SUPPORTING OBJECTS IN VOXEL-BASED DESIGN ENVIRONMENTS

STEVE KUAN, THOMAS KVN
The Department of Architecture, The University of Hong Kong, Pokfulam Road, Hong Kong S.A.R., P.R. China.
kskuan@graduate.hku.hk, tkvan@hku.hk

Abstract. Inspired from the idea of using voxels in the conceptual stages of architectural design, a synchronous collaborative design system, CoBlocks, was developed. This paper raises the problem that simply adopting the voxel representations would take voxels as graphic primitives and offer design operations at the voxel level. We introduced the object descriptor to group voxels to form higher-level semantic elements. A test of its efficacy is reported in this paper. The results indicate that the objects facilitated more discussions on high-level design issues and supported more effective design operations. This suggests designers can benefit from the use of objects in voxel-based design environments.

1. Introduction

Much research has been carried out in investigating and developing synchronous drawing programs that allow designers to share their ideas with others even when they are in geographically distributed locations. Some examples are NetDraw (Qian and Gross, 1999), GroupSketch, GroupDraw (Greenberg et al., 1992), Commune (Bly and Minneman, 1990). However, the applications are limited to 2D shared drawings and the issues arising in synchronous 3D collaborative design have been explored much less.

Collaborative design in 3D virtual environments enhances exploration and communication of volume and space. Significant inhibitors in their adoption are usability issues and technical problems which stem from synchronizing large 3D datasets (Schnabel and Kvan, 2002). Nam and Wright (2001) developed a synchronous collaborative design system called Syco3D based on conventional interaction techniques in commercial 3D CAD systems. To support conceptual design activities, design systems require tools different than those used in the production of construction documents. The interaction techniques and representations in current commercial CAD packages are too rigid and precise for exploration and communication of ideas in the earlier phases of design process (Dorsey and McMillan, 1998). Simpler free form drawing tools are one approach to satisfying this need.

Many studies have investigated 3D modelling techniques for their potential as
simple and easy sketching environments. A widely reported tool is DDDoolz (de Vries and Achten, 2002), a voxel-based design tool developed for mass study and spatial design in the early stage of architectural design. Similar to a pixel in 2D, a voxel describes attributes of a point in the 3D world (Foley et al., 1997). In DDDoolz, voxels are visualized as blocks and users can create different shapes composed of many of these blocks. Inspired by the idea of using voxels in conceptual design and the simple modelling techniques introduced in DDDoolz, we developed a system called CoBlocks that uses voxels to support collaborative conceptual design. However, we examined that the way voxels are organized, in particular looking at the support of editing on a voxel-by-voxel basis as in DDDoolz, there is room for improvement to incorporate voxels into design systems.

Many computer-aided design applications have exploited the advantages of voxel-based modelling techniques, for example, virtual sculpting, constructive solid modelling, simulation of amorphous phenomena, etc. (Kaufman, 2000). VoxDesign (Donath and Regenbrecht, 1996) is another tool similar to DDDoolz that demonstrated the use of voxels to support conceptual architectural design. Although these applications have offered new ideas to support design in 3D environments, many of these applications simply adopt the voxel representation into a design tool. This would take voxels as graphic primitives and provide design operations at the voxel level. We observe, however, that treating voxels as graphical primitives is probably not adequate for design. It is common practice that designers work not at the discrete component level but by grouping components into higher level semantic elements which then re-composed to form a design (Mitchell, 1990). A shortcoming of voxel representations is a lack of structure to preserve such higher level elements formed in the design process. Thus, editing operations are restricted to the voxel level. The process of design is limited to initial creation but depends heavily upon editing and revision of existing designs (Lawson, 1994). A hypothesis of our work is that limiting designers to editing of their design at such a low level is likely to inhibit the design process.

This paper introduces the organization of voxels using object descriptors and outlines the design operation offered in CoBlocks. A test is carried out to examine if the users can benefit from the introduction of objects in voxel-based design environments.

2. Supporting objects in CoBlocks

CoBlocks has been developed to support synchronous collaborative conceptual design. It provides a synchronized collaborative design environment in which users can collaboratively work on the 3D models and communicate via the chat-line (Figure 1). The design environment in CoBlocks adopts similar principles described in DDDoolz (2002) except in CoBlocks the notion of objects are preserved by
object descriptors in order to avoid designers working at the voxel level. In CoBlocks, users can define objects which are made of blocks and edit objects as primitives.

To support objects in CoBlocks, a data structure called object descriptor is derived based on the idea suggested by Leu and Chen (1999). Instead of having one voxel dataset to store all objects, each object descriptor links to a voxel dataset which is used to describe the geometry of object. Position, orientation, scale, and colours of objects are not stored separately in voxels but stored as attributes of objects in the descriptor.

This organization allows applying design operations at the object level easily by simply changing the attributes maintained in the object descriptor without the need of re-sampling the transformed object. The geometry of objects are also independent between objects. So the geometry can be preserved even after the objects have been overlapped and separated. The shortcoming is that the rendering process will become more time and resource consuming and special rendering techniques are needed for large voxel datasets (Chen et al., 2000). Since CoBlocks visualizes these voxels as coarse blocks, the problem is not a significant issue in this application.

While incorporating object descriptors in CoBlocks, voxel-based models are managed as objects and design operations in CoBlocks are offered at the object level. Users define objects, composed of a set of blocks. Although users can modify the shape by adding blocks to or removing blocks from the object, they are not allowed to move, scale or colour an individual block. These operations can only apply on the objects. Objects can be copied or removed by using the copy and
object removal tools. Functions are available for copying or removing the object.

CoBlocks supports the grouping of several objects into a composition by means of a composition descriptor linking these sub-components together into an object hierarchy. Figure 2 illustrates an example of hierarchy of objects, formed by two grouping operations. From the user perspective, a composition behaves in the same manner as an object. Design operations can apply on compositions as a unit rather than the objects individually. Compositions can be ungrouped in order to obtain back the individual objects.

![Figure 2. A hierarchy of objects formed by a number of object descriptors.](image)

3. Experiment

This experiment is designed to examine the hypothesis that restricting the editing to the voxel level inhibits the use of voxel-based design systems. CoBlocks supports operations in two modes: Object Mode and Block Mode. The major difference between the two modes is that Object Mode uses an object descriptor as described above, whereas Block Mode does not. In Block Mode, each voxel is treated as an individual block, therefore blocks can be moved, scaled and coloured independently. Since there are no objects in Block Mode, there are no operations provided for copying or removing an object. Participants are required therefore to remove or copy models on a voxel basis. With these exceptions, the user interfaces of two systems were similar (Figure 3).
3.1. PROCEDURE

Testing was carried out by means of paired collaborative design sessions. Each test session was scheduled for an hour and consisted of three stages: Tutorial (15 minutes), Design (30 minutes) and Debriefing (15 minutes). Each member of the pair was assigned to work in a different room in order to simulate a distributed working environment in which verbal communication was not feasible; however, the configuration of the computer in both rooms were identical. The task given was to design a processional gate or archway over a road that can be used in a celebratory procession of Olympic athletes.

In the Tutorial stage, participants were given 15 minutes to learn CoBlocks by reading a paper-form tutorial and trying out the system. After the tutorial, the two users were connected together so that their drawing tools were synchronized. They were only allowed to communicate over the chat-line provided with the system (Vera et al., 1998). Participants were asked to work collaboratively on a design task using the CoBlocks for 30 minutes. During the experimental session, the following data were collected:

1. The chat-line protocol, providing data on the types of design activity undertaken.
2. The timestamps and types of operations performed, offering data on the operation of the system.
3. A debriefing questionnaire, in which each participant was asked to rate the
effectiveness of functions available in the system on a scale from one (‘Not effectiveness at all’) to seven (‘Very effective’) with an additional ‘Not applicable’ option.

3.2. PARTICIPANTS

Thirty-four participants (16 male and 18 female) were recruited and grouped into 17 pairs. The age of the participants ranged from 19 to 22. All of whom were Year Two students studying in the Department of Architecture at the University of Hong Kong. They had not seen the CoBlocks before the experiment. Eight pairs of participants were assigned to use the CoBlocks in the Object Mode and nine pairs were assigned to use the CoBlocks in the Block Mode.

4. Results

Data were analyzed to study the design activities and operating time of design operations during the design. The perceived effectiveness of design operations was obtained from analyzing the questionnaires.

4.1. DESIGN ACTIVITIES

The chat-line protocols were coded according to the Design Process Model proposed by Vera et al. (1998) to examine design activities. According to this Model, communicative exchanges can be encoded into four categories: High-Level Design (HLD), Low-Level Design (LLD), Task-Focused (TF) and Interface-Specific (IS).

The percentages of design exchanges are shown in Figure 4. Participants in Object Mode discussed more high-level design issues. The analysis showed that 36% of exchanges in Object Mode were HLD exchanges in contrast to 26% of exchanges in Block Mode. The independent t-test indicated that the difference was significant. 

![Figure 4. Percentages of communicative exchanges in the design task.](image-url)
significant at the 0.05 level; there were no significant differences in other communication exchanges.

4.2. OPERATING TIME

Operating times for user activity were analyzed. The operating time is defined as the amount of time spent in using a type of operation by a participant in the design task. As adding blocks, removing blocks, move and scale require mouse-drag action in the operation, the duration of these operations was measured to identify changes in use and hence trace how the introduction of objects affected designers in using these operations (Figure 5).

![Figure 5. Time spent in operations.](image)

We observe that there was a great difference in the operating time of block removal and move. Participants in Object Mode move more than those in Block Mode (p<0.01) where a participant spent about 15 seconds in move operation. Participants remove blocks less often in Object Mode than in Block Mode (p<0.01).

4.3. PERCEIVED EFFECTIVENESS

Perceived effectiveness of design operations was measured by the questionnaires given in the debriefing session. Table 1 summarizes the responses in which 7 represented ‘Very effective’ and 1 represented ‘Not effective at all’.

The mean score obtained by these operations were all higher in Object Mode than in Block Mode. The perceived effectiveness of move was significant greater in Object Mode than in Block Mode (p<0.01). Block removal was also perceived as more effective (p<0.05) in Object Mode. Among all operations, the block removal
in Block Mode was considered as the least effective design operation.

<table>
<thead>
<tr>
<th>Operations</th>
<th>Object Mode</th>
<th>Block Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Std. Dev</td>
</tr>
<tr>
<td>Block creation</td>
<td>5.19</td>
<td>1.223</td>
</tr>
<tr>
<td>Block removal</td>
<td>5.38</td>
<td>1.360</td>
</tr>
<tr>
<td>Move</td>
<td>5.44</td>
<td>1.750</td>
</tr>
<tr>
<td>Scale</td>
<td>4.94</td>
<td>1.340</td>
</tr>
<tr>
<td>Colour assignment</td>
<td>5.63</td>
<td>1.708</td>
</tr>
</tbody>
</table>

1: Not effective at all, 7: Very effective

5. Discussion

The result demonstrated that the designers benefited from Objects in a voxel-based design environment. The protocol analysis shows that participants in Object Mode engage in more High-Level Design issues than those in Block Mode. High-Level Design exchanges are discussions such as planning the site, layouting the major components and deciding the overall theme of the design that will significantly affect their later design decisions. As Vera (1998) stated, the High-Level Design exchanges play an important role in design quality, and an effective design process should have a high percentage of High-Level Design exchanges found in the protocol. That implies participants can design more effectively with objects.

As revealed in the questionnaire data, participants perceived that the Move operation was effective. It is observed that they moved objects more often in their exploration of different designs. The Remove operation was perceived as slightly less effective but still more effective in Object than Block mode. We note from the operation times that participants removed blocks more often in Block Mode. This is likely to be due to the need to remove several voxels in order to delete a design component; the more efficient operation of the Object mode allowed for rapid removal of several voxels together.

While voxel-based design systems are apparently capable of supporting simple sketching in 3D, their use is inadequate if the systems work only at the voxel level. Limiting editing operations to low-level components such as voxels seem to be inadequate for design. Since editing is important in the design process, it is essential for design systems to have operations that will permit designers to work with semantically richer constructs. Incorporating the object descriptors and offering operators at the object level appear to better facilitate the design process.
Acknowledgements

We would like to thank all members in the Digital Design Research Group for their comments to this study and Marc Aurel Schnabel for his help on the experiment. We would also like to thank all the architectural students who participated in the experiment.

6. References


