# Non-Isomorphic 3D Rotational Techniques

**Ivan Poupyrev [1], Suzanne Weghorst [2], Sidney Fels[3]**

[1] ATR MIC Research Laboratories
2-2 Hikaridai, Seika, Soraku-gun
Kyoto 619-02, Japan
(774) 95-1432
poup@mic.atr.co.jp

[2] HIT Lab, University of Washington
Box 352142
Seattle, WA 98195, USA
(206) 685-3215
weghorst@hitl.washington.edu

[3] Department of ECE,
University of British Columbia
Vancouver, BC Canada, V6T 1Z4
(604) 822-5338
ssfels@ece.ubc.ca

**ABSTRACT**

This paper demonstrates how non-isomorphic rotational mappings and interaction techniques can be designed and used to build effective spatial 3D user interfaces. In this paper, we develop a mathematical framework allowing us to design non-isomorphic 3D rotational mappings and techniques, investigate their usability properties, and evaluate their user performance characteristics. The results suggest that non-isomorphic rotational mappings can be an effective tool in building high-quality manipulation dialogs in 3D interfaces, allowing our subjects to accomplish experimental tasks 13% faster without a statistically detectable loss in accuracy. The current paper will help interface designers to use non-isomorphic rotational mappings effectively.

**Keywords:** 6DOF input devices, interactive 3D rotations, 3D user interfaces, interaction techniques, motor control.

## INTRODUCTION

Three-dimensional (3D) computer graphics has advanced from a subject of research curiosity to an indispensable tool in many areas of human activities. While visual quality and rendering efficiency have been rapidly improving, the design of efficient interfaces for 3D applications remains a practical concern for application developers and a vexing problem for researchers in industry and academia [12].

Direct manual control has a very special place in 3D user interfaces design and research: human hands remain the dominant channel of interaction not only for 3D interfaces, but also for traditional 2D graphical user interfaces (GUI) as well for our everyday interaction with the physical world. The quality of interface components that enable users to manipulate objects and scenes in virtual environments has a profound effect on the quality of the whole interface – if the user cannot manipulate effectively, many specific application tasks simply cannot be performed. Consequently, a large amount of research has already addressed various issues in multidimensional manipulation, e.g., designing and evaluating multiple degrees-of-freedom (DOF) input devices, innovating new manipulation interaction techniques, investigating implications of human motor skills on 3D interface design, and many others [e.g. 1, 8, 12, 22, 25].

The design of 3D mappings and interaction techniques, which translate user-operated device motions into object movements in virtual environments, is certainly one of the core issues in designing manipulation interfaces. The challenge was very well defined by Sheridan (cited from [24]): "How do the geometrical mappings of body and environmental objects, both within the virtual environment and the true one, and relative to each other, contribute to the sense of presence, training, and performance? … In some cases there may be a need to deviate significantly from strict geometric isomorphism because of hardware limits, or constraints of the human body. At present we do not have design/operating principles for knowing what mapping … is permissible, and which degrades performance."

Designing and investigating non-isomorphic mappings for 3D spatial user interfaces has recently been an area of active research [e.g. 2, 8, 12, 13, 14, 15], and the current paper adds to this body of work. In particular, we explore how non-isomorphic rotational mappings can be designed and used to enhance 3D rotations of objects and scenes in virtual worlds. The paper attempts to close a current gap in the literature on multidimensional interaction, where 3D mappings and interaction techniques have been used only with the translation components of multiple DOF input. When it comes to 3D rotations, most researchers, as well as producers of commercial devices and software, have used only the simplest one-to-one (*isomorphic*) mapping between rotations of the multiple DOF controller and virtual objects. In fact, even the basic equations of control-display (C-D) gain for 3D rotations and their properties have not been reported.[1] In comparison, C-D mappings for translation tasks have been used and studied since the early 1940s.

This paper demonstrates how non-isomorphic 3D rotational mappings can be constructed and effectively used to design 3D interfaces. First, we introduce a basic mathematical framework that allows design of both linear and non-linear C-D mappings between device rotations and rotations in 3D interface space. This framework is based on the idea of extrapolating the orientation of a multiple DOF device on a quaternion sphere in four dimensions. Second, we identify basic idiosyncratic properties of rotational mappings, such as relations between the mappings and device form-factor, and discuss issues of interaction techniques design. Finally, we report experiments which have shown that by using our technique, subjects could accomplish an experimental task 13% faster without any significant loss in accuracy.

## BACKGROUND AND RELATED WORK

Any interface between humans and machines that uses continuous manual control includes three basic components: 1) input devices, which capture user actions, 2) display de-

[1] We reported preliminary results in the CHI'99 late-breaking paper [16].



ACM CHI 2000, Copyright ACM

vices, which present the effect of these actions back to the user, and 3) transfer functions, often referred to as control-display mappings, which map the movements of the device into the movements of controlled elements of the system or interface [11, 24] (Figure 1). The goal is to design input devices, displays and transfer functions that facilitate high user performance and comfort, while diminishing the impact from human and hardware limitations [11].

The design of mapping functions for manual control and studies of their impact on operator performance stretch back to the 1940s [10]. It has also been an active research area in 3D user interfaces where two philosophies have emerged [24]: The *isomorphic* view suggests a strict geometrical isomorphism (i.e. one-to-one mapping) between motions in the physical and virtual worlds, on the grounds that it is the most natural and therefore is better for users. The results of early human factor studies indicated that while isomorphism is, indeed, often more natural [overview in 11], it also has important shortcomings. First, isomorphic mappings are often impractical because of constraints in the input technologies, e.g., the limited tracking range of input devices. Second, isomorphism is often ineffective due to the limitations of human operators, e.g., anatomical constraints. Finally, it has been argued that 3D interfaces can be more effective, intuitive and richer if, instead of imitating the physical reality, we create mappings and interaction techniques that are specifically tailored to virtual environments, providing in some sense a "better" reality [e.g. 21].

Hence, the *non-isomorphic* approach suggests that manipulation mappings and techniques can significantly deviate from strict realism, providing users with "magic" virtual tools, e.g., laser rays, rubber arms, Voodoo Dolls [13, 14] and others. These non-isomorphic mappings and techniques allow users to manipulate objects quite differently than in the physical world, yet rather effectively [2, 15]. In fact, the majority of 3D direct manipulation techniques today are non-isomorphic techniques.

We should note, however, that non-isomorphic mappings are not an entirely new idea; they have been used for decades in a variety of everyday controls, e.g., dials, pedals, handlers, and wheels, where our input is scaled, shifted or integrated using different mapping or *transfer functions* [11]. Traditionally, the human factors literature categorizes transfer functions by a number of integrations, applied to the user input [11, 24]. Thus, in *zero-order* mappings, displacement of the input device results in displacement of the controlled element, while in *first-order* mappings, it results in a change of its velocity. Consequently, they are often referred to as position and rate control, respectively.

The simplest example of zero-order mapping is a linear control-display gain function which scales the user input:

$$D_d = kD_c, \qquad (1)$$

where $D_c$ and $D_d$ are displacements of the controller and displayed elements, respectively, and $k$ is a ratio of scaling. The zero-order control should not necessarily be linear; for example, various dials in consumer electronic devices often use non-linear mappings. Non-linear position control has also been used in VR interaction techniques for object manipulation [14] and navigation [20]. A good example of first-order control is the steering wheel of a car, where the displacement of the steering wheel results in the change of the car's angular velocity.

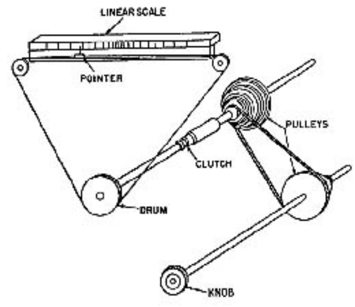

Figure 1: Basic components of any direct manipulation system: input device, output device, and transfer function (in this figure: knob, pointer, and pulleys respectively) [10].

The design of non-isomorphic mappings cannot be accomplished without considering the properties of input devices. The most important device property is the number of the degrees of freedom: early research on 3D user interfaces was often concerned with the design and evaluation of techniques for performing 3D tasks with 2D input devices, e.g., ARCBALL or Virtual Trackball techniques, which use a mouse to rotate 3D objects [9, 18]. In multiple DOF input, additional device properties have to be considered. For example, studies by Zhai [24, chapter 2] have shown that isometric devices, such as force-resistant joysticks, allow for better rate control performance, while isotonic devices, such as free-moving magnetic trackers, are preferable for position control. Given that the same device permits a large variety of mappings, device-mappings compatibility is an important and interesting research direction.

The non-isomorphic mappings and interaction techniques have been designed, until now, only for translation components in multiple DOF input. When it comes to 3D rotations, most researchers, as well as producers of commercial devices and software, use only the simplest one-to-one mapping between the 3D rotation of the device and virtual objects. In fact, even the basic equations of mappings that would linearly amplify the device rotations, i.e., linear C-D gain, have not been reported.

What advantages can we gain by using non-isomorphic 3D rotational mappings? Indeed, it can be argued that, unlike translations, the rotation space is limited to 360 degrees, so any desired orientation can be easily achieved. This issue, however, may well be moot. First, the entire 360 degrees of rotations cannot always be tracked; for example, in computer vision-based tracking, the range of rotations that can be reliably measured is often less then 180 degrees [16]. The non-isomorphic mappings would allow a more effective use of this limited tracking range.

Second, the effective range of rotations in manual control is naturally constrained by human anatomy: our joints can only rotate up to a certain angle. Hence, controlling the large range of rotations is difficult and requires *clutching*, i.e., releasing a virtual object, re-adjusting the hand, and continuing the manipulation. Clutching, however, is frustrating and can noticeably degrade user performance. While an appropriate device form-factor can reduce clutching [25], it cannot eliminate it. Thus, non-isomorphic mappings can be used to decrease clutching in 3D rotations.

Finally, the introduction of non-isomorphic mappings for 3D rotations would provide interface designers with an additional tool for fine-tuning 3D user interfaces and creating new mappings and 3D interaction techniques.

## CONTROL-DISPLAY MAPPINGS IN 3D ROTATIONS

In this section we introduce a basic mathematical framework that allows design of both linear and non-linear C-D mappings between device rotations and rotations in 3D interface space. The design of these mappings is not obvious and requires a consideration of the fundamental mathematical properties of rotations in space. The resulting framework is based on the idea of extrapolating multiple DOF device orientations on a quaternion sphere in 4 dimensions.

### Rotations in space

Rotations in 3D space are significantly more confusing then they appear, since they do not follow familiar laws of Euclidean geometry. For example, rotate an object in some direction and it would eventually return to its initial starting orientation, something which cannot happen in a vector space. This is because the space of rotations is not a vector space but a closed and curved surface, a manifold, in four dimensions, which can also be represented as a 4D sphere.

The connection between spatial rotations and spherical geometry is quite natural and can be illustrated using a simple physical example. Imagine rotating a rigid physical object, e.g., a pencil, about a fixed point. Apparently, the tip of the pencil would travel on the surface of a sphere and each *orientation* of the pencil can be identified as a *point* on this sphere. Furthermore, a pencil *rotation* around an axis would draw an *arc* and if the pencil has unit length, then the length of this arc equals the rotation angle. Thus, the orientation of the body can be conveniently represented as a point on a unit sphere, while rotation can be represented as an arc on a sphere, connecting the starting and final body orientations.

This example is illustrative, albeit not quite correct: a point on a 3D sphere specifies a family of rotations, since twisting the pencil along the longest axis would not draw any arcs. Since a sphere in 3D specifies only two degrees of rotational freedom, we need to move into a higher, fourth dimension to specify all three degrees of rotations. This is exactly what *unit quaternions* allow us to do.

### Quaternions

Quaternions were discovered by Hamilton in 1843 [17]. Since then, they have been widely used in robotics, avionics and any other application field that requires an efficient way to describe and operate 3D rotations. Introduced into computer graphics and interface design by Shoemake [17, 18], today quaternions are a standard tool in the arsenal of the interactive computer graphics professional.

Quaternion $q$ is a four-dimensional vector often represented as a pair ($\mathbf{v}$, w), where w is a real number and $\mathbf{v}$ is a 3D vector. Given quaternion $q$, we can compute its length $|q|$ and inverse $q^{-1}$; given quaternion $q'$, we can compute their multiplication $qq'$ and a dot product $q \cdot q'$. A quaternion of unit length can be used to represent a single rotation about unit axis $\hat{\mathbf{u}}$ by angle $\vartheta$ in two equal forms as follows:

$$q = (\sin\frac{\vartheta}{2}\hat{\mathbf{u}}, \cos\frac{\vartheta}{2}) = e^{\frac{\vartheta}{2}\hat{\mathbf{u}}}.$$

Rotating a vector $\mathbf{v}$ about axis $\hat{\mathbf{u}}$ by angle $\vartheta$ can be computed as the double quaternion multiplication $\mathbf{v}' = q\mathbf{v}q^{-1}$. A sequence of rotations $q_1, q_2 \ldots q_n$ can be easily computed as the multiplication $q_n \ldots q_2 q_1$ (notice the reversed order; see Appendix for operation definitions).

The set of all unit quaternions forms a unit sphere in four dimensions and each point on its surface represents an orientation of a rigid body. It was proven by Euler that a combination of any number of rotations can be represented as a single rotation from an reference orientation. A unit quaternion represents this single rotation as a *great arc* connecting the reference and current body orientations on quaternion sphere. The length of this arc equals ½ of the rotation angle. Thus, just as we use vectors to represent translations, we also can use spherical arcs to represent 3D rotations. If the reference orientation is not explicitly specified, a quaternion defines the rotation from the identity quaternion $\mathbf{1} = (\bar{0}, 1)$, which has a special meaning as a zero orientation or no rotation – an equivalent to the origin in a vector space.

### Linear zero-order C-D gain for 3D rotations

Given an orientation of the multiple DOF input device, what mapping allows us to amplify or scale this orientation in a manner similar to scaling translations of the device?

Amplifying rotation means changing the amplitude while preserving the direction of rotation. Let $q_c$ be the orientation of a multiple DOF input device:

$$q_c = (\sin\frac{\vartheta_c}{2}\hat{\mathbf{u}}_c, \cos\frac{\vartheta_c}{2}) = e^{\frac{\vartheta_c}{2}\hat{\mathbf{u}}_c},$$

where $\hat{\mathbf{u}}_c$ is the axis of rotation and $\vartheta_c$ is the angle. The zero-order C-D gain should amplify the angle of rotation $\vartheta_c$ by coefficient $k$ while leaving axis $\hat{\mathbf{u}}_c$ intact:

$$q_d = (\sin\frac{k\vartheta_c}{2}\hat{\mathbf{u}}_c, \cos\frac{k\vartheta_c}{2}) = e^{k\frac{\vartheta_c}{2}\hat{\mathbf{u}}_c} = q_c^k.$$

Therefore, the basic equation for the zero-order linear C-D gain for spatial rotations is a *power* function of the form:

$$q_d = q_c^k, \qquad (2)$$

where $q_c$ is the device rotation, $q_d$ is the displayed orientation, and $k$ is the C-D gain coefficient.

Quaternion $q_c$ in Equation 2 specifies device orientation as a rotation from an unspecified initial orientation designated by identity quaternion $\mathbf{1}$. However, it is often important to amplify rotation relative to some explicitly specified reference orientation $q_0$. This can be done by calculating the rotation that connects $q_0$ and $q_c$, amplifying it, and combining it with reference orientation $q_0$:

$$q_d = (q_c q_0^{-1})^k q_0. \qquad (3)$$

Notice that Equation 3 is identical to the *slerp* function introduced by Shoemake for rotation interpolation [17]. This should not come as a surprise; indeed, while Shoemake *interpolates* quaternions using a great arc on a quaternion

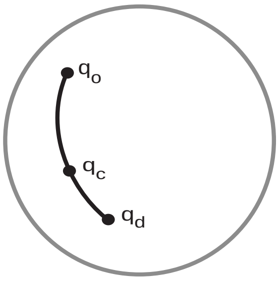


Figure 2: Extrapolating device orientation $q_c$ on a quaternion sphere; $q_0$ and $q_d$ are initial and displayed orientations.

sphere, we *extrapolate* the device orientation using a great arc connecting $q_0$ and $q_c$ (Figure 2)[2]. Therefore, we can use an equivalent formula that is easier to apply [17]:

$$q_d = q_0 \frac{\sin((1-k)\Omega)}{\sin(\Omega)} + q_c \frac{\sin(k\Omega)}{\sin(\Omega)}$$

where $\Omega$ can be obtained from $\cos\Omega = q_c \cdot q_d$.

Equations 2 and 3 are fundamental equations of rotational C-D gain. They are fundamental in the sense that they represent a basic form of zero-order C-D mappings between rotations of the device and rotations in a 3D interface space and provide a generic method for constructing a variety of rotational techniques suitable for particular application.

These mappings are *linear*, since the rate of amplification does not change no matter how far the user rotates the device. The non-linear mappings, which might be useful, are not discussed here; we refer the interested reader to [16].

## INTERACTION TECHNIQUES: DESIGN GUIDLINES

In the previous section, we derived the basic equations of mapping that allow us to linearly amplify rotations of multiple DOF input devices. This section discusses how these equations can be used to design non-isomorphic techniques for rotating objects in VEs. At the center of this discussion are important and non-intuitive differences between absolute and relative mapping schemes in 3D rotations.

### Absolute and relative mappings: it makes a difference

Typical isotonic multiple DOF devices, such as magnetic trackers, are absolute devices, i.e., they measure and return the absolute displacement of the device relative to the initial, zero orientation [4]. Hence, the easiest method for implementing non-isomorphic techniques is to map the absolute orientation of device $q_{c_i}$, measured on *i*-th cycle of the simulation loop, using Equations 2 or 3:

$$q_{d_i} = q_{c_i}^k,$$

and apply the resulting absolute orientation $q_{d_i}$ to virtual objects, scenes, and virtual viewpoints.

An alternative way to implement non-isomorphic techniques using the same equations is to amplify only relative changes in the device orientation, i.e. on *i*-th cycle of the simulation loop, we calculate the relative rotation of device rotation from its orientation on the *i-1*th cycle and amplify it. The orientation of virtual object $q_{d_i}$ is then calculated by combining this amplified relative rotation with the orientation of virtual object on the *i*-1 step of the simulation loop:

$$q_{d_i} = (q_{c_i} q_{c_{i-1}}^{-1})^k q_{d_{i-1}}. \qquad (4)$$

Hence, the difference between these two mapping schemes is that in the first one we amplify the absolute orientation of the device, while in the second one we amplify its relative rotations. Consequently, we will refer to them as *absolute* and *relative* non-isomorphic rotation mappings.

Differentiating between absolute and relative mappings in spatial rotations is important for two reasons. First, they are different from a mathematical point of view: *given the same rotation path of the device, these two mappings produce different rotation paths of the displayed object* [3]. This might be unexpected; indeed, in the case of translations, relative and absolute mappings would obviously yield the same trajectory of movement. This, however, is yet another example of the peculiar nature of curved rotational space.

Second, *absolute and relative mappings are very different from the usability point of view*. The "feel" of the manipulation largely depends on the choice between relative and absolute mappings. The next section compares and contrasts the usability characteristics of relative and absolute mappings and their implications for 3D interface design.

### Usability properties of absolute and relative techniques

Our ability to self-regulate motor movements, e.g., object manipulation, depends on spatial and temporal correspondence between a large variety of sensory feedbacks: visual, tactile, kinesthetic, proprioceptive and others. If the computer response, e.g., visual feedback, conflicts with kinesthetic or proprioceptive feedback produced by the human motor system, then the user performance degrades [19]. Therefore, the effectiveness of manipulation techniques depends on whether they preserve *compliances* between the user motor movements and the sensory feedback s/he receives, i.e., a stimulus-response (S-R) compatibility [6, 19].

In this section, we examine whether absolute and relative rotational mappings preserve two particular compliances which are important for effective direct manipulation: 1) compliance between the rotation directions of the input device and virtual object, i.e., *directional compliance* and 2) compliance between initial orientations of the object and input device, which we refer to as a *nulling compliance*.

#### *Directional compliance*

Directional compliance in spatial rotations simply means that as the user rotates the multiple DOF input device, the virtual object rotates in the same direction, i.e., around the same axis. Directional compliance ensures correspondence between visual, kinesthetic, proprioceptive and other feedbacks of motor movement [7, 19]. Britton [3] introduced

---

[2] Spherical arcs have also been used in Arcball [18]. However, using spherical arcs to represent 3D rotations is a standard practice while the purpose and realization of Arcball are different from the present work.

[3] To prove this, we need to show that if for a sequence of $n$ incremental rotations $q_n q_{n-1} \ldots q_1 = q$ then generally $q_n^k q_{n-1}^k \ldots q_1^k \neq q^k$ (*). Although an analytical proof is beyond the scope of the paper, it can be easily tested empirically: for $n$=3, $k$=2, $q_1$=(0.8,0.6,0,0), $q_2$=(0.8,0,0.6,0) and $q_3$=(0.64,-0.48,-0.48,-0.36), the left part of equation (*) is (0.7, 0.3,-0.5,-0.2), while the right part is (0,0,0,1).

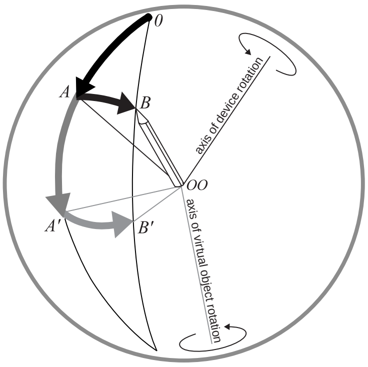


Figure 3: Directional *in*compliance in absolute mapping: (black – actual rotations; gray – amplified rotations).

directional compliance to computer graphics as a principle of kinesthetic correspondence. We can show that:

*1) Relative non-isomorphic mappings always maintain directional compliance* between rotations in physical and virtual spaces. As shown in Equation 4, in each cycle of the simulation loop, the virtual object is incrementally rotated in the same direction as the device, though with a different amplitude. *2) Absolute non-isomorphic mappings generally do not preserve directional compliance* between rotations of the input device and virtual objects. To illustrate this let us consider a simple physical example. Suppose our input device is a pencil fixed in point *00* and it rotates from the initial orientation *0* to *A* and then to *B* (Figure 3). The absolute orientation of the pencil can be represented as an arc connecting initial orientation *0* and the tip of the pencil. The absolute mapping would always scale this arc *relative to the initial orientation 0,* and then apply the resulting amplified orientation to the virtual pencil, which would rotate from *0* to *A'* and then to *B'*. It is obvious from Figure 3 that rotation *AB* of the physical pencil and rotation *A'B'* of the virtual pencil will happen around different axes.

### *Nulling compliance*

Nulling compliance ensures that nulling the device, i.e., rotating it into an initial, zero orientation [4], would also rotate the controlled virtual object into a zero orientation. Nulling compliance preserves the consistent correspondence between the origins of the coordinate systems in physical and virtual spaces. We can show that:

*1) Absolute non-isomorphic mappings strictly preserve nulling compliance*. This follows directly from Equation 2; indeed, raising the identity quaternion, i.e., zero rotation, to a power will always yield the identity quaternion. *2) The relative mappings do not generally preserve nulling compliance*. Nulling the device would not necessarily rotate the virtual object into the expected initial state, but rather into some unpredictable orientation. This follows directly from the discussion in footnote 2.

How important is nulling compliance for direct manipulation in 3D interfaces? The answer depends on the other design variable – the form-factor of the input device.

### *Non-isomorphic mappings and device form-factor*

The shape of a multiple DOF input device can make its manipulation easier or harder, by involving different muscle groups [25]. The device form-factor can also provide cognitive clues to the user on how it can be used [8]. In addition, we observe that different device form-factors provide different sensory feedbacks on the physical orientation of the device. For example, a device mounted on the user's hand, e.g., a data glove, provides the user with strong kinesthetic and proprioceptive feedback on the device's orientation. The user inherently *feels* the device orientation, hence the inconsistency between the zero orientation of the device and the zero orientation of the virtual object, will be noticed and may degrade user performance.

On the other hand, devices that are not worn on the body but are freely rotated in the user's fingers do not inherently provide any kinesthetic or proprioceptive feedback on their orientation. In cases where the device's geometrical shape can be recognized through tactile feedback, the inconsistencies between orientations of the device and virtual objects can still be perceived. However, a homogeneous form, such as a sphere, provides very little sensory information about its actual physical orientation. For such a device, all orientations are equivalent and it is difficult, if impossible, for the user to discriminate between them. Hence, the nulling compliance becomes unnecessary.

To conclude, multiple DOF devices, which have a spherical form and can be manipulated in the fingers, such as Zhai's Finger Ball [25], are not subjected to nulling compliance constraints. Such devices can be effectively used with non-isomorphic relative mappings, as experimental studies reported later in the paper will demonstrate.

### **Design trade-off in 3D rotational techniques**

The difference between using relative and absolute C-D mappings in designing rotational techniques is a question of a trade-off between the directional and the nulling consistency: we cannot have both (Table 1).

Absolute non-isomorphic mappings can only have a limited use since they do not always preserve the directional com-

| | *Absolute mapping* | *Relative mapping* |
|---|---|---|
| **Directional compliance** | **No,** virtual object does *not* always rotate in the same direction as the device | **Yes,** virtual object *always* rotates in the same direction as the device |
| **Nulling compliance** | **Yes,** nulling device *always* returns virtual object to initial orientation | **No,** nulling device does *not* necessarily return virtual object to initial orientation |
| **Device form factor** | **Worn on body or easy recognizable shape**  | **Homogeneous, sphere**  |

Table 1: Design trade-off in 3D rotational techniques

pliance, and therefore do not allow the user to consistently predict the response of the virtual object on the device rotations. These mappings, however, can be useful when the device rotations do not change the axis much. For example, we have used them with satisfactory results for viewpoint control using head rotations tracked by a camera [16].

Relative non-isomorphic mappings can be very efficient in manual control tasks if the multiple DOF input device provides little tactile and kinesthetic feedback on its actual orientation and can be freely rotated in the fingers, such as in the case of a Finger Ball and to a certain degree the Polhemus Space Ball. We will support this observation by presenting the results of the following experimental studies.

## EXPERIMENTAL USABILITY STUDY

An experimental usability evaluation was conducted to investigate the performance characteristics of a relative non-isomorphic rotational technique compared with conventional one-to-one mapping in a 3D object rotation task. Based on the results of pilot studies, the following preliminary hypotheses were formulated prior to the experiments.

*$H_1$: A relative amplification of multiple DOF input device rotations will allow subjects to accomplish a rotation task faster than with traditional isomorphic mappings when a large range of rotations is required. A non-isomorphic mapping will not have a significant effect on subject performance for a small range of rotations.*

A large range of rotations usually requires clutching or involves larger muscles of the arms and shoulders, and this decreases user performance [25]. We suggest that a non-isomorphic interaction technique with moderate amplification of rotations will allow subjects to use their fingers more effectively, reduce the need for clutching, and therefore result in faster task completion. We hypothesize that non-isomorphic mapping would not result in better performance for small rotations, and we were interested in whether there would be a decrease in user performance.

*$H_2$: Non-isomorphic techniques with moderate amplification of rotations will decrease rotational accuracy.*

The higher the sensitivity of a device, the more difficult it is to rotate the device precisely into the required orientation. While it seems logical to assume that accuracy will suffer, we were interested in how significant the decreases in rotational accuracy would be.

Finally, we were interested in estimating subjects' preferences for rotation techniques. The rotation task has an inherently limited range – 360 degrees – and can be accomplished with or without amplification. Strong subject preferences for some of the techniques would indicate that the choice of mapping does make a difference and therefore should be considered in the design of spatial interfaces.

### Subjects and apparatus

Twenty unpaid subjects, eighteen male and two female, all right handed, age range from 19 to 35, were recruited from the laboratory subjects pool. None of the subjects had previous experience with 6DOF input devices.

The experiments were conducted in desktop environments using the SGI O2 workstation with a 17" 1280x1024 pixels true color monitor. The update rate was controlled between 19 and 25 Hz. The Polhemus SpaceBall 6DOF magnetic sensor was selected as the input device, and a mouse was used as the trigger device. The coefficient of amplification in non-isomorphic technique was chosen empirically at 1.8.

### Experimental task

The experimental task design followed the design of orientation matching experiments used by Chen [5] and Hinckley [8]. Participants were instructed to rotate a solid shaded 3D model of a house from a randomly generated orientation into a requested, a priori-specified target orientation (Figure 4). The target orientation was a front of the house, indicated by the front door, facing the user. The house model was made to provide maximum clues to understanding its orientation, e.g., asymmetric location of chimney and windows.

The user picked up and released a house by pressing and releasing the mouse button with the non-dominant hand. The user could rotate the house iteratively using clutching – pick, rotate, release, re-adjust the hand, and re-pick the house as many times as necessary to orient it within the threshold of the specified accuracy. When the error of orientation fell below the threshold, which was approximately 18 degrees ($\pi/10$), the house would disappear, cueing subjects that the task had been accomplished successfully. The next trial was presented after a three-second delay.

This task design differed from Chen's in two respects. First, Chen required subjects to rotate a house from a fixed initial orientation into a randomly generated one. We slightly simplified the task by reversing it: the user rotated the house from the initial random orientation into a known one. Second, Chen rated and scored the user's completion accuracy after each task as "Excellent," "Good match," and so on. We used the accuracy threshold instead, because it allowed us to implicitly control the difficulty of the task as well as to provide participants with clear criteria of task completion.

### Experiment design and procedure

The repeated measures-within subject experimental design was used. The independent variables were *interaction technique* (one-to-one and non-isomorphic mapping) and *amplitude* of rotation, defined as the shortest rotation required to rotate the house into the target orientation. The amplitude variable had two levels: *small* (a random angle from 20 to 60) and *large* (a random angle from 70 to 180).

The dependent variables were *completion time* and *orientation error*. The completion time was measured from the

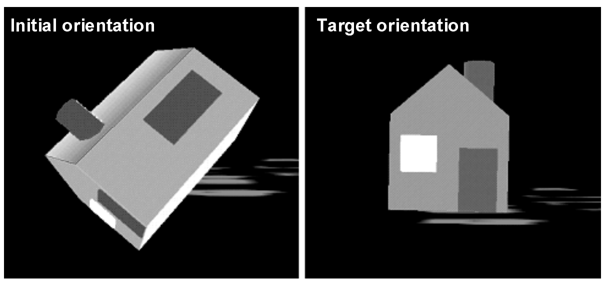


Figure 4: Task required users to rotate house model from randomly generated initial orientation (left) to target orientation, with front of the house facing user (right).

moment the user picked up a house until the moment the house was oriented with the required accuracy. Error was measured as the angular difference between the final orientation of the house and target orientation.

The experiments started with an explanation of the techniques, experimental task and procedure, followed by a 15 to 20 minute training to stabilize the manipulation performance and ensure understanding of the task and techniques. The training was followed by the experimental session consisting of two blocks of trials, one with one-to-one and the other with non-isomorphic mapping. Each block consisted of ten trials: five with large and five with small amplitude of rotation, randomized. All subjects matched the same randomly generated orientations. To control for order effect, half of the subjects started with one-to-one mapping while the other half started with the non-isomorphic technique.

In a questionnaire administered after completion of the experiments, subjects were asked to rate the techniques on a scale from 0 to 4 (0 = very bad, 1 = bad, 2 = OK, 3 = good, and 4 = excellent) and explain their choices. The experiments took from 45 minutes to 1 hour for each subject.

**Results**

A repeated-measures two-way analysis of variance (ANOVA) was performed for each of the dependent variables with *interaction techniques* and *amplitude* as independent variables. Data for completion time was transformed using a natural logarithm, since analysis revealed that the data was skewed away from a normal distribution.

Table 2 outlines the main effects of independent variables as well as their interaction for each dependent variable. Both technique and amplitude significantly affected the completion time. The interaction technique, however, was not a significant factor for the orientation error. A significant interaction between technique and amplitude for completion time suggests that the effect of the interaction technique depends on rotational amplitudes.

A separate comparison of techniques for small and large rotations shows that the non-isomorphic mapping was on average 13.4 percent faster when a large amplitude was required ($F_{1,19} = 7.3$, $p < 0.01$, Figure 5), while no significant difference was found for small rotations ($F_{1,19} = 0.03$, $p < 0.87$). This finding supports the first hypothesis. Both techniques resulted in almost the same orientation error: the average was 6.9 and 6.6 degrees for non-isomorphic and one-to-one mappings, respectively (Figure 5). This difference is insignificant both statistically and qualitively.

**Subjects preferences**

In the questionnaire 18 subjects (95%) preferred non-isomorphic to one-to-one mapping; on average they were rated

| | ***t**echnique* | ***a**mplitude* | *interaction **t*a*** |
|---|---|---|---|
| **Time** | $F_{1,19} = 5.52$, $p < 0.03$ | $F_{1,19} = 113.932$, $p < 0.0001$ | $F_{1,19} = 5.68$, $p < 0.028$ |
| **Error** | $F_{1,19} = 2.29$, $p < 0.15$ | $F_{1,19} = 15.7$, $p < 0.001$ | $F_{1,19} = 0.026$, $p < 0.874$ |

Table 2: The main effects and interaction

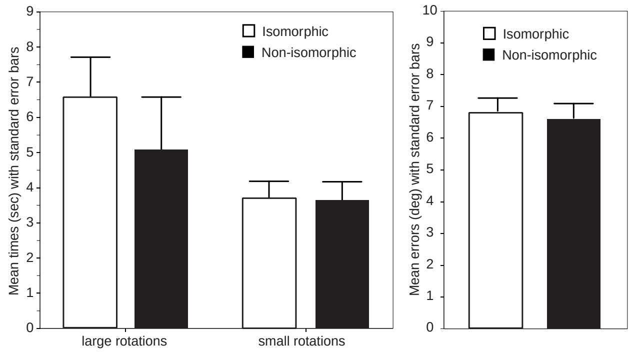


Figure 5: Left: mean completion times; right: mean errors of orientation, collapsed for all amplitudes

3.15 and 2.3, respectively, on a scale from 0 to 4. A paired *t*-test confirmed that this difference was statistically significant: $t_{19} = 3.9$, $p < 0.001$. Subjects noted that non-isomorphic mappings allowed them to rotate objects faster, with little re-adjustment of hand or device and less physical effort when a large range of rotations was required. Three subjects specifically commented that amplified rotations allowed them to use their fingers over a larger range of rotations, which they found was more efficient.

Many subjects reported that it was slightly more difficult to precisely control device rotations with non-isomorphic mapping. However, many of them noted that it was a question of experience and practice. Two subjects suggested that it would be useful to be able to control the sensitivity of mapping and use slower rotations, especially when accuracy was important. The cable of the SpaceBall tracker was found to be the most disturbing factor in the experiments.

**Discussion**

The experiments demonstrated that a non-isomorphic interaction technique, which linearly amplifies rotations of a multiple DOF input device, allowed the subjects to accomplish the experimental task 13% faster compared with one-to-one mapping for a large range of rotations. The performance for a small range of rotations was the same. The subjects' strong preferences for the non-isomorphic techniques also suggest that rotational mappings are an important design variable in constructing 3D user interfaces.

Furthermore, mappings had no effect on the accuracy of rotation, or at least none that could be detected with 20 subjects. If we compare our results with Hinkley's experiments [8], who used a similar experimental design, our subjects averaged 6.8 degrees of error, while Hinkley's 24 subjects averaged 6.7 degrees of error. Although our experimental design emphasized speed, our results closely replicated Hinkley's, even though his experiments emphasized accuracy instead. This supports Hinkley's observation that the accuracy of rotation might be less affected by the manipulation capabilities of the interface then by the difficulties subjects had in perceiving and adjusting the rotation error. Furthermore, the experiments of Ware and Rose [23] demonstrated that even when subjects rotated ordinary physical objects in real world, there was a natural limit in accuracy that averaged 4.64 degrees. In 3D interfaces, the accuracy can deteriorate further due to insufficient depth cues or lag.

For completion time, Hinkley's subjects averaged 17.8 seconds while our subjects averaged 5.15 seconds for one-to-one mapping. This difference can be explained, first, by the emphasis of accuracy in Hinkley's experiments, i.e., his subjects spent more time trying to match orientation; second, in the training level, i.e., his subjects were given as little instructions as possible, while we tested the stabilized manipulation performance. The experiments of Ware and Rose, on the other hand, resulted in quite comparable subject performance: their 4.96 versus our 5.15 seconds. Thus, we believe that our experiments produced quite accurate estimates of user rotational performance and accuracy.

## CONCLUSIONS

This paper demonstrates how non-isomorphic 3D rotational interaction techniques can be constructed and used to design effective spatial user interfaces. We attempted to provide a thorough treatment of this subject, by designing the mathematical foundations of rotational mappings, investigating their usability properties, and evaluating their user performance characteristics. Our results suggest that non-isomorphic rotational mappings are an effective tool in building high-quality manipulation dialogs in 3D interfaces. This paper will help designers to use them effectively.

## ACKNOWLEDGMENTS

Research reported in this paper has been partially conducted as part of the first author's Ph. D. work at Hiroshima University. We are thankful to many people for their suggestions, discussion and help. The basic idea for this work emerged from a discussion between the first author and Jock McKinlay. Michael Svinin and Horst Kraemer have provided invaluable help in discussing the subtle aspects of the mathematics involved in 3D rotations. Takeo Igorashi, Mark Billinghurst, Michael Kowalski and Prof. Ichikawa provided crucial feedback that helped us present these materials in the best possible way. We are also thankful to all subjects who participated in the experiments as well as anonymous reviewers for their comments.

## REFERENCES


1. Boritz, J., Booth, K., A study of interactive 3D point location in a computer simulated virtual environment. *Proceedings of VRST'97*. 1997. ACM. pp. 181-187.
2. Bowman, D., Hodges, L., An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. *Proceedings of I3DCG*. 1997. ACM. pp. 35-38.
3. Britton, E., Lipscomb, J., Pique, M., Making nested rotations convenient for the user. *Proceedings of SIGGRAPH'78*. 1978. ACM. pp. 222-227.
4. Buxton, W., There's more to interaction then meets the eye: some issues in manual input. In *User Centered System Design: New Perspectives on Human-Computer Interaction*, D. Norman and S. Draper, Editors. 1986, Lowrence Erlbaum, pp. 319-337.
5. Chen, M., Mountford, S., Sellen, A., A study in interactive 3D rotation using 2-D control devices. *Proceedings of SIGGRAPH'88*. 1988. ACM. pp. 121-129.
6. Fitts, P., Jones, R., Compatibility: spatial characteristics of stimulus and response codes. *Journal. of Experimental Psychology*, 1953 (46).
7. Gould, J., Smith, K., Angular displacement of the visual feedback in motion. *Science*, 1962(137): pp. 619-620.
8. Hinckley, K., Tullio, J., Pausch, R., et al., Usability analysis of 3D rotation techniques. *Proc. of ACM UIST'97*. 1997. pp. 1-10.
9. Hultquits, J., A Vrtual Trackball. In *Graphics Gems I*. 1990, Academic Press. pp. 462-463.
10. Jenkins, W., Connor, M., Some design factors in making settings on a linear scale. *Journal of Applied Psychology*, 1949. 33(4): pp. 395-409.
11. Knight, J., Manual control and tracking. In *Handbook of human factors*, Salvendy, Ed. 1987, John Wiley&S. pp. 182-218.
12. Mine, M., Brooks, F., Sequin, C., Moving objects in space: exploiting proprioception in virtual-environment interaction. *Proceedings of SIGGRAPH'97*. 1997. ACM. pp. 19-26.
13. Pierce, J., Stearns, B., Pausch, R., Voodoo Dolls: Seamless interaction at the multiple scales in virtual environments. *Proceedings of I3DCG'99*. 1999. ACM. pp. 141-145.
14. Poupyrev, I., Billinghurst, M., Weghorst, S., Ichikawa, T., Go-Go Interaction Technique: Non-Linear Mapping for Direct Manipulation in VR. *Proc. of UIST'96*. 1996. ACM. pp. 79-80.
15. Poupyrev, I., Weghorst, S., Billinghurst, M., Ichikawa, T., Egocentric object manipulation in virtual environments: empirical evaluation of interaction techniques. *Computer Graphics Forum, EUROGRAPHICS'98 issue*, 1998. 17(3): pp. 41-52.
16. Poupyrev, I., Weghorst, S., Otsuka, T., Ichikawa, T., Amplifying rotations in 3D interfaces. *Proc. of CHI'99 Conference Abstracts, Late Breaking Results*. 1999. ACM. pp. 256-257.
17. Shoemake, K., Animating rotations with quaternion curves. *Proceedings of SIGGRAPH'85*. 1985. ACM. pp. 245-254.
18. Shoemake, K., ARCBALL: a user interface for specifying three-dimensional orientation using a mouse. *Proceedings of Graphics Interface'92*. 1992. pp. 151-156.
19. Smith, T., Smith, K., Feedback-control mechanisms of human behavior. In *Handbook of human factors*, G. Salvendy, Editor. 1987, John Wiley and Sons. pp. 251-293.
20. Song, D., Norman, M., Nonlinear interactive motion control techniques for virtual space navigation. *Proceedings of VRAIS'93*. 1993. IEEE. pp. 111-117.
21. Stoakley, R., Conway, M., Pausch, R., Virtual reality on a WIM: interactive worlds in miniature. *Proceedings of CHI'95*. 1995. pp. 265-272.
22. Wang, Y., MacKenzie, L., Object manipulation in virtual environments: relative size matters. *Proceedings of CHI'99*. 1999. ACM. pp. 48-55.
23. Ware, C., Rose, J., Rotating virtual objects with real handles. ACM *Transactions on Computer-Human Interaction*, 1999. 6(2): pp. 162-180.
24. Zhai, S., Human performance in six degrees of freedom input control, Ph.D. Thesis, *Department of Industrial Engineering*. 1995, University of Toronto, Canada.
25. Zhai, S., Milgram, P., Buxton, W., The influence of muscle groups on performance of multiple degree-of-freedom input. *Proceedings of CHI'96*. 1996. ACM. pp. 308-315.


## APPENDIX: QUATERNIONS

The definitions of the quaternion operations used in this paper are as follows:

$$q^* = (-\mathbf{v}, w);\ |q| = \sqrt{x^2 + y^2 + z^2 + w^2};\ q^{-1} = \frac{q^*}{|q|}$$

$$qq' = \mathbf{v} \times \mathbf{v}' + w\mathbf{v}' + w'\mathbf{v}, ww' - \mathbf{v} \cdot \mathbf{v}';\ q \cdot q' = \mathbf{v} \cdot \mathbf{v}' + w \cdot w'$$

More information on quaternions as well code samples can be found at http://www.hitl.washington.edu/people/poup/ or http://www.mic.atr.co.jp/~poup/