# A View-Oriented Interface for Block-Based Modeling

Jun Fujiki*
Graduate School of Design
Kyushu University

Taketoshi Ushiama†
Faculty of Design
Kyushu University

Kiyoshi Tomimatsu‡
Faculty of Design
Kyushu University

## 1. Introduction

We have developed block-based modeling software called Incompatible BLOCK (Fig. 1) which has an interface allowing the user to position cubes at expected three-dimensional positions with a drag operation. We expect that not only will the Incompatible BLOCK interface be useful for three-dimensional positioning that conforms to a mental model that is likely to exist in reality, but it will also increase the intellectual curiosity of users by allowing them to examine objects from positions that would be impossible in the real world, such as trompes l'oeil and optical illusions. Recent methods for building and editing desired three-dimensional shapes from mouse or pen inputs have mainly employed gesture input for intuition [1,2]. We are interested in a technique whereby shapes can be positioned at expected three-dimensional positions simply and pleasantly by moving them with a drag operation. Here, we consider cubes, and using the properties of cubes and two-dimensional information about their destinations, we set their three-dimensional positions associable with the image. We describe the concepts of our system and the implementation procedure below.

## 2. Concepts

In the method of positioning building blocks which we consider here, without being conscious of three-dimensional space, blocks are placed on a floor serving as a base or are raised in mid-air, and similar building blocks are joined together. In SKETCH [1], based on a basic three-dimensional interpretation of a two-dimensional image, cubes are drawn on the floor and a three dimensional object is constructed by stacking similar objects to join the cubes. In Incompatible BLOCK, using a similar interpretation, cubes are also moved onto the floor(Fig. 2 left) and placed in contact with other cubes; however, because users generally desire to arrange the cubes in an orderly fashion, the three-dimensional positions are determined so that similar cubes are accurately aligned when placed at candidate positions in three dimensions. By determining the candidate position from the six faces of a candidate cube to be joined to, it is possible to position it on the rear face of the cube(Fig. 3 left). At this time, if a cube is added upward to the bottom of a form adjacent to the floor, the added cube looks as if it is under or over the floor(Fig. 4 left). In the real world, however, a form buried in a floor cannot be seen. By considering this, the Incompatible BLOCK shifts the whole form up so that the form will be adjacent to the top of the floor. In addition, the height of a cube can be changed using the three-dimensional interpretation obtained when the cube is moved to a position in mid-air where nothing is drawn(Fig. 5 left).

## 3. Implementation

As a preliminary step, the Z-buffer algorithm is used for all cubes and the floor which are subjected to perspective transformation, except for the movement target, to create a screen size buffer (ID buffer) filled with ID values of the cubes. Next, the ID buffer is compared, pixel by pixel, with the cube moved on the screen, and processing is executed according to the following decision criteria.

**Decision Criterion 1.** Obtained ID indicates the floor (when a cube is placed on the floor)

The cube is positioned at the intersection of the plane of the floor and a straight line pointing in the line-of-sight direction from the cursor position (Fig. 2).

**Decision Criterion 2.** An ID indicating the cube is included in the obtained buffer ID (when joining to a cube)

Intersection points of the six faces of the candidate cube to be joined to and a straight line pointing in the line-of-sight direction from the cursor position are determined. The position that minimizes the distance to a similar cube when the cube is positioned at that intersection point is used (Fig. 3). If it bumps against another cube when moved to this position, the cube is positioned at a three-dimensional position with the next-smallest distance. If it bumps against the floor, it is moved towards the viewing point so that the entire shape which connects to the cube at the destination position is placed on the floor (Fig. 4).

**Decision Criterion 3.** No corresponding ID buffer exists (when a cube is placed at a position where nothing exists)

The cube is positioned at the intersection of a plane which passes through a three-dimensional position before moving and whose camera direction is oriented perpendicular to the floor and a straight line pointing in the line-of-sight direction from the cursor position (Fig. 5).

## 4. Conclusion

Based on comments received from many users, we found that although the manipulation of building blocks in Incompatible BLOCK is not precise, it is possible to quickly place blocks at the target positions and the operation for doing so is fun. We believe that our proposed interface will be effective in manipulating building blocks.

## 5. References

[1] Zeleznik, R., "SKETCH: An Interface for Sketching 3D Scenes", SIGGRAPH'96 Proceedings, pp.163-170, 1996.

[2] Igarashi, T., Matsuoka, S., Tanaka, H., "Teddy: A Sketching Interface for 3D Freeform Design", SIGGRAPH'99 Proceedings, pp.409-416, 1999.

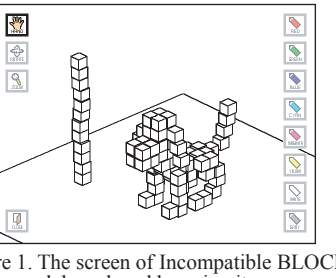

Figure 1. The screen of Incompatible BLOCK with a model produced by using it.

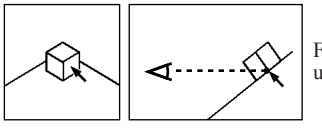

Figure 2. Moving on the floor (left: user view, right: side view).

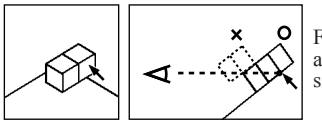

Figure 3. Movig to connect to another cube(left: user view, right: side view).

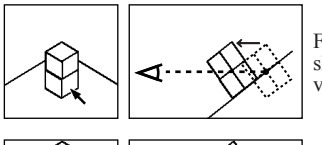

Figure 4. Moving the entire shape(left: user view, right: side view).

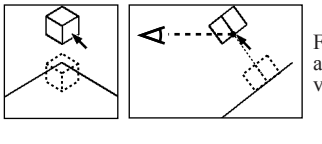

Figure 5. Moving to a point in mid-air(left: user view, right: side view).

*e-mail: fujiki@gsd.design.kyushu-u.ac.jp
†e-mail: ushiama@design.kyushu-u.ac.jp
‡e-mail: tomimatu@design.kyushu-u.ac.jp