Queryball：没入型VRシステムのための新しい問合せモデル

渡辺知恵美† 大杉あゆみ‡ 増永良文* 城和貴†
†奈良女子大学大学院 人間文化研究科
‡セコム株式会社 ＊お茶の水女子大学 理学部

要旨：
本稿では、没入型VRシステムにおける新しい問合せの1手法としてQueryballを提案する。Queryballは半透明の球体であり，探索条件および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持つ。ユーザは生成されたQueryballを仮想世界のオブジェクトに対して重ね合わせることで，Queryballの中にあるオブジェクトに対して問合せを適用させることができる。また、ボールを持って動かす、大きさを変える、複数のQueryballを組み合わせるなどという直感的な操作により試行錯誤的に問合せを行うことができる。

Queryball：A New Model for Querying in Immersive VR sytems

Chiemi Watanabe† Ayumi Osugi‡ Yoshifumi Masunaga* Kazuki Joe†
†Nara Women's University, Graduate School of Humanity and Science,
‡Secom Co. LTD,
*Ochanomizu University, Faculty of Science

Abstract— **In this paper, we propose a powerful and convenient querying model, "Queryball", for users to query in immersive VR systems. Queryball is a new style of querying model for intuitive and heuristic query interaction. Queryball, which is a translucent ball, is defined as a quadruple of area specification, search condition, and two types of display methods. The search condition can be applied only to the virtual objects inside of the Queryball. The query results are shown to the user according to the display method. The position and radius of the Queryball changes when the user moves the Queryball interactively with observing the query results.**

1. Introduction

In recent years, visualization systems for scientific data are becoming indispensable. 3D visualization is used in various application fields because users can watch objects from various viewpoints by "walking through" to understand intuitively[3]. Especially, immersive VR systems are expected as an important visualization tool for intuitive understanding by virtual experience where users enter into 3D visualized space to look around the visualized objects[2].

Besides the effective presentation tool, immersive VR systems are also expected as a visual data mining[4] tool, which explores unreachable knowledge by vision and intuition. Interactive analysis process is the key part for visual data mining since new knowledge is explored by human vision and intuition. In an interactive analysis process, a user tries to discover new knowledge from 3D visualized simulation or real data by changing his/her viewpoints and applying other visualization methods interactively, intuitively, and freely. Such user interactive operations can be considered some query interaction. Thus, database query interaction in immersive VR systems is very useful and essential for the interactive analysis process. However, useful query interaction tools are not provided for immersive VR systems. Without typical input devices for computers such as keyboard and mouse, it is difficult to ask the immersive VR systems for queries with easy operations using a pair of data gloves or a stylus.

We propose a powerful and convenient query model "Queryball" for users to query in an immersive VR system. Fig.1 shows an image of a tool using Queryball. Queryball, which is a translucent ball, is defined as a quadruple of area specification, search condition and two types of display methods. The search condition can be applied only to the virtual objects inside of a Queryball. The query results are visualized according to the specified display methods. By locating a Queryball on the region of interest, a user can observe just the objects he/she is interested in with emphasizing them and having other objects disappear. He/she can change the region of interest by changing the area specification of the Queryball interactively.

Furthermore, by the simple operations of overlapping multiple Queryballs, users can make a more complicated query that corresponds to his/her intention interactively.

In this paper, we define the Queryball query model and describe a prototype system. In section 2, we describe a prototype system as the first step for the query model using Queryball with some examples of query interaction by Queryball. In section 3, we give formal definitions of the Queryball query model.

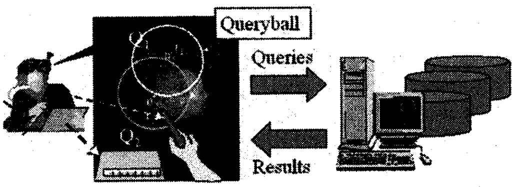

Fig. 1. Overview of Queryball System

2. Prototype System

Although Queryball makes a most sense for immersive VR systems, Queryball is effective for simplified VR applications (not immersive, no special input devices) such as Web3D. We implement the first prototype system using a simplified VR application. We call the prototype system *Queryball system for non-immersive VR*. We have implemented this VR application by JAVA3D, and we use objectstore PSE for JAVA, which is an object-oriented database system, as the back-end DBMS.

2.1 Query Interaction by using Single Queryball

In this example, we assume a virtual world of 5x5x5 green boxes. On the back-end database, we define a class schema for the boxes as shown Fig.2. Each box class object has the attribute of *pos*, which means the center position of the object, and the attributes of *val* and *val2*. There are some boxes of which attribute *val* is more than ten while the attribute *val* of the other boxes is less than or equal to ten. Fig. 3 shows the state of query interaction by a single Queryball, which shows the boxes whose attribute *val* is more than ten with emphasizing the boxes by coloring red. We can see the state change that the other boxes, of which attribute *val* is less than or equal to ten, disappear in the Queryball.

2.2 Query Interaction by Overlapping Several Queryballs

Next, we show an example of query interaction by multiple Queryballs. We show an example that a big white box "Tbox" covers the 125 green boxes of the previous example. The class definition of Tbox is the same as the Box class. We prepare three Queryballs as follows:

- Q_1: Transparentize the part of Tbox that covers green boxes. In order to peer the green boxes without transparentising the whole figure of Tbox, the search condition is sent toward the surface of Tbox. Transparentize all surfaces inside the Q_1.
- Q_2: Display the boxes whose attribute *val* is more than ten and emphasis the boxes by coloring red.
- Q_3: Display the boxes whose attribute *val2* is less than fifty and emphasis the boxes by coloring red.

Fig. 4 shows the state of the virtual world during applying Q_1, Q_2 and Q_3.

When Q_1 is overlapped on the Tbox, the surfaces inside of the Q_1 are transparentized, and we can see several green boxes inside of Tbox. Next, Q_2 is overlapped on the Q_1. Several boxes are emphasized by coloring red and the other boxes are transparentized. Furthermore, when Q_3 is overlapped on the Q_1 and Q_2, Q_3 narrows down the search condition of one of Q_2; the boxes whose attribute *val* is more than ten and attribute *val2* is less than fifty are emphasized while the other boxes are transparentized.

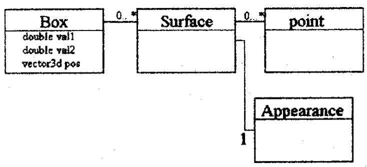

Fig.2 Class Hierarchical for Box object

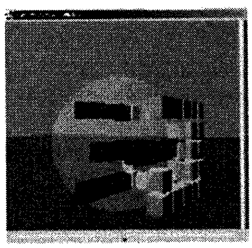

Fig.3 Query interaction using single Queryball

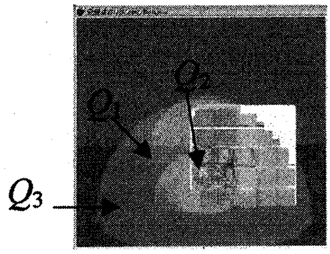

Fig.4 Superposing Multiple Queryballs Q_1, Q_2 and Q_3

3. Definitions for Queryball

3.1 Queryball

Queryball Q is defined by a quadruple $Q = (B,S,Ps,Pu)$, where B is area specification, S is search condition, Ps is a display method for the objects which are explored with S, and Pu is a display method for the objects which are not explored with S. Queryballs are distinguished by subscript numbers as

$$Q_i=(B_i,\ S_i,\ Ps_i,\ Pu_i).$$

Area specification B_i is represented as $dist(pos,O_i)<r_i$, where O_i is the center position of Q_i, r_i is the radius, *pos* is the center position of an object in the virtual world. Search condition S_i describes the way to issue to a back-end database system. The query descriptions are written with a first order logic. To query the Box objects of which attribute "*val*" is larger than 10, S_i is described as

$$(^{\exists}b)(Boxes(b) \wedge b.val>10). \qquad (1)$$

Ps and *Pu* describe the way to display the objects that are explored and not explored with the search condition, respectively. Each method is a user or system defined procedure. Our prototype system provides the following system defined procedures.

emphasize(col)*: emphasize the objects by changing their color to col*

transparency(alpha)*: set transparency value of the objects to alpha*

wireframe()*: display the objects by a wireframe model.*

For example, Queryball Q_j, which is represented as description (2), emphasizes the explored objects whose attribute "val" is larger than 10 by coloring red, with displaying the other (non-explored) objects by wireframe.

$$Q_j=(dist(pos,O_j)<r_j, (^{\exists}b)(Boxes(b) \wedge b.val>10), emphasize(255,0,0), wireframe()) \qquad (2)$$

3.2 Superpose two Queryballs

Multiple Queryballs can be superposed to interactively narrow and/or expand explored objects. Namely, combinations of primitive operations (superposing multiple Queryballs) make the users obtain more complicated query results by interest.

For example, a user superposes a Queryball Q_A on a Queryball Q_B so that he/she can narrow down the search condition of Q_B by the search condition of Q_A, or he /she can overlap the query result of Q_A and Q_B.

A query region is a set of the objects that are explored by one or more superposed Queryballs. When Q_j is superposed on Q_i, we define $Q_i{}^{\theta}{}_{\cup}Q_j$ as

$$Q_i{}^{\theta}{}_{\cup}Q_j=(Q_i\text{-}Q_j)\text{UNION}(Q_j\text{-}Q_i)\text{UNION}(Q_i{}^{\theta}{}_{\cap}Q_j) \qquad (3)$$

where Q_i-Q_j, Q_j-Q_i and $Q_i{}^{\theta}{}_{\cap}Q_j$ correspond to Fig.5 (a), (b), (c), respectively. Note that θ represents disjunction($\vee$) or conjunction($\wedge$).

Query region R is formally defined as a list of an area specification and some pairs of a search condition and a display method.

$$R = (B, (S_1,P_1),\ldots,(S_{n-1},P_{n-1}),(\neg S_n,P_n)) \qquad (4)$$

The region R_i which is given by a Queryball $Q_i=(B_i, S_i, Ps_i, Pu_i)$ is expressed as:

$$R_i=(B_i,(S_i,Ps_i),(\neg S_i,Pu_i)). \qquad (5)$$

$Q_i{}^{\theta}{}_{\cup}Q_j$ expresses a superposing relationship between Queryballs. The relationship can be illustrated as a tree structure.

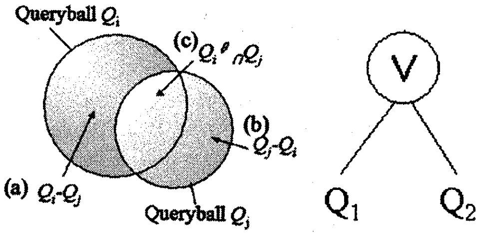

Fig.5 $Q_1{}^{\theta}{}_{\cup}Q_2$

As described above, $Q_i{}^{\theta}{}_{\cup}Q_j$ consists of three parts of query regions: $Q_i - Q_j$, $Q_j - Q_i$, and $Q_i{}^{\theta}{}_{\cap}Q_j$. These three regions have the following area specifications, search conditions and display methods.

- $Q_i - Q_j$
  - area specification: $B_i \wedge \neg B_j$
  - pairs of search condition and display method: $(S_i,Ps_i),(\neg S_i,Pu_i)$
- $Q_i{}^{\theta}{}_{\cap}Q_j$
  - area specification : $B_i \wedge B_j$

There are two cases for pairs of search condition and display method: $\theta = \vee$, and $\theta = \wedge$.

In the case of $Q_i{}^{\vee}{}_{\cap}Q_j$, superposing Q_j on Q_i by disjunction is regarded as the overlay query result by Q_j on the query result by Q_i. A straightforward expression for pairs of search condition and display method is shown below.

$$(S_i, Ps_i), (\neg S_i, Pu_i), (S_j, Ps_j), (\neg S_j, Pu_j)$$

The above expression has two problems:

1. When an object satisfies S_i and not S_j, the object is displayed as Pu_2 to disappear in the most case.
2. When an object satisfies both S_i and S_j, the object must be displayed with another method rather than Ps_i and Ps_j.

To solve the first problem, the above expression is modified as shown below

$$(S_i, Ps_i), (S_j, Ps_j), (\neg(S_i \vee S_j), Pu_i) \quad (6)$$

This formula applies display method Pu_i to the objects that satisfy neither S_i nor S_j. To solve the second problem, formula (6) is extended to formula (7).

$$(S_i \wedge \neg S_j, Ps_i), (S_j \wedge \neg S_i, Ps_j), (S_i \wedge S_j, Ps_k), (\neg(S_i \vee S_j), Pu_i) \quad (7)$$

Ps_k is provided by the system automatically. For example, suppose Ps_i and Ps_j emphasize the query results by coloring red (emphasize (255,0,0)) and blue (emphasize(0,0,255)), respectively. In this case, the system automatically defines Ps_k emphasizing the query results by coloring yellow (emphasize(255,255,0)) for the objects that satisfy both S_i and S_j.

In the case of $Q_i{}^{\wedge}{}_{\cap}Q_j$, superposing Q_j on Q_i using conjunction is regarded as narrowing down the search condition S_i by the search condition S_j. The pair of search condition and display method is:

$$(S_i \wedge S_j, Ps_i), (\neg(S_i \wedge S_j), Pu_i) \quad (7)$$

3.3 Superpose more than three Queryballs

In this subsection, we define the overlap of more than three Queryballs. When n Queryballs overlap, they generate up to $\Sigma_n C_i$ query regions.

When three Queryballs overlap each other, they generate the following seven query regions.

$$(Q_i{}^{\vee}{}_{\cup}Q_j)^{\wedge}{}_{\cup}Q_k) = (Q_i\text{-}Q_j\text{-}Q_k) \text{ UNION } (Q_j\text{-}Q_i\text{-}Q_k) \text{ UNION } (Q_k\text{-}Q_i\text{-}Q_j) \text{ UNION } ((Q_i{}^{\vee}{}_{\cap}Q_j)\text{-}Q_k) \text{ UNION } ((Q_i{}^{\vee}{}_{\cap}Q_k)\text{-}Q_j) \text{ UNION } ((Q_j{}^{\vee}{}_{\cap}Q_k)\text{-}Q_i) \text{ UNION } ((Q_i{}^{\vee}{}_{\cap}Q_j)^{\wedge}{}_{\cap}Q_k) \quad (8)$$

When more than three Queryballs overlap, we must give formal definitions for the intersection of multiple query regions. For example, $(Q_i{}^{\vee}{}_{\cap}Q_j)^{\wedge}{}_{\cap}Q_k$ is constructed by superposing Queryball Q_k on the query region of $Q_i{}^{\vee}{}_{\cap}Q_j$.

To generalize the definitions in the previous subsection, we obtain formal definitions for query region $R_i{}^{\theta}{}_{\cap}R_j$, where $R_i = (B_i, (S_{i1}, P_{i1}), \ldots, (S_{i(m-1)}, P_{i(m-1)}), (\neg S_{im}, P_{im}))$ and $R_j = (B_j, (S_{j1}, P_{j1}), \ldots, (S_{j(n-1)}, P_{j(n-1)}), (\neg S_{jn}, P_{jn}))$.

- $R_i - R_j$

  area specification: $B_i \wedge \neg B_j$

  pairs of search condition and display method: $(S_{i1}, P_{i1}), \ldots, (S_{i(m-1)}, P_{i(m-1)}), (\neg S_{im}, P_{im})$

- $R_i{}^{\theta}{}_{\cap}R_j$

  area specification : $B_i \wedge B_j$

In the case of $R_i{}^{\vee}{}_{\cap}R_j$, superposing R_j on R_i using disjunction is regarded as the overlay query result by R_j on the query result by R_i.

$$R_i{}^{\vee}{}_{\cap}R_j = (B_i \wedge B_j, (S_{i1} \wedge \neg(S_{j1} \vee \ldots \vee S_{j(n-1)}), P_{i1}), \ldots, (S_{i(m-1)} \wedge \neg(S_{j1} \vee \ldots \vee S_{j(n-1)}), P_{j(m-1)}), (S_{j1} \wedge \neg(S_{i1} \vee \ldots \vee S_{i(m-1)}), P_{j1}), \ldots, (S_{j(n-1)} \wedge \neg(S_{i1} \vee \ldots \vee S_{i(m-1)}), P_{j(n-1)}), (S_{i1} \wedge S_{j1}, P_{k1}), \ldots, (S_{ii} \wedge S_{jj}, P_{kk}), \ldots, (S_{im} \wedge S_{jn}, P_{k(m \times n)})(\neg(S_{i1} \vee \ldots \vee S_{in} \vee S_{j1} \vee \ldots \vee S_{jn}), P_{im})) \quad (9)$$

In the case of $R_i{}^{\wedge}{}_{\cap}R_j$, superposing R_j on R_i using conjunction is regarded as narrowing down the search condition S_i by the search condition S_j.

$$R_i{}^{\wedge}{}_{\cap}R_j = (B_i \wedge B_j, ((S_{i1} \wedge S_{j1}) \vee \ldots \vee (S_{i1} \wedge S_{j(n-1)}), P_{i1}), \ldots, ((S_{im} \wedge S_{j1}) \vee \ldots \vee (S_{i(m-1)} \wedge S_{j(n-1)}), P_{i(m-1)}), (\neg((S_{i1} \wedge S_{j1}) \vee \ldots \vee (S_{i1} \wedge S_{j(n-1)}) \vee \ldots \vee (S_{ii} \wedge S_{jj}) \vee \ldots \vee (S_{im} \wedge S_{jn})), P_{in})) \quad (10)$$

4. Conclusion

In this paper, we proposed a new style query model using Queryball. We defined a query model for the interaction by combination of Queryballs, and described the first implementation of a prototype system. This model provides the simplest query that is intuitionally understandable with non-immersive VR system users. Especially combination of multiple Queryballs makes complex search condition and display quite easy.

References

[1] D. Bowman and L. Hodges: User Interface Constraints for Immersive Virtual Environment Applications, Graphic, Visualization and Usability Center Technical Report GIT-GVU-95-26(1995).

[2] S. Bryson: Virtual reality in scientific visualization, in Comm. of the ACM, 39(5), pp. 62-72 (1996).

[3] P. Cignoni, E. Puppo, R. Scopigno: Representation and Visualization of Terrain Surfaces at Variable Resolution, The Visual Computer, 13(5), pp.199-217(1997).

[4] D. A. Keim, H. P. Kriege: Visualization Techniques for Mining Large Databases: A Comparison, Transactions on Knowledge and Data Engineering, Special Issue on Data Mining, pp. 923--938 (1996).