CoBlocks: An Improved Voxel-based Design Tool by Object Structuring of Voxel Models

Steve Kuan and Thomas Kvan

CoBlocks: An Improved Voxel-based Design Tool by Object Structuring of Voxel Models

Steve Kuan and Thomas Kvan

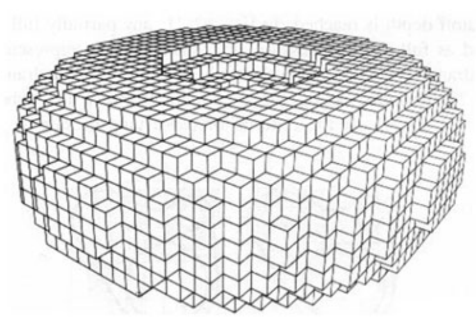
This paper introduces a voxel-based collaborative modelling system called CoBlocks which was developed to support designers in building models together in a synchronized virtual environment. This is due to the fact that voxel models are gaining more attention in computer-aided design (CAD) systems as they support simple and intuitive modelling for the early design phases. However, due to the discrete nature of voxels, it is common practice in most voxel-based design systems that the voxel modelling methods have limited users to manipulate models at the level of individual voxels. From the literature, however, we might expect that voxel modelling would benefit from higher-level interaction as supported by the object structuring of such models. In light of these, a controlled lab study was therefore carried out to examine the benefits of the structuring of these models in voxel-based design systems. The results show that users prefer working with structured voxels and that they can interact better with them.

1. INTRODUCTION

Recently, three-dimensional (3D) digital media are playing an increasingly important role [1] while architects engage in a range of design media. Specifically, Schnabel and Kvan [2] identify that designing with 3D digital representations enhances designers' understanding of and communication with architectural spaces. Moreover, advancements in networking infrastructure have facilitated the development of design systems to support "group modelling" that allows physically dispersed designers to jointly create and edit 3D voxel models [e.g., 3, 4, 5]. However, the designs of these systems are primarily based on interaction techniques and solid representations similar to those used in conventional single-user 3D modelling systems. Even if most of them provide sophisticated modelling capabilities, they appear to be too rigid and precise for design interaction with the initial ideas [6].

CAD research has explored a variety of methods to support 3D modelling that is perceived as simple and intuitive. One direction is to exploit the distinctive properties of voxel representation to innovate modelling techniques. Unlike the 3D solid representations used in standard CAD systems such as 3D Studio Max, ArchiCAD, FormZ, and so on, voxel representations decompose solids into identical cells called voxels, which are usually arranged in a fixed, regular grid (Figure 1). Voxel representations have been used extensively in geophysics, medical imaging, and scientific visualization. However, in the field of CAD, other types of solid representations such as Boundary Representations are still dominant in representing 3D graphics. The use of voxels in design is thus still relatively unexplored.

► Figure 1: Decomposition of a solid into voxels



Since voxel models are discrete in nature, in contrast to the continuous surfaces used in conventional CAD systems, this discreteness innovates ways for CAD systems to afford intuitive and effective modelling techniques. For example, voxel representations enable virtual sculpting which allows the creation of a model by the metaphor of traditional sculpting [7, 8]. This was demonstrated in the commercial modelling software *FreeFrom* developed by Sensable Technologies (http://www.sensable.com). This system, usually aided with the haptic device PHANTOM, enables designers to quickly and freely modify 3D forms by sculpting on virtual clay. FreeForm has been reported to be useful in exploring ideas in the early design phases [9].

DDDoolz [10] is another tool based on voxel representation developed for mass study in the early stage of architectural design. This suggests a distinctive feature allowing for the direct creation of geometry and is fulfilled by describing a 3D form using a set of coarse voxels. By visualizing the voxels as cubic blocks, it enables interaction through the metaphor of "3D painting". Furthermore, although the use of such blocks results in very rough forms and hinders the process of shaping the details of the models, the technique enables users to create models with less effort and is adequate for the early stage of design [11]. Another tool that aims to support conceptual architectural design, voxDesign [12], also makes use of voxel representations, but does so in an immersive virtual environment supporting 3D sketching. Empirical studies on DDDoolz and voxDesign suggest that the low-resolution models represented by coarse voxel blocks leave the details undefined and allow ambiguity for different interpretations [11, 12].

One possible extension of existing voxel-based design systems is to support multiple users in modelling collaboratively in the early design stage [13]. This is because the ease and intuitiveness of voxel modelling techniques allow rapid design communication, and it appears that very few studies have employed voxel representations in group modelling systems. As such, we developed a system to incorporate voxel modelling techniques to support group modelling in the early stage of design and investigated the effect this had on design collaboration. In the following section, however, it is speculated that the unstructured nature of voxel models make the interaction between designers ineffective. We propose, therefore, that structuring voxel models into objects facilitates collaborative design. In the third section of this paper, we describe the design of CoBlocks, a design system that utilizes voxel representations to support group modelling in the early design stage. A usability test is reported in Section 4 to examine the effects of the object structuring in voxel-based collaborative systems, and finally, we draw conclusions from this study in Section 5.

2. UNSTRUCTURED VOXELS

It is apparent from the literature reported above that most voxel-based design systems do not structure the 3D geometries represented; users are required to manipulate the models at the level of individual voxels. While this atomistic representation is effective in supporting intuitive modelling methods such as "3D painting" in DDDoolz [10], the drawbacks inherent in such the discreteness of voxel models are significant. Mitchell [14] identfies that it is uncommon for designers to manipulate discrete components such as voxels during design, suggesting that our design thinking tends to group primitives into objects that correspond to higher-level design semantics which are move meaningfully manipulated.

This behaviour is reflected in empirical studies of sketching in which the production of sketches is observed to be guided by the semantics of the drawn objects. For example, Scrivener and Clark [15] developed a system called ROCOCO that could record and play back every stroke which occurred during the drawing process. Analyzing the recorded sequences of strokes and segmented them into "drawing acts", they observed that each act appears to be a meaningful and complete component rather than an arbitrary fragment of entities. Kavakli et al. [16] likewise show that sketching activity is largely oriented to the geometrical semantics of the imagined or recalled item. In this study, they observed how the participants draw a chair. If the participant draws a component completely and then moves on to draw another component, this is described as drawing "part-by-part". On the other hand, if a participant draws a portion of a component but then moves on to draw a portion of another component, this is described as drawing "non-part-by-part". The result of this study shows that 87% of the drawing acts were done by the participants in a part-by-part manner.

These observations show that semantics play an important role in our design thinking. While these examples cover geometrically rich descriptions, similar effects can be expected in geometrically limited models such as the coarse voxel models used in some voxel-based systems [e.g. 10, 12]. However, the design of voxel-based systems, whether low-resolution or high-resolution, does not address the need to represent the semantics of voxel models. This is shown in existing voxel-based systems which offer design operations that only work at the level of voxels. For example, with the FreeForm system, users can perform various sculpting techniques to modify a model but cannot move or scale any component of that model. In addition, DDDoolz users can move the geometries but only in a voxel-by-voxel manner [17]. As de Vries et al. [11] note:

> "The problem of a low-level data model, such as used here, is that operations on a higher aggregation level (such as 'wall' or 'floors') are only possible by manipulating the component cubes of that object."

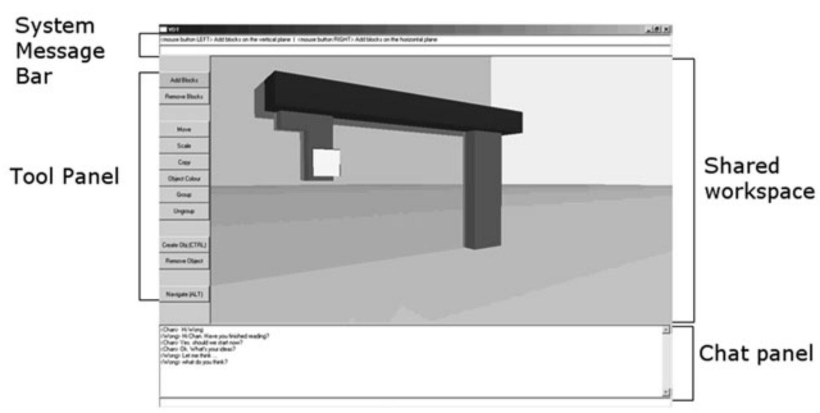
As such, they suggest a higher-level interaction called "Edge Drag" for moving an edge which is made up of a set of blocks connected in a row. However, this feature does not solve the problem if the designer intends to move an object composed of blocks that are not arranged in a row. Furthermore, de Vries, Jessurun, and van Wijk [11] suggest implementing the "group" feature that allows the grouping of blocks into a building element to further enhance the capability of DDDoolz in the later version.

Grouping techniques in current VR systems, such as those described above, have not yet been demonstrated to be successful in allowing the intuitive grouping of such discrete blocks [18]. Instead of requiring users to "select and group", the formation of objects can be achieved more simply by exploiting the part-by-part drawing behaviours found in the studies done by Kavakli [16] and Scrivener and Clark [15]; that is, designers tend to draw a meaningful part (or object) completely before starting to draw another part. If the part-by-part drawing behaviour found in sketching is also dominant in the voxel modelling activities, this may provide a basis for voxel-based design systems to which will allow the definition of consecutively created voxels as an object rather than the "selection and grouping" of the voxels individually. This allows users to manipulate the defined, meaningful objects directly instead of low-level individual voxels.

From the literature, we can expect voxel modelling to benefit from higher-level interaction as supported by the object structuring of voxel models. To date, however, no voxel systems have provided such support and there is no evidence to suggest that expected benefits would be realised. Hence, the objective of this study is to examine whether or not the object structuring of voxel models can improve the usability of voxel-based design systems.

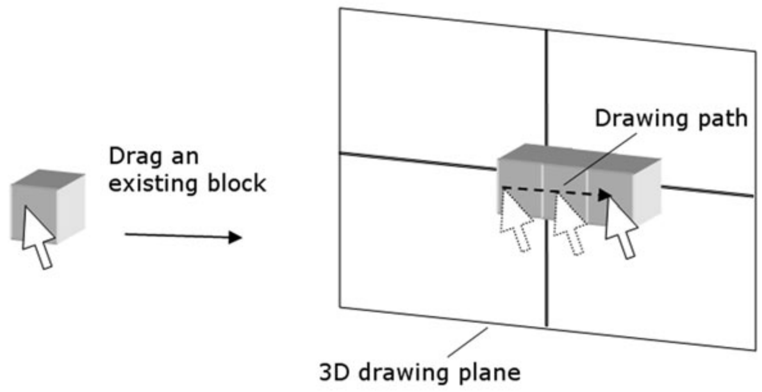
3. PROTOTYPE SYSTEM – COBLOCKS

CoBlocks was developed from the design of DDDoolz [10] as this system has been successfully used to demonstrate the use of low-resolution voxels to support architectural design in the early stages. However, unlike DDDoolz, CoBlocks allows two distributed designers to interact on the 3D models synchronously. Figure 2 illustrates the user interface which consists of four main components: shared workspace, tool panel, chat panel, and system message bar.



◀ Figure 2: The user interface of CoBlocks.

The modelling techniques are based on the two modelling principles suggested in DDDoolz [11]. First, all geometries are composed of coarse blocks. Second, users use the "Drag and Copy" technique to "draw" 3D geometries quickly. The dragging block is copied and placed along the drawing path on a specified drawing plane (Figure 3). The steps in deleting blocks are the same as those in drawing blocks.



▶ Figure 3 The "Drag and Copy" technique to draw blocks.

3.1. Object Mode and Block Mode

While most studies on voxel-based design systems have not addressed the needs of structure, CoBlocks is implemented in such a way that it can operate in either Object Mode or Block Mode. Block Mode requires users to work on individual voxels because it lacks the structure for semantically storing the models, just like most voxel-based design systems as described in Section 2. On the contrary, Object Mode allows users to define objects during modelling so that the system knows the objects semantics given by the users. With this information, users can modify the design with operations at a higher level, like "remove this object" instead of "remove this set of voxels". However, the system requires a different data scheme and design operations from Block Mode in order to support Object Mode.

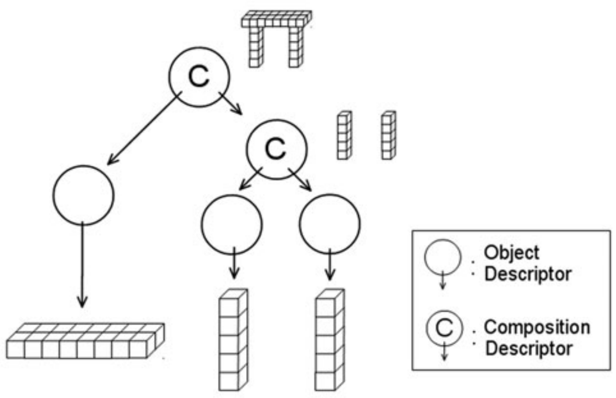
Data scheme

Technically, the differences between two modes are derived from the manner in which the models are represented computationally. As in other voxel-based design systems, CoBlocks represents geometries in Block Mode in a data set which does not maintain objects associations once the input is voxelized. In order to maintain the notion of objects, Object Mode adopts a different data scheme that is capable of storing the representations of object semantics with multiple data sets, as described in the work of Leu and Chen [19].

For each object defined in CoBlocks, an object descriptor is instantiated to maintain the object. Each object descriptor contains an independent voxel data set to store the object's geometry. The object descriptor also defines the object's attributes, such as its position, orientation, and scale. With the use of object descriptors, Object Mode can provide design operations at the object level. The operations are also computationally efficient since this requires simply changing the attributes maintained in the object descriptor without the step of re-sampling the transformed object [20]. However, the shortcoming is that the rendering process becomes more time- and resource-consuming. Special rendering techniques are needed for large voxel data sets [21], but since CoBlocks uses low-resolution voxel models, this problem is not a significant issue here.

Additionally, CoBlocks in Object Mode 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 4]

Figure 4 illustrates an example of the hierarchy of objects which is formed by two grouping operations. From the user perspective, a composition behaves in the same manner as an object to which design operations can then be applied. Compositions can be ungrouped in order to revert to the constituent objects.



◀ Figure 4: A hierarchy of objects formed by a number of descriptors.

Design operations

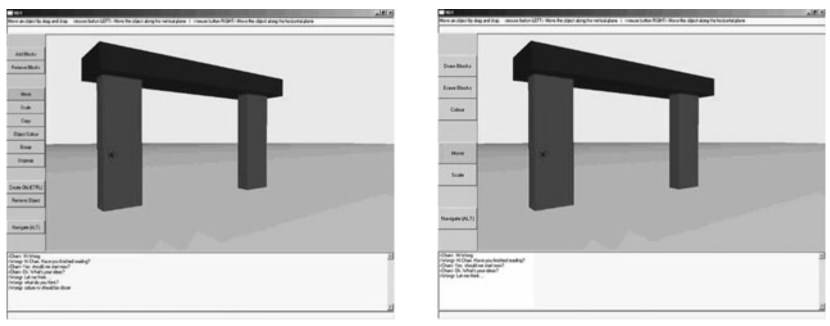
More operations are provided compared with those in Block Mode (Table 1), since additional tools are needed for managing the objects in Object Mode. For example, Object Mode provides "Create Object" and "Remove Object" tools. Developing the observation, noted in Section 2, that part-by-part drawing behaviours are dominant in the production of sketches, CoBlocks is designed to allow users to define the consecutively created voxels as an object. To define a new object in Object Mode, users select the "Create Object" tool (or hold the "Ctrl" key while "3D painting" the blocks) to define a new object through sequential drawing actions. Users can then use "Add Blocks" or "Remove Blocks" tools to refine the shape of the object. The "Copy" tool is supported in the Object Mode for users to clone objects while "Group" associates several objects together to form a composition. Upon grouping, the operations such as move, scale, and colour, can be applied to the composition.

▶ Table 1: Operations supported in Object Mode and Block Mode.

Operations	Supported in Object Mode	Supported in Block Mode
Add Blocks	√	√
Remove Blocks	√	√
Move	√	√
Scale	√	√
Colour	√	√
Create Objects	√	χ
Remove Objects	√	χ
Copy	√	χ
Group	√	χ
Ungroup	√	χ

The "Add Blocks" and "Remove Blocks" tools, which are based on the "3D painting" technique, are used similarly in both Object Mode and Block Mode. However, the "Move", "Scale", and "Colour" tools vary in the two modes. Since the objects are defined in Object Mode, these tools are used to move, scale, and colour an object, while in Block Mode, they function on individual voxels and not on objects.

The user interfaces of Block Mode and Object Mode are very similar (Figure 5) except for the layout of the buttons on the tool panels, since fewer tools are provided in Block Mode.



▶ Figure 5. Left: The user interfaces of CoBlocks in Object Mode. Right: The user interfaces of CoBlocks in Block Mode.

3.2. Implementation

CoBlocks was developed using the Python (version 2.3) programming language which was chosen because it offers quick program development and is powerful enough to build complex systems. Although Python is a cross-platform tool, the system was tested primarily in the Microsoft Windows XP environment.

CoBlocks adopts a replicated approach for the storage of design data in order to provide high system responsiveness. This means that the voxel models are stored locally in each node and not in a centralized server. When users update the model, CoBlocks first operates on the local repository so that users can view the effects instantly. Then CoBlocks replicates the design operation and transmits it to the remote sites in order to update the remote models. In this case, local users can view the update instantly without being affected by any network latency. CoBlocks uses the bandwidth more effectively since it transmits the operations instead of the models for synchronization. However, this introduces the problem of locking for concurrent operations.

CoBlocks uses the locking method to manage concurrent operations and implements optimistic locking that does not require users to wait for the reply of a lock request during collaboration [22]. CoBlocks requests and releases locks imperceptibly when users are working on the models. Unless a lock request is denied when there are concurrent operations, the executed operations would be automatically undone to restore to the original state, and a message would be displayed to inform the users.

4. USABILITY EVALUATION

We conducted an experiment to investigate the effects on usability of CoBlocks when voxels are structured into objects. The goal of this evaluation is to test the hypothesis that structuring voxel models into objects improves the usability of voxel-based design systems.

4.1. Method

The usability test was designed as a controlled laboratory study. Pairs of participants used CoBlocks to design collaboratively either in Object Mode or Block Mode. The purpose of having these two modes was to collect data for a comparison of collaborative designing at the object level (Object Mode) and at the voxel level (Block Mode).

Our goal is to look at the improvement in three aspects of collaboration: product, process, and satisfaction [23]. Our product measure assesses the quality of the design outcomes, while the process measures examine the conversations and design operations used during a collaborative session. Meanwhile, the satisfaction measures collect a participant's perception of the system. Table 2 illustrates the six measures used in this study.

Type of Measure	Measures Used
Product measures	(a) Quality of design outcomes
Process measures	(b) Richness of design conversations (c) Use of design operations
Satisfaction measures	(d) User satisfaction with the system usability (e) Perceived effectiveness of design operations (f) Open-ended questions

◀ Table 2: Measures used in this study.

Subjects

Thirty-four participants (16 male and 18 female) were recruited and randomly grouped into 17 pairs. The age of the participants ranged from 19 to 22, all of whom were second year undergraduate students studying in the Department of Architecture of the University of Hong Kong. They had not encountered CoBlocks prior to the experiment but were all users of similar experience in CAD tools. Eight pairs of participants were assigned to use Object Mode, and nine pairs were assigned to use Block Mode.

Procedures

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). One 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 computers in both rooms was identical.

In the Tutorial stage, the participants were given 15 minutes to learn CoBlocks by reading a printed tutorial and trying out the system. The previously carried out pilot test showed that 15 minutes was adequate for users to learn the system. Upon successful completion of the tutorial, the two CoBlocks users were connected so that their shared workspaces were synchronized and the task, to design a processional gate or archway over a road that can be used in a celebratory march of Olympic athletes, was assigned. The participants were asked to work together on the design task for 30 minutes during which time they were able to discuss their work over the chat-line provided within the system. The participants were then requested to complete a questionnaire upon the completion of the design session.

4.2. Results

In this section, we report the data obtained by the six measures in the experiment. The findings are subsequently discussed in in Section 4.3.

Quality of Design Outcomes

Three tutors from the Department of Architecture were invited to assess the design outcomes [24]. Each teacher classified the outcomes as Good, Average, and Poor by assessing whether or not the solution showed innovation and met the requirements specified in the design brief. Since the design task required only a sketch of the solution, it was difficult for the tutors to make fine distinctions in the quality of outcomes; the tutors were therefore requested to rank the outcomes into three categories, from 1 (poor) to 3 (good). The quality of each design outcome was quantified by calculating the mean from ratings given by the three teachers. The results indicate that the mean score of the outcome for Object Mode is slightly higher than that for Block Mode (Table 3) but the t-Test indicated that there is no significant difference between the scores obtained for the two modes of operation ($p = 0.771$). Moreover, since there is no significant finding in this measure, it is unclear how the modes affected design quality.

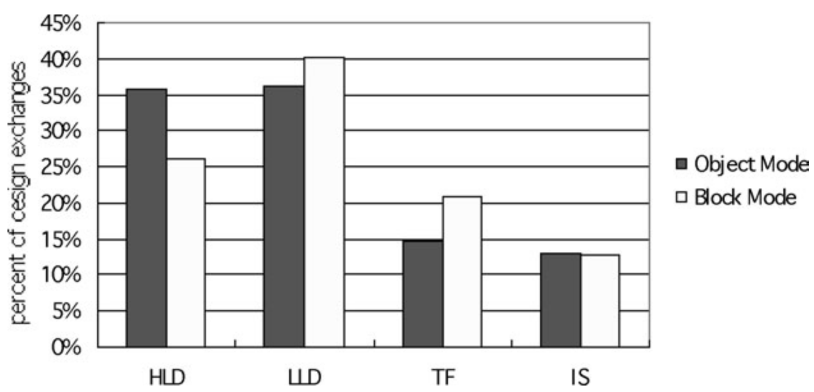
	Scale: 1 (Poor) to 3 (Good)			
	Mean	S.D.	Min	Max
Object Mode	1.833	0.309	1.33	2.33
Block Mode	1.778	0.441	1.00	2.33

◀ **Table 3: Descriptive statistics for the quality of design outcomes.**

Richness of design conversations

To examine if the introduction of objects influences the process of design exploration, the conversations were encoded according to the Design Process Model proposed by Vera et al. [24]. According to this model, communicative exchanges can be classified into four categories: High-Level Design (HLD), Low-Level Design (LLD), Task-Focused (TF), and Interface-Specific (IS). As High-Level Design exchanges are those that significantly affect subsequent design actions, the number of HLD exchanges can be considered an indicator for measuring the richness of design exploration during the collaborative process.

The analysis of the design communication shows that the participants in Object Mode have richer design exploration. Particularly, Figure 6 shows the percentages of exchanges where it can be noted that the participants in Object Mode have significantly more communications on high-level design issues ($p < 0.05$). The data show that 36% of exchanges in Object Mode were HLD exchanges in contrast to 26% of exchanges in Block Mode. For other design activities, there were no significant differences. This finding suggests that the participants who collaborated at the object level had a more extensive or meaningful exploration of the design than those who worked at the voxel level.

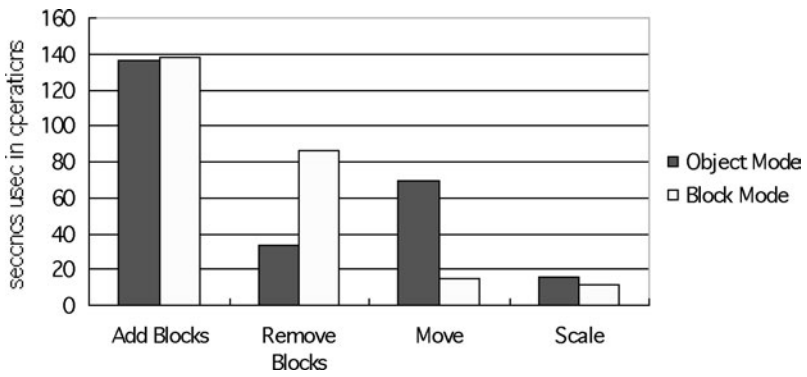


◀ Figure 6: Percentages of communicative exchanges in the design task.
Uses of design operations

Uses of design operations

The uses of design operations could be determined by looking at the duration of use for each modelling tool by a participant in the design session. The duration of use is defined as the time of using a modelling tool that starts when the user presses the mouse button and ends when the user releases it. Since the "Colour" tool is just a one-click operation, the use of it in the two modes was not compared in this study. Instead, the duration of use of the four operations – "Add Blocks", "Remove Blocks", "Move", and "Scale" – supported in both Object Mode and Block Mode, were examined.

The results show that the time to operate "Remove Blocks" was significantly less ($p < 0.01$), while the time it took to operate "Move" in Object Mode was significantly more ($p < 0.01$) (Figure 7). There is no significant difference found in the "Add Blocks" and "Scale" operations. The data from this measure reveal that the structure of objects influenced the participants' use of the design operations. Specifically, the participants tend to use the "Move" operation more and "Remove Blocks" operations less when the objects were available.



► Figure 7: Duration of use for each modelling tool in a session.

User satisfaction with the system usability

A questionnaire which was primarily adapted from the IBM Post Study System Usability Questionnaire (PSSUQ), and which was reported to be reliable, valid, and sensitive, was used to determined user satisfaction [25]. Lewis [26] reported the data of his five-year study using PSSUQ in which he concluded that practitioners can confidently use the PSSUQ for measuring user satisfaction.

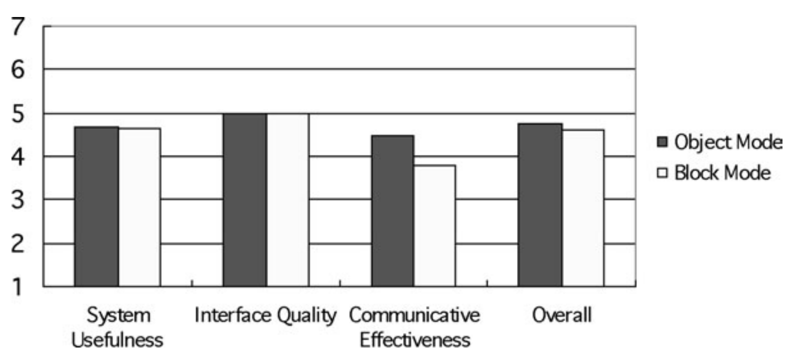
Table 4 shows the questionnaire which was designed for this study to collect data on user satisfaction in three aspects of the system: System Usefulness, Interface Quality, and Communicative Effectiveness. Most questionnaire items in the System Usefulness and Interface Quality were adapted from PSSUQ, while some items were excluded because they were not applicable to this study. In addition, our questionnaire contained items (items 12 to 14) which ask about the effectiveness of communicating concepts using the blocks. All items in the questionnaire were rated using a scale ranging from one ("Strongly disagree") to seven ("Strongly agree"), including a "Not applicable" option.

1.	It was easy to use this system.
2.	I was able to complete the design task quickly using this system.
3.	I could effectively complete the design task using this system.
4.	It was simple to use this system.
5.	It was easy to learn to use this system.
6.	I believe I could become productive using this system.
7.	This system proved useful to complete the task.
8.	Whenever I made a mistake using the system, I could recover easily and quickly.
9.	The interface of this system was pleasant.
10.	I liked using the interface of this system.
11.	This system has all the functions and capabilities I expect it to have.
12.	It was quick to exchange ideas with these "blocks".
13.	I could use these "blocks" to communicate my idea clearly to my partner.
14.	It was easy to communicate using these "blocks".
15.	Overall, I am satisfied with this system.

◀ **Table 4: Questionnaire items for measuring user satisfaction with the system usability.**

In order to determine the number of factors in the data, discontinuity analysis [27] was used. The identified factor structure was similar to what was expected but with some difference. According to this analysis, Figure 8 shows the overall score and the factor scores that were calculated as follows:

- System Usefulness: averaging the ratings of Items 1, 2, 3, 6, 7
- Interface Quality: averaging the ratings of Items 4, 5, 9, 10, 11
- Communicative Effectiveness: averaging the ratings of Items 8, 12, 13, 14
- Overall: averaging the rating of Items 1 to 15



◀ Figure 8: Scores of the factors and overall questionnaire.

The Cronbach alpha reliability of the overall questionnaire was 0.81, and the reliabilities of the factors of System Usefulness, Interface Quality, and Communicative Effectiveness are 0.73, 0.77, and 0.86, respectively. The Cronbach alpha reliabilities are over 0.7, indicating that the measurements are reliable [28].

The scores of Communicative Effectiveness in Object Mode and Block Mode are 4.47 and 3.81, respectively, and the differences in these factor scores were deemed significant ($p < 0.05$). As the layout of the user interfaces of the two modes were not much different, the similar scores obtained in Interface Quality were expected (Object Mode: 4.99; Block Mode: 5). Moreover, the differences of the scores in System Usefulness are non-significant. It is worth noting, however, that the Communicative Effectiveness of Block Mode scored lowest among all the factors. The comparison of factor scores clearly shows that the object structuring improved user satisfaction in using voxel models for design communication.

Perceived effectiveness of design operations

The effectiveness of design operations was also obtained using questionnaires that were distributed at the debriefing session. This questionnaire asked the participants to rate the effectiveness of each design operation on a scale of one ("Not effective at all") to seven ("Very effective"), with an additional "Not applicable" option.

Table 5 summarizes the responses for the modelling tools offered in Object Mode and Block Mode.

The mean score obtained by these tools were all higher in Object Mode than in Block Mode, and the perceived effectiveness of "Move" was significantly greater in Object Mode than in Block Mode ($p < 0.01$). The "Remove Blocks" operation was also perceived as more effective ($p < 0.05$) in Object Mode. In general, the data suggest that the participants preferred the design operations provided in Object Mode than those in Block Mode.

► Table 5: The mean and standard deviation of the ratings for the modelling tools

	Scale: 1 (Not effective at all) to 7 (Very effective)			
Operations	**Object Mode**		**Block Mode**	
	Mean	**S.D.**	**Mean**	**S.D.**
Add Blocks	5.19	1.223	4.67	1.237
Remove Blocks	5.38	1.360	3.61	1.378
Move	5.44	1.750	4.06	1.713
Scale	4.94	1.340	4.39	1.378
Colour	5.63	1.708	4.61	1.685
Create Object	5.63	1.147	N/A	
Remove Object	6.06	1.063	N/A	
Copy	5.19	1.515	N/A	
Group	5.50	1.317	N/A	
Ungroup	5.25	1.291	N/A	

Responses to the open-ended questions

Two open-ended questions were included in the questionnaires in addition to using the Likert scales to obtain the participants' satisfaction. The first question asked users to describe the occasions when they think the system is particularly good, while the second question asked users to describe the occasions when they think the system is particularly poor.

Participants in Block Mode addressed two shortcomings that participants in Object Mode did not mention. First, eight participants reported that they suffered from difficulty in manipulating shapes block-by-block. Second, eight participants commented that they could not draw the blocks at the desired position. This problem was due to the perspective distortion when mapping the 3D scene onto the 2D user interface. However, none of the participants in Object Mode mentioned this although this problem was also encountered in Object Mode.

4.3. Discussion

In summary, the findings of this study show significant improvements when the objects were used, and no measure reports that the usability was worsened. This clearly supports the hypothesis that structuring voxel models into objects can improve the usability of the voxel-based design system. The following sections discuss how structuring influences the usability of the system in three aspects: the process, the user satisfaction, and the product of collaboration.

Effects on the collaborative design process

While structuring the voxel models into objects, the codes of conversations reveal that the participants have more design activities. The statistical analysis on the coding shows that the participants in Object Mode have more high-level design discussions which significantly affect their design solutions (Figure 6). On the contrary, participants who work on the voxel level tend to focus more on low-level design and issues such as digesting the tasks. This suggests that the participants collaborate more effectively when the objects are supported.

Additionally, the study finds that the two groups of participants use "Remove Blocks" and "Move" differently. As noted in Figure 7, when the operations were supported at the level of objects (Object Mode), the participants use more "Move" than "Remove Blocks" operations. This difference is due to the fact the semantics of the design operations is affected accordingly when the voxel models are structured into objects. For example, the "Move" operation in Object Mode allows participants to move the objects to a more meaningful position. The capability to move objects meaningfully facilitates the recomposition of the design. Similarly, users in Object Mode can remove objects instead of removing blocks, which is a more effective editing operation than low-level voxel deletion. Clearly, this was engaged widely, as reflected in the decrease in the use of "Remove Blocks" in Object Mode because the participants applied "Remove Object" instead. Thus, users were engaged at a semantic level of design when objects were available.

Effects on user satisfaction

It is revealed from the responses in the questionnaires that there are differences in user satisfaction between the two groups of participants; user satisfaction was higher when given the opportunity to interact or work on objects.

The score of Communicative Effectiveness is significantly higher in Object Mode than in Block Mode (Figure 8). This score reflects how users can effectively communicate ideas using the blocks, which in turn means that the participants have the perception that they could communicate design ideas more effectively with object structuring.

In addition, participants favoured the operations in Object Mode than those in Block Mode. Particularly, they preferred the "Remove Blocks" and "Move" operations in Object Mode. Comparing this finding with the uses of design operations as discussed in the previous section, both the measures of collaborative design process and user satisfaction show that the "Remove Blocks" and the "Move" operations benefit from the introduction of object structuring. It appears that the objects make participants more willing to use "Move" and "Remove Blocks" in editing their design. Compared with the "Move" and "Remove Blocks" at the voxel level, the same operations at the object level allowed them to modify their design more efficiently.

On the other hand, the dissatisfaction with the design operations provided at the voxel level is further supported by the responses to the open-ended questions. Many participants mentioned the difficulty of manipulating models in a block-by-block manner; they also reported that they were dissatisfied with the poor control in drawing the blocks at the desired positions. What is particularly interesting here is that this problem is due to the perspective distortion of moving in 3D space with the 2D user interface which can happen in both Object Mode and Block Mode. It is significant that none of the participants in Object Mode mentioned the problem although they encountered it which is probably because the participants in Object Mode can edit the objects efficiently and move them to the right location if the objects are not positioned correctly at the first time. However, since editing is more difficult in Block Mode, drawing the geometries at the right location is more of a concern in this case.

Effects on the design quality

While both the collaborative design process and user satisfaction benefit from using objects for voxel modelling, the improvement to design quality is statistically not clear. The scores obtained for Object Mode are slightly higher than those for Block Mode, but the number of outcomes (17 in total) produced in this study appears insufficient to conclude any significance. However, it is likely that structuring voxels into objects does not reduce the quality of the designs, although it cannot be concluded that participants either in Block Mode or in Object Mode produce better design outcomes.

Further studies are needed to examine the effect on the design quality of structuring voxels into objects. It may be that the task needs to be more complex so that the effect on design quality can be more obvious, although the voxel environment does place limitations on the complexity of the design task. To overcome this, it may be possible to extend the nature of the task to require the inclusion of some textual descriptions about the outcomes instead of only presenting a sketch so that more information can be given to the evaluators to assess the quality.

Limitations

Although the study provides evidence that the structuring of voxel models can be useful in supporting the early stages of design, there are some limitations that affect the generalizability of the findings. First, this experiment provided just one design task for the participants to execute, and although the design task was created by an experienced architectural teacher, and it is representative of the design tasks used in the early stages of design, the effects of different tasks on usability measures are unknown.

Another limitation is related to the sample used. Obviously, the second year architectural students chosen to be the participants in this study have much less architectural knowledge compared with experts. As the performance of using digital design tools may be influenced by the design expertise of the users [29], the findings of this study may therefore not be applicable to expert users.

CoBlocks follows the "3D painting" methods used in DDDoolz. However, other voxel-based systems use different modelling techniques. Some systems, such as FreeForm, may support modelling using 3D haptic devices, and therefore, a variety of modelling techniques and input devices used in the voxel-based system may influence the usability findings differently. However, this study did not explore the impacts of these factors in details.

5. CONCLUSION

Most voxel-based design systems use unstructured voxel models, and while most existing voxel-based design systems are not responsive to the needs of any structure in the voxel models, we examine whether the structuring of the voxel models would improve the usability of voxel-based design systems. Specifically, this paper reports the development and usability evaluation of our collaborative design system, CoBlocks, which is distinct from other voxel-based design systems because it can structure voxel models into objects. In order to examine the usability improvements, a controlled experiment was conducted by a comparison of participants collaborating with the structured voxel models and with the unstructured voxel models. The results show that users can interact more effectively and are more satisfied with the structured models.

However, we believe that there is still much room for improvement even if structuring voxel models into objects has been demonstrated to be useful in CoBlocks. For example, CoBlocks allows users to group objects and ungroup the grouped objects to redefine their semantics. Instead of grouping and ungrouping the objects, future research could explore other structuring possibilities that may further facilitate the design interaction such as "slicing" in order to segment an arbitrary portion of an object.

If CoBlocks were not limited to cubic forms, modelling non-cubic forms would be technically possible by reducing the size of the voxels and using rendering methods such as ray casting and marching cubes. Such capabilities are available in some voxel modelling tools (e.g., FreeForm) enabling users to create models with curved surfaces. Future work can hence enhance the "3D painting" method so that it can be used to create curved surfaces.

Finally, as compared to other CAD systems, CoBlocks offers very basic functions to develop conceptual design quickly. This limited functionality may not satisfy many experienced CAD users, and although these additional functions are worth looking into for the purpose of improving the usability of CoBlocks, it is appropriate for tools for conceptual design to be kept simple and effective. As addition features are made available, it is possible that similar problems will be encountered as in conventional CAD systems in which systems are perceived as overly rigid and too complicated to be used in the early stage of design.

Acknowledgement

We would like to thank all the members of the Digital Design Research Group at the Department of Architecture of The University of Hong Kong for their help and support.

References

1. Mitchell, W. J. and McCullough, M., *Digital Design Media : A Handbook for Architects and Design Professionals*, Van Nostrand Reinhold, New York, 1991.
2. Schnabel, M. A. and Kvan, T., *Design, Communication & Collaboration in Immersive Virtual Environments*, International Journal of Design Computing, 2002, 4.
3. Nam, T.-J. and Wright, D., The Development and Evaluation of Syco3d: A Real-Time Collaborative 3d Cad System, in: *Design Studies*, 2001, 22, 557–582.
4. Greco, J., The CAD Collaboration Trials, in: *Computer Graphics World*, 2001.
5. Tay, F. E. H. and Roy, A., Cybercad: A Collaborative Approach in 3D-CAD

Technology in a Multimedia-supported Environment, in: *Computers in Industry*, 2003, 52, 127–145.

6. Dorsey, J. and McMillan, L., Computer Graphics and Architecture: State of the Art and Outlook for the Future, in: *SIGGRAPH Computer* Graphics, 1998, 32, 45–48.
7. Galyean, T. A. and Hughes, J. F., Sculpting: An Interactive Volumetric Modeling Technique, in: *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques*, 1991, 267 – 274.
8. Wang, S. W. and Kaufman, A. E., Volume Sculpting, in: *Proceedings of the 1995 Symposium on Interactive 3D Graphics*, ACM Press, Monterey, California, United States, 1995, 151 –156.
9. Senar, B., Pedgley, O., Wormald, P. and Campbell, I., Incorporating the Freeform Haptic Modelling System into New Product Development, in: Oakley, I., O'Modhrain, S. and Newell, F., eds., *EuroHaptics 2003*, Dublin, Ireland, 2003, 26–39.
10. De Vries, B. and Achten, H. H., DDDoolz: Designing with Modular Masses, in: *Design Studies*, 2002, 23, 515–531.
11. De Vries, B., Jessurun, A. J. and van Wijk, J. J., Interactive 3d Modeling in the Inception Phase of Architectural Design, in: Roberts, J. C., ed., *Eurographics 2001*, United Kingdom, 2001, 265–271.
12. Donath, D. and Regenbrecht, H., Using Virtual Reality Aided Design Techniques for Three-Dimensional Architectural Sketching, in: McIntosh, P. and Ozel, F., eds., *Design Computation: Collaboration, Reasoning, Pedagogy*, Acadia, Tucson, Arizona, USA, 1996, 201–212.
13. Achten, H. and de Vries, B., Multiple Sketch Users in Dddoolz, in: Stellingwerff, M. and Verbeke, J., eds., *Accolade – Architecture, Collaboration, Design*, Delft University Press, Delft, The Netherlands, 2001, 153–162.
14. Mitchell, W. J., *The Logic of Architecture : Design, Computation, and Cognition*, MIT Press, Cambridge, Mass., 1990
15. Scrivener, S. A. R. and Clark, S. M., Sketching in Collaborative Design in: MacDonald, L. W. and Vince, J., eds., *Interacting with Virtual Environments*, John Wiley & Sons Ltd, Chichester,1994, 95–117.
16. Kavakli, M., Scrivener, S. A. R. and Ball, L. J., Structure in Idea Sketching Behaviour, in: *Design Studies*, 1998, 19, 485–517.
17. Achten, H., de Vries, B. and Jessurun, J., DDDoolz – a Virtual Reality Sketch Tool for Early Design, in: *Proceedings of the Fifth Conference on Computer Aided Architectural Design Research in Asia*, Singapore, 2000, 451–460.
18. Oh, J.-Y. and Stuerzlinger, W., A System for Desktop Conceptual 3d Design, in: *Virtual Reality*, 2004, 7(3–4), 198–211.
19. Leu, A. and Chen, M., Modeling and Rendering Graphics Scenes Composed of Multiple Volumetric Datasets, in: *Computer Graphics Forum*, 1999, 18, 159–171.
20. Jense, G. J., Voxel-Based Methods for Cad, in: *Computer-Aided Design*, 1989, 21, 528–533.
21. Chen, M., Tucker, J. V. and Leu, A., Constructive Representation of Volumetric Environments in: Chen, M., A. E. Kaufman and R. Yagel, (eds.), *Volume Graphics*, Springer,2000.
22. Greenberg, S. and Marwood, D., Real Time Groupware as a Distributed System: Concurrency Control and Its Effect on the Interface, in: *Proceedings of the 1994 ACM Conference on Computer-supported Cooperative Work*, ACM Press, Chapel Hill, North Carolina, United States, 1994, 207–217.
23. Gutwin, C. and Greenberg, S., The Effects of Workspace Awareness Support on the Usability of Real-Time Distributed Groupware, in: *ACM Transactions on*

Computer–Human Interaction, 1999, 6, 243–281.

24. Vera, A., Kvan, T., West, R. and Lai, S., Expertise and Collaborative Design, in: *CHI '98 Conference Proceedings*, ACM, Los Angeles, 1998, 503–510.
25. Lewis, J. R., IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use, in: *International Journal of Human–Computer Interaction*, 1995, 7, 57–78.
26. Lewis, J. R., Psychometric Evaluation of the PSSUQ: Using Data from Five Years of Usability Studies, *International Journal of Human–Computer Interaction*, 2002, 14(3/4), 463–488.
27. Coovert, M. D. and McNelis, K., Determining the Number of Common Factors in Factor Analysis: A Review and Program, in: *Educational and Psychological Measurement*, 1988, 48, 687–693.
28. Nunnally, J. C., *Psychometric Theory*, McGraw-Hill, New York, 1978
29. Verstijnen, I. M., Hennessey, J. M., van Leeuwen, C., Hamel, R. and Goldschmidt, G., Sketching and Creative Discovery, in: *Design Studies*, 1998, 19, 519–546.

Steve Kuan
The University of Hong Kong
Department of Architecture
Pokfulam Road, Hong Kong S.A.R., P.R. China

kskuan@graduate.hku.hk

Thomas Kvan
The University of Sydney
Faculty of Architecture
Sydney, Australia

tkvan@arch.usyd.edu.au