の問合せ領域に完全に入っていないが、本システムでは、QueryBall の中心と、オブジェクトの重心との距離が、QueryBall の半径以下であるものが問



(a) (b)

図 10 QueryBall2 個の合体による問合せ例 2 に使用した仮想世界



① ②

図 11 Q_1 を用いて問合せを行った様子

合せ領域内に入っているものとしているので、強調表示されている。

図 9(d) は、QueryBall を更に右に移動させ、ユーザ(アバタ)も右に移動し、積み上げられたオブジェクトを見ているところである。右上角も “type”の属性値が A であるので、1 つ QueryBall の問合せ領域内に入り、強調表示されている事がわかる。

図 9(e) は、QueryBall の問合せ領域をスライダを用いて大きくした様子である。積み上げられたオブジェクトをすっぽりと覆う程 QueryBall の問合せ領域が大きくなったため、左上、左下、右上、右下の角、左から 2 列目、上から 3 行目、右から 2 列目、上から 3 行目に位置する “type”の属性値が A のオブジェクトがすべて赤く強調表示され、それ以外は、消えている事がわかる。

5.4 QueryBall2 個の合体による問合せの実例

QueryBall2 個の合体による問合せを行う。属性名に “parts” を持ち、直方体の箱の中に部品が入っている仮想世界オブジェクトのインスタンスを 1 つ仮想世界に存在させた(図 10(a))。直方体の箱の属性値を Tbox とした。図 10(b) は、この外側の直方体の箱を取り除いて、中の部品のみ見られるようにしたものである。青い縦長の直方体の “parts”の属性値を A、緑色の横長の直方体の “parts”の属性値を B、黒いコードの “parts”の属性値を Wire とした。

この直方体の箱のどの部分に部品 A があるのか知りたいユーザは、以下のような QueryBall Q_1 を生成した。

- 問合せ領域
半径 2 メートルのボール
- 探索条件

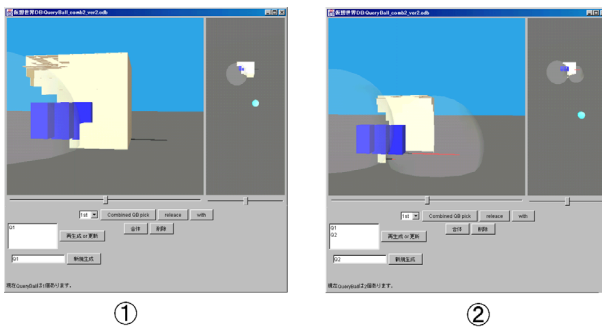


図 12 Q₂ を仮想世界に生成する前と生成後問合せを行っている様子

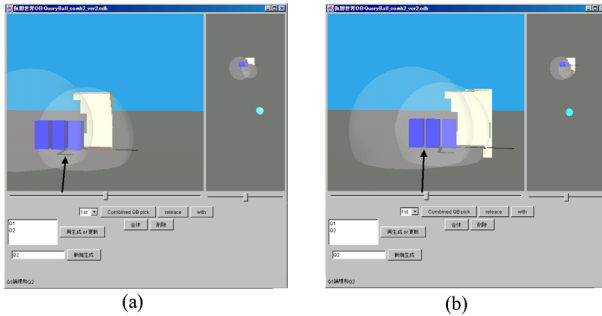


図 13 Q₁ と Q₂ を合体させて問合せを行った様子

```
parts== "A "
```

- 表示条件
 - 探索条件に該当するオブジェクト
 - 非強調表示
 - 探索条件に該当しないオブジェクト
 - 透明度 100%

図 11 は、Q₁ を用いて問合せを行っている様子である。①は、遠くから見た様子。②は、近くから見た様子である。部品 A が見え、その他のオブジェクトが消えている事がわかる。

次に、ユーザが右へ移動すると、箱から出ているコードがある事に気付いた(図 12①)。そこで、部品 A につながっているコードを見たいと考え、以下のような QueryBall Q₂ を生成した。

- 問合せ領域
 - 半径 1.5 メートルのボール
- 探索条件


```
parts== "Wire "
```
- 表示条件
 - 探索条件に該当するオブジェクト
 - 強調表示
 - 探索条件に該当しないオブジェクト
 - 透明度 100%

図 12②は、ユーザがオブジェクトに近付き、Q₂ を仮想世界に生成し問合せを行っている様子である。Q₂ 内部に入り、” parts ”が Wire のオブジェクトは強調表示され、以外のオブジェクトは、透明に表示されている事が分かる。

Q₁ と Q₂ を合体させ、重なった部分の探索条件を論理和として問合せを行ったのが、図 13 である。なお、この場合の表示方法は、優先度の高い Q₁ の表示方法としている。

図 13(a) は、右から Q₂ を移動させて Q₁ に合体させた様子である。矢印の部分に、今まで見られなかったコードが表示されたことがわかる(図 13(a) 矢印)。

図 13(b) は、ユーザと Q₂ が左に移動して問合せを続けている様子である。図 13(a) では見られなかったコード(図 13(b) 矢印)が Q₂ を移動させた事によって表示されている事が分かる。

6. 結 論

6.1 ま と め

本研究では、仮想世界データベースシステムにおいて、ユーザが仮想世界に対して細かい範囲指定を行いながらインタラクティブに問合せを行うための新しい問合せ手法として、QueryBall の提案とシステムの実装を行った。

QueryBall は、仮想世界に問合せ領域に見立てた球を生成し、それを移動させてその内部に入ったオブジェクトにのみ問合せを行う方法である。この QueryBall は、問合せ領域(球の半径)、探索条件、表示条件で構成され、ユーザがそれぞれを指定するものとした。また、QueryBall を複数仮想世界に生成し、合体させる問合せ方法も提案した。そして、単一の QueryBall に関する問合せと、QueryBall を合体させた問合せの有効性について論じた。

さらに、QueryBall システムを現在我々が開発している仮想世界データベースシステム VWDB の一機能として実装した。現在のシステムでは、単一の QueryBall による問合せと QueryBall を 2 個用いて合体させた問合せを行う事ができる。また、本システムを用いて、単一の QueryBall を用いた問合せの実例と、2 個の QueryBall を合体させた問合せの実例の紹介を行った。

6.2 今後の課題

6.2.1 実行性能の向上

現在、一定間隔(5秒毎)でデータベースシステムに問合せを発行している。QueryBall を複数生成した場合、処理速度が著しく低下する場合があった。そこで、前回の問合せ結果オブジェクトと現在の問合せ結果オブジェクトと照らし合わせ、前回描画したオブジェクトは、再描画を行わないという方法を用いて実行性能の向上を行っている。

今後、更に実行性能の向上を行うために、以下の 2 つの方法の導入が考えられる。

1. 前回の問合せに該当したオブジェクトに対してデータベースシステムに再度問合せを行わない方法
 2. QueryBall の移動距離が長い場合は問合せ発行間隔を短くし、移動距離が短い場合は問合せ発行間隔を長くする方法
- この 2 つの方法を検証し、システムに導入する。

6.2.2 QueryBall の形状

本研究では、問合せ範囲を球に見立てて QueryBall 問合せを行ってきた。球の形状の利点は、パラメタが半径しかないため、生成が容易で、かつ大きさの変更も容易であるという点である。また、パラメタの指定が容易な分、仮想世界における大きさや距離感を得やすいという利点もある。

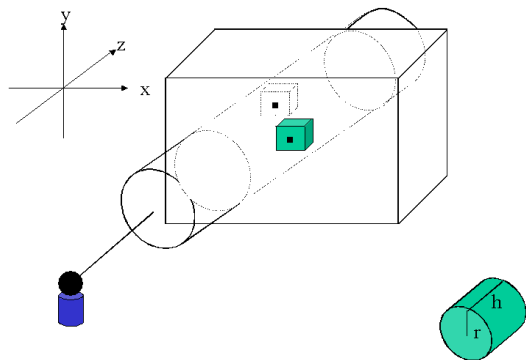


図 14 円柱を利用して問合せの方法

一方で、パラメタが 1 つしかない事から、奥行きのあるオブジェクトに対して問合せを行う場合、QueryBall がめりこんでしまい、ユーザと球の間にあるオブジェクトで視界がふさがれてしまう欠点がある。これを回避しようとする、オブジェクトの奥行きに合わせて半径を大きくした球を生成するか、QueryBall がオブジェクトからはみ出しているならば、ユーザがそこまで移動しなければならなくなる。

奥行きのあるオブジェクトに対する問合せには、円柱を問合せ範囲として利用する方法が考えられる(図 14)。半径パラメタに加え、高さパラメタを加える事で、高さを大きくする事で、奥行きのあるオブジェクトを貫通する事ができ、なおかつ半径パラメタの部分で部分的な範囲指定も行えると考えられる。

しかし、円柱を生成する場合、パラメタが増えた分、逆に生成や、大きさ変更が複雑となり、また、仮想世界に対する円柱の大きさや距離感を得にくいという欠点もある。

形状には、利点・欠点があるため、問合せを行う仮想空間や、仮想世界オブジェクトに合わせて、ユーザの要望にあった問合せ範囲を表す形状を提供する事が大切であると考えられる。

そこで、利用できると思われるのは、CG で行われる技術である CSG (Constructive Solid Geometry) である。これは、基本的な物体 (プリミティブ) を和集合、積集合、差集合といった演算で組み合わせて形状のモデリングをする技術である。図 15 は、球 (オブジェクト A)、円柱 (オブジェクト B) を用いて、CSG 技術を利用して形状モデリングを行った例である。

QueryBall システムで基本的な形状と CSG 技術を用いた形状モデリングが行える機能を提供すれば、ユーザがそれを利用して要望にあった問合せ範囲を表す形状を生成する事ができると考えられる。

6.2.3 マルチモーダルインタラクションの導入

現在 QueryBall システムでは、マウスのドラッグ操作を行って QueryBall を移動させている。マウスのドラッグ操作は、一定の操作方法があるため、思い通りに QueryBall を移動させる方法を習得するまでにユーザに少し負担がかかると考えられる。また、現在、合体させた後のオブジェクトを合体させた形状のまま移動させる事は可能であるが、マウスのドラッグ操作の制約により、1 つの QueryBall を選択して移動させている間

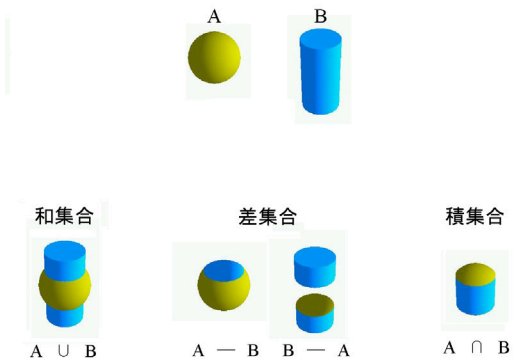


図 15 CSG 技術を用いたモデリング例

は、他の QueryBall を選択して移動させる事ができない。

そこで、ユーザの思い通りに QueryBall を移動させるためには、データグローブを両手に装着し、両手を使って同時に QueryBall を移動できる方が効率よく合体ができると考えられる。

また、現在、GUI の画面を用いて探索条件や表示条件を入力しているが、音声認識による入力を用いれば、仮想世界にいらがらにして問合せを行う事ができる。

よって、QueryBall システムにマルチモーダルインタラクションを導入する事は有意義であると考えられる。

文 献

- [1] Kamiura, M., Oiso, H., Tajima, K. and Tanaka, K.: Spacial Views and LOD-based Access Control in VRML-object Database, *Worldwide Computing and Its Application*, pp.210-225, LNCS 1274, Springer(1997).
- [2] Gausemeir, J., Krumm, H., Molt, T. A database driven server for an Internet based plant layout presentation system, *Proceedings of the fifth symposium on Virtual reality modeling language (Web3D-VRML)*, pp.17 - 22, (2000).
- [3] Alex van Ballegooij and Anton Wliens: Navigation by Query in Virtual Worlds, *Proceedings of the sixth international conference on 3D Web technology*, pp.77-83, (2000).
- [4] "Three-dimensional Restitution via Internet of Digital Elevation Networks in Towns" <http://ias.trident3d.net/ucm/demos.html>, (2003).
- [5] Watanabe, C. and Masunaga, Y.: VWDB2: A Network Virtual Reality System with a Database Function for a Shared Work Environment, In *Proceedings of the IASTED Information Systems and Databases Conference (ISDB)*, pp.190-196 (2002).
- [6] 渡辺知恵美, 大杉あゆみ, 佐藤こず恵, 増永良文: 仮想世界データベースシステム: VWDB における共有型作業環境のためのトランザクション概念の導入, *情報処理学会論文誌*, Vol.43, SIG9(TOD15), pp.55-67 (2002).
- [7] Watanabe, C. and Masunaga, Y.: A Framework for a Multimodal Query Interaction of the Virtual World Database System, In *Natural Science Report of the Ochanomizu University*, Vol.53, No. 1, pp.101-106 (2002).
- [8] 渡辺知恵美, 大杉あゆみ, 佐藤こず恵, 増永良文: VWDB: 仮想世界データベースシステム-基本的な概念とシステム構成-, *情報処理学会第 62 回全国大会講演論文集*, 2W-1(2001).
- [9] 高橋 和子, 竹沢 寿幸, (ATR): マルチモーダル対話システムにおけるインタラクション機構, *Proceedings of Interaction'99*, pp. 89-97 (1999).