A Comparison of Real and Virtual 3D Construction Tools with Novice Users

Hooman Baradaran, Wolfgang Stuerzlinger
Human-Computer Interaction Lab
Department of Computer Science
York University
Toronto, ON
www.hoomanb.com, www.cs.yorku.ca/~wolfgang

1. ABSTRACT
In this paper, we present the results of a comparative evaluation of a "virtual Lego" system against real Lego bricks with novice users. The virtual Lego system was designed to emulate the behavior of real Lego bricks as close as possible, while still allowing for efficient operation. We implemented two different user interfaces for the virtual Lego system, namely controlling it with a 2D mouse, based on an efficient mapping from 2D to 3D, as well as with a 3D haptics device, which provides haptic feedback to the user. The results of our study show that real Lego is still significantly faster for first-time users (i.e. users with minimal training). A more surprising result is that the 2D mouse condition and the 3D haptics condition did not differ significantly - even though the 3D haptics condition provides much richer feedback. We discuss the results and speculate about the underlying reasons.

1.1. Keywords
Comparing real and virtual, haptics, 3D construction.

1.2. 1 INTRODUCTION
This work is based on system for 3D scene construction, which is targeted at the conceptual design session. Conceptual design happens in an early stage in the design process. Here, the designer explores the problem/solution space to produce an optimum solution that satisfies design constraints. For this it is necessary to have tools that allow the user to rapidly build and iteratively modify a model without much commitment to detail. Another important issue is the ability to perform structural modifications efficiently. The 3D construction system is based on the concept of Lego™ bricks. A simple way to describe this system is to say that it simulates real Lego with virtual bricks that behave similarly to real Lego bricks, i.e. they attach rigidly to each other, and one can stack them in various ways. However, additional functionality, such as re-coloring and resizing, is also available.

The fundamental idea behind building a system on the concept of Lego is that most novice users are at least to some degree familiar with Lego blocks as a basic construction tool. From a user's standpoint, Lego is very simple and versatile, in that various models can be rapidly built using a small variety of simple blocks. From a system developer's standpoint the choice of bricks makes it is easy to test ideas for interaction techniques, while still keeping relatively close to the context of a real application.

The 3D construction system underlying this work is normally used with a standard mouse and a normal desktop monitor. The "translation" of mouse actions to 3D manipulation is handled via algorithms that emulate the behavior of real Lego so that manipulation of virtual objects is similar to the behavior of real Lego bricks. For this study, we added support for a 3D haptics device to the system. Such devices enable 3D input and also provide force feedback to the user, thus enabling the user to feel when they hit a virtual brick. The basic motivation for this is that force feedback provides better feedback for the placement of new blocks and hence one might argue that this would enable users to work more efficiently.

1.3. 1.1 Previous Work

While there are many commercial systems targeted at the creation of 3D content, few of them can be considered easy-to-use. In the last years, several researchers have presented several systems that try to present simpler user interfaces for this task (see e.g. [2],[3],[4],[5],[6]). A related topic is user interfaces for virtual reality systems, which is surveyed by a recent book [1].

In general, there are 2 main approaches to the problem of specifying modifications to the 3D structure of the underlying geometric model. In general, specifying the position of an object requires 6 dimensions (3 translations and 3 rotations). However, we choose to ignore rotations in this context, as this simplifies the discussion as well as the experimental design greatly - while still allowing for interesting construction tasks. To specify the 3D position of an object, either a 2D input device together with algorithms to map 2D motion to 3D motion is used or a 3D input device is used.

Usually a mouse is used as a 2D input device. Mapping 2D input to 3D manipulation is a problem that has is at the core of many examples of previous work. Usually, the position of the cursor is used to define a ray from the viewer into the scene and the first visible object is then manipulated. For movement, one can either move objects along coordinate system axis, parallel to the camera plane, or on the surfaces of other objects. Moving objects along coordinate axis (an approach often used by commercial 3D software) simplifies implementation and gives the greatest degree of flexibility, but in not necessarily efficient. Moving objects parallel to the camera plane is also easy to implement but quickly leads to surprising results as most naïve users have trouble understanding this concept. Finally, moving objects on the surfaces of other objects requires more computation, but corresponds better to the real world, where no object can (permanently) float in space. This last approach was chosen by most researchers to implement parts of their easy-to-use 3D construction systems [2],[3],[5],[6].

Another approach is to use 3D input devices, often called 3D/6D trackers. Examples include elelctromagnetic, ultrasonic, optical, and inertial, and hybrid systems. Most of these systems suffer from jitter, need precise calibration to achieve a moderate degree of accuracy and require extensive infrastructure. They are typically used together with Virtual Reality equipment, i.e. immersive display systems. As all of immersive systems require head-gear, designers have not accepted these systems in general. The most precise 3D/6D input devices are mechanical arms. Kitamura [4] used such a mechanical tracking device in a constraint-based 3D construction system.

Very few of these systems have been evaluated against real-world construction tasks. The most relevant comparisons are the studies performed by Kitamura *et al.* [4] and Oh *et al.* [5]. Both of these studies compared the performance of users with real construction blocks and with a virtual 3D construction system used with a 3D tracker, but Oh also compared the performance with a (2D) mouse. The task is usually the construction of a simple object consisting of 5-10 primitives. Kitamura's results show that the constraint-based virtual system used with a 3D mechanical arm tracker was approximately 50% slower than the same task with real blocks, after a training period of unspecified length. The condition with a 3D tracking device without constraints proved to be even worse and was more than 150% slower than real blocks (all data approximated from Figure 16 in [4]). Oh [5] tested also a (electromagnetic) 3D tracker in a pilot study, corresponding to the "without constraints" condition of Kitamura, but found that with naïve users that performance was 260% slower. A more formal evaluation of the (2D) mouse condition showed that naïve users were approximately 170% slower with the virtual system, compared to the real construction task. The authors of this work cite the need to select blocks from a menu as one of the main sources of this drop in performance, as well as the jitter of the magnetic tracking device.

Figure 1. Virtual Lego and the Phantom

Recently, haptics devices have been introduced, which basically extend a mechanical tracking device with force-feedback capabilities. This enables the user to feel the contact of an object with other objects in the scene. One might argue that this would provide a better from of feedback to the user as to where a 3D object is precisely. Hence, we expect that the addition of force-feedback to a 3D construction system would increase performance.

2. 2 IMPLEMENTATION

For this comparison, we used an implementation of a virtual 3D construction system, which is similar to the virtual Lego system presented by Oh *et al.* [5]. The user can simply pick up and drop blocks by dragging them with the left mouse button. As in [5], the mouse condition used the foremost visible surface under the cursor to determine where to place the currently manipulated block. To help the user in placing a block into a position that is not visible from the current viewpoint, a simple navigation scheme is used, which allows the user to rotate the viewpoint on a sphere of fixed size around the object, while still looking towards the center. Navigation was mapped to the right mouse button. To eliminate potentially confounding factors, our implementation did not provide the ability to rotate, re-color or resize blocks.

As 3D input device, we used a Phantom haptics device (by SensAble Technologies). As the device already delivers 3D coordinates, we used a one-to-one mapping to map user input to 3D manipulation of the current block. Appropriate force-feedback was generated whenever the software (supplied by SensAble Technologies) reported a contact/collision between the moving object and the rest of the scene. As one can move behind objects with a 3D input device, we disabled navigation for this condition.

3. 3 USER STUDY

To verify the usability of the system, we conducted a user study. Eight paid participants (3 females, 5 males, all in their 20's) out of a pool of undergraduate university students were recruited. All participants used computers on day-to-day basis. Seven out of eight participants had experience with 2D authoring tools and most of them claimed to be comfortable with the use of a mouse for design or draw tasks (an average of 5.5 on a 7 point Likert scale). Only one participant had limited experience with virtual reality systems (except 3D games and amusement parks). Two participants mentioned limited experience with Autocad and 3D Studio, but both were not using these systems on a regular basis. None of the participants had any previous experience with our system.

3.1. 3.1 Test Procedure

The test consisted of a practice session and three evaluation sessions. First, participants were introduced to the general operation of the virtual Lego system and were asked to practice the movement of a single 3D object with the help of the instructions. Both input conditions (the mouse and the Phantom) were practiced until participants felt comfortable. The average time period for the whole practice session was 10 minutes.

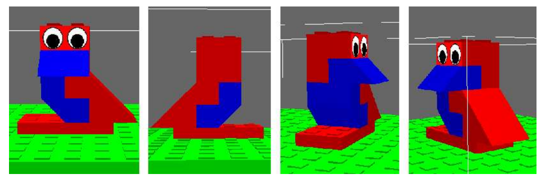

Figure 2. View of target object. Lines to clearly delineate blocks were added to the handout by hand for clarity (not shown).

In the evaluation sessions, the participants had to perform the same task with the three different conditions. The task was to build an object (a simple duck) out of 10 given Lego blocks. The participants were presented with a printout of a screen shot of the fully constructed duck, so they would not know what the object would look like in real Lego blocks. However, since the Virtual Lego system does not show clear boundaries between Lego blocks we added these to the screen shots by hand.

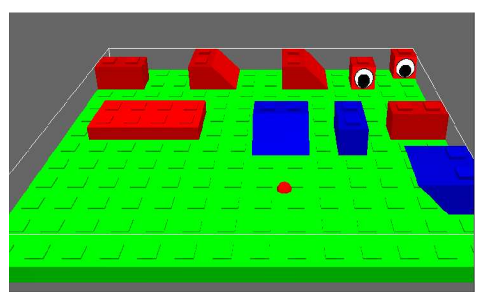

Figure 3. Initial position of the board and the Lego blocks at start of a trial. For the real Lego conditions pieces were placed on a sheet of paper in the same layout.

In the three conditions, the participants had to build this duck using real Lego blocks and using our virtual Lego system, once using the mouse and once with the Phantom. The order of the conditions was counterbalanced among subjects to avoid potential learning effects.

To avoid potentially confounding factors, we always started a trial with the 10 blocks in the same positions on the board as shown in figure 3. Hence, there was no need to rotate blocks, and the user just had to assemble the blocks using drag operations. For the trials with real Lego blocks we drew the position of the blocks on a sheet of paper in the same positions as in figure 3 and placed the blocks on these positions before each trial.

3.2. 3.2 Task Completion Times

An ANOVA analysis of the trial task completion times show that the effect of the condition is significant ($F_{7,2} = 7.10$, $p < .01$). A Tukey-Kramer multiple-comparison test reveals that the difference between real Lego and the two conditions based on the virtual system is significant. However, there is no significant difference between the mouse and Phantom conditions.

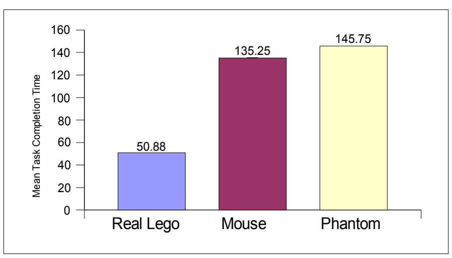

Figure 4. Mean task completion times.

The average completion times are shown in figure 4. It is evident that the mouse condition is approximately 165% slower than the real Lego condition, which is in line with result of previous experiments performed with the virtual Lego system. It is also evident that the Phantom condition did not perform better than the Mouse condition.

3.3. 3.3 Qualitative results and more observations

After finishing all tasks, each participant rated the Phantom input device on a 7-point Likert scale on the following questions: “How does the Phantom feel in comparison to real Lego blocks?” and “How likely are you to use the Phantom over the mouse if you were to design another object or use a similar system”. The answer to both questions was mostly positive with an equal average of 4.75 for each question.

Six out of eight participants commented that the “feel” and true 3D movement of the Phantom as opposed to the 2D to 3D “mapping” for the mouse made it easier for them to design objects. Two participants even found the Phantom easier to use than the mouse, but most of them commented that they are not yet as familiar with the Phantom as they are with mouse and some commented they don't find the Phantom “user-friendly” enough. Almost all participants made many mistakes with the Phantom, while in the mouse condition they hardly had to repeat a movement.

The only case where some participants made more than a few mistakes with the mouse was where the mouse was used in the first experiment. Similarly, in one case real Lego was slower than the mouse. In this case the subject was presented with the real Lego evaluation first and we observed that the subject was having trouble building the model for the first time. We attribute this to the fact that participants had to spend time to understand how the object was to be put together.

Compared to the mouse condition, three participants performed better in the Phantom condition. In two of these cases, the mouse condition was the first trial and we observed that the participants spent most of the time correcting and rebuilding parts of the duck to learn how to build the model. The third participant who was faster with the Phantom had also relatively little computer experience, both according to the questionnaire and our observation, and had to make lots of corrections for both input devices.

As explained earlier, the mouse condition necessitates a mapping from 2D to 3D combined with the ability to navigate. According to our observations, some (but not all) of the “lost” time can be explained by the fact that the user may have to navigate when they can’t see the position where the next block goes (e.g. the block behind the “beak” in a frontal view).

Some participants who first used the mouse condition and then the Phantom condition commented that they would have liked the ability to navigate in this condition, too.

4. 4 DISCUSSION

It is evident from the results that virtual construction systems are still slower than real construction kits. Clearly, part of this result is due to the fact that we used novice users. One of the authors of this paper has been using the system frequently and informal timings show that a highly trained user can *sometimes* achieve the same speed with virtual and real systems. However, we still believe that one of the grand challenges of 3D user interface design is to come up with solutions that allow an average user to achieve the same performance as a real construction task. We hasten to point out that virtual construction systems have many abilities that do not exist real kits (e.g. the ability to play back, resize, etc.), but it would help adoption of new systems greatly if the overhead would be smaller or even negligible.

One factor that might have biased our result maybe the fact that using real Lego, participants are allowed to use both hands while in either virtual technique only use of one hand is allowed. In addition, in real Lego the user holds a brick with multiple fingers and hence has better feedback than in the virtual system, where the user has to manipulate each brick as if it were mounted at the end of a hand-held short stick.

One potential explanation of the results of our study is that the relative greater familiarity with a mouse may be the reason behind the relatively inferior performance of the force-feedback device. However, today it is hard to find subjects that are not familiar with standard desktop input devices. A better solution would be to provide training for the Phantom device. One way to do this is to do a longer-term study with repeated trials with various objects. In theory, we would expect that after a while a cross-over point would be reached. We intend to investigate this in future work.

Another possible explanation for some of the observed effects are the various implementation choices we made for the mouse and Phantom conditions. This includes the details of the movement algorithm, constants for snapping distances and other internal thresholds. However, our results seem to be reasonably consistent with previous work. Hence, we do not believe that this was an issue that influenced the results in a major way.

5. 5 CONCLUSIONS AND FUTURE WORK

In this paper, we presented an evaluation of a virtual 3D construction system, which allows novice users to rapidly construct 3D models. We compared the performance of first-time users with a 2D input device and a 3D force-feedback device with a real construction task, performed with Lego blocks. The results show that real Lego condition was approximately 2.7 times faster than the virtual system.

We plan to perform a longer-term evaluation of the system in the future, to see when the force-feedback condition becomes faster than the mouse condition. Another issue that we intend to address in future work is the fact that even though we used a fairly simple 3D object, some participants had problems rapidly understanding the structure and potential assembly sequences, which led to errors.

6. ACKNOWLEDGMENTS

The authors would like to thank J.-Y. Oh for help with the implementation, L. Sergio for the loan of the equipment and all the participants in the experiment.

7. ABOUT THE AUTHORS

Hooman Baradaran is a computer science senior at York University in Toronto, Canada. He can be reached at www.hoomanb.com .

Wolfgang Stuerzlinger is an associate professor of computer science at York University. His web site is www.cs.yorku.ca/~wolfgang .

8. REFERENCES

1. Bowman D., Kruijff E., LaViola J., Poupyrev I., *3D User Interfaces: Theory and Practice*, Addison-Wesley, 2004.
2. Bukowski, R. & Sequin, C. (1995), Object associations: a simple and practical approach to virtual 3D manipulation, *Symp. Interactive 3D Graphics 1995*, pp. 131-138.
3. Igarashi, T., Matsuoka, S. & Tanaka, H. (1999), Teddy: A sketching interface for 3D freeform design, *SIGGRAPH 1999*, pp. 409-416.
4. Kitamura, Y., Yee, A., Kishino, F., A sophisticated manipulation aid in a virtual environment using dynamic constraints among object faces, *Presence*, 7(5), pp. 460-477.
5. Oh, J.-Y., Stuerzlinger, W. A System for Desktop Conceptual 3D Design, *Virtual Reality*, vol. 7 (2004), pp. 198-211.
6. Zeleznik, R., Herndon, K., Hughes, J., SKETCH: An interface for sketching 3D scenes, *SIGGRAPH 1996*, pp. 17-21.