The Bent Pick Ray: An Extended Pointing Technique for Multi-User Interaction

Kai Riege*
Fraunhofer Institute for Media Communication

Gerold Wesche†
Fraunhofer Institute for Media Communication

Thorsten Holtkämper‡
Fraunhofer Institute for Media Communication

Bernd Fröhlich§
Bauhaus University Weimar

Abstract

This paper presents a collaborative pointing technique for co-located multi-user interaction in projection-based virtual environments. Our approach uses bent pick rays to allow users to collaboratively work together without locking objects. Moreover, a user can manipulate distant objects in immediate reach, using a Scaled-Grab technique. The main purpose of the bent pick ray is to provide continuous visual user feedback, keeping a user informed about the collaborative manipulation.

CR Categories: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Interaction styles; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual Reality

Keywords: pointing technique, 3D interaction, multi-user input, manipulation, collaboration, 3D user interfaces, virtual environments

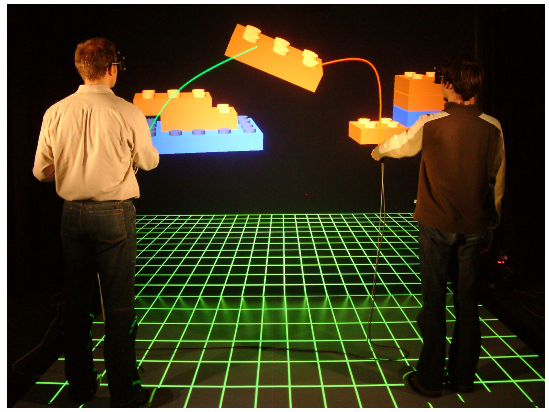

Figure 1: Two users collaboratively dragging an object.

1 Introduction

Projection-based virtual environments commonly support perspectively correct viewing for one user at most. Only very few systems have been developed to support multiple tracked users using a shared display, among those the Two-User Responsive Workbench™ [1], which supports two individually tracked users, and the recently developed multi-viewer system at the Bauhaus University Weimar [5], which supports up to four tracked users. In these co-located environments multiple users share the same manipulation space, which facilitates natural interaction. Applications in these environments require collaborative techniques for all kinds of 3D interaction tasks, such as selection, navigation, manipulation, and system control.

In this paper we concentrate on 3D co-located collaborative selection and manipulation techniques based on the ray casting metaphor. In most collaborative applications, the selection of the same object by multiple users is forbidden. After the first user has selected an object, it is locked for all other users while the first user is interacting. However, in the real world people are able to touch an object currently held by another person and help that person perform a certain task. Based on this metaphor, our Bent Pick Ray technique preserves the visual feedback during multi-user interaction and allows collaborative manipulation (see figure 1). The possibility of multiple selections and concurrent manipulation requires merging the inputs from multiple users. We present two approaches to solve this problem, which avoid locking selected objects during the interaction. The techniques are described for two collaborating users, but they can be easily extended to support multiple users.

We implemented and tested our techniques in our projection-based two user display system, which displays a dedicated stereoscopic image for each of the two users. Since both users perceive the virtual objects exactly at the same spatial location, the real as well as the virtual space are shared. Tasks like pointing and manipulating are often performed simultaneously and both users are fully aware of each other's actions. First experiences with this system show that our collaborative techniques are easy to use and appreciated by our users.

2 The TwoView display system

Our two user system, called The TwoView display[3], consists of two Barco Galaxy™ 6000 DLP projectors, which each project a stereoscopic image for one user. Circular polarization filters are used to separate the users' views; stereo shutter glasses are used for eye separation. To combine both techniques we use BARCO's modified StereoGraphics ChrystalEyes™ shutter glasses, which have circular polarization filters laminated on the top of the LCD shutter. The effective light absorption of such glasses is less than a simple combination of a circular polarization filter and a LCD shutter since the linear polarizing property of the LCD shutters is used as one half of the circular polarization filter. We use a back projection screen at a size of 3×2.40 meters which maintains polarization. The 3D images are generated on a two-headed PC with a video refresh rate of 108 Hz and with 1280x1024 resolution and are synchronously output by the DLP projectors.

A simple alternative to multi-user virtual environments are so called multi-viewpoint images [12]. In a multi-viewpoint image different image elements are projected from multiple viewpoints and

*e-mail: kai.riege@imk.fraunhofer.de
†e-mail: gerold.wesche@imk.fraunhofer.de
‡e-mail: thorsten.holtkaemper@imk.fraunhofer.de
§e-mail: bernd.froehlich@medien.uni-weimar.de

they are combined in a single stereoscopic image. This technique is used to project interaction elements for individual users such that from a user's point of view these elements are in the correct position and depth. The tracked positions of interaction devices match the virtual position of visual and functional interaction elements such as pointers, menus or intersection rays. Though multi-viewpoint images enable multiple users to interact on the same virtual scene, it does not overcome the incorrect spatial perception for the main scene. In addition the interaction elements viewed from the perspective of another user are also displayed incorrectly.

3 Description of the technique

The Bent Pick Ray is essentially a method for providing a useful visual feedback in situations in which objects are manipulated simultaneously by two or more users. The technique is derived from single-user environments: in most cases there is a visible connection from the input device to the selected object *and* this connection is a straight extension of the picking direction; in our case a colored straight line is used. As soon as the second user is grabbing an object already picked by the first user, the pick rays of the two users are deformed so that they still emanate from the input device, tangentially to the pointing direction, and touch the selected object at the point of intersection; in other words they have to be bent. In case we would have chosen straight lines, the pick ray would not correspond to the pointing direction. In the worst case, the pick ray would even emanate from the back of the input device. We consider mediating the correct direction of the hand held input device as an important feedback information since this reflects a natural extension of the hand. A flexible pointer for two-handed selection was proposed by Olwal and Feiner [7]. Compared to our technique, they use it for pointing at occluded objects in a single user application.

The order of actions during a collaborative picking/manipulation task is as follows:

1. The first user selects an object, and drags it using a modified scaled-grab technique (see 3.1).
2. The second user selects the same objects by pointing at it. Both pick rays are straight lines at this moment.
3. The object is moved simultaneously by both users, using techniques for merging the input from both users' hands. We derive a single object transformation from both hand motions as described in 3.2. At the same time, both pick rays are bent such that, firstly, they still emanate from the input devices, and secondly, they are connected to the dragged object at their respective hit positions (see 3.3).
4. One user releases the object. The corresponding pick ray is immediately reset to a straight line. However, the pick ray of the other user does not change, it is still bent.
5. The other user drags the selected object, whereas the pick ray gradually becomes straight again (see 3.4). This technique is used to avoid a sudden transition from a bent ray into a straight line and allows a continuous manipulation of the object .

3.1 A Scaled-Grab Technique

There are different techniques to dynamically adjust the control/display ratio (as defined in [2]), which use scaling to optimize interaction in large display environments. The Go-Go interaction technique [8] scales the movements of a user's hand in order to modify the position of a virtual hand. The Scaled-World Grab technique [6] scales down the whole environment around a user's position to perform direct interaction at a closer distance. Frees and Kessler [4] presented the so-called PRISM technique for adaptively under-scaling the interaction to optimize the precision of manipulation. The Scaled-Grab, introduced by Simon et al. [11], applies the dynamically scaled hand movement to the selected object in all translation dimensions. This way a user is able to select objects at arbitrary distance and rapidly move them to any position in the virtual environment with just one arm movement. Since the scaling is applied to all translation directions every movement that deviates from the pick ray direction breaks the connection of the pick ray to the object. To circumvent this Simon et al. turn off the pick ray during interaction.

Our approach modifies the Scaled-Grab technique in such a way that we can keep the pick ray visible during the interaction, maintaining a consistent visual feedback. This is achieved by only scaling that portion of the movement which points into the direction of the pick ray. Thus the scaling is only performed along the pick ray.

As shown in figure 2 we project the movement of the input device between successive frames onto the current pointing direction.

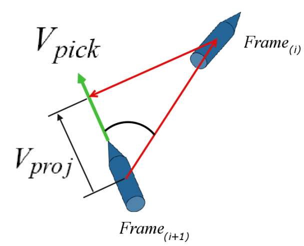

Figure 2: Projection of the device movement onto pick ray direction.

The ratio of the object distance D_{obj} to the user's arm length L_{arm} results in the the scale factor s as described in equation 1. The new distance of the object on the pick ray is defined by the length of V_{new}, calculated in equation 2.

$$s = \begin{cases} \frac{D_{obj}}{L_{arm}} \geq 1 : & \frac{D_{obj}}{L_{arm}} \\ \frac{D_{obj}}{L_{arm}} < 1 : & 1 \end{cases} \quad (1)$$

$$\mathbf{V_{new}} = s \cdot \mathbf{V_{proj}} + \mathbf{V_{pick}} \quad (2)$$

3.2 Merging Multiple Inputs

The transformation of the input device is used to drag the selected object. Collaboratively manipulating the same object results in more than one matrix describing the motion of each input device. In case of the TwoView display system we have to merge two transformations, one for each user.

Assuming that both users are equated our first approach averages the inputs of the two users as shown in figure 3. We simply decompose the two intented target transformations into rotations and translations, according to [10], representing rotations as quaternions. The rotations are interpolated using the slerp algorithm [9] with a fixed weight, while the translations are interpolated using a

```
for (each input) {
  calculateTargetMatrix();
  decomposeTargetMatrix();
  trans_result += translation;
  slerp(1./numOfInputs, rot_result, rotation);
}
trans_result *= (1./numOfInputs);
```

Figure 3: Interpolation with fixed weight.

weighted sum. The weight is the inverse of the number of currently concurrent input devices (in our case 0.5). Note that this method merges absolute states of transformations.

Tests of the first approach showed that the activity of both users is unequal most of the time. Therefore, our second approach weights the influence of interaction according to the amount of hand movement a user does with the input device. For example, if two users are connected to an object and one is currently not manipulating, the influence of this user is reduced. In order to assign a weight to a user's input, first the two object transformations are examined independently (as for a single-user interaction). A larger amount of hand movement, corresponding to a larger object drag, results in a bigger weight. The object drag matrices are compared to the initial state in which the collaborative dragging has started to calculate relative transformation offsets. The interpolation of the translation component is applied to these offsets; the rotations are weighted according to the first approach. Note that the interpolation is calculated using only the offset transformations, so the object is moved incrementally.

```
offsets[], weights[];
for (each input) {
  target = calculateTargetMatrix();
  offsets.add(calculateDeltaMatrix(target,
      reference));
}
weights = calculateTranslationWeights(offsets);
for (each offset){
  decomposeOffsetMatrix(offset);
  sumUpTranslation(weight);
  slerp(1./numOfInputs, rot_result, rotations);
}
```

Figure 4: Interpolation with weighted offsets.

The calculation of the weights based on the lengths of the translation offsets L_1 and L_2 is calculated by equation 3.

$$\omega_1 = \begin{cases} L_1 > L_2: & \frac{1}{4}\cdot\left(\frac{L_1-L_2}{L_2}+\frac{1}{2}\right) \\ L_1 < L_2: & \frac{1}{4}\cdot\left(\frac{L_2-L_1}{L_1}+\frac{1}{2}\right) \end{cases} \quad (3)$$

$$\omega_2 = 1-\omega_1$$

With this method the maximum weight is reached if one user's offset is four times greater than the other user's offset.

3.3 Bending the pick ray

Ideally interaction techniques should provide continuous feedback to the user during the whole interaction cycle. Since the pick ray is well-known and an established visual feedback for single-user interactions, the idea is to use it for multi-user interactions as well. If more than one user is connected to the object, the pick ray gets bent in order to merge the multiple inputs. So users are continuously informed about their connection to the object and they also get direct feedback from the input merging process (see section 3.2). The pick ray bending is calculated according to figure 5.

We require that the unbent pick ray is tangential to the circular arc, which describes the bending and lies within the plane spanned by the object direction V_1 and the pick ray direction V_2. We calculate A_1 representing the radius r and the center M according to equation 4 and 5.

$$\mathbf{A_1} = \frac{\mathbf{V_2}\cdot\cos\alpha\cdot|\mathbf{V_1}|}{|\mathbf{V_2}|} - \mathbf{V_1} \quad (4)$$

$$\mathbf{M} = \mathbf{P_0} - \frac{|\mathbf{V_1}|}{2\cdot\cos\left(\frac{\pi}{2}-\alpha\right)}\cdot\frac{\mathbf{A_1}}{|\mathbf{A_1}|} \quad (5)$$

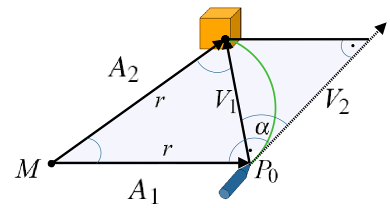

Figure 5: Calculate the bending of the pick ray.

Now we can draw the circular arc by using piece-wise linear combinations of the spanning vectors A_1 and A_2 extended to the length r.

Figure 1 shows two users interacting with one object. Caused by different pointing directions you can clearly see that the merged inputs result in bent pick rays.

3.4 Unbending the pick ray

If the pick ray is still bent while all but one user released the object we slowly transition back into the normal single-user manipulation mode, i.e. into a straight pick ray. It is quite clear that straightening the pick ray, keeping the object selected, would cause an unintended movement of the object. Deselecting the object automatically seems not useful, either. Therefore, we keep the pick ray connected to the object, but gradually straighten the ray every time the movement of the user's hand decreases the angle to the object, whereas the object's position is unchanged. Hand movements not decreasing that angle drag the selected object as in single-user manipulation. This way the pick ray gets unbent in a continuous and transparent way, which intuitively resolves the issues of the feedback of multi-user interaction, shown in figure 6 and 7.

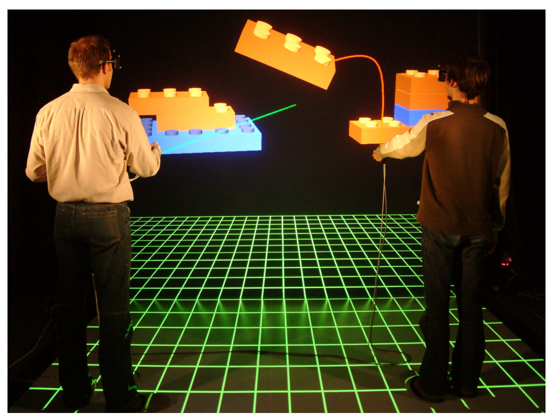

Figure 6: One user releases object, remaining pick ray still bent.

4 Conclusions and Future Work

We proposed the Bent Pick Ray technique, which supports picking and dragging for multi-user environments in a transparent way. It involves no overhead actions, such as mode switches and avoids imposing restrictions on users, e.g. object locking. We achieve a fluent transition from single-user to collaborative two-user dragging

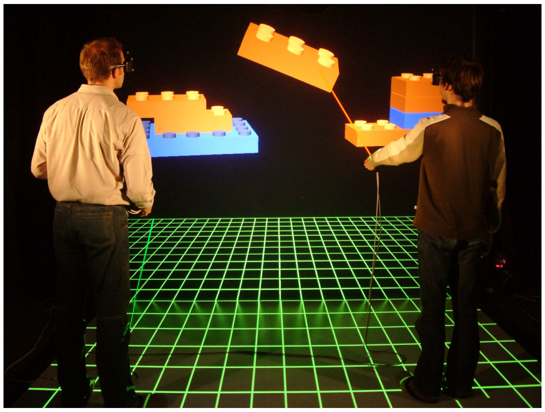

Figure 7: While moving towards the hit point the remaining pick ray gets unbent.

and back. A permanent visual feedback for the effects of simultaneously dragging an object selected by several users is provided. This, additionally, enables users to align their input devices in such a way that the bending of the pick ray nearly disappears. This allows a user either to follow or to guide the interaction. We decided against using a physically-based simulation to avoid the associated computational overhead. Moreover, a straight forward implementation of physically correct behavior would result in unwanted effects when switching back into single user interaction, since one of the forces imposed on the object would suddenly be lost.

Although we did not yet intensely evaluate the Bent Pick Ray technique, we can report about first user experiences. Most notably, users liked the transparent way of connecting and disconnecting to an object, independent of whether it had already been selected by the collaborating user or not, due to the fact that explicit mode changes are not necessary. Additionally they liked the very distinct feedback, generated by the bending of the ray, visualizing two users controlling the movement of an object. The gradual unbending of the pick ray happening after the collaborating user had released the object was surprising at first but did not seem to be disturbing after some time of use. The mode of collaborative dragging was considered as questionable by some users. The averaging drag mode was intuitively usable and understandable for most users, whereas the maximum mode caused some confusion: users reported that they felt a loss of control while dragging a selected object this way. It was not always clear to them how to move the input device to achieve a desired effect. Currently we are working on the evaluation of the introduced techniques. We plan to perform a more detailed user study that compares the performance of two users in a collaborative 6 DOF docking task.

Although the bent pick ray was originally designed just to circumvent the need to lock objects, it introduces new possibilities for true collaborative tools, two of them presented in this paper. Our next steps will be investigating these potentials with regard to new collaborative tools. Based on our first experiences and the results of our upcoming user study, we will further refine our collaborative dragging and manipulation methods. In addition, we will also explore the potential for collaborative tools and techniques for other interaction tasks than the ones presented here.

References

[1] Maneesh Agrawala, Andrew C. Beers, Ian McDowall, Bernd Fröhlich, Mark Bolas, and Pat Hanrahan. The two-user responsive workbench: support for collaboration through individual views of a shared space. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 327–332, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[2] James D. Foley, Victor L. Wallace, and Peggy Chan. The human factors of computer graphics interaction techniques. *IEEE Comput. Graph. Appl.*, 4(11):13–48, 1984.

[3] Fraunhofer Institute for Media Communication (IMK). The TwoView Display System. *http://www.imk.fraunhofer.de/en/twoview*, Last Access: 30/11/2005.

[4] Scott Frees and G. Drew Kessler. Precise and rapid interaction through scaled manipulation in immersive virtual environments. In *IEEE Virtual Reality 2005*, pages 99–106, Bonn, Germany, 2005.

[5] Bernd Fröhlich, Jan Hochstrate, Jörg Hoffmann, Karsten Klüger, Roland Blach, Matthias Bues, and Oliver Stefani. Implementing multi-viewer stereo displays. In *WCSG (Full Papers) 2005*, pages 139–146, Plzen, Czech Republic, 2005.

[6] Mark R. Mine, Jr. Frederick P. Brooks, and Carlo H. Sequin. Moving objects in space: exploiting proprioception in virtual-environment interaction. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 19–26, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[7] Alex Olwal and Steven Feiner. The flexible pointer - an interaction technique for selection in augmented and virtual reality. In *Conference Supplement of UIST '03 - ACM Symp. on User Interface Software and Technology*, pages 81–82, Vancouver, BC, 2003.

[8] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. The go-go interaction technique: non-linear mapping for direct manipulation in vr. In *UIST '96: Proceedings of the 9th annual ACM symposium on User interface software and technology*, pages 79–80, New York, NY, USA, 1996. ACM Press.

[9] Ken Shoemake. Animating rotation with quaternion curves. In *SIGGRAPH '85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254, New York, NY, USA, 1985. ACM Press.

[10] Ken Shoemake and Tom Duff. Matrix animation and polar decomposition. In *Proc. of the Graphics Interface '92*, pages 258–264, Vancouver, Canada, 1992.

[11] Andreas Simon, Armin Dressler, Hans-Peter Krüger, Sascha Scholz, and Jürgen Wind. Interaction and co-located collaboration in large projection-based virtual environments. In *IFIP Interact*, pages 364–376, Rome, Italy, 2005.

[12] Andreas Simon and Sascha Scholz. Multi-viewpoint images for multi-user interaction. In *VR '05: Proceedings of IEEE Virtual Reality 2005*, pages 107–113, Bonn, Germany, 2005.