

Efficient Bimanual Symmetric 3D Manipulation for Bare-Handed Interaction

Markus Schlattmann and Reinhard Klein*

*Institute of Computer Science II - Computer Graphics
University of Bonn

Roemerstrasse 164, 53117 Bonn, Germany

phone, email: {markus | rk}@cs.uni-bonn.de

www: <http://cg.cs.uni-bonn.de/hand-tracking>

Abstract

Recently, stable markerless 6 DOF video based hand-tracking devices became available. These devices simultaneously track the positions and orientations of both user hands in different postures with at least 25 frames per second. Such hand-tracking allows for using the human hands as natural input devices. However, the absence of physical buttons for performing click actions and state changes poses severe challenges in designing an efficient and easy to use 3D interface on top of such a device. In particular, for coupling and decoupling a virtual object's movements to the user's hand (i.e. grabbing and releasing) a solution has to be found. In this paper, we introduce a novel technique for efficient two-handed grabbing and releasing objects and intuitively manipulating them in the virtual space. This technique is integrated in a novel 3D interface for virtual manipulations. A user experiment shows the superior applicability of this new technique. Last but not least, we describe how this technique can be exploited in practice to improve interaction by integrating it with RTT DeltaGen, a profes-

sional CAD/CAS visualization and editing tool.

Keywords: Virtual reality, interaction techniques, bimanual interaction

1 Introduction

For interaction with a virtual environment, handtracking is one of the favorite approaches, because it directly exploits the ease and perfection with which humans employ their hands in everyday life. In order to support immersive user experience, markerless real-time hand-tracking without the need of special initialization procedures gained a lot of interest in recent years. Presently, methods fulfilling these properties are capable of tracking up to 6 continuous degrees of freedom (DOF) of both hand poses (global positions and orientations) simultaneously and recognizing several stiff postures for each hand.

As suggested in [BKLP05] we strictly distinguish between the notions pose, posture and gesture. The pose is meant to be the combination of a rigid body's 3D position and orientation (e.g. the hand's global pose). A hand posture is defined as a specified configuration of finger limb positions and orientations relative to the hand pose (e.g. the fist posture). A hand gesture (or free-hand gesture) describes a predefined movement of the hand pose(s) (e.g. writing a letter in the air).

Given the pose(s) of one or both hands simultaneously, the movements of a virtual object can be coupled to the movements of the user's hand(s). While the pose of one single hand is in principle enough for sufficiently controlling an object, the use of both

Digital Peer Publishing Licence

Any party may pass on this Work by electronic means and make it available for download under the terms and conditions of the current version of the Digital Peer Publishing Licence (DPPL). The text of the licence may be accessed and retrieved via Internet at <http://www.dipp.nrw.de/>.

First presented at the Virtual Reality International Conference 2009, extended and revised for JVRB

hands can significantly improve certain manipulation tasks (according to [OKF⁺05]). For example the symmetric bimanual technique named grab-and-twirl proposed by Cutler et al. [CFH97] (a virtual object is moved as if gripped between the hands) gives more control over an object; the center of rotation can intuitively be chosen and a considerably higher rotational accuracy is achieved.

In this paper we focus on interaction techniques for two-handed manipulation of an object's pose as an extension for a one-handed interface. We assume fundamental mechanisms such as selection of objects and movement of the virtual camera as well as single-handed interaction to be already solved and available. Nevertheless, two major problems have to be solved in order to build a 3D interface on top of the basic hand-tracking technology.

First, working with both hands simultaneously induces additional strain to the user, so permanently working with two hands is not desired and therefore an interface should additionally enable easily transiting between single and two-handed interaction.

Second, an interface for two-handed interaction has to provide mechanisms for grabbing and releasing the virtual object, which are inherent tasks during 3D interaction sessions due to the following reasons: The virtual world is typically significantly larger than the working volume of the user (the 3D region the hand is tracked in). Therefore, to enable users to move a virtual object to every position in the virtual space it has to be possible to grab it and release it in order to move it step by step. Scaling the working volume to the whole virtual world is not an option, because the accuracy would decrease too heavily. Similar, the range of angular movements of the human wrist is limited. In order to fully rotate and inspect an object from all viewing directions grabbing and releasing are indispensable. In interfaces that employ a standard 2D mouse, the problem of grabbing and releasing is solved either by lifting the mouse (while it is lifted a mouse movement induces no object movement) or by exploiting button states (usually the object is only moved if a button is held down). But in contrast to a standard 2D mouse no direct equivalent exists for markerless hand-tracking. Simple solutions for the realization of a grab and release cycle in the absence of physical buttons are disposing one degree of freedom (DOF) of one of the hand poses or applying different postures for different button states. Unfortunately, these approaches suffer from severe drawbacks.

Exploiting a DOF of one hand pose reduces the available degrees of freedom (e.g. if z-coordinate value is used for triggering grabbing). The use of different postures, one for grabbing and another for releasing, is significantly more demanding for the user than simply pressing a mouse button, because the physical effort as well as the coordination complexity of changing postures is considerably higher, especially if various specified postures are concurrently needed. Moreover, a posture change always induces an unintended pose change mainly in rotation in current markerless hand-tracking systems. This is due to the problem that the tracking state is temporarily undefined during a posture change. Therefore, it would nearly be impossible to instantly stop an object's movement by switching to another posture.

An easy to use solution of these problems is crucial for usability and efficiency of the interface. Such approaches have to be adapted to the users' capabilities and incapacabilities, the applications' specifications as well as the requirements and drawbacks of the used hand-tracking method. This diversity of demands poses several challenges in the design of easy to use interfaces.

The main contributions of this paper are summarized in the following. We introduce a novel technique for symmetric object manipulations, which allows for efficient and intuitive two-handed grabbing, moving and releasing of objects without the need of posture changes. This technique is well suited as an extension for one-handed object manipulation in order to enable a superior object control in complex and high precision tasks. We show that performing transitions between single and our two-handed interaction can be established easily and effectively. We further introduce several simple solutions for compensating some general drawbacks that appear if vision based markerless twohand-tracking is employed. Then, a short evaluation and discussion of this technique is as well presented, which is based on a user study, user questioning and our observations. Last but not least, we present our work about using bare-hand-tracking for controlling the commercial CAD application RTT DeltaGen and present a suitable interaction metaphor including our new bimanual manipulation technique.

2 Related Work

The large amount of literature on interaction techniques makes it practically impossible to give a full

review of the previously reported methods here; elaborate analysis can be found in [BKLP05], or [JS07] for multi modal interaction. We will only discuss the most related methods that are designed for or can be applied to 3D interaction interfaces based on markerless handtracking as an input device.

2.1 Object Grabbing and Releasing Techniques

According to Zachmann [Zac00] grabbing an object (i.e. attaching the object to the hand(s)) can be realized in (at least) three different ways: single-step, two-step or naturally. Single-step grabbing attaches the object at a certain event (e.g. a spoken command like “grab thing”). Two-step grabbing can be further divided into the following interaction steps:

- Some event (e.g. a posture or spoken command) switches the grabbing mode on; only in this mode, objects can be grabbed.
- The object is attached to the hand(s) at another event.

To release the object usually the same event as in the first step is used. In the grabbing mode natural grabbing is typically realized by conditioning collisions of virtual hand representations with the object (e.g. two virtual hand models must collide with the object). The object’s movements will be coupled directly to the hands’, when the object is touched this way. Note that if the virtual scene is rendered into the hands’ working area/volume the virtual hand representations are normally not visualized, because the real hands can represent themselves. Otherwise, the virtual representations of the user’s hands have to be rendered. However, in both cases precise two-handed manipulation of a small object is nearly impossible, because the distance between the hands has to be very small for grabbing and holding the object.

The easiest solution for triggering an attach or release action is extending the hand-tracking interaction interface with additional physical buttons as for example floor pedals. However, this kind of interaction turned out to be awkward and slow (according to [GFGB04]).

Another approach is using a dwell time threshold for triggering a grab or release event as for example used in [WP03] and [GFGB04]. Although this is simple, it introduces a constant lag in the interaction.

A further approach is to use speech to signal grabbing or releasing [Bol80]. But this is especially excessive if several grab and release cycles are needed.

Therefore, most approaches adopt grabbing postures to determine whether an object is attached to the hand or not (e.g. [MF04] or [BI05]). Unfortunately, it turned out to be quite difficult to release an object at a precise position [Osa06]. The reasons for this are: first, it is demanding for people to fix their hands precisely in midair without having physical support. Second, judging the release point without tactile feedback can be difficult. Third, the finger movements of a grabbing action often induce involuntary changes of the hand’s global pose. To solve the third problem of using grabbing postures Osawa [Osa06] proposed an approach to automatically adjust the release pose of a virtual object based on the relative speed of the two grabbing fingers (usually the thumb and one forefinger).

Furthermore, in current markerless hand-tracking systems most of the different postures humans exploit for grabbing (e.g. the 3-point pinch grab exploiting the thumb and two other fingers or the power grab exploiting the whole hand [Zac00]) are not available; typically only one grabbing posture is supported per hand. Additionally, in order to ensure a stable tracking, this posture must be formed very exactly and clearly. We observed this to be cumbersome for most users.

2.2 Object Manipulation Techniques

In a hand-tracking based 3D interface commonly virtual representations of the user’s hands are shown in the 3D scene. When one or both (depends on the employed technique) virtual representations intersect with an object, the object can be grabbed. Once grabbed, the movements of the virtual hand(s) are directly applied to the object in order to move, rotate or deform it. This is called the virtual hand metaphor, which is the most common direct manipulation technique for manipulating objects [1]between the physical world (hand or device) and the virtual representations work well, this interaction technique turns out to be very intuitive, since it is similar to every-day manipulation of objects. The main drawback of the virtual hand metaphor is the scaling problem; the limited workspace of the user’s limbs or the input device, which makes distant objects unreachable.

To solve this problem various approaches were reported. For example the Go-Go technique [PBWI96]

simulates an interactively non-linear growing of the user's arm(s). When a user's hand is close to her/him, the mapping of real hand pose to virtual object pose is one to one. As she/he extends her/his hand and arm beyond a certain range, the mapping becomes nonlinear and the virtual arm "grows". Thus, she/he is able to reach objects out of her/his range. Several other solutions to the scaling problem were introduced as for instance the World-in-Miniature technique [SCP95], HOMER (hand-centered object manipulation extending ray-casting) [BH97], Scaled-World Grab [MBS97] or Voodoo Dolls [PSP99]. However, none of these techniques can be identified as the "best" solution; their performance depends on the task and environment.

Two-handed 3D interaction techniques are commonly distinguished (according to [Gui87]) into bimanual asymmetric, where the hands perform different actions (e.g. the non-dominant hand holds something while the other manipulates it), or bimanual symmetric, where the hands perform identical actions (e.g. pulling a rope, typing on the keyboard). As in this paper a technique for bimanual symmetric interaction is introduced, we will not discuss asymmetric techniques here, an overview of asymmetric interaction can be found in [BKLP05]. In the following the symmetric 3D interaction techniques most relevant to our work are outlined.

Two-handed scaling of objects is a popular example for symmetric bimanual interaction. This can be solved as follows: the user picks up two sides of an object and can scale the object by moving her/his hands apart or together (e.g. [ZFS79, ML04]).

As well traveling in the virtual environment can be solved bimanually symmetric. In the Polyshop system [MM95] a user can travel by performing a rope gesture. By pulling on an invisible rope with both hands the user can pull her/him through the environment.

Using both hands for moving an object is most related to our work. In this context Cutler et al. [CFH97] introduced several bimanual symmetric techniques: the grab-and-twirl technique enables the user to pick up two sides of an object and then to carry and turn it around with both hands. By fixing one or more DOFs of the applied transformation several other techniques were derived such as the grab-and-carry technique (no roll around the line connecting the two hands is allowed) or the turntable technique (only turning around a fixed axis of rotation is allowed). These techniques turned out to be both intuitive and efficient. How-

ever, these techniques implicitly assume the employment of the virtual hand metaphor for determining the two pivot points (i.e. the two points where the object is grabbed), which still suffers from the scaling problem.

3 The Hand-Tracking Device

To markerless track the 6 continuous DOFs of the user's both global hand poses in several different stiff postures we implemented the method of Schlattmann et al. [SK07]. We decided to use this approach due to the real-time capability and automatic initialization. In contrast to [SK07], we connected all three cameras to one and the same computer (Intel E6600, Geforce 8800) where as well the hand-tracking is computed. The intersection of the viewing volumes of the three cameras defines the working volume of the handtracking. It describes the physical space the user's hand poses and postures are determined in (approximately 80cm 50cm 50cm). Our hardware prototype is shown in Fig. 1.

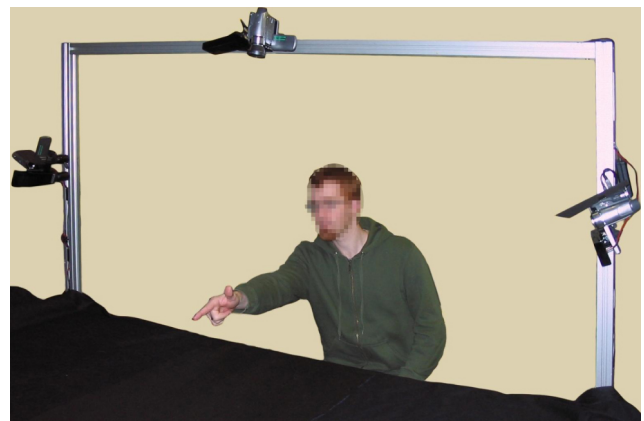


Figure 1: Our hand-tracking prototype.

In this method the camera images are first segmented and then the resulting masks are combined into a 3D voxel grid by computing the visual hull (see Fig. 2(Left)). This is performed on the GPU as described in [LMS03]. The angle between the viewing directions of two adjacent cameras is chosen to be 60 degrees. Having computed the visual hull as a coarse 3D reconstruction of the hand (see Fig. 2(Top right)), one or two fingertips are extracted (see Fig. 2(Bottom right)). Because the permitted stiff gestures are chosen appropriately (see Table 1), the fingertips to be identified always belong to protruding parts of the visual hull (the thumb and the index or middle finger). Hence, these

fingertips can be detected by first computing a small set of feature candidates (see Fig. 2(Middle right)), that are located on the convex hull of the visual hull, and then evaluating local volume properties of these candidates to distinguish the corresponding fingertips. Using further local volume properties the fingertips are classified and the posture (see Table 1) and pose (see Fig. 2 (Bottom right)) are determined. Note that due to the dependency of the pose on the fingertip positions movements of the stretched fingers slightly influence the derived hand pose. In our interface we restrict ourselves to the employment of the 'pointingA' posture (see Table 1) as it can be tracked with the highest accuracy.

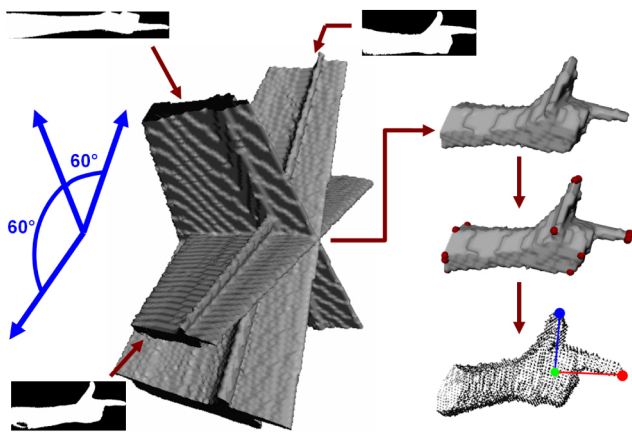


Figure 2: The major parts of the pose estimation algorithm. First, the visual hull of the hand is constructed from the three 2D hand silhouettes (From left to top right). Then, feature candidate points are detected (Middle right). Last but not least, these points are classified and the hand center is estimated for the determination of the final pose (Bottom right).

'palm'	'picking'	'pointingA'	'pointingB'

Table 1: The postures with “most protruding” fingertips.

For enabling the simultaneous tracking of two hands, first, the number of hands present in the working volume is detected based on the numbers of regions of interest (ROI) in the segmented camera images. If two hands are present, one consistent set of

ROIs is identified for each hand, respectively, and each hand is separately tracked as described above. Subsequently, the elbow joint position for each tracked hand is estimated. This way, it is determined which tracked pose/posture belongs to the left or right hand, respectively, exploiting the fact that the right elbow joint is usually located further right than the left elbow joint.

4 Bimanual Symmetric Grab

For our grabbing and releasing technique we assume an object has been selected and now the user's intention is manipulating the object's pose by exploiting both hands in order to gain a high precision and control. In this case an additional suitable mechanism is needed for determining when the objects movements shall be coupled to the hands' movements or not (grabbed or released). This mechanism should enable precise releasing of objects and should be manageable efficiently and intuitively. To this end, we introduce an approach based on different velocities of the grabbing/releasing action. The idea is to trigger a grabbing action if the hands move together while a release action is performed if the hands move apart as illustrated in Fig. 3.



Figure 3: Illustration of real two-handed grabbing, manipulating and releasing. The red arrows indicate either a grabbing gesture (Left) or a releasing gesture (Right).

Grabbing and releasing of an object can be expressed by a state machine comprising the two states 'Grabbed' and 'Released' and the transitions in between. A reasonable formulation of the exact conditions is crucial in order to avoid involuntary grabbing or releasing actions. The user's actions have to be suitably analyzed to faithfully distinguish grabbing, manipulation and releasing. To this end, the conditions are based on the grabbing velocity v_G^i and the manipulation velocity v_M^i . v_G^i denotes the signed velocity with which the hands moved together (positive) or apart (negative) between the previous frame $i - 1$ and

$$v_G^i := \frac{\|p_L^{i-1} - p_R^{i-1}\| - \|p_L^i - p_R^i\|}{t^i - t^{i-1}}, \quad (1)$$

where p_L^i, p_L^{i-1}, p_R^i and p_R^{i-1} are the tracked positions of the left and right hand in frame i and $i-1$, respectively. t^i and t^{i-1} denote the times of frame i and $i-1$. The manipulation velocity v_M^i denotes the sum of the hands' translational velocities that will potentially be used for object manipulation. In other words v_M^i is the sum of the translational velocities of both hands minus the grabbing velocity and is defined as:

$$v_M^i := \frac{\|p_L^i - p_L^{i-1}\|}{t^i - t^{i-1}} + \frac{\|p_R^i - p_R^{i-1}\|}{t^i - t^{i-1}} - |v_G^i|. \quad (2)$$

Now, in order to discover whether the user wants to perform a grab/release action or simply an object manipulation, the grabbing and manipulation velocities are analyzed. To this end, we introduce the modified signed grabbing velocity \tilde{v}_G^i which is defined as:

$$\tilde{v}_G^i := \begin{cases} \text{sgn}(v_G^i)(|v_G^i| - v_M^i) & , \text{if } |v_G^i| > v_M^i \\ 0 & , \text{else} \end{cases} \quad (3)$$

Thereby, sgn denotes the sign function. Note that \tilde{v}_G^i is zero while the manipulation velocity is either dominant or equal to the grabbing velocity. If the grabbing \tilde{v}_G^i velocity is dominant is either positive (the hands perform a grab movement) or negative (the hands perform a release movement). Now, our grabbing and releasing technique can be expressed by the state machine depicted in Fig. 4. T_G and T_R are thresholds for triggering a grab or release action, whereby T_G has to be a positive and T_R a negative real value. These parameters are used to adjust the velocity that has to be performed by the user to grab or release an object. In our current setting we use symmetric thresholds with $T_G = 10 \frac{cm}{s}$ and $T_R = 10 \frac{cm}{s}$. These values were determined in a short test scenario, where several users should perform this kind of grabbing and releasing gestures and manipulation movements while we recorded the employed velocities. We categorized the velocities manually to belong to the 'Grabbed' or 'Released' state. By analyzing several of these sequences we identified the threshold values, that would lead to the fewest errors. An error can either be a falsely triggered grab/release action or an intentional grab/release action that was not triggered.

Using this formulation the grabbing and releasing of a virtual object can be performed with only a slight

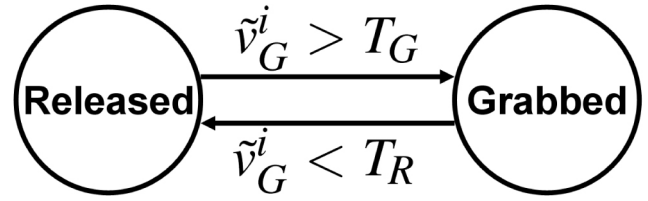


Figure 4: State machine illustrating how an object can be grabbed and released. Thresholds T_G and T_R are applied to the modified signed grabbing velocity \tilde{v}_G^i .

movement; no spacious and therefore uncomfortable gestures are needed. Moreover, employing the manipulation velocity to inhibit grab/release actions avoids involuntary grabbing/releasing during a manipulation task. For the same reason the velocity thresholds T_G and T_R can be chosen with such a low value which enables grabbing/releasing with slow movements. Note that these thresholds have to be greater than zero because otherwise unintentional grab/release actions could occur due to the natural tremor of human hands or slight inaccuracies of the hand-tracking device.

Furthermore, this formulation allows for grabbing and holding an object with arbitrary distance between the hands. This supports the adaptation to different demands of the current task, e.g. if a very precise rotation shall be performed a greater distance between the hands is superior while a smaller distance is more convenient for large translational manipulations due to the limited work space. Note that a simple solution as for example using a specified distance between the hands to trigger grabbing and releasing would not have these features.

Using this criterion for releasing operations enables fairly precise positioning of objects. However, humans in general do not perform release operations with perfect symmetric movements of both hands and only in the grabbing direction. Hence, sometimes slight unintentional movements occur while the user wants to release the object but the velocity threshold T_R is still not exceeded. To overcome this problem, we inhibit object movements while $\tilde{v}_G^i > 0$ causing the object to stand still while the grabbing velocity is dominant and the hands perform very slow releasing movements. This way, the object's movements stop immediately when the user starts to perform a release action. Note that this condition generally imposes no restriction on intended object manipulations, because in these cases the manipulation velocity is dominant and \tilde{v}_G^i is equal

to zero. Applying this simple modification to our twohanded grabbing/releasing technique the precision for positioning virtual objects can further be improved.

Additionally, our technique enables performing grab and release actions in a comfortable and effective way. Furthermore, due to the close relation to natural two-handed grabbing it is intuitive and no long practicing is needed.

5 Object Movement Modification

To allow for bimanually moving an object we developed a manipulation technique that is inspired by the grab-and-twirl technique introduced by Cutler et al. [CFH97]. The grab-and-twirl technique can be formalized as follows. The translational object movement is determined by the average translational movements of both hands. The rotational object movement is determined by two different perpendicular rotations: the rotation of the line connecting the two hand positions and the average hand rotations around this line. Both rotations are applied to the object. Note that Cutler et al. [CFH97] proposed to use not the average hand rotations but instead the rotation of only one hand. However, because we observed the combination of both enabling more precise rotations we preferred to use the average.

In contrast to the grab-and-twirl technique we do not trigger grabbing and releasing of an object by moving the hands' virtual representations until they both intersect the object and then performing a grabbing posture. Instead, using the technique introduced in Sec. 4 grabbing and releasing can take place independent of the size or position of the virtual object or the distance between the hands. Therefore, grabbing and releasing is significantly easier and more comfortable, but in order to preserve the intuitiveness of object movements the hands' physical movements must not be mapped to the virtual world in the same way as done in the grab-and-twirl technique. In the grab-and-twirl technique an object is translated by a certain mapping of the physical center between the two hands to the virtual center of the object. This is reasonable, because the virtual hand positions coincide with the two pivot points where the object is grabbed (see Fig. 5(Top left)). In our technique, the user also anticipates two pivot points being located close to the object's surface where it is grabbed, but as in our case the distance between the hands' positions is not fixed to the object size, the same mapping as in Fig. 5(Top left) of phys-

ical to virtual movements would lead to unnatural results (see Fig. 5(Top right)). To obtain reasonable results the mapping has to be modified as illustrated in Fig. 5(Bottom left).

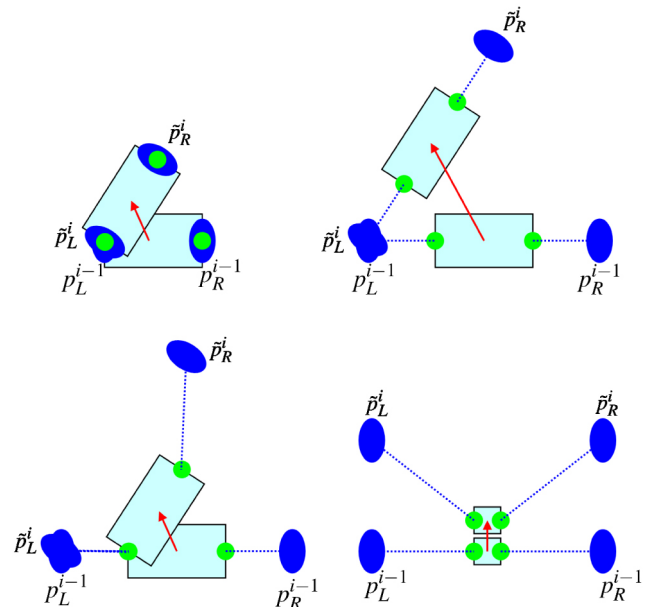


Figure 5: Illustration of the problem of mapping real to virtual coordinates with fixed distance between the hands. An object (rectangular box) is moved by exploiting the positions of both hands (blue ellipses). The pivot points are illustrated in green and the applied translation vector in red. Top left: distance between hands is equal to distance between pivot points. Top right: hand distance is greater than distance between pivot points but the same mapping is used as in (Top left). Bottom left: hand distance is greater than distance between pivot points and a suitable mapping is used. Bottom right: hand distance is much greater than distance between pivot points and a scaling by the ratio of pivot point distance to hand position distance is used.

Unfortunately, if a solution which simply scales the translation by the ratio of the distance between the two pivot points (determined by the two intersection points of the object's convex hull with the line through the object center having the direction of the line connecting the two hand positions) to the distance between the hands would be applied, small objects could hardly be translated any more as illustrated in Fig. 5(Bottom right). Our solution consists in two steps: first, we adjust the hands' positions such that their distance is normalized and second, we split the translational move-

ment up into an asymmetric part and a symmetric part which are scaled differently.

First, the left and right hand positions p_L^i and p_R^i in the current frame i are adjusted to have the same distance as in the previous frame $i-1$. The adjusted position for p_L^i is obtained by moving p_L^i in direction of p_R^i with a certain signed distance. This signed distance is equal to the difference between the hand distances d^i and d^{i-1} in frame i and $i-1$, respectively, multiplied by a suitable weighting factor λ_L^i . The scheme for obtaining the adjusted left hand position \tilde{p}_L^i is defined as:

$$\tilde{p}_L^i := \lambda_L^i (d^i - d^{i-1}) \frac{p_R^i - p_L^i}{\|p_R^i - p_L^i\|} + p_L^i. \quad (4)$$

The weighting factor λ_L^i is defined to be the ratio of the translational amount of the right hand to the sum of translational amounts of both hands and can be formalized as follows:

$$\lambda_L^i = \frac{\|p_R^i - p_R^{i-1}\|}{\|p_R^i - p_R^{i-1}\| + \|p_L^i - p_L^{i-1}\|}. \quad (5)$$

\tilde{p}_R^i is determined analogously by swapping all L 's and R 's in both equations. This way the hand that moved slower becomes stickier for the object (i.e. the object tends to stay in touch with this hand). If for example one hand remains at the same position the object can not be translated toward the grabbing direction by movements of the other hand. Furthermore, no more translational object movements are induced by decreasing or increasing the distance between the hands. Note that λ_L^i is equal to $(1 - \lambda_R^i)$ and $\lambda_L^i \in [0, 1]$, therefore the distance between the adjusted hand positions is guaranteed to be equal to the distance between the previous hand positions.

Second, the translational movement is split up into an asymmetric part (difference of the hand translation amounts) and a symmetric part (dissimilarity of the hand translation amounts). The asymmetric part describes the amount of translation that influences the position of the center of rotation. This part has to be applied in a one-to-one mapping (i.e. it has to be mapped as in Fig. 5(Bottom left)). The symmetric part describes the amount of translation that has no influence on the displacement of the rotation center. This part can be applied to the object translation with an arbitrary scaling (independent of the object size or the hand distance) without affecting the intuitiveness. The two parts can be computed as follows: if t_L^i and t_R^i are the left and right hand translations with

$t_{L/R}^i = \tilde{p}_{L/R}^i - p_{L/R}^{i-1}$ in frame i , then $t^i := \frac{1}{2}(t_L^i + t_R^i)$ describes the total amount of translation. We then define λ^i to be the percentage of symmetric translation with

$$\lambda^i = \frac{\min(\|t_L^i\|, \|t_R^i\|)}{\max(\|t_L^i\|, \|t_R^i\|)} \quad (6)$$

Now, the resulting modified total translation \tilde{t}^i can be written as:

$$\tilde{t}^i = s\lambda^i t^i + r(1 - \lambda^i)t^i, \quad (7)$$

where r is defined to be the ratio of object extend to the current distance between the hands and s can be chosen arbitrarily to alter the mapping from real to virtual coordinates. This way, if an asymmetric rotation is performed (e.g. one hand is moved fast, the other stands still) the object is rotated intuitively around the pivot point touching the object and corresponding to the hand that was not moved (see Fig. 6(Left)). On the contrary, if a symmetric movement of both hands in equal directions is performed the object translation does not depend on the object size or distance between the hands (see Fig. 6(Right)).

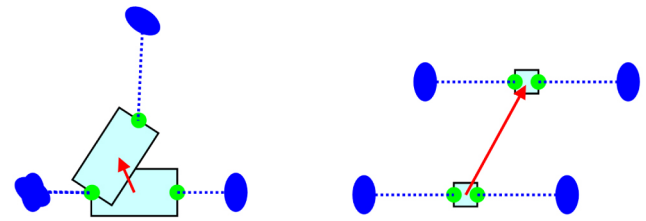


Figure 6: Illustration how our technique maps real to virtual movement. Left: Asymmetric movement. Right: Symmetric movement.

In order to provide a visual clue about where the object is grabbed and how it can be turned around, the resulting two pivot points are visualized in the 3D interface. Two spheres are drawn at the corresponding positions as long as the object is grabbed. Additionally, this gives a direct visual feedback about whether the object is grabbed or released. An illustration is given in Fig. 7.

With this technique both human hands are exploited for simultaneous manipulation of the whole 6 DOFs of an object's pose in order to gain a significantly improved rotational precision and a generally higher control over the object. These advantages are due to the following reasons: first, the rotation depends on the

distance between the hands leading to an increased precision if the distance is increased. Second, the center of rotation can be chosen more intuitively than in one-handed interaction, because object rotations are determined by hand translations (e.g. fixing one hand to a certain position leads to rotating an object around this point when the other hand is moved).

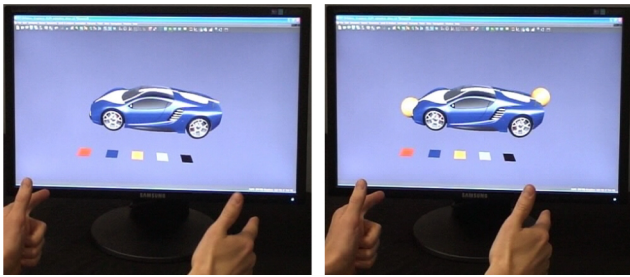


Figure 7: Visual feedback for grabbing and releasing. Left: The object is released. The pivot points are not visualized. Right: The object is grabbed. The two yellow spheres indicate the pivot points and that the object is grabbed. (Car model courtesy of RTT AG)

6 Integration with One-Handed Techniques

We argue that symmetric two-handed object manipulation is primarily suitable as an extension of one-handed manipulation. Only if high precision and/or control of the object is needed the application of both hands can significantly improve the current task. A reasonable scenario could be as follows: a user selects an object and moves it to the approximate pose by employing only one hand. Only then she/he uses both hands for fine adjustment of the object pose. In order to allow for such scenarios in an effective way, transitioning from single-handed to two-handed interaction and backwards must be manageably fast and uncomplicated.

The first case of switching from single to two-handed interaction can simply be solved by automatically enabling the two-handed interaction mode when the second hand enters the working volume.

The second case of changing from two to single-handed interaction is more complicated. This is due to the problem that a hand could have left the working volume unintentionally. This occurs if the user currently performs an object manipulation at the edge of the working volume and moves one hand outside

by mistake. To this end, we distinguish between two cases: first, if the object was grabbed when the hand left the working volume we assume leaving was unintended and prompt a message on the screen to move the hand back inside. Second, if the object was already released we assume the user wants to switch back to the one-handed manipulation mode which is therefore automatically enabled in this case. This distinction turned out to be intuitive and easy to handle.

7 Stabilizing the Object Movement

Exploiting the human hands as direct input devices by using vision based markerless hand-tracking has several great advantages (e.g. intuitive, non-obtrusive, etc.). However, as well some problems arise when such devices are exploited. In the following we explain some major drawbacks and present our solutions.

One drawback of vision based markerless hand-tracking is the dependency on the hand segmentation in the camera images. Performing a good segmentation becomes even harder when two hands are tracked simultaneously due to reciprocal occlusions. Such occlusions occur more often if the distance between the hands decreases. We observed the tracking stability of the hands' orientations to be primarily sensitive to such hand configurations. Note that in the employed two-handed object manipulation technique the hand orientations are only exploited for rotating the object around the line connecting the two hand positions (see Sec. 5). In order to avoid distracting rotational jumps during object manipulation we propose the following simple stabilization procedure: if the angular velocity of one hand exceeds a certain threshold (currently we use $2\pi \frac{rad}{s}$) its rotation is set to the identity. This threshold value was determined by analyzing the typical velocities of rotational jumps induced by an incorrect tracking. This way, the jumps of the hands' orientations are not applied in the current step leading to a stable object manipulation. In return, very fast rotations around this axis can not be performed any more. This is a clear limitation, but as it counts only for velocities higher than $2\pi \frac{rad}{s}$ the interaction is marginally affected. In practice, we discovered such high angular velocities to be exploited for less than 1% of object rotations in typical manipulation tasks.

Another problem induced by segmentation issues is that tracking two hands is impossible when they can not be separated in each camera image. In this case, a message is prompted requesting the user to increase

the distance between her/his hands.

In particular relevant for two-handed object manipulations is the problem of limited work space; the hands can only be tracked if they can be seen by enough cameras. If an object shall be translated a long distance the hands have to be moved an accordingly way in the working volume. If in this case the user grabs, holds and moves an object with a large space between her/his hands, the range of available translational movement becomes even smaller and one hand often leaves the working volume unintentionally. Therefore, we prompt a hint suggesting the user to decrease the distance between the hands if one hand leaves the working volume and the distance between the hands was higher than *30cm* in the preceding frame. This threshold value was used, because we observed that larger distances do not significantly improve the accuracy of object manipulations.

8 Experiments

We conducted an experiment to evaluate the performance of the proposed technique for two-handed grabbing and releasing virtual objects. Our hypothesis was that our method would be superior to other two-handed techniques commonly applied for hand-tracking devices. Moreover, we expected that our technique would generally be superior to one-handed techniques in high precision tasks.

In order to investigate these hypotheses, we compared our two-handed grabbing and releasing technique to three other approaches: the exploitation of a grabbing posture for triggering grabbing/releasing (i.e. only if the subject forms the grabbing posture with one or both hand(s) the virtual object moves according to her/his hands), the use of a standard 6 DOF controller (3D mouse) and the application of our recently developed one-handed technique; the Jerky Release technique (see [SNK09]). The idea of this one-handed technique is to grab the object by default and only release it while the user performs a fast and jerky hand movement. Note that in a short user study this one-handed technique outperformed a grabbing posture based technique and showed similar results as a standard 3D mouse in 6 DOF manipulation tasks.

8.1 Experimental Setting

Two connected PC-based systems were used in the experiment, one coupled to the cameras for tracking the

hand (see Sec. 3) and another (Intel E6600, Geforce 8800) for running the virtual environment application. The application was visualized on a standard 19 TFT-Display. Additionally, a 3D connexion SpaceNavigator was connected to the second PC.

An illustration of how this system works together with our bimanual technique is given in Fig. 8).

8.1.1 Experimental Task

In the experimental task a virtual object had to be moved to a specified position and orientation (see Fig. 9) by using the different 6 DOF techniques/controllers. The time needed for completing one task was split up into the approach time and the precision time. The approach time is measured when the object is approximately approached (see Fig. 9)); when it is moved roughly to the target pose (i.e. less than 6 degrees rotational error and 4 units translational error). The period from this moment on until the user releases the object almost precisely (i.e. less than 1, 5 degrees rotational error and 0, 5 units translational error) at the target pose (see Fig. 9(Right)) is defined to be the precision time. This way, both the capabilities of performing rough/spacious as well as precise/fine manipulations are measured for each technique/controller. To reach the target pose several grab and release cycles had to be performed for the hand-tracking based techniques. No snapping algorithm (the object snaps to the desired pose, when it is near by) was applied.

8.1.2 Participants

Ten participants (all males, all university students) took part in the experiment. They had little or no virtual reality experience.

8.1.3 Procedure

Each participant had to solve the task four times by employing each of the four controllers. Thereby, the sequence of the employed controllers was permuted according to ten columns of three balanced mutually orthogonal Latin squares of size 44. All tasks had to be finished until the next controller was adopted. Before starting the test for each controller, its mode of action was explained and the subjects could familiarize with it in a short preparation time (two minutes).

8.2 Results and Discussion

The average approach and precision times including standard deviations are illustrated in Fig. 10. The average total times of the different controllers were 83 (two handed), 48.3 (3D Mouse), 59.6 (single handed) and 44.5 (our technique) seconds. Note that our technique also performed best in average for the entire task.

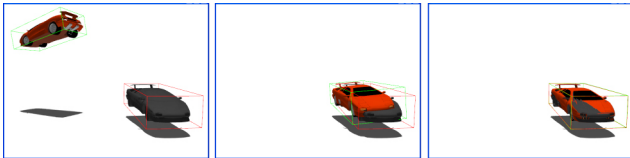


Figure 9: Illustration of the experimental task that had to be solved. Left: Initial situation. Middle: Roughly approached target pose (the approach time is measured). Right: Solved task (the precision time is measured).

The high standard deviations in Fig. 10 are caused by two major reasons:

1. Different manual skills of the individuals. A subject that performed *slowly/fast* with a certain controller mostly performed *slowly/fast* with the other controllers, too. Thereby, *slow/fast* means having high/low average per controller timings with respect to the other participants.
2. Different individual controller preferences. While several users could easily handle one controller some others were nearly incapable to work with it. In particular precisely positioning by using the 3D mouse had very inconsistent results. For this reason we argue that in an end user interface redundancies between different techniques (if possible) and/or selectable alternatives should be available.

Despite the high standard deviations, some statistical significances could be derived by performing the Students t-test. In the following discussion we consider a confidence greater than 95% to be significant.

Roughly approaching an object was superior if only one hand or the 3D mouse was employed instead of a two-handed technique. This is mainly because more grab and release cycles were used for two-handed translational movements due to the higher need of workspace (both hands have to be located in the working volume) and the slightly reduced anatomical range

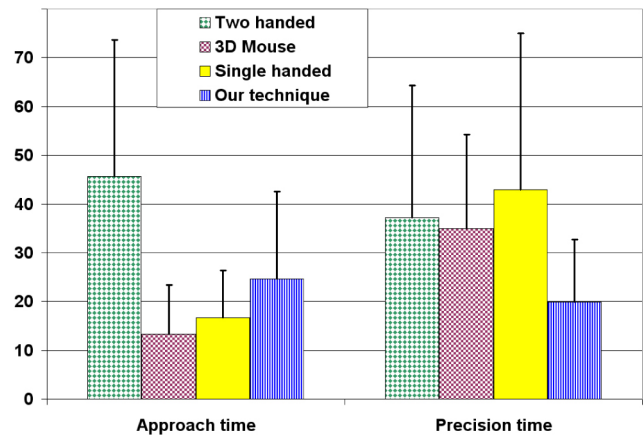


Figure 10: Completion times (in seconds) and standard deviations (error bars).

(the shoulder has to remain straight if both arms are stretched). Only marginal difference was found between the 3D mouse and the single-handed object approaching. However, all techniques have significant superior results to the two-handed posture based technique. We observed the subjects to have some problems using different postures for grabbing and releasing, primarily because a higher concentration was needed for switching between the standard and the grabbing posture. The differences between the one-handed techniques and our two-handed technique is only significant with a confidence greater than 90% for the 3D mouse and 75% for the single-handed technique.

The best results in precisely positioning an object were achieved applying our two-handed technique. Furthermore, the differences between our two-handed technique and the two one-handed techniques are significant. The difference between our two-handed technique and the two-handed posture based technique are only significant with a confidence greater than 90%. Surprisingly, the precision times of the two-handed posture based technique are similar high as of the 3D mouse and the one-handed technique, respectively. This is caused by overhasty hand movements during a release operation while the hands were still holding the virtual object. Note that in this case the involuntary object movement leads to loosing the target pose so the subject must repeat the high precision positioning.

After the user experiments, we questioned the subjects for their subjective impression concerning the performance of the different controllers. For precisely positioning eight participants would prefer our two-

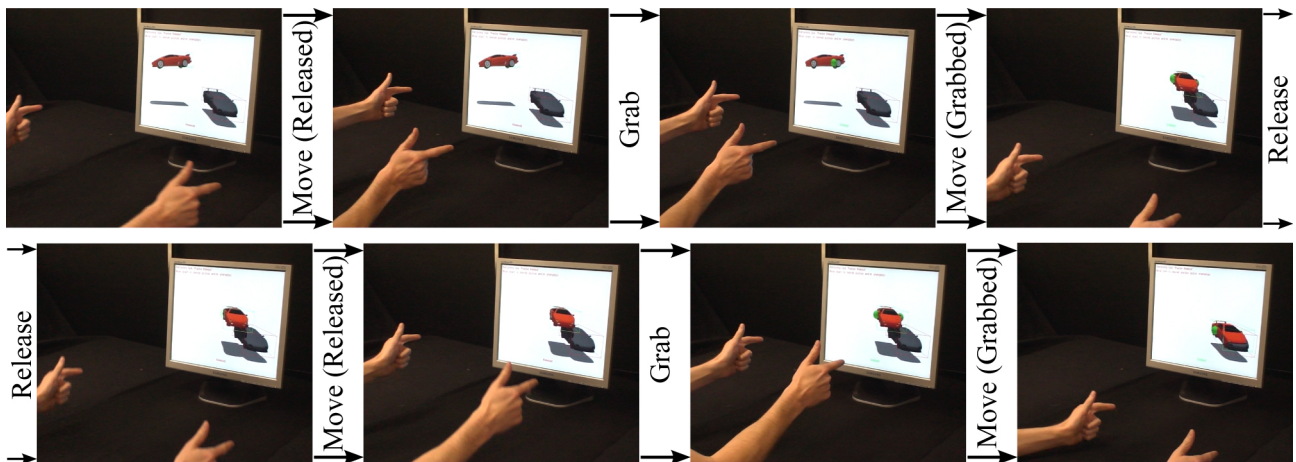


Figure 8: Image sequence of our system. Based on our bimanual symmetric interaction technique a virtual object (red car) can be grabbed (move hands together), manipulated (as if gripped between hands) and released (move hands apart).

handed technique, one would prefer the twohanded posture based technique and another one the 3D mouse. For approaching the target they rated both the single-handed technique and the 3D mouse best with four votes for each. The other two participants voted for our two-handed technique. Note that these ratings correspond to the measured times.

9 Integration with RTT DeltaGen

RTT DeltaGen is a professional tool enabling real-time visualization and editing of professional CAD/CAS datasets. We integrated the hand-tracking device with Version 8.5 released 2009 by the Realtime Technology AG. One marked strength of using bare-handed tracking for 3D interaction is the ease and quickness that people become able to handle even complex tasks. Therefore, a scenario easily enabling normal people to interact with CAD models was the focus in the integration with RTT DeltaGen. Obviously, suitably editing CAD models needs a lot of practice, independent on the adopted interaction device. Therefore, we concentrated on the visualization part of RTT DeltaGen. The user should be able to easily inspect a 3D object (e.g. a car) from all viewpoints as well as to switch between different variants of the object (e.g. lacquer, rims) or activate predefined animations (e.g. open a door). The interaction should be as easy to handle as possible.

To this end, at least two different interaction modes are needed: one mode for 3D object manipulation/ inspection, where the object or the camera moves ac-

ording to the hand(s), and another mode like menu navigation, where the user can select a variant/ animation and can issue a switch/activate command. In this context, enabling easy switching between these modes is crucial. Moreover, the user should always be aware of the mode she/he is currently in. To this end, the two modes should be distinguishable and easy to switch. We experimented with typical different purely one-handed solutions as for example different postures for different modes or splitting up the working volume in different regions. Using postures led to two main problems: the users were distracted by ensuring that they form the correct posture and the object movement could not be stopped precisely due to movements induced by the posture change. Splitting up the working volume often led to involuntary mode switching because no tactile feedback is available. Moreover, the working space for 3D manipulation is reduced in this case. Therefore, we also experimented with two-handed interaction and came to a solution, where one-handed interaction is used for menu navigation and two-handed for 3D manipulation. This solution performed best in several informal experiments and is implemented as follows.

If two hands are concurrently located inside the working volume, the Bimanual Symmetric Grab technique (see Sec. 4) is adopted to translate and rotate the 3D object. This way, the object can intuitively be inspected from any point of view. See Fig. 11 for an illustration.

In the one-handed interaction mode the user con-

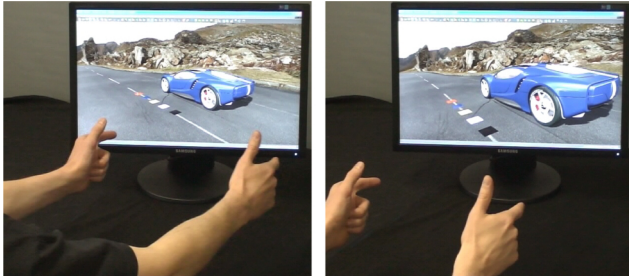


Figure 11: Two screenshots of using the Bimanual Symmetric Grab in RTT DeltaGen. By moving the hands from the front (left image) to the back (right image) the car and surrounding is moved accordingly. (Car model courtesy of RTT AG)

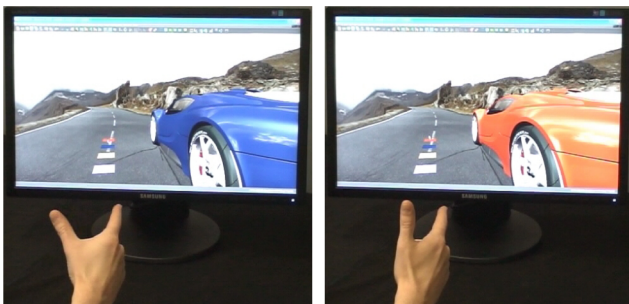


Figure 12: Two screenshots of using the 2D mouse cursor in RTT DeltaGen. By clicking (using the Roll Click technique) on the red squares plane (used as buttons) the color of the car is switched from blue (left image) to red (right image). (Car model courtesy of RTT AG)

trols the 2D mouse cursor and can click on object parts or other items to start animations or switch the design variant. This way also the menu of RTT DeltaGen can be handled in order to change arbitrary settings. The 2D mouse cursor is moved by using the pointing direction of the index fingers (the yaw axis angle for the X-coordinates and the pitch axis angle for the Y-coordinates). Clicking can be performed via the Roll Click technique [SNK09], where a small hand rotation around the roll axis is used to trigger a click event. In Fig. 12 two screenshots illustrate such a clicking operation used for changing the lacquer of a car model.

Switching between one and two-handed interaction is solved according to the mechanism proposed in Sec. 6.

By using one and two-handed interaction for distinguishing cursor control and 3D manipulation, mode switching is easy and the user is always aware of the mode she/he is currently in. In addition, because only easily manageable interaction techniques are adopted, also unpracticed users are able to use it without training.

10 Conclusions and Future Work

We presented a novel technique for two-handed grabbing and releasing virtual objects which can be used for efficient symmetric object manipulations. We further introduced some modifications needed for preserving the intuitiveness of these symmetric manipulations. We show how our two-handed technique can reasonably be integrated with a single-handed interface and introduced some stabilization procedures needed for compensating some drawbacks of markerless two hand-tracking. Furthermore, we performed a user experiment to compare our two-handed technique to another two-handed technique, to a single-handed technique as well as to a standard 6 DOF controller. Last but not least, we described how our technique was integrated with a commercial CAD-application and why this is superior to other approaches.

The results of our user study confirmed our hypotheses that single-handed manipulation performs better for coarse tasks and two-handed for high precision tasks. In order to benefit from both advantages a possible solution would be combining a single-handed technique with our two-handed technique. To discover which single-handed technique best fits to our twohanded technique we plan to further investigate the performance of different combinations.

Another future direction might be adapting different bimanual asymmetric techniques for the special problems and requirements of markerless hand-tracking.

References

- [BH97] Doug A. Bowman and Larry F. Hodges, *An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments*, SI3D '97: Proceedings of the 1997 symposium on Interactive 3D graphics, ACM, 1997, ISBN 0-89791-884-3.
- [BI05] Christoph W. Borst and Arun P. Indugula, *Realistic virtual grasping*, VR '05: Proceedings of the 2005 IEEE Conference 2005 on Virtual Reality, IEEE Computer Society, 2005, ISBN 0-7803-8929-8.
- [BKLP05] Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev, *3d user interfaces: Theory and practice*, Addison Wesley Longman Publishing Co., Inc., 2005, ISBN 978-0201758672.
- [Bol80] Richard A. Bolt, "put-that-there": *Voice and gesture at the graphics interface*, SIGGRAPH Comput. Graph. **14** (1980), no. 3, 262–270, ISSN 0097-8930.
- [CFH97] Lawrence D. Cutler, Bernd Fröhlich, and Pat Hanrahan, *Two-handed direct manipulation on the responsive workbench*, SI3D '97: Proceedings of the 1997 symposium on Interactive 3D graphics, ACM Press, 1997, pp. 107–114, ISBN 0-89791-884-3.
- [GFGB04] C. Grätzel, T. Fong, S. Grange, and C. Baur, *A non-contact mouse for surgeon-computer interaction*, Technol. Health Care **12** (2004), no. 3, 245–257, ISSN 0928-7329.
- [Gui87] Yves Guiard, *Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model*, Journal of Motor Behavior **19** (1987), no. 4, 486–517, ISSN 0022-2895.
- [JS07] Alejandro Jaimes and Nicu Sebe, *Multimodal human-computer interaction: A survey*, Comput. Vis. Image Underst. **108** (2007), no. 1-2, 116–134, ISSN 1077-3142.
- [LMS03] M. Li, M. Magnor, and H.-P. Seidel, *Hardware-accelerated visual hull reconstruction and rendering*, Proc. Graphics Interface (GI'03), 2003, pp. 65–71, ISBN 1-56881-207-8.
- [MBS97] Mark R. Mine, Jr. Frederick P. Brooks, and Carlo H. Sequin, *Moving objects in space: exploiting proprioception in virtual-environment interaction*, SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co., 1997, pp. 19–26, ISBN 0-89791-896-7.
- [MF04] N. Murray and T. Fernando, *An immersive assembly and maintenance simulation environment*, DS-RT '04: Proceedings of the 8th IEEE International Symposium on Distributed Simulation and Real-Time Applications, IEEE Computer Society, 2004, pp. 159–166, ISBN 0-7695-2232-7.
- [ML04] Shahzad Malik and Joe Laszlo, *Visual touchpad: a two-handed gestural input device*, ICMI '04: Proceedings of the 6th international conference on Multimodal interfaces, ACM Press, 2004, pp. 289–296, ISBN 1-58113-995-0.
- [MM95] D. Mapes and J. M. Moshell, *A twohanded interface for object manipulation in virtual environments*, Presence: Teleoperators and Virtual Environments **4** (1995), no. 4, 403–416, ISSN 1054-7460.
- [OKF+05] Russell Owen, Gordon Kurtenbach, George Fitzmaurice, Thomas Baudel, and Bill Buxton, *When it gets more difficult, use both hands: exploring bimanual curve manipulation*, GI '05: Proceedings of Graphics Interface 2005 (School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, Canadian Human-Computer Communications Society, 2005, pp. 17–24, ISBN 1-56881-265-5.

- [Osa06] Noritaka Osawa, *Automatic adjustments for efficient and precise positioning and release of virtual objects*, VRCIA '06: Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications, ACM, 2006, pp. 121–128, ISBN 1-59593-324-7.
- [PBWI96] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa, *The go-go interaction technique: non-linear mapping for direct manipulation in vr*, UIST '96: Proceedings of the 9th annual ACM symposium on User interface software and technology, ACM, 1996, pp. 79–80, ISBN 0-89791-798-7.
- [PSP99] Jeffrey S. Pierce, Brian C. Stearns, and Randy Pausch, *Voodoo dolls: seamless interaction at multiple scales in virtual environments*, I3D 99: Proceedings of the 1999 symposium on Interactive 3D graphics, ACM, 1999, pp. 141–145, ISBN 1-58113-082-1.
- [SCP95] Richard Stoakley, Matthew J. Conway, and Randy Pausch, *Virtual reality on a wim: interactive worlds in miniature*, CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM Press/Addison-Wesley Publishing Co., 1995, pp. 265–272, ISBN 0-201-84705-1.
- [SK07] Markus Schlattmann and Reinhard Klein, *Simultaneous 4 gestures 6 dof real-time two-hand tracking without any markers*, ACM Symposium on Virtual Reality Software and Technology (VRST '07), 2007, pp. 39–42, ISBN 978-1-59593-863-3.
- [SNK09] Markus Schlattmann, Tanin Na Nakorn, and Reinhard Klein, *3d interaction techniques for 6 dof markerless handtracking*, International Conference on Computer Graphics, Visualization and Computer Vision (WSCG '09), 2009, ISBN 978-80-86943-93-0.
- [WP03] Andrew Wilson and Hubert Pham, *Pointing in intelligent environments with the worldcursor*, INTERACT, 2003, ISBN 978-1-58603-363-7.
- [Zac00] G. Zachmann, *Virtual reality in assembly simulation - collision detection, simulation algorithms, and interaction techniques*, Ph.D. thesis, Darmstadt University of Technology, Germany, Department of Computer Science, 2000.
- [ZFS79] Robert C. Zeleznik, Andrew S. Forsberg, and Paul S. Strauss, *Two pointer input for 3d interaction*, SI3D '97: Proceedings of the 1997 symposium on Interactive 3D graphics, ACM, 1979, pp. 115–ff, ISBN 0-89791-884-3.

Citation
Markus Schlattmann and Reinhard Klein, <i>Efficient Bimanual Symmetric 3D Manipulation for Bare-Handed Interaction</i> , Journal of Virtual Reality and Broadcasting, 7(2010), no. 8, October 2010, urn:nbn:de:0009-6-26686, ISSN 1860-2037.