# Exploring Brick-Based Navigation and Composition in an Augmented Reality

Morten Fjeld[1], Fred Voorhorst[1], Martin Bichsel[2], Kristina Lauche[3],
Matthias Rauterberg[4], and Helmut Krueger[1]

[1] Institute for Hygiene and Applied Physiology (IHA),
Swiss Federal Institute of Technology (ETH), Clausiusstr. 25, CH-8092 Zurich
{fjeld, voorhorst, krueger}@iha.bepr.ethz.ch
www.fjeld.ch
[2] Institute for Design and Construction Methods (IKB),
Swiss Federal Institute of Technology (ETH), Tannenstr. 3, CH-8092 Zurich
mbichsel@ikb.mavt.ethz.ch
[3] Institute for Work and Organisational Psychology (IfAP),
Swiss Federal Institute of Technology (ETH), Nelkenstr. 11, CH-8092 Zurich
lauche@ifap.bepr.ethz.ch
[4] Center for Research on User-System Interaction (IPO),
Technical University Eindhoven (TUE), Den Dolech 2, NL-5612 AZ Eindhoven
g.w.m.rauterberg@tue.nl

**Abstract.** BUILD-IT is a planning tool based on computer vision technology, supporting complex planning and composition tasks. A group of people, seated around a table, interact with objects in a virtual scene using real bricks. A plan view of the scene is projected onto the table, where object manipulation takes place. A perspective view is projected on the wall. The views are set by virtual cameras, having spatial attributes like *shift*, *rotation* and *zoom*. However, planar interaction with bricks provides only position and rotation information. Object height control is equally constrained by planar interaction. The aim of this paper is to suggest methods and tools bridging the gap between planar interaction and three-dimensional control. To control camera attributes, *active* objects, with intelligent behaviour are introduced. To control object height, several real and virtual tools are suggested. Some of the solutions are based on metaphors, like *window*, *sliding-ruler* and *floor*.

## 1 Introduction

BUILD-IT is a planning tool based on computer vision technology, with a capacity for complex planning and composition tasks [19] [20]. The system enables users, grouped around a table, to interact in a virtual scene, using real bricks to select and manipulate objects in the scene (Fig. 1). A *plan view* of the scene is projected onto the table. A perspective view of the scene, called *side view*, is projected on the wall. The plan view

H.-W. Gellersen (Ed.): HUC'99, LNCS 1707, pp. 102-116, 1999.

contains a storage space with originals, allowing users to create new objects. Object selection is done by putting a brick at the object position. Once selected, objects can be positioned, rotated and fixed by simple brick manipulation. They are de-selected, and stay put, when the brick is covered or lifted off the table. Objects brought back to the storage space are deleted from the views.

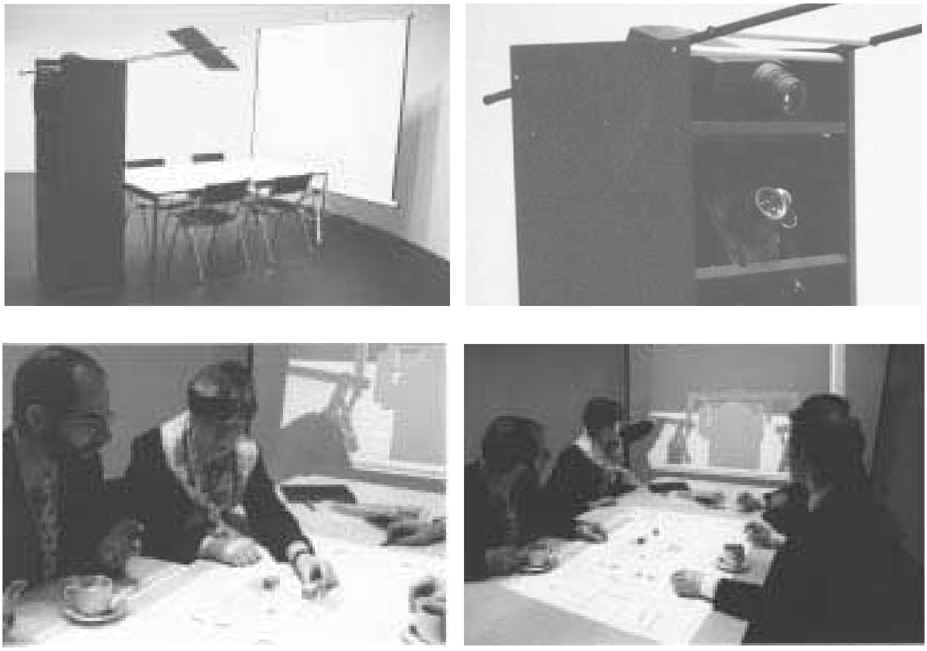


**Fig. 1.** BUILD-IT consists of a rack, mirror, table, chairs and a screen (top left). In addition to a high-end personal computer (PC), the rack contains two beamers, a video camera and a light-source (top right). The system offers a horizontal *plan view* for combined action and perception (bottom left), and a vertical *side view* with a perspective of the situation (bottom right).

Using the BUILD-IT system as a research platform for graspable interaction [8], our exploration takes the following path: Section 2 introduces some of the problems related to working in *real* and *virtual* environments, then indicates a few guidelines to achieve what we call *natural interaction.* Section 3 gives more details about the *interaction content*, which is configuration and planning tasks. Section 4 presents the interaction form, Augmented Reality (AR). Section 5 gives more detail on designing hand-held bricks for interaction. Section 6 presents some new implementations for three-dimensional (3D) navigation. Section 7 introduces alternative ways to control object height in the BUILD-IT system. Section 8 discusses the outcome of our design activity and suggests ways to advance the issues presented in Sections 6 and 7.

## 2 Working in Real and Virtual Environments

When people act, and interact, in a setting of real, tangible objects, we prefer to call this *natural behaviour*. Computer-supported group-work actually tends to bring people, physically, farther away from each other, setting them at remove from the concrete world. They interact with *virtual* objects and may even be totally engrossed in a *virtual world*. The larger aim of this project is to study ways to integrate real-world, *natural behaviour* with *computer-mediated work* in order to draw on the advantages of both these worlds.

First, we consider the characteristic features of natural behaviour. When people act in a real setting, they make *eye-contact*, *speak to each other* and *communicate through body language*. They can also *touch* each other and directly *manipulate* real-world objects. They may use the physical world as part of their memory and explore that world by direct interaction [25]. This fosters cooperative thinking, individual expression, and visualisation of ideas.

Using this sketch of natural behaviour as a guide, we sought feasible means of making interaction with virtual worlds more *natural*. We found it important to:

- support body motions and everyday skills [5]
- support haptic feedback [17]
- use real interaction handles with intelligent features
- connect real interaction-handles with virtual objects in a clear manner
- give consistent user feedback

Second, we chose to make the distinction between interaction *content* and *form*. *Content* refers to the kind of tasks that are being solved and the social setting under which they are performed. *Form* refers to such things as the layout of the human-computer interface, which kind of handles are being used and what kind of interaction features are offered.

*Content*: In this project, the kind of tasks to be solved are composition tasks with existing objects. Such objects have analogues in the real world, like chairs, machines and buildings, but neither molecules nor proteins. We study how subjects solve such tasks, both individually and in groups. The important criteria for successful task solving are given by users' task solving performance, which can be defined in various ways depending on the task.

*Form*: We work in the context of Augmented Reality (AR) where virtual objects and real objects are manipulated in one, coincident interaction-space. We study different ways of implementing real interaction handles and their connection to the virtual world. Also, we study different manners of navigation in the virtual word. Finally, we are interested in two-handed operations and how interaction can be designed to accommodate the dominant and non-dominant hand.

Based on the said dichotomy of content and form, we set out to explore possible implementations. The long-term aim is to determine what is a *good design strategy* for interfaces supporting *natural behaviour*. Thus we need a way to construe *natural* in terms of guidelines pertaining to interface design. Some of these *design guidelines* may be as follows:

- give visual feedback consistent with user expectation
- assure ease of navigation
- support exploratory behaviour and ensure that ’trying out’ is a low risk activity
- draw on bimanual handling, adapted to dominant and non-dominant hand
- use real handles with clear explanations of their connection to the virtual world
- offer real handles as extended user memory

## 3 Interaction Content: Configuration and Planning Tasks

For most planning tasks in engineering and architecture systems, drawings and two-dimensional (2D) models have been replaced by Computer-Aided Design (CAD) system. This change has given raise to a range of supportive tools for drawing and information processing. However, this also entails less immediate contact among CAD users, planning experts and sales people.

First, we performed a task analysis with potential user groups for our system. We observed that they spent a great deal of time in discussions with their clients and noticed that off-line CAD support is hardly available during their sales trips. This lack of support sometimes caused misunderstandings with the designers at home, trying to communicate their solutions to the travelling sales people. Also, some of the customers were not familiar with 2D layout techniques; they were unable to imagine what an object would look like in three dimensions. Therefore, an easy-to-handle, three-dimensional (3D) planning tool proved to be attractive for planning experts and sales people. A distributed, networked system would additionally allow for interaction among users located at different sites.

Actually, modern management concepts like *simultaneous engineering* are based on dynamic interaction among cooperating experts. Simultaneous engineering goals are realised through an early involvement of different functions, like marketing and manufacturing. Many companies are forming multi-functional teams as a solution to rapid product development goals. For these teams to be significantly more successful than other organisational approaches, it is necessary to build a process around them that is appropriate for this new type of organisation structure. The process should encourage team-based cooperation rather than a one-person-one-screen set-up. Such requirements can hardly be met by existing technologies like video conferencing. Adequate solutions must offer more intuitive, natural interaction.

All of these considerations were taken into account in the designing of the BUILD-IT technology. The result was a system supporting early offering and design processes. This apparatus is not intended as an alternative, but rather as a complement to CAD systems. It allows for ready-made applications in various fields, such as machine configuration, city and urban planning, architecture and interior design.

## 4 Interaction Form: Brick-Based Augmented Reality

Computer-supported group work has allowed for distant and asynchronous communication between people and has helped build bridges in our global economy. This has brought about many well-advertised advantages, ranging from economic benefit to less status-oriented network communication. However, with many computer applications users hardly interact with their physical environment. They deal with virtual objects only, which is also the case for most single-user applications. Sometimes users are even embedded in a fully virtual world, unable to draw on any attributes of the tangible world. Much of the users' mental capacity is employed to adapt to the virtual world, leaving less capacity for actual task solving.

An alternative approach is to bring the *virtual* world of computers into the *real* world of everyday human activity. This approach includes aspects of natural communication which serve as mediators for mutual understanding: eye-contact, body language and physical object handling. It is *non-intrusive*, using no gloves or helmets, and thereby respects the body-space [21]. At the same time users can still draw on the advantages of a virtually enriched world, which is of particular importance to planning tasks. The activity of planning is intrinsically virtual because it involves reflecting on and modifying objects that only exist in the future. Virtual objects can be more easily changed than physical objects, can be stored in external computer memory and can be visualised for interaction purposes. Thus both physical and virtual methods have their rightful place in a planning process. A specific aim of our project is to study ways to integrate the real-world and computer-mediated activity. This is how we came to work within the tradition of AR, where computer-generated and real-world objects are handled in one workspace.

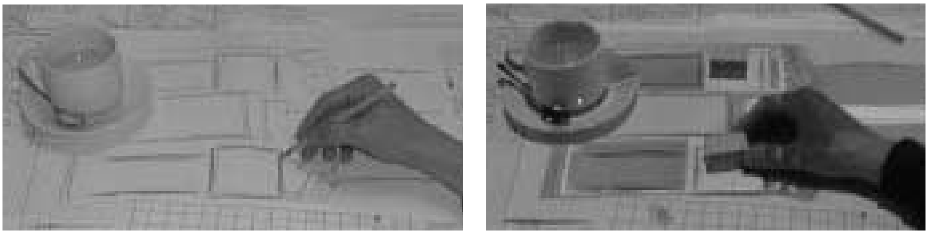


**Fig. 2**. AR means that a real workspace (left) is augmented, or enriched, by a virtual world (right). Even when users interact with the projected, virtual objects, they do not leave the real world context (e.g. sketching) and tools (e.g. pencil).

AR was first described by Wellner et al. [24]. The goal of AR is to ”allow users to continue to use the ordinary, everyday objects they encounter in their daily work and then to enhance or augment them with functionality from the computer” [16]. According to Mackay [16], AR means that computer information is projected onto drawings so that users can interact with both the projected information and the paper drawing (Fig. 2). The first brick-based AR system was described by Firtzmaurice et al. [6] using tethered bricks. Wireless brick-based system were also described by Ishii and Ullmer [12] and Underkoffler and Ishii [22].

### 4.1 The BUILD-IT System

Compared with physical, model-based layout systems, BUILD-IT offers cheaper, quicker and more exact object representation. Based on a 3D multimedia framework [MET++, 1], the system can read and display arbitrary objects. Employing Virtual Reality Modelling Language (VRML), these objects are sent from a CAD system to BUILD-IT. After a planning session, the results can be sent back to the CAD system [10].

Geometry is not the only aspect of product data. With a growing need to interact in other dimensions, such as cost and configurations, the system has been engineered to send and receive numerous forms of meta-data [10]. The potential of computer-mediated work is made readily available through automatic calculation of prices and time-to-delivery. Also, animation of objects, like robots and laser welders, combined with plant configuration, allow for live simulation [9] of production cycles.

So far, the system is a multi-user, single-site interaction tool. However, the *interaction methods* realised thus far are capable of supporting distributed networking systems. Based on this system, Sections 5-7 offer a tour through some of the recent design cycles.

## 5 Designing and Using Bricks

In BUILD-IT mediation between users and virtual worlds follows a cyclic order (Fig. 3). Users select an object by putting the brick at an object position. The object can be positioned, rotated and fixed by simple brick manipulation. An object is deselected by covering the brick. Then, another object is selected or the brick is left idle inside or outside the plan view. It was shown that a brick-based interface is significantly easier to use and more intuitive than a mouse-keyboard-screen [18].

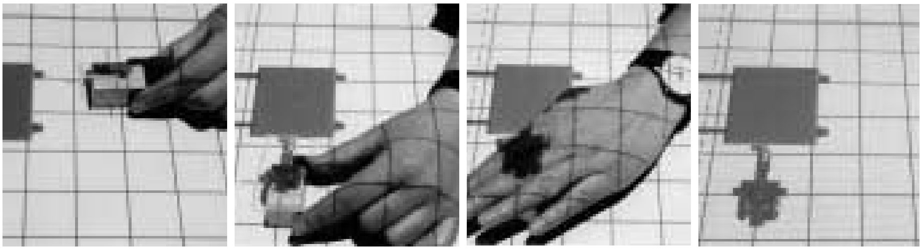


**Fig. 3.** The basic steps for user manipulations with the brick.

Bricks are handheld mediators linking a virtual world with the physical world. We aim for brick forms inviting users to perform manipulations like grasping, moving, rotating and covering. Since all the steps in the described cycle are reversible, the cost of making a mistake is low. Thus, exploratory epistemic and goal-directed pragmatic actions [14] are equally supported. Bricks can be left and picked up later anywhere in the plan view, serving as ubiquitous extended user memory (Fig. 2).

The detection of the brick is achieved by using infra-red light and retro-reflective paper [4]. The light is reflected by the paper, then picked up by an infra-red camera which in turn triggers an update of the computer image. There are only a few constraints on *brick size* and *shape*. For instance, the reflective area must exceed a two by three centimetre size. With such modest technical limitations, the major question is how to design bricks so that people can move real bricks and simultaneously immerse themselves in a virtual setting. We performed some exploratory brick modelling [15], based on different materials, forms and metaphors (Fig. 4, left).

The first feasible brick was the *block* (Fig. 4, right). The advantages of this shape is that users grasp the brick easily and that image detection is simple. However, due to the height of the brick, the image projected upon it is distorted. Therefore the brick is not always suited to mediate fluently between users and the virtual world.

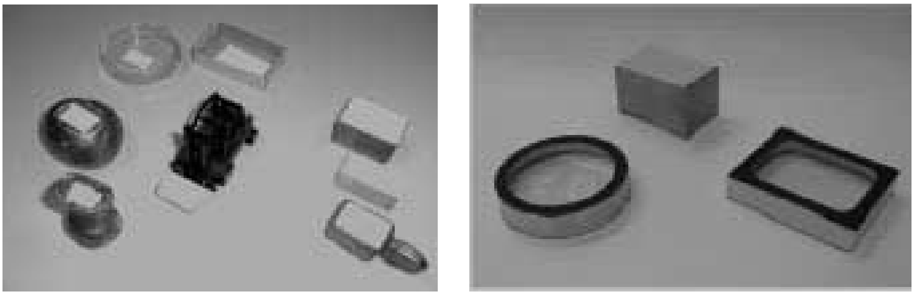


**Fig. 4.** In the beginning we experimented with different bricks, based on different materials, forms and metaphors (left). The first feasible brick was the *block* (right). The more recent *circular* and *rectangular* bricks (right) are based on the principle of *reduction screen*.

To solve such a problem Kaptelinin [13] suggested: ”deal with two interfaces instead of one user interface, with two borders, separating (1) the user from the computer and (2) the user *and* the computer from the outside world”. This situation is similar to Bateson’s [3] blind man’s stick dilemma: where is the boundary between the individual who uses a tool and the external world? Does it coincide with the individual-tool or with the tool-world boundary? In our case, the tool is the brick, mediating between individual and virtual worlds. We want to design a brick so that users perceive interaction with the virtual world, not interaction with the real world of the bricks.

A solution to this problem can be found by employing a *reduction screen* [23]. Already in common use, such screens are *borders* placed in front of a monitor to reduce flattening cues. The same idea can be used to make the brick like an open box with a narrow black border on the edge (Fig. 4, right). For users, the bottom of the box-like brick is indistinguishable from the table surface and image distortion is minimised.

A final aspect of using bricks, is how they actually connect with the virtual world. At the moment of selecting a virtual object with a real brick, a planar relation - in terms of position and rotation - is established between real and virtual. We call this a *locking mechanism*. A *locking mechanism* determines how a real brick and a virtual object stay connected from the moment of selection until the moment of de-selection.

For different brick forms appropriate locking mechanisms must be defined. For instance, the virtual object may align with the brick, its centerpoint may move to the centerpoint of the brick or both.

## 6 Spatial Navigation: Controlling Shift, Rotation and Zoom

Some basic aspects of 2D, brick-based interaction were previously explored [7]. Bimanual camera control and object manipulation in 3D graphics interfaces were also explored [2], using two mice, keyboard and screen. The innovative feature of BUILD-IT, beyond the brick-based interaction, is that the objects are part of a 3D scene. The use of the multimedia framework [MET++, 1], allows for full 3D interaction, including *shift, rotation, zoom, tilt* and *roll*. However, planar interaction with bricks provides only position and rotation information. Hence, there is a need to bridge the gap between planar interaction and 3D view control.

The exploration of an environment, or a product, is important in a range of composition and planning tasks, e.g. design of production lines, architecture and industrial design. To explore a 3D virtual world, it is essential to assume different point of views, to take an overview and to look at things in detail. This, at least, calls for a direct control of *shift, rotation* and *zoom* in *both views.*

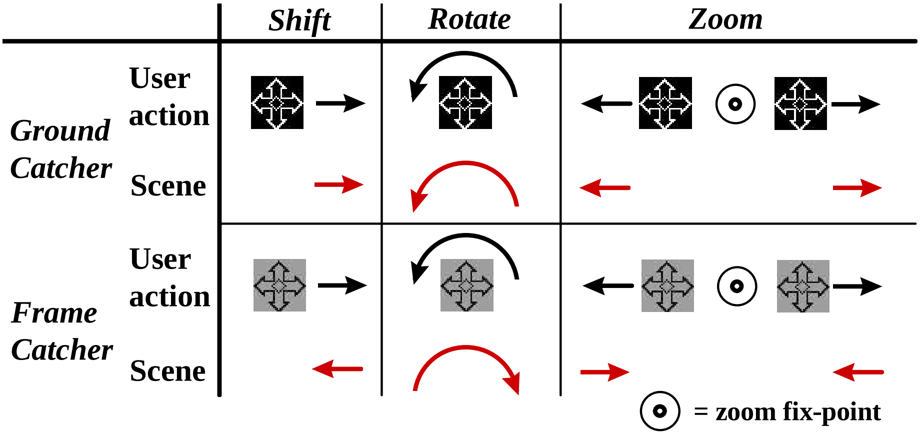
One strategy which we considered was the use of a specialised brick, which would control a side view camera. This would require extending the properties sensed by the computer vision input. We want to explore software solutions, so this approach was not pursued. Instead, *active* virtual objects were introduced. *Active* objects feature *intelligent behaviour* and support *complex operations.*

Employing active objects, plan- and side view control were implemented. For each view, two alternative methods will be explored: GroundCatcher and FrameCatcher for the plan view, Camera and Window for the side view. Two of the methods, GroundCatcher and Window, are based on *scene* handling. The other two, FrameCatcher and Camera, are based on what shall be called *observer* handling.

### 6.1 Plan View Control: GroundCatcher and FrameCatcher

One brick offers shift and rotation, a second brick adds zoom (Table 1). Zoom is given by brick movements along a connecting line; other movements give shift and/or rotation. GroundCatcher (Fig. 5) updates the scene according to user action. One quits by removing bricks. FrameCatcher (Fig. 6) updates a frame of interest according to user action. When bricks are removed, the scene adjusts to the frame. Objects have no real-world analogues.

**Table 1.** GroundCatcher and FrameCatcher



| | | *Shift* | *Rotate* | *Zoom* |
|---|---|---|---|---|
| *Ground Catcher* | User action | ■ → | ↶ ■ | ← ■ ⊙ ■ → |
| | Scene | → | ↶ | ← → |
| *Frame Catcher* | User action | ▨ → | ↶ ▨ | ← ▨ ⊙ ▨ → |
| | Scene | ← | ↷ | → ← |

⊙ = zoom fix-point

**Fig. 5.** GroundCatcher: Introducing, using, and removing bricks (left to right).

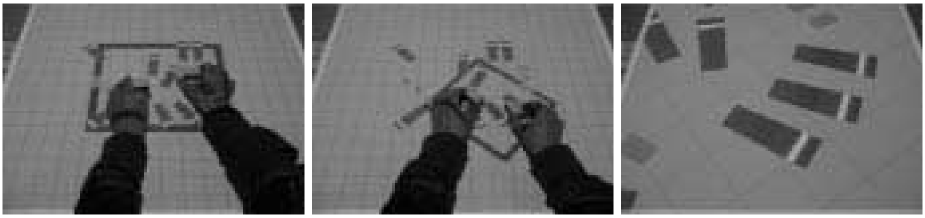
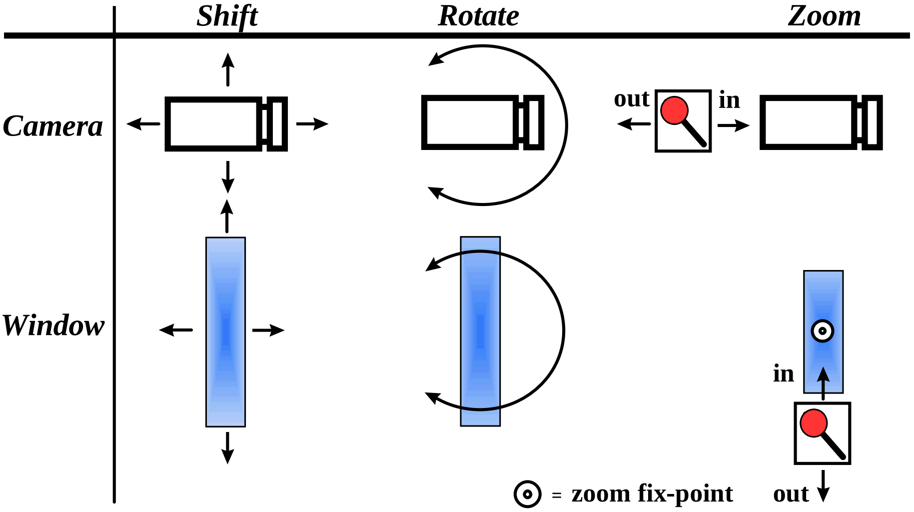


**Fig. 6.** FrameCatcher: Introducing, using, and removing bricks (left to right).

### 6.2 Side View Control: Camera and Window

Both these objects are fixed at the approximate height of a person's eyes, 1.6 meter above the ground. One brick offers shift and rotation, a second brick adds zoom (Table 2). Camera (Fig. 7) sets the virtual camera of the side view. Handle distance sets zoom. Window (Fig. 8) sets the side view border. Scaling it sets the zoom.

**Table 2.** Camera and Window.



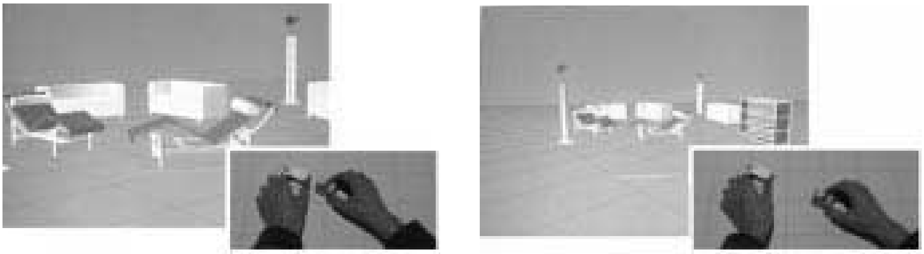


**Fig. 7.** Camera handling: Zooming in (left) and out (right).

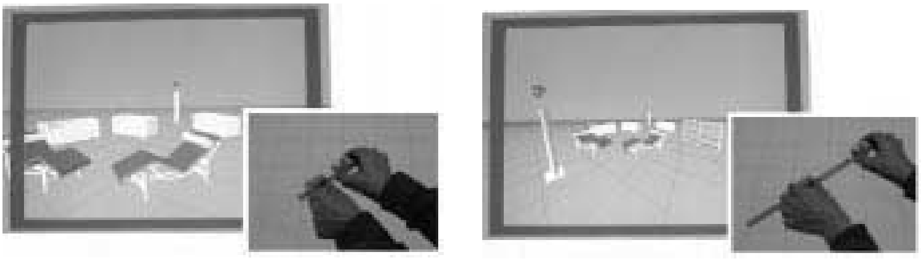


**Fig. 8.** Window handling: Zooming in (left) and out (right).

# 7 Object Height Control

The need to set object height, or more generally, to handle a ground level where objects sit, is our concern for the rest of this paper. The instrument was developed from a virtual tool, used for scrolling the side view and for setting object height, to a set of real tools working on a grid-based floor, before integrating all these features into one single virtual tool.

## 7.1 A Virtual Prototype for Height Manipulation

Side view control was first achieved by copying a section of the side view, called *height slice*, onto the table (Fig. 9). Scrolling the side view is performed by putting a brick at the upper (scroll up) or lower (scroll down) edge of the *height slice*. When the brick is away from these edges, objects can be selected, and moved up and down.

**Fig. 9.** Scrolling the side view by putting a brick at the height slice corner; moving an object in the height slice; perceiving the moved object in the side view (left to right).

## 7.2 Using Real Tools to Control the Third Dimension

Based on real tools (Fig. 10), we explored three ways to *physically* handle object height. We tried to get closer to the principle of *coinciding action and perception spaces* [18]. *Digit*, a digital controller, works on selected objects and sets their height according to up-down buttons. *Tower* offers the same buttons but is combined with a luminous scale showing current height. The up-down buttons and the visual feedback are organized along the height axis, so action and perception are partly coincident. *Slider* is a vertical sliding-rule, where height is handled *and* indicated by an up-down handle. With *Slider*, handling and height cues are both organized along the height axis; thus action and perception are fully coincident. The actual height handling in the virtual setting is carried out by a virtual floor, which is called *Floor* and described below. The problematic aspect of this solution was that these tools require serial interfaces (RS 232, bi-directional, 9600 Baud) with the BUILD-IT system. Also, they rely on battery driven remote control. Hence, we looked for a software solution.

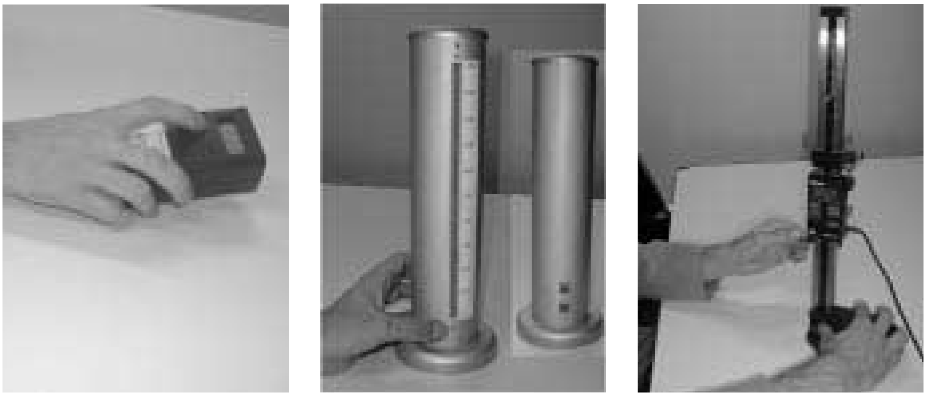


**Fig. 10.** Real height tools: *Digit,* a box with up-down buttons; *Tower,* a luminous scale with up-down buttons; *Slider,* a sliding-ruler with up-down handle and digital display (left to right).

## 7.3 Back to the Virtual: a Concept for Floor Handling

The resulting solution for placement and manipulation of objects among multiple stories is a fully virtual solution. It reuses the *height slice* from the first solution and *Floor* from the second solution. Hence, it integrates the knowledge acquired in the previous steps.

*Floor* is handled in the *height slice*, located along one edge of the plan view. The objects selected are all assembled on one storey. This assembly of selected objects then moves vertically as a whole when *Floor* is moved upwards or downwards (Figs. 11-12). De-selected objects are unloaded at the desired storey. Only objects at or above *Floor* are visible. To support multi-storey planning, we also offer a virtual ceiling, which is called *Ceiling*. *Ceiling* is also handled in the *height slice* and only objects below *Ceiling* are visible. Hence, by using *Floor* and *Ceiling*, objects within one storey can be *focused* and handled in the plan view. In the case of the side view, all stories are visible. *Ceiling* does not affect the side view.

A requirement for *Floor* and *Ceiling* to be developed was that the system should offer what we will call *dynamic control of clipping planes*. This technology is employed in the following way: *Floor* and *Ceiling* are each connected with a clipping plane so that only *focused* objects are visible.

When *Camera,* or *Window,* is selected, it also follows *Floor*, updating the side view content (Fig. 12).

In some cases work with a complete building where roof and walls are part of the CAD model. To see different levels, an empty *Floor* can be raised to traverse different stories of the building (Fig. 13). When *Floor* is raised *Ceiling* automatically follows.

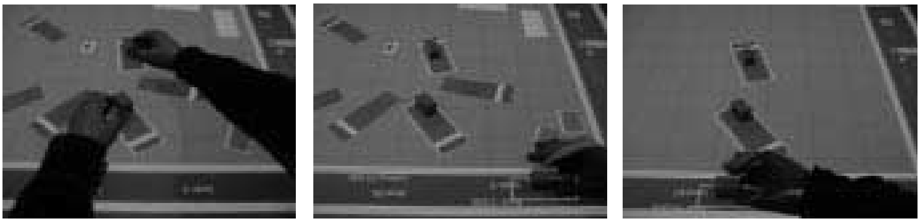


**Fig. 11.** Floor and object handling in the plan view: Normal object handling; selecting *floor*; raising *floor* with selected objects (left to right).

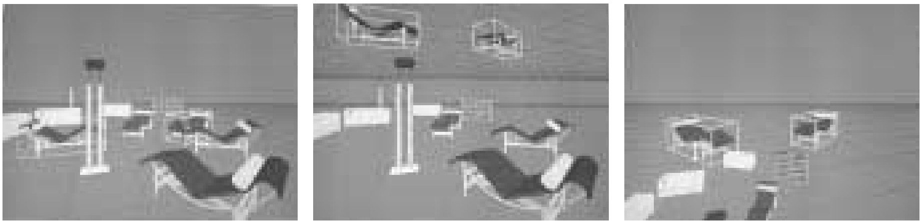


**Fig. 12.** Floor handling as seen in the side view: Normal object handling; raising *floor* with selected objects; raising *floor* with objects *and* Camera selected (left to right).

**Fig. 13.** Floor handling as an aid for inspection: Selecting *floor*; raising *floor* to roof-level; raising *floor* through roof-level (left to right).

## 8 Discussion and Perspectives

The navigation methods (Section 6) will be followed up by an evaluation of the different design strategies. First, *inspection tasks*, conducive to exploration, will be developed. With these tasks, users are offered *one* (plan *or* side view), *two* (plan *and* side view), or *alternative* (more than two) control methods. For each set-up, mean task completion time will give *quantitative data*. For *subjective evaluation*, participants will be asked to rate their preference of each method after the experiments. We conjecture that methods based on *scene* handling are better than methods based on *ob-*

*server* handling. Second, *composition task* experiments, following the same set-up, are planned.

Rotation with one brick requires an oriented brick form. GroundCatcher or FrameCatcher using two bricks do not rely on brick orientation. So it may be of interest to fit *form* to *operation*, by employing rectangular and circular bricks.

*Zoom* may be controlled by *one* or *both* bricks, raising the topic of *asymmetry* [11]. The same applies to the functions *tilt* and *roll*. Sets of these functions, and their relation to one- or two-handed interaction, will be explored in future research. The concept of time- and space-multiplexed input [7] may prove fruitful in this regard.

The need to control object height (Section 7) was first met with a virtual method, and then triggered the realisation of real tools, before it was answered with virtual floor handling. This *cycle* may indicate a future design strategy. To evaluate the outcome of that *cycle*, usability studies must be carried out. Also, a seamless integration of Floor with the spatial navigation methods (Section 6) is required.

The structure of *scene* and *observer* handling (Section 6.1) may actually be valid for virtual floor handling as well. In the plan view, Floor and clipping-plane are connected, so scene *and* observer are handled together (Figs. 11, 13). In the side view (Fig. 12), *scene* is always handled, *observer* is only handled when Camera, or Window, is selected.

In this paper, the names of certain tools were somewhat metaphorical, such as *window*, *sliding-ruler* and *floor*. It may be of interest to know more about the consequences of using metaphors, for user behaviour *and* for our design activity.

Guided by the suggested *design guidelines* (Section 2), the tools and methods presented are subject to usability studies. At the same time, the value of the indicated guidelines may turn out clearer.

## References

1. Ackermann, P.: Developing Object-Oriented Multimedia Software. dpunkt Verlag für digitale Technologie, Heidelberg (1996)
2. Balakrishnan, R., Kurtenbach, G.: Exploring bimanual camera control and object manipulation in 3D graphics interfaces. Proceedings of the CHI'99, ACM, NY (1999) 56-63
3. Bateson, G.: Steps to an ecology of Mind. Ballantine Books, NY (1972)
4. Bichsel, M.: Personal communications (1997)
5. Campbell, J.: Training design for performance improvement. In Campbell, J, Campbell, R. (eds.): Productivity in organisations. San Francisco, Jossey-Bass. (1988) 177-216
6. Fitzmaurice, G., Ishii, H., Buxton, W.: Bricks: Lay-ing the Foundations for Graspable User Inter-faces. Proceedings of the CHI'95. ACM, NY (1995) 442-449
7. Fitzmaurice, G. W.: Graspable User Interfaces. PhD at the University of Toronto, Toronto (1996) Ch. 7
8. Fjeld, M., Bichsel, M., Rauterberg, M.: BUILD-IT: An Intuitive Design Tool Based on Direct Object Manipulation. In: Wachsmut, I., Frölich, M. (eds.): Gesture and Sign Language in Human-Computer Interaction. Lecture Notes in Artificial Intelligence, Vol. 1371. Springer-Verlag, Berlin Heidelberg NY (1998) 297-308

9. Fjeld, M., Jourdan, F., Bichsel, M., Rauterberg, M.: BUILD-IT: an intuitive simulation tool for multi-expert layout processes. In Engeli, M., Hrdliczka, V. (eds.): Fortschritte in der Simulationstechnik (ASIM). vdf Hochschuleverlag AG, Zurich (1998) 411-418
10. Fjeld, M., Lauche, K., Dierssen, S., Bichsel, M., Rauterberg, M.: BUILD-IT: A Brick-based integral Solution Supporting Multidisciplinary Design Tasks. In Sutcliffe, A., Ziegler, J., Johnson, P. (eds.): Designing Effective and Usable Multimedia Systems (IFIP 13.2). Kluwer Academic Publishers, Boston (1998) 131-142
11. Guiard, Y.: Asymmetric division of labour in human skilled bimanual action: The kinematic chain as a model. Journal of Motor Behaviour, Vol. 19. (1987) 486-517
12. Ishii, H., Ullmer, B. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. Proceedings of the CHI'97. ACM, NY (1997)
13. Kaptelinin, V.: Computer-Mediated Activity: Functional Organs in Social and Developmental Contexts. In B. Nardi (ed.) Context and Consciousness, MIT Press, MA (1996) 50
14. Kirsh, D., Maglio, P. On Distinguishing Epistemic from Pragmatic Action. Cognitive Science, 18, (1994) 513-549
15. Lauche, K.: Personal communications (1998)
16. Mackay, W.E., Pagani, D.S., Faber, L., Inwood, B., Launiainen, P., Brenta, L., Pouzol, V.: Ariel: Augmenting Paper Engineering Drawings. Proceedings of the CHI'95, video program, ACM, NY (1995)
17. Mackenzie, C., Iberall, T.: The grasping hand. Elsevier. (1994) 15
18. Rauterberg, M., Mauch, T., Stebler, R.: What is a promising candidate for the next generation of interface technology. Proceedings of the 5th International Conference INTERFACE to Real and Virtual Worlds. EC2 & Cie., Montpellier (1996) 95-103
19. Rauterberg, M., Bichsel, M., Meier, M., Fjeld, M.: A gesture based interaction technique for a planning tool for construction and design. Proceedings of the RO-MAN'97. IEEE. (1997) 212-217
20. Rauterberg, M., Fjeld, M., Krueger, H., Bichsel, M., Leonhardt U., Meier M.: BUILD-IT: A Planning Tool for Construction and Design. Proceedings of the CHI'98, video program, ACM, NY (1998)
21. Rauterberg, M.: Personal communications (1999)
22. Underkoffler, J., Ishii, H.: Illuminating Light: An Optical Design Tool with Physical Interface. Proceedings of the CHI'98, ACM, NY (1998)
23. Voorhorst, F.: Personal communications (1999)
24. Wellner, P., Mackay, W., Gold, R.: Computer Augmented Environments: Back to the Real World. In Communications of the ACM, July, Vol. 36, No. 7, (1993) 24-26
25. Winograd, T., Flores, C. F.: Understanding computers and cognition: a new foundation for design. Ablex publishing corporation. (1986) 97

## Acknowledgements

We thank Peter Troxler (peter@klapt.ch) for his photos (Figs. 1-3, 8) and Scott Stapleford (Philosophy, www.uwo.ca) for his editorial assistance. BUILD-IT is funded by The Swiss Federal Commission for Technology and Innovation (KTI), The Center for Integrated Production Systems (ZIP) at ETH Zurich, and The Institute for Design and Construction Methods (IKB) at ETH Zurich. Morten Fjeld thanks the Research Council of Norway (www.forskningsradet.no) for his Ph.D. fellowship. BUILD-IT is a registered trademark of TellWare (info@tellware.com).