M. Fjeld, N. Ironmonger, S. Guttormsen Schär & H. Krueger (in press): Design and Evaluation of Four AR Navigation Tools Using Scene and Viewpoint Handling. In proceedings of INTERACT 2001, Tokyo, Japan, July 9-13, 2001. 10 pages.

Design and Evaluation of Four AR Navigation Tools Using Scene and Viewpoint Handling

M. Fjeld[1], N. Ironmonger[2], S. Guttormsen Schär[1] & H. Krueger[1]

[1]IHA, Swiss Federal Institute of Technology, Clausiusstr. 25, CH-8092 Zurich

www.fjeld.ch, morten@fjeld.ch, [2]ndi@apte.net

Abstract: In an Augmented Reality (AR) system using a brick-based tangible user interface, we present and evaluate alternative techniques for scene navigation. Going from two-dimensional brick-based input to three-dimensional navigation presents design issues. There are two fundamental methods for scene control: scene handling or viewpoint handling. The system has two views (plan and side), presenting action-perception spaces which are coincident and separate. Four tools were developed to explore design solutions, testing the alternative methods in each view. In a quantitative user experiment with a search-and-position task, we evaluated the four tools, measuring performance by trial completion time. Results showed that scene and viewpoint handling performed equally well in the plan view. In the side view, scene handling performed better. Subjective ranking showed that scene handling was always preferred to viewpoint handling. Results indicate that when action-perception spaces are coincident, the choice of handling method is less critical than when separate.

Keywords: augmented reality, tangible user interface, bricks, 3D navigation, usability evaluation

1 Introduction

The context of our research is Augmented Reality (AR), which aims to bring interaction with virtual environments out into the physical world. One domain in AR research is Head Mounted Displays (HMD), another is Tangible User Interfaces (TUI), which we study here. In TUIs, physical objects are used as handles to represent and interact with models in a virtual scene. Previous studies have investigated the use of bricks as input medium for TUIs. Handling of models using brick-based TUIs for three-dimensional (3D) graphics has been explored for simple tasks. The integration of input and display devices, being termed *action* and *perception spaces*, is a major concern in the design of TUIs. Such spaces may be coincident or separate, and this is a design issue.

Putting a TUI into effective practice presents new challenges. In real-world applications, the size of the virtual environment often exceeds the physical interface and hence users need means to navigate the scene. Systems with two-dimensional input operating on virtual environments require a mapping between the 2D control surface and the 3D scene. Control of the positioning of a virtual scene may employ two alternative fundamental methods, these being *scene handling* (SH) and *viewpoint handling* (VH). To investigate these issues, we extended an existing brick-based TUI to perform navigation in a 3D virtual scene.

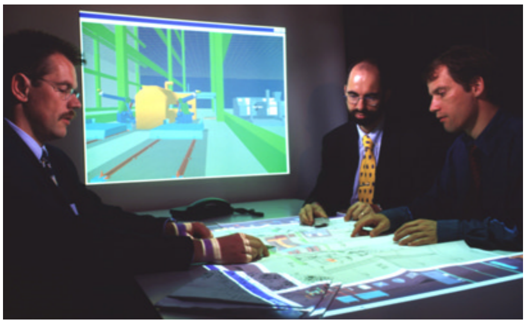

Figure 1: The TUI system: Plan and side view.

The system we worked with has two views, called plan and side view, presenting action-perception spaces which are coincident and separate (Fig. 1). Four new tools were developed in order to explore design solutions, testing the alternative methods in each view. In extending the TUI, the development of the tools was based on two principles for interaction design, being:

- Bimanual interaction (Fitzmaurice et al., 1997)
- Pragmatic and epistemic action (Kirsh et al., 1994)

In a quantitative usability evaluation with a search-and-position task, the four tools were evaluated using protocol data and subjective rankings. We defined performance as trial completion time. Our main interest was to find which handling method will give better performance and which method will be preferred by users. We studied each view separately, but did not compare the views. To examine the handling methods in more detail, we defined and measured user actions in terms of bimanual interaction and epistemic action.

2 Background

2.1 Augmented Reality (AR)

Augmented Reality was introduced by Wellner (1993). The goal of AR, as described by Mackay (1995), 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" (like Fig. 2). In the AR research of Mackay, computer information is projected onto drawings so that users can interact with both the projected information and the paper drawing (Mackay et al., 1995). An early brick-based AR system was described by Fitzmaurice et al. (1995). A more recent example showing how AR can support urban planning is given by Arias et al. (2000). A spatially continuous workspace called Augmented Surfaces (Rekimoto et al., 1999) and BUILD-IT (Rauterberg et al., 1997) are two other AR examples.

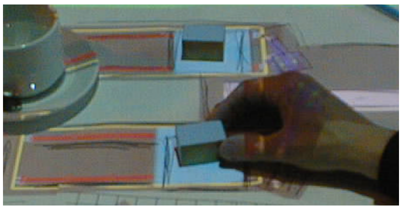

Figure 2: Interacting in an AR system using physical objects.

2.2 Tangible User Interfaces

Tangible User Interfaces (TUI) have been studied by various interaction researchers as a new kind of input medium. An extensive survey of TUIs was given by Ullmer et al. (2000). Tethered bricks, requiring wires, were investigated in the Active Desk by Fitzmaurice et al. (1995). Wireless bricks, detected by a tablet, were later developed by Fitzmaurice et al. (1997). In a more recent use of tangible bricks, the surface is detected from above (Rauterberg et al., 1996; Ullmer et al., 1997). It was shown that a brick-based interface was significantly more effective than a mouse-keyboard-screen interface or a touch-screen (Rauterberg et al., 1996).

2.3 Bimanual Interaction

Two-handed, termed bimanual, interaction calls upon everyday coordination skills such as aligning and grouping (Fitzmaurice et al., 1997). Bimanual brick-based interaction in two dimensions has already been investigated (Fitzmaurice et al., 1995; Ullmer et al., 1997). Bimanual viewpoint control and model handling in 3D graphics interfaces have been studied using two mice, a keyboard, and a screen (Balakrishnan et al., 1999b). The relation between two-handed movements and input performance in separate action and perception spaces has been examined (Balakrishnan et al., 1999a). Here, we make use of bimanual brick-based interaction for 3D scene navigation, and in this way, our work is novel compared to the state-of-the-art.

2.4 Epistemic Action

Depending on users' acquaintance with virtual environments, navigation can range from *epistemic action* to *pragmatic action*. Epistemic (exploratory) action is performed to unveil hidden information or to gain insight that would otherwise require considerable reflection, pragmatic action directly leads to goal attainment (Kirsh et al., 1994). Due to ease of use and direct interaction, Tangible User Interfaces (TUI) may encourage more use of epistemic action (Fitzmaurice et al., 1997) than traditional ones.

2.5 The BUILD-IT System

The TUI we use is based on computer vision technology and is the interface for a multi-user planning tool called BUILD-IT (Fig. 1). Following the tradition of AR, projected light replaces the use of screens as the output medium. Grouped around a table and employing tangible physical bricks, users can select and manipulate virtual models within the scene which they are planning (Fig. 1). The system enables its users to cooperate in a virtual environment for planning a real-world project, such as a room, a school, or a factory. In the BUILD-IT system, the position and orientation of each brick on the table top is determined by a computer vision system. Multiple bricks allow for uses such as groupware and bimanual interaction. Here, bimanual interaction is studied for single users.

Users have at all times two up-to-date views of the scene they are creating and manipulating: the plan view and the side view (Fig. 1). The plan view is the bird's eye view from above – which is projected onto the table. The side view is projected onto a screen near the table. In the case of the side view, a virtual camera, which can be located in the virtual scene either outside or inside the plan view, allows the users to choose from which position the side view is to be captured. It can also be zoomed. In the case of the plan view, the entire projection of the scene can be moved around, rotated, or zoomed. The interaction surface also contains a virtual storage space – or menu – for models not in immediate use (Fig. 3). It allows users to create new model instances. Models returned to the menu are deleted.

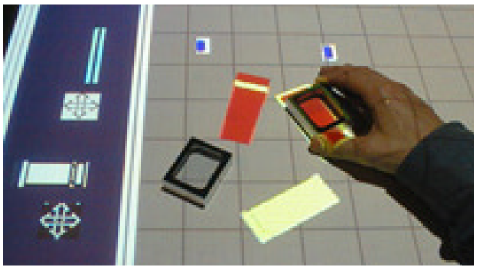

Figure 3: The system in use: menu with navigation tools (left) and model handling (right).

2.6 TUIs and the Need for Navigation

Several TUIs employ front-projected tables while others employ back-projected workbenches, both giving a plan view (Ullmer et al., 2000). Similar to the BUILD-IT system, some of these offer a second display, giving a perspective side view. TUIs have been applied to a range of application domains, such as visualisation, simulation, modelling, collaborative work, and education. Features such as orientation, scaling, and navigation were offered only by a few of the systems, requiring specialised physical handles. Given the wide use and diverse domains of applications for TUIs, further exploration of navigation tools is worthwhile to study.

Navigation of 3D scenes in TUIs requires viewpoint control (pan, rotation, zoom). However, planar input using physical handles provides only position and rotation. The chosen tool designs have to bridge this disparity between the two-dimensional control surface and the three-dimensional virtual environment.

3 Design of Navigation Tools

To explore alternative handling methods, a series of design choices had to be taken. In 3.1, we first introduce the design options. In 3.2, we explain our initial decisions which led to the experimental factors. In 3.3, we assign update mechanisms and finally, in 3.4, we show how the tools were implemented and are used.

3.1 Design Options

The design options were as follows:

- **Action and perception spaces** may be *coincident* or *separate* according to input and display devices.

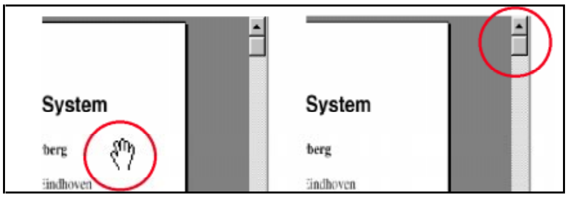

Figure 4: 'Hand' directly handles *scene* positioning (l.), 'scrollbar' handles *viewpoint* positioning (r.).

- **Handling methods** concern the relationship between user input actions and the resulting effect on the displayed image. One may chose between handling the placement of the scene itself, termed *scene handling* (SH), and handling the point of view, termed *viewpoint handling* (VH). Equivalent handling can be seen in two-dimensional browsers like Acrobat Reader 3.0 (Fig. 4).
- **Update mechanisms** concern the update of the display of the scene. They may be *continuous* or *discrete*. Continuous update responds directly to user's handling actions. Discrete update is triggered by the user after handling and then updates the scene.
- **Degrees of control** may be *shift*, *rotation*, *zoom*, *tilt*, and/or *roll*, as with a camera.
- **Physical handles** concerns the input devices in a TUI, and how they represent and affect virtual models and tools. They may have a *generic* or *specialised* function.
- **Number of physical handles available** may be *one*, *two* or *many*.

3.2 Deciding Experimental Factors

In the context of the system we used, we next settled each of the design options. Two of the options became experimental factors; one was set accordingly; the reminder were fixed.

- **Action and perception spaces:** For plan view interaction, action-perception spaces are coincident. For side view interaction, these spaces are separate. We decided to investigate both views, making this issue an experimental factor.
- **Handling methods:** Within each view of the system, we were interested in alternative handling methods. This issue is the second experimental factor.
- **Update mechanisms:** The update mechanism for each handling method and view was chosen based on

logistical requirements of the implementation. Hence, update mechanism did not act as an independent experimental factor. We elaborate on this in 3.3.

The remaining three design options were fixed for the whole experiment as follows:

• **Degrees of control:** First, a pilot-study where users could control all five degrees of control (shift, rotation, zoom, tilt, roll) of the side view showed that they felt uncomfortable. Some reported that the horizon of the side view was too unstable. Second, the plan view is a horizontal work-bench, implying shift, rotation, and zoom as the only feasible degrees of control. Third, we tried to make control of each view as similar as possible. Altogether, this gave us grounded reasons for excluding tilt & roll and to offer pan, rotation & zoom in both views.

• **Physical handles:** The use of specialised *physical handles* for navigation of TUIs has been studied in different frameworks. For instance, in a TUI called metaDESK (Ullmer et al., 1997), a physical input device was specially designed for combined zoom and rotation of the plan view. This device showed the limits in the use of specialised input devices, and indicated that the use of generic input devices, such as bricks, might be suited for navigation in TUIs. For the BUILD-IT system, various height tools were evaluated and generic ones preferred (Fjeld et al., in press). Based on these observations and aiming to keep hardware complexity low and software flexibility high, we chose to use *generic handles*, these being rectangular bricks.

• **Number of physical handles available:** With generic bricks and with the need to control shift, rotation, and zoom, each view required two bricks. Since navigation of each view are orthogonal functions and may be performed at the same time, at least four bricks must be available at a time.

3.3 Assigning Update Mechanisms

In cross-combining the two factors, being handling method and view, we got four tools. For each tool, the update mechanism used was decided as follows:

• **Scene handling of the plan view**: Update had to be continuous, since discrete update would have given insufficient feedback on user actions and led to a breakdown of the coincident action-perception space.

• **Viewpoint handling of the plan view**: Since the viewpoint is a virtual camera not visible in the scene, it needs to be represented. The representation had to be a model which could be shifted, rotated and scaled. Such model handling could not be solved with a simple scroll bar and needed to operate within the plan view. In this case, continuous update is not possible, hence discrete update was used, employing a reference frame model.

• **Scene handling of the side view**: Since we tried to make handling methods as similar as possible in both views, we chose to perform side view scene handling using continuous scene update.

• **Viewpoint handling of the side view**: To make handling methods as similar as possible in both views, viewpoint handling of the side view was chosen to use discrete scene update, also employing a reference frame.

3.4 Tool Implementation and Use

In the following, we give details on the use of the four tools (Table 1). Further details on design and use were given in a video (Fjeld et al., 2000).

	Scene Handling	Viewpoint Handling
Plan view control	***GroundCatcher* (2)**	***FrameCatcher* (2)**
Side view control	***Camera*** ***zoom***	***ViewFrame*** ***zoom***

Table 1: SH and VH for each view, showing bimanual use.

The tools can be activated in the menu (Fig. 3). When employing one brick, called the *first brick*, shift and rotation of the controlled view can be set. Adding a second brick, called the *zoom brick*, activates zooming as well. The use of two bricks is called bimanual navigation. For plan view navigation, these two bricks have equal functionality, for the side view they have different functionality. For each tool we explain how to perform bimanual navigation.

• **Scene handling of the plan view: GroundCatcher**

A *GroundCatcher* (Fig. 5) is selected from the menu and as soon as it is placed, it locks to the scene. Subsequent handling controls the plan view with continuous update. A second *GroundCatcher* is likewise selected and locks to another part of the scene. De-selecting the bricks quits the tool.

Figure 5: *GroundCatcher*; here zooming and rotating the plan view: zooming in (l.) and zooming out (r.).

• Viewpoint handling of the plan view: FrameCatcher

When a *FrameCatcher* (Fig. 6) is selected from the menu, the scene automatically zooms out to show a wider context, and a frame appears. As soon as the *FrameCatcher* is placed within the frame, it locks to the frame. Subsequent handling controls the frame, selecting the desired part of the scene. A second *FrameCatcher* is likewise selected and locks to another part of the frame. De-selecting the bricks triggers a discrete scene update of the framed region.

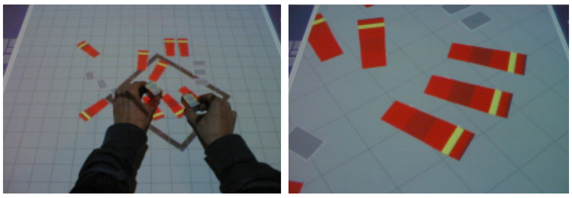

Figure 6: *FrameCatcher*; here zooming and rotating the plan view: frame control (l.) and updated scene (r.).

• Scene handling of the side view: Camera

By selecting the *Camera* (Fig. 7) from the menu, the part of the scene shown in the side view can be continuously set. Zoom is selected with a second (here: right hand) brick. By moving the *zoom brick* and the *Camera* further apart, the side view can be enlarged; by moving them closer, the side view can be focused. De-selecting the *zoom brick* freezes the zoom. If a second *Camera* is selected from the menu, the first one disappears.

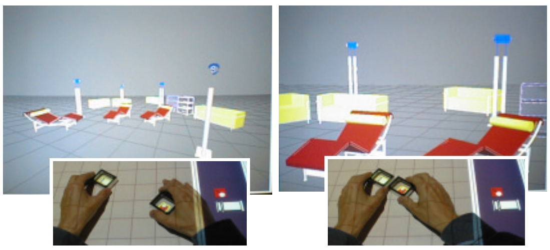

Figure 7: *Camera*; here zooming the side view: zooming out (l.) and zooming in (r.).

• Viewpoint handling of the side view: ViewFrame

When the *ViewFrame* (Fig. 8) is selected from the menu, the side view automatically zooms out to show a wider context. The desired part of the scene can be selected. Zoom is selected with the second (here: right hand) brick, thus resizing the window as with the *Camera*. De-selecting the *zoom brick* freezes the zoom. De-selecting the *first brick* triggers a discrete scene update of the framed region. If a second *ViewFrame* is selected from the menu, the first one disappears.

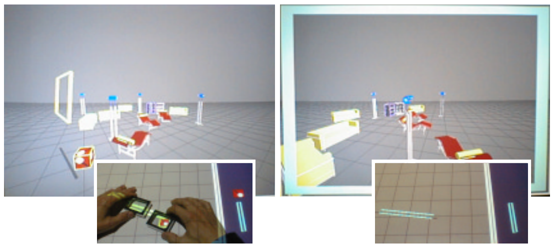

Figure 8: *ViewFrame*; here zooming and rotating the side view: frame control (l.) and updated scene (r.).

4 Hypotheses

A pilot-study indicated that the strengths of *scene handling* (SH) using continuous update are direct feedback, lower number of menu selections, and intuitive use. The strengths of *viewpoint handling* (VH) using discrete update are better tolerance for graphics latency and improved overview for users.

The hypotheses were stated in the conventional null form, as appropriate for statistical testing (Table 2).

H1. No difference in performance between SH and VH
H2. No difference in bimanual interaction between SH and VH
H3. No difference in epistemic action between SH and VH
H4. No difference in subjective preference between SH and VH

Table 2: Null hypotheses H1– H4.

H1: Due to the established benefits of direct feedback, we conjectured that SH would perform better than VH.
H2: According to the reported benefits of bimanual interaction (Fitzmaurice et al., 1997), we conjectured more bimanual use for the expected higher performing method, this being SH.
H3: According to results concerning epistemic action (Kirsh et al., 1994), we conjectured more use of epistemic action for the expected higher performing method, this being SH.
H4: Considering user preferences, we drew upon the indication that epistemic action may provide an

"enhanced sense of engagement of the 3D scene" (Balakrishnan et al., 1999b). Additionally, SH appears to be closer than VH to users' natural expectations for performing search-and-position problems in the real world. Hence, we conjectured that user preferences would favour SH to VH.

5 Usability Evaluation

5.1 Participants and Apparatus

Sixteen graduate students, four women and twelve men, aged 24 to 35, volunteered to participate, being paid a small fee. They had no former experience with BUILD-IT and had to acquire a significant number of skills within the experiment.

The BUILD-IT software runs on a standard 400 MHz PC with 128 MB RAM, 10GB disk space, a frame grabber card, and two OpenGL graphic cards. The computer reads table images with a video camera and provides output via a pair of standard projectors, all sitting in a rack together with an IR source and a mirror. There were five rectangular bricks with reflective material, all identical. All experiments took place in a room without daylight and dimmed ceiling lighting.

5.2 Task Scene and Task

We chose a 3D search-and-position task, which is one of the simplest tasks typically used in studying human performance in computer input control. Other typical tasks, like path following and pursuit tracking, are more difficult (Balakrishnan et al., 1999b). The task was to search for models in a maze and to position each of them at their correct place, requiring the use of both views and bimanual navigation.

The task scene consisted of a maze and an elevated cubic box with four faces and face replicas on the ground (Fig. 9). Each face was split into nine cells, one of which contained a stimulus, identified by colour (red, yellow, green) and form (disk, square, triangle). Due to shields, each stimulus was visible only when viewed from the front of the face. On the ground was a replica of each face for model placement. For each stimulus, a model with the same colour and form was hidden in the maze. Since a pilot study showed that four models were too time consuming, we simply required any two of the four models to be found and positioned. The task was defined as follows:

For any two of the four stimuli, not necessarily sequentially (Figs. 9a-d):

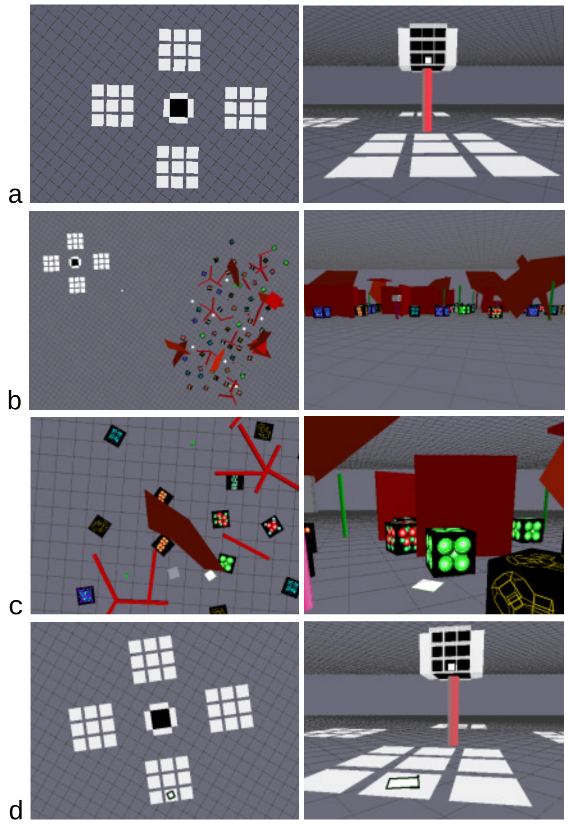

Figure 9: Four stages of a trial; left half is plan view, right half is side view: a) start-up situation with cubic box containing stimulus, b) searching for models in the maze, c) a model has been found, and d) the model is positioned at the correct cell of the face replica on the ground.

i) View colour and form of stimulus (a).
ii) Search for and retrieve matching model (b-c).
iii) Position model at the correct cell of the face replica on the ground (d).

We generated twelve variations of the task differing in colour and form of the stimuli and in position and orientation of the cubic box and models. Four of these were used as demo and practice tasks, eight were used as main tasks. The main tasks were permuted giving eight experimental trials per participant.

5.3 Operationalization

H1. Good *performance* was operationalized by low trial completion time, measured from stimulus appearance until the last model or tool was de-selected.

H2. To select *zoom brick* from the menu and thereby to start a bimanual interaction sequence, the system requires a *first brick* to be activated and stay selected. Hence, *bimanual interaction* in navigation was measured by the number of zoom selections from the menu.

H3. We assume that users performing pragmatic action will not stop for reflection in the middle of a movement. When they stop, we consider them to reflect and the action is classified as *epistemic*. This can be classified by the number of step-by-step brick movements and was measured when a brick was re-positioned after a pause, ignoring pauses shorter than half a second. (This is more than the system latency and the sampling interval for protocol logging.) The *first brick* and the *zoom brick* were registered separately.

H4. Preference was operationalized by asking participants to i) rate their satisfaction with each tool on a [very low, low, high, very high] scale indexed by grades [4,3,2,1] and ii) indicate one preferred tool per view and justify their preference.

H1 is one-sided using two categories. H2 and H3 are one-sided using ordinal data with several categories. H4 is one-sided using interval data.

5.4 Design

A two-by-two within-group design, enabling the investigation of subjective tool preference, was used. Pairing each plan view tool with each side view tool resulted in four experimental conditions (Table 3).

Experimental condition	**Plan view**		**Side view**	
	SH:*Ground Catcher*	VH:*Frame Catcher*	SH: *Camera*	VH:*View Frame*
1st	x		x	
2nd	x			x
3rd		x	x	
4th		x		x

Table 3: The two-by-two design gives four conditions.

Each participant was given a permuted sequence of the four conditions. To equalise potential learning effects, sequences were chosen so that always two new tools appeared in the first condition, one new tool appeared in the second and third condition, and no new tools in the fourth condition. This resulted in 16 of the 24 possible permutations, hence the number of participants. The presentation order of the main tasks was chosen by the latin square procedure so that task-condition combination was counterbalanced. The order of the demonstration and practice tasks did not vary.

5.5 Procedure

Each participant was welcomed by the investigator and performed the experiment seated next to him at the table.

Introduction: The investigator explained the system by loading a *furniture scene*. Hardware components were briefly pointed out: rack, mirror, camera, computer, two projectors, and five bricks. Operation was demonstrated in terms of plan view, side view, menu, model selection, positioning, rotation, and de-selection. Assisted practice with one, then five bricks, was given. Navigation was introduced and handling methods, as found in Acrobat Reader (Fig. 4), were explained. It was shown how SH and VH had been implemented for the plan and the side view (Table 2) and how the four tools combine into four conditions (Table 3). Taking the tools given by the first condition for the participant, navigation was demonstrated for each of the plan and side views, first with one, then with two bricks. *Introduction* lasted approx. 25 minutes.

Trials sets: The *task scene* was now loaded. There were four trial sets according to the sequence of tool pairing conditions given for the participant. For each trial set, the investigator explained and performed a demonstration task with one model (Fig. 9); the participant did an assisted practice task with two models; then performed two different main tasks as fast as possible. The investigator checked task completion and initiated the next task. The eight main tasks formed the experimental trials. *Trial sets* lasted approx. 45 minutes.

Subjective rating: At the end of the experiment, the participant rated their satisfaction with each tool, then selected the preferred tool per view and justified their choice. *Subjective rating* lasted approx. 10 minutes.

5.6 Logging

The software was instrumented to log brick movements in real time, giving the position and kind of virtual model the bricks operated on. A sampling rate of approx. 0.3 sec. was used. From the log of each trial we extracted i) trial completion time (tct), and for each tool ii) number of zoom selections (nzs), and iii) number of re-positionings of *first brick* (nfrp) and *zoom brick* (nzrp).

6 Experimental Results

Our aim was to examine within each of the views differences in performance and use between the alternative tools, these being based on SH and VH.

However, participants employed two complementary tools at a time, one for each view, giving the four experimental conditions (Table 3). In order to test the hypotheses H1 – H3 (Table 2), the analysis needed to reveal individual differences within each view between the alternative tools. Consequently, much like Balakrishnan et al. (1999b), we used a multiway ANOVA, here with a General Linear Model (GLM). The five independent variables were plan view method (SH, VH), side view method (SH, VH), trial, task, and user. The four dependent variables were trial completion time (H1), zoom selections (H2), *first brick* and *zoom brick* re-positionings (H3). For H2 and H3, we analyzed these separately for plan and side view navigation. The Bonferroni method was used with k = 7, alpha = 0.05/k = 0.007. Significant effects are shown by p-values less than alpha are marked by a star (*). Below, we give the ANOVA results for the strong cases (Tables 4 – 6) and for the significant effects note the supporting data averaged over trials.

6.1 Trial Completion Time (H1)

Indep. variable	df	F-ratio	p
Plan view method	1	0.391	p = 0.533
Side view method	1	8.144	p = 0.005 *
Trial	7	5.210	p < 0.001 *
Task	7	3.146	p = 0.005 *
User	15	2.063	p = 0.018

Table 4: Trial completion time: Significant effects for side view method, trial, and task.

Plan view method (Table 4): No significant effect and H1 is upheld.
Side view method (Table 4): SH (tct=150 s.) gave better performance than VH (tct=183 s.) and H1 is rejected.
Other effects (Table 4): Trial (learning effect) and task had a significant effect.

6.2 Bimanual Interaction (H2)

Indep. variable	df	F-ratio	p
Plan view method	1	11.885	p < 0.001 *
Side view method	1	0.053	p = 0.818
Trial	7	0.583	p = 0.768
Task	7	4.376	p < 0.001 *
User	15	1.715	p = 0.061

Table 5: Zoom selections in plan view navigation: Significant effects for plan view method and task.

Zoom selections in plan view navigation (Table 5): More zoom selections per trial with SH (nzs=2.4) than VH (nzs=1.5) and H1 is rejected. Task was significant.
Zoom selections in side view navigation: No significant effects.

6.3 Epistemic Action (H3)

Indep. variable	df	F-ratio	p
Plan view method	1	14.188	p < 0.001 *
Side view method	1	2.177	p = 0.143
Trial	7	3.119	p = 0.005 *
Task	7	3.365	p = 0.003 *
User	15	4.731	p < 0.001 *

Table 6: *First brick* re-positionings in plan view navigation: Significant effects for plan view method, trial, task, and user.

First brick re-positionings in plan view navigation (Table 6): Fewer *first brick* re-positionings per trial with SH (nfrp=14.9) than VH (nfrp=21.6) and H3 is rejected. Trial (less use), task, and user had a significant effect.
Zoom brick re-positionings in plan view navigation: More *zoom brick* re-positionings with SH (nzrp=13.0) than VH (nzrp=5.1) and H3 is rejected. Task was significant.
First brick re-positionings in side view navigation: Significant effects for trial (less use) only. H3 is upheld.
Zoom brick re-positionings in side view navigation: No significant effects and H3 is upheld.

6.4 Subjective Preference (H4)

Tool	Rating				Mean rating
	v.low, -2	low, -1	high, 1	v.high, 2	
GroundCatcher	0	1	8	7	1.31
FrameCatcher	1	4	8	3	0.50
Camera	0	1	5	10	1.50
ViewFrame	1	5	7	3	0.38

Table 7: Overall tool rating selections.

Plan view	*GroundCatcher*	12	*FrameCatcher*	4
Side view	*Camera*	14	*ViewFrame*	2

Table 8: Tool preference per view.

Table 7 gives subjective ratings; showing for each tool the number of participants selecting each rating. The indices are shifted to a balanced scale [-2,-1,1,2] and the mean rating is given. We note that the mean rating of SH tools was higher than for VH tools. Table 8 gives for each view the number of participants who preferred each tool. In both cases, SH tools were preferred to VH tools for both views and H4 is rejected.

6.5 Further Subjective Statements

Participants were also asked to justify their choice and to comment on overall system usability. We classified their statements as positive [+], negative [–], or neutral [±], giving (in brackets) the number of participants making the same statement if more than one:

- **GroundCatcher**
  - + : feedback (7), ease of use (4), combined shift and zoom, engagement of 3D scene, underpins search
  - – : 3D graphics slow, zoom requires learning
- **FrameCatcher**
  - + : provides overview, gives a steady image
  - – : plan view orientation
- **Camera**
  - + : feedback (10), plan view orientation (5), ease of use (2)
  - – : separate action and perception spaces, zoom not needed
- **ViewFrame**
  - + : side view orientation (2), provides overview
  - – : plan view orientation (2)
- **Overall system usability**
  - + : engagement of the 3D scene, interesting tools
  - – : unwanted de-selection through shadowing of bricks (4), operation tiresome, 3D graphics is slow, active state of plan view tool unclear, no undo
  - ± : needs practice (2), frame with SH may be of interest

7 Discussion

We discuss the major results according to the views.

7.1 Plan View

Scene handling (SH) and *viewpoint handling* (VH) performed equally well for plan view navigation – being against our expectations. So the strengths of direct feedback in SH did not appear as performance benefits. However, SH was preferred to VH in subjective ratings and obtained more positive user statements, particularly related to *feedback*. There is more *bimanual interaction* in SH than VH. *Epistemic action* shows opposite effects for *first brick* and *zoom brick*. For the *first brick*, less *epistemic action* in later trials indicates a learning effect.

7.2 Side View

SH outperforms VH for side view navigation – confirming our expectations. Also users rated SH higher than VH, mainly due to better SH feedback. *Bimanual interaction* and *epistemic action* gave equal results for SH and VH and do not explain the difference in performance. This leaves the subjective factor of direct feedback in SH as a likely cause. Again, for the *first brick*, less *epistemic action* in later trials indicates a learning effect.

8 Conclusion

We presented the design of four tools for navigation in an Augmented Reality system; two of them based on *scene handling* (SH), the other two based on *viewpoint handling* (VH). The two system views, called plan and side view, present action-perception spaces which are coincident and separate. One tool of each handling method was used to control each view.

A usability evaluation of the tools was undertaken with 16 users, recording trial completion time, user actions and preferences. We tested hypotheses regarding performance, bimanual interaction, epistemic action, and subjective preference. Expressed in terms of handling methods, and generalising from the views to action and perception spaces, our three main findings for this system were:

1) When action and perception spaces coincide, SH and VH perform equally well.
2) When action and perception spaces are separate, SH performs better than VH.
3) Users prefer SH in both cases.

The results indicate that when action-perception spaces are coincident, the choice of handling method is less critical than when they are separate. For future research, the design choice of update mechanism and its influence on task solving efficiency is worthy of further evaluation. Due to space, we do not report on details related to symmetry in bimanual interaction. We leave it for others to determine how far our findings apply to other Tangible User Interfaces.

Acknowledgements

We thank Scott MacKenzie, George Fitzmaurice and Terry Winograd for commenting on an earlier version of this paper. We also thank the INTERACT reviewers for constructive feedback. Valuable help came from Domenico Signorello (task design), Martin Bichsel (development), Fred Voorhorst (design), Christoph Schierz (experimental design), Georg Kralidis (statistical analysis), and Matthias Rauterberg (interaction design and PhD-mentor). BUILD-IT is a trademark of TellWare GmbH.

References

Arias, E., Eden, H., Fischer, G., Gorman, A. & Scharff, E. (2000). Transcending the Individual Human Mind – Creating Shared Understanding through Collaborative Design. *TOCHI*, **7**(1), pp. 84-113, ACM.

Balakrishnan, R. & Hinckley, K. (1999a). The Role of Kinesthetic Reference Frames in Two-Handed Input Performance. *Proc. UIST'99*, pp. 171-178, ACM.

Balakrishnan, R. & Kurtenbach, G. (1999b) Exploring Bimanual Camera Control and Object Manipulation in 3D Graphics Interfaces. *Proc. CHI'99*, pp. 56-63, ACM.

Fitzmaurice, G., Ishii, H. & Buxton, W. (1995). Bricks: Laying the Foundations for Graspable User Inter-faces. *Proc. CHI'95*, pp. 442-449, ACM.

Fitzmaurice, G. & Buxton, W. (1997). An Empirical Evaluation of Graspable User Interfaces: towards specialized, space-multiplexed input. *Proc. CHI'97*, pp. 43-50, ACM.

Fjeld, M., Voorhorst, F., Bichsel, M., Krueger H. & Rauterberg, M. (2000). Navigation Methods for an Augmented Reality System. *Video program CHI 2000*, pp. 8-9, ACM.

Fjeld M., Lauche K., Bichsel M., Voorhorst F., Krueger H., & Rauterberg, M. (in press). Physical and Virtual Tools: Activity Theory Applied to the Design of Groupware. *CSCW*, Kluwer. See also at: www.fjeld.ch/cscw

Kirsh, D. & Maglio, P. (1994). On Distinguishing Epistemic from Pragmatic Action. *Cognitive Science*, 18, pp. 513-549.

Mackay, W.E., Pagani, D.S., Faber, L., Inwood, B., Launiainen, P., Brenta, L. & Pouzol, V. (1995). Ariel: Augmenting Paper Engineering Drawings. *Video program CHI'95*, pp. 421-422, ACM.

Rauterberg, M., Mauch, T. & Stebler, R. (1996). What is a promising candidate for the next generation of interface technology? *Proc. 5th INTERFACE to Real and Virtual Worlds*, pp. 95-103, EC2 & Cie.

Rauterberg, M., Bichsel, M., Leonhardt, U. & Meier, M. (1997). BUILD-IT: a computer vision-based interaction technique of a planning tool for construction and design. *Proc. INTERACT'97*, pp. 587-588.

Rekimoto, J. & Saitoh, M. (1999). Augmented Surfaces: A Spatially Continuous Workspace for Hybrid Computing Environments. *Proc. CHI'99*, pp. 378-385, ACM.

Ullmer, B. & Ishii, H. (1997). The metaDESK: Models and Prototypes for Tangible User Interfaces. *Proc. UIST'97*, pp. 223-232, ACM.

Ullmer, B. & Ishii, H. (2000). Emerging frameworks for tangible user interfaces. In IBM Systems Journal, 39(3 & 4), pp. 915-931, IBM.

Wellner, P., Mackay, W. & Gold, R. (1993). Computer Augmented Environments: Back to the Real World. *Comm. of ACM*, 36(7), pp. 24-26, ACM.