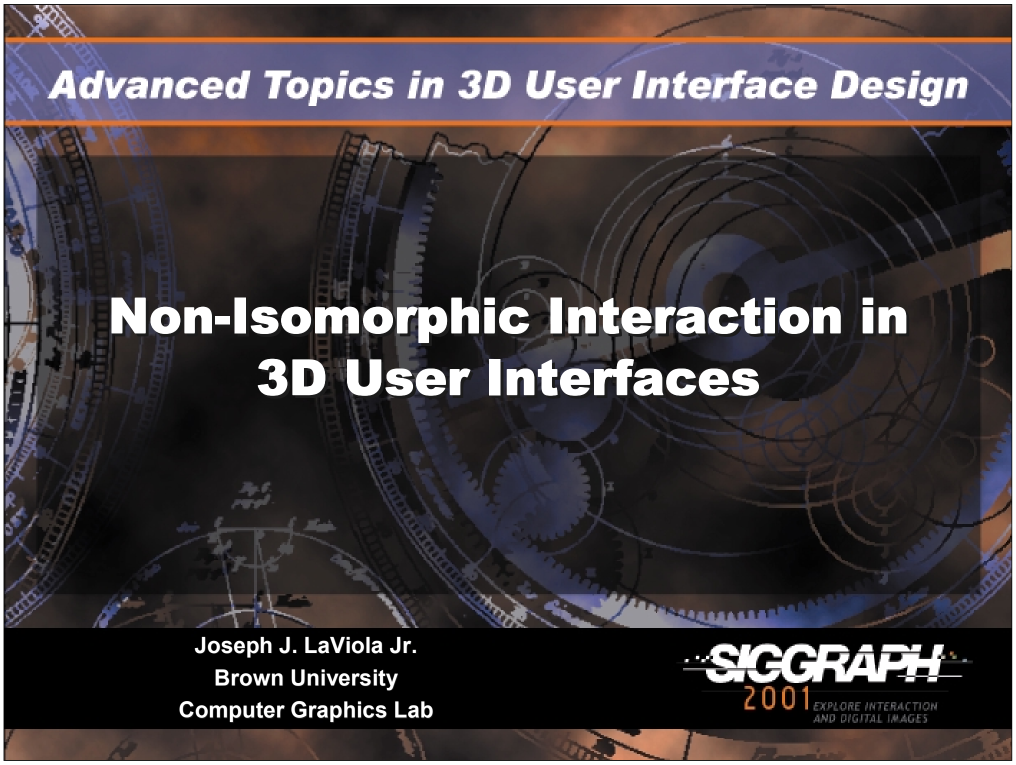
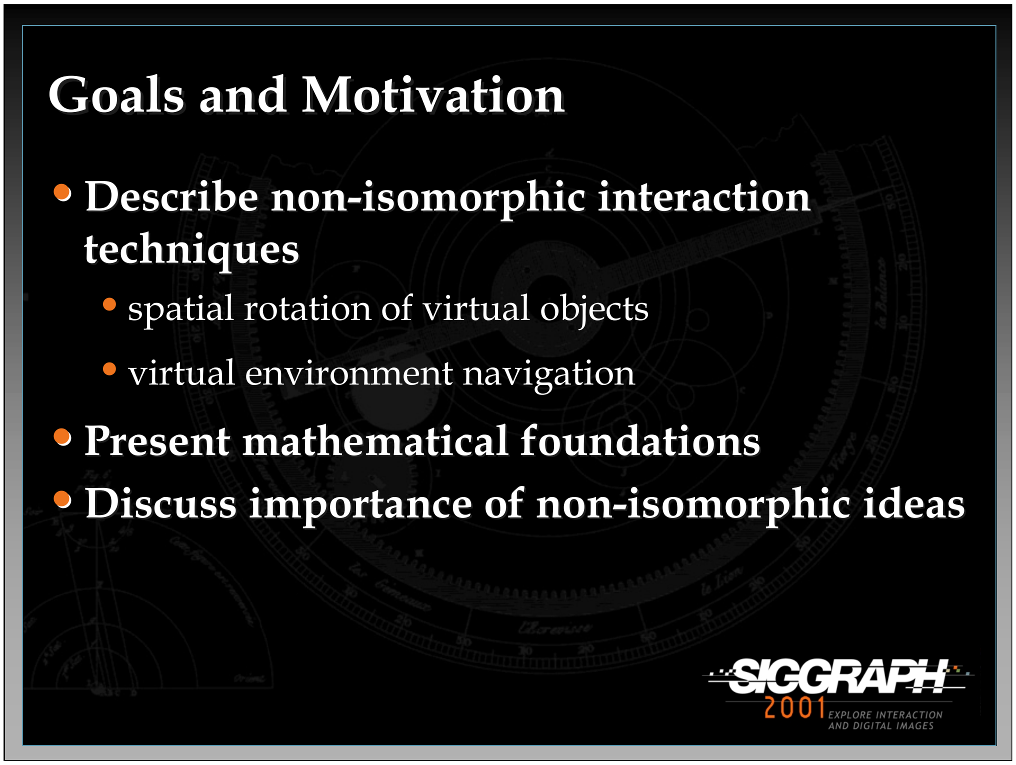


Non-Isomorphic Interaction in 3D User Interfaces

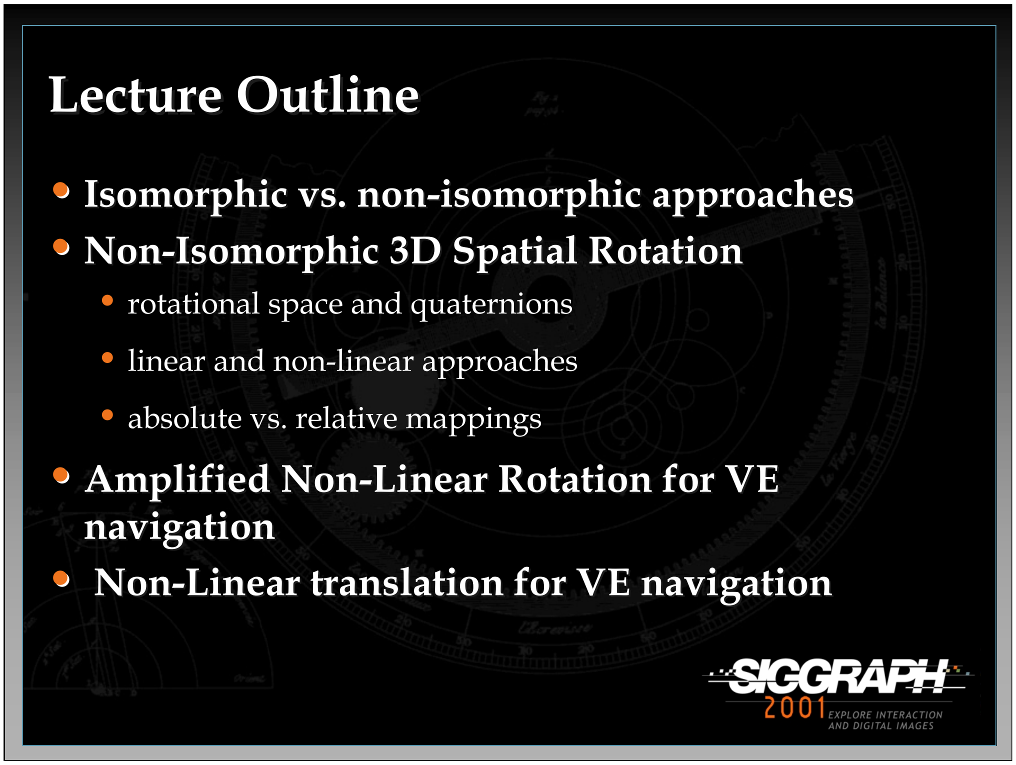
Joseph J LaViola Jr
Ph.D Candidate
Brown University,
Department of Computer Science
Providence, Rhode Island

Lead Consultant and Founder
JJL Interface Consultants, Inc.

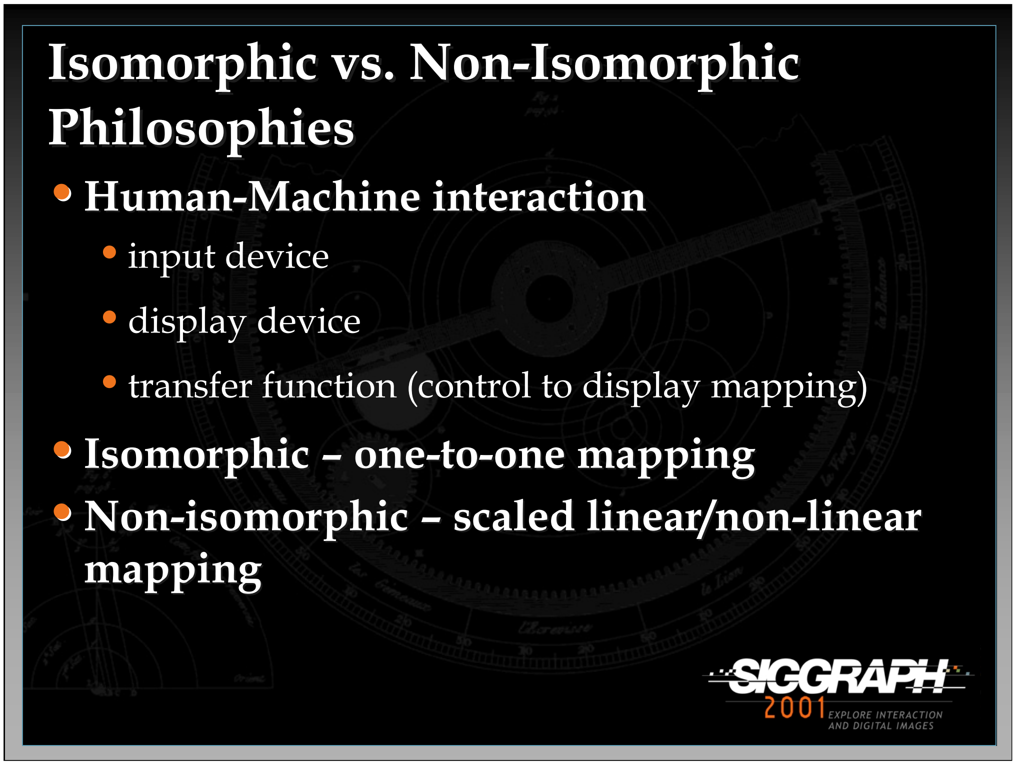
http: http://www.cs.brown.edu/people/jjl
email: jjl@cs.brown.edu



In this lecture, we are going to discuss non-isomorphic interaction techniques and present specific examples for navigation and object rotation in 3D user interfaces. We will also present some mathematical foundations for developing these techniques and examine their utility in 3D applications.



After describing the differences between isomorphic and non-isomorphic interaction philosophies, we will present a framework for developing non-isomorphic object rotation techniques, discuss design trade-offs, and present some experimental evaluations of a specific technique. Next, we describe a specific technique for amplifying a user's rotation in a surround screen virtual environment (SSVE). Finally, we present a technique for moving through a virtual environment using a leaning metaphor and non-linear translations.



There are three basic components of any human-machine interface:

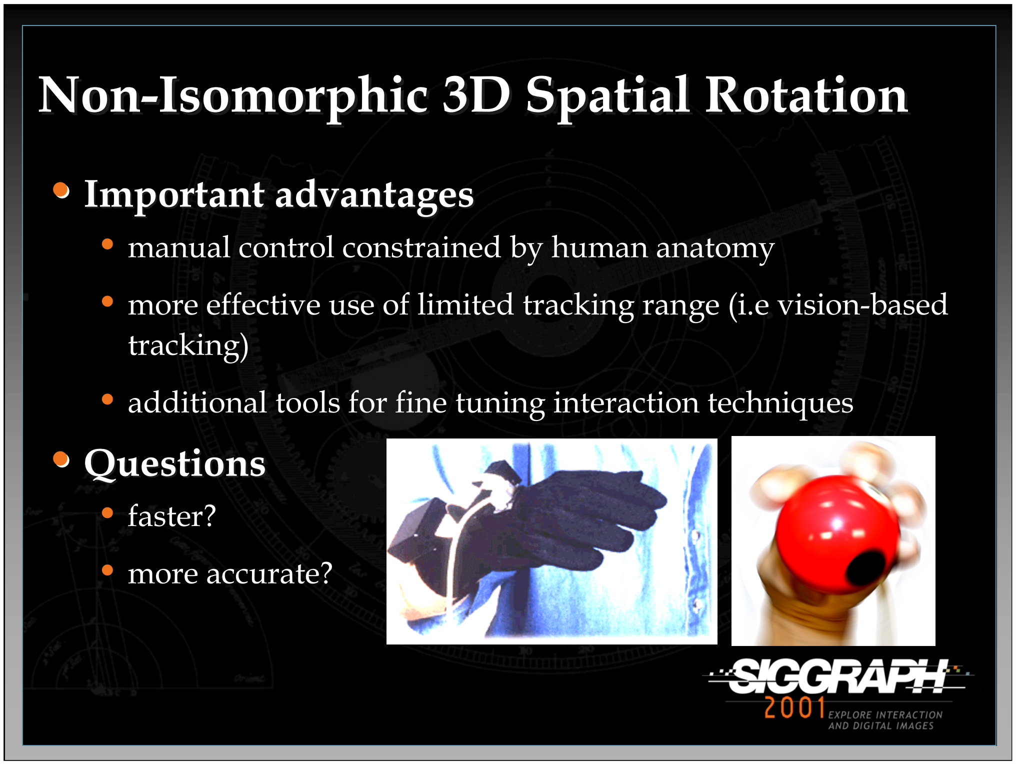
1) input devices which capture user actions

2) display devices which present the effect of these actions back to the user

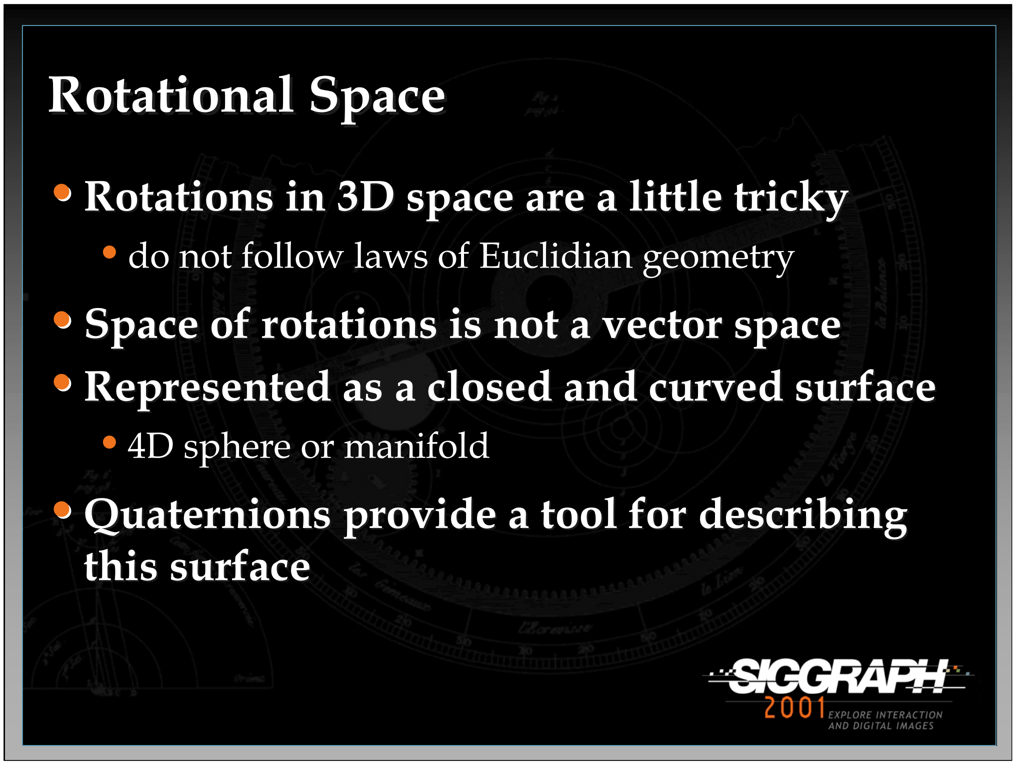
3) transfer functions (control to display mappings) which map device movements into movements of controlled display or interface elements.

These control to display mapping functions (CD mappings) significantly impact user performance and have been an active research area in 3D user interfaces where two competing philosophies have emerged. The isomorphic approach suggests a strict geometric isomorphism (i.e. one-to-one mapping) between physical and virtual objects on the grounds that it is the most natural and therefore the most useful. Although isomorphism is often more natural, it has number of shortcomings. First, isomorphic mappings are often impractical because of input device constraints such as limited tracking range. Second, isomorphism is often ineffective due to limits of human operators such as anatomical constraints.

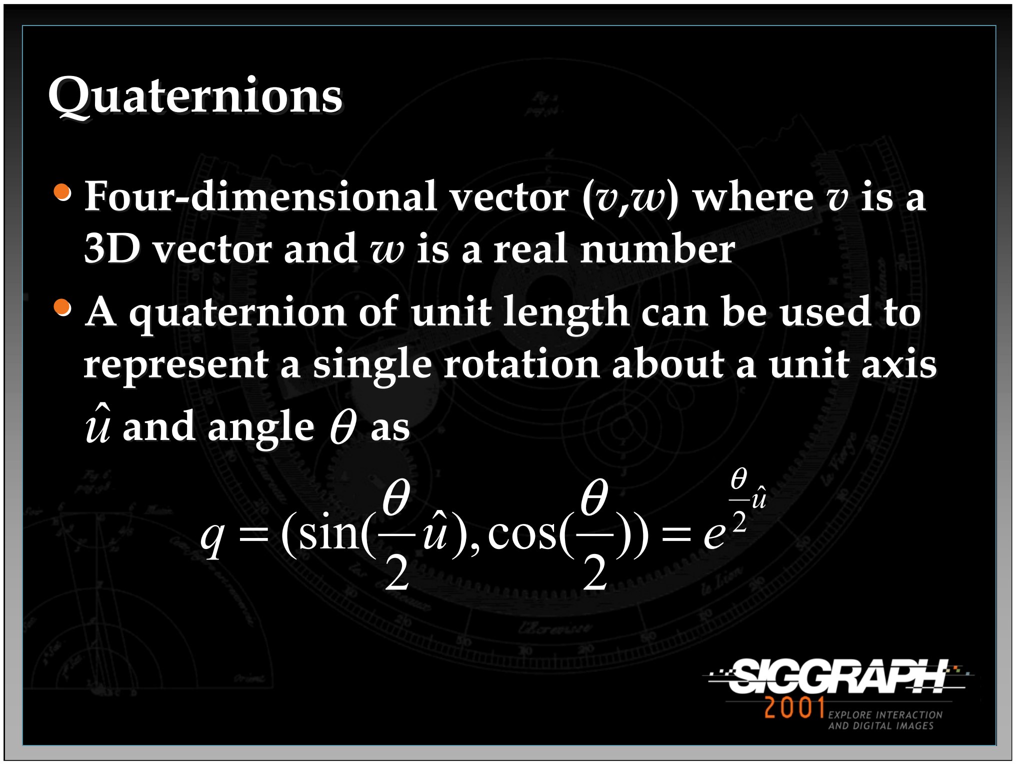
In contrast, the non-isomorphic approach suggests that manipulation mappings and techniques can deviate from strict realism, providing users with "magic" virtual tools. Instead of one-to-one isomorphic mappings, non-isomorphic techniques use scaled linear and non-linear mapping functions which, in effect, give users more power to manipulate virtual objects and navigate through 3D worlds.



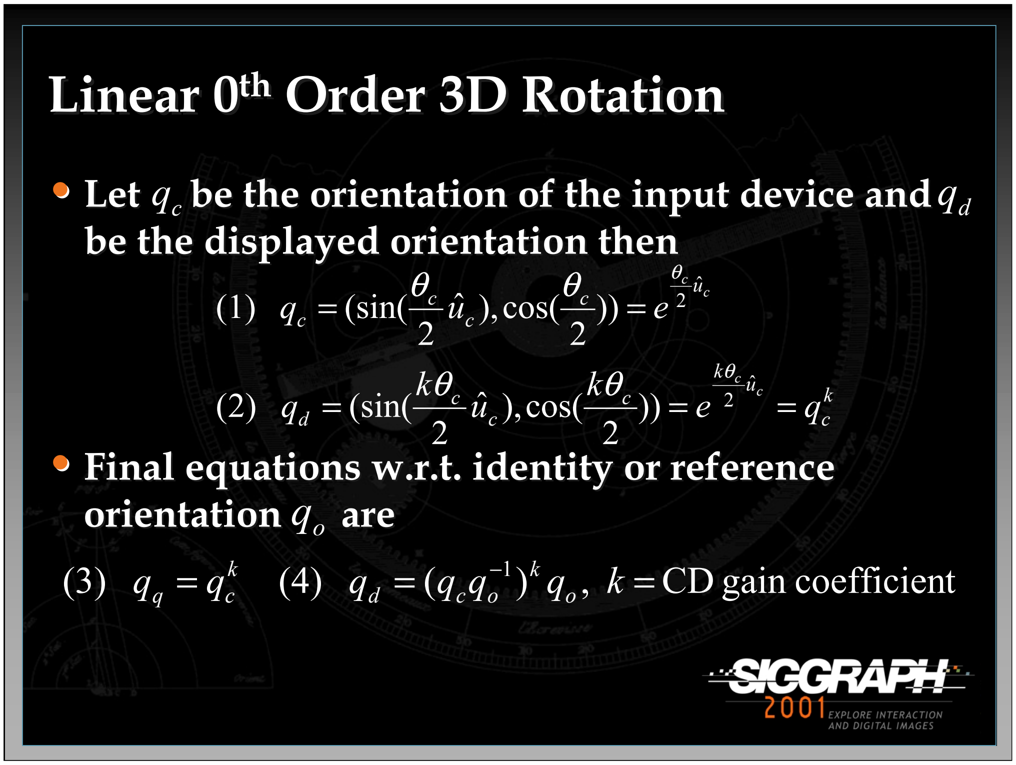
Non-isomorphic mappings are not an entirely new idea; they have been used for decades in a variety of everyday controls such as dials, wheels, and levers. In the context of 3D user interfaces, non-isomorphic techniques have been primarily designed for dealing with translation components in multiple DOF input. For 3D rotations, most techniques only use the simple one-to-one mappings between virtual objects and the 3D input device. What advantages can we gain from using a non-isomorphic approach to 3D spatial rotation? One of the biggest gains that the non-isomorphic approach provides is a method for handling manual control constraints. For example, with a isomorphic approach, rotating an object a full 360 degrees is extremely difficult in one motion since our arms have limited rotational movement about the elbow. A non-isomorphic mapping could eliminate this problem by mapping the user's, clearly limited, interaction space onto the full 360 degree virtual object space. Other advantages of the non-isomorphic approach include the ability to make better use of limited tracking range some input devices have such as a desktop VR camera tracking system. Another question we must consider is how these interaction approaches improve or hinder user performance; a question that we will begin to answer with some usability evaluations described later on in the lecture.



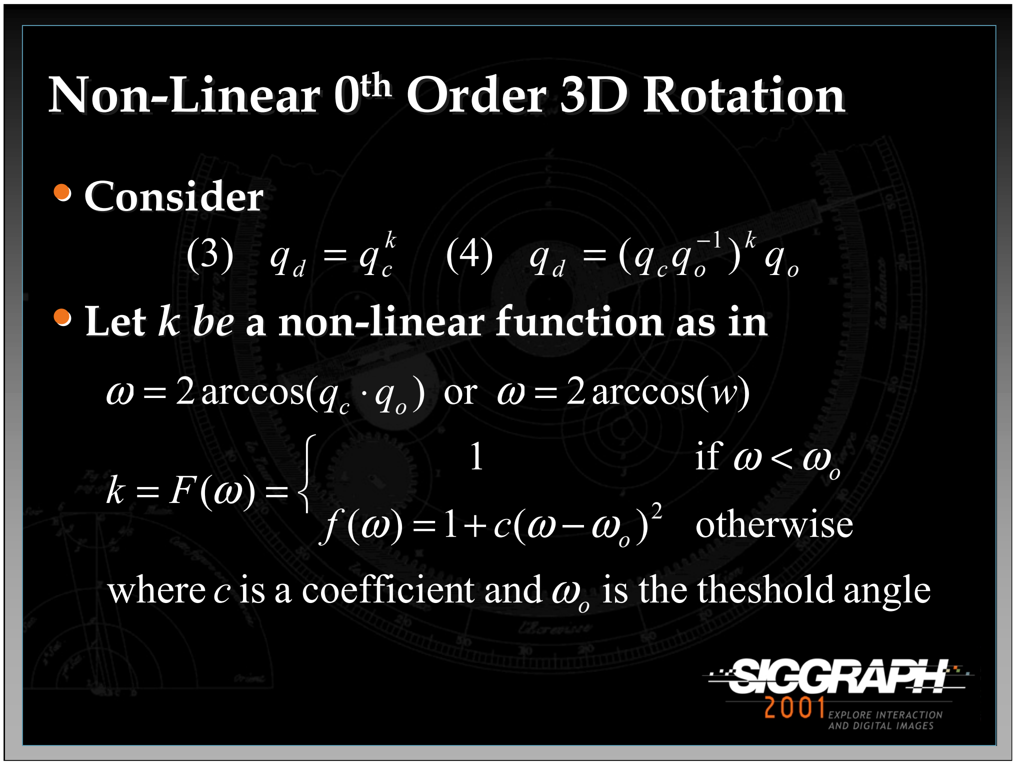
Rotations in 3D space are more confusing than they appear since they do not obey the laws of Euclidean geometry. For example, rotate an object in a certain direction and it will eventually come back to its initial orientation. The reason for this is that the space of rotations is not a vector space, but a closed and curved surface that can be represented as a 4D sphere. Quaternions provide us with a tool for describing rotations within the context of this 4D sphere and we will use them to develop a basic mathematical framework for designing non-isomorphic transfer functions.



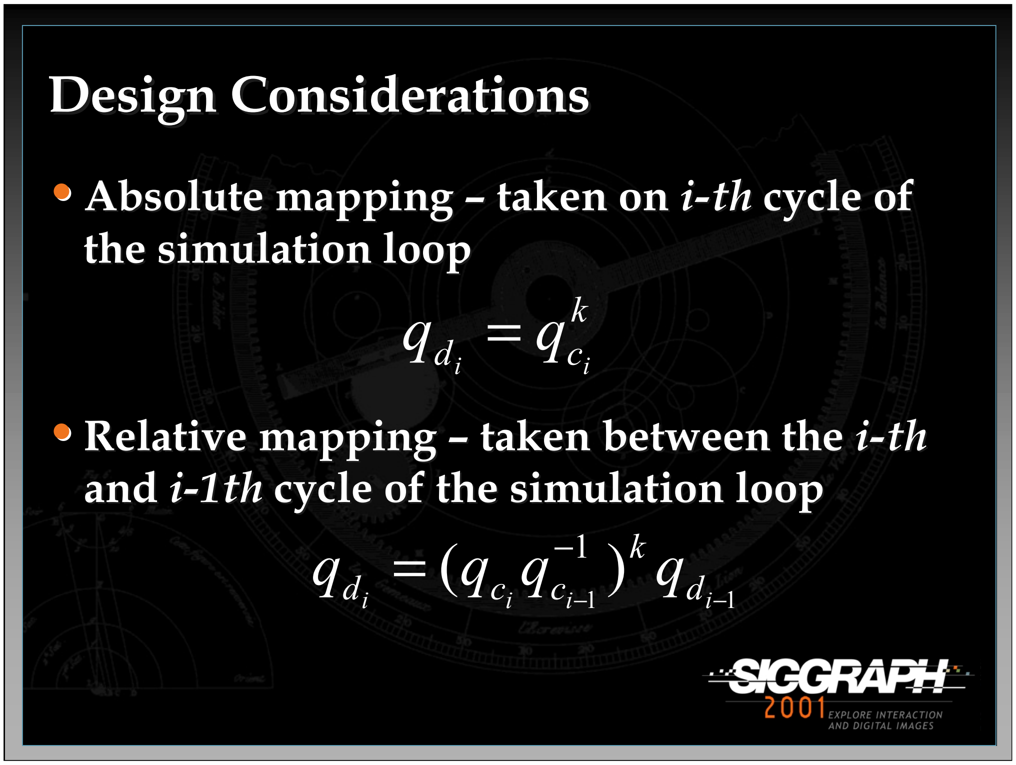
Quaternions provide an efficient mechanism to describe and operate 3D rotations. Quaternion *q* is a four-dimensional vector often represented as a pair (***v***,*w*) where ***v*** is a 3D vector and *w* is a real number. Given *q*, we can compute its length and inverse; given quaternion *q'*, we can compute their multiplication and dot product. The set of all unit quaternions forms a unit sphere in four dimensions and each point on the surface of the sphere represents an orientation of a rigid body. Euler showed that a combination of any number of rotations can be represented as a single rotation from a reference orientation. If no reference orientation is explicitly specified , a quaternion defines the rotation from the identity quaternion which is an analog to the origin in a vector space.



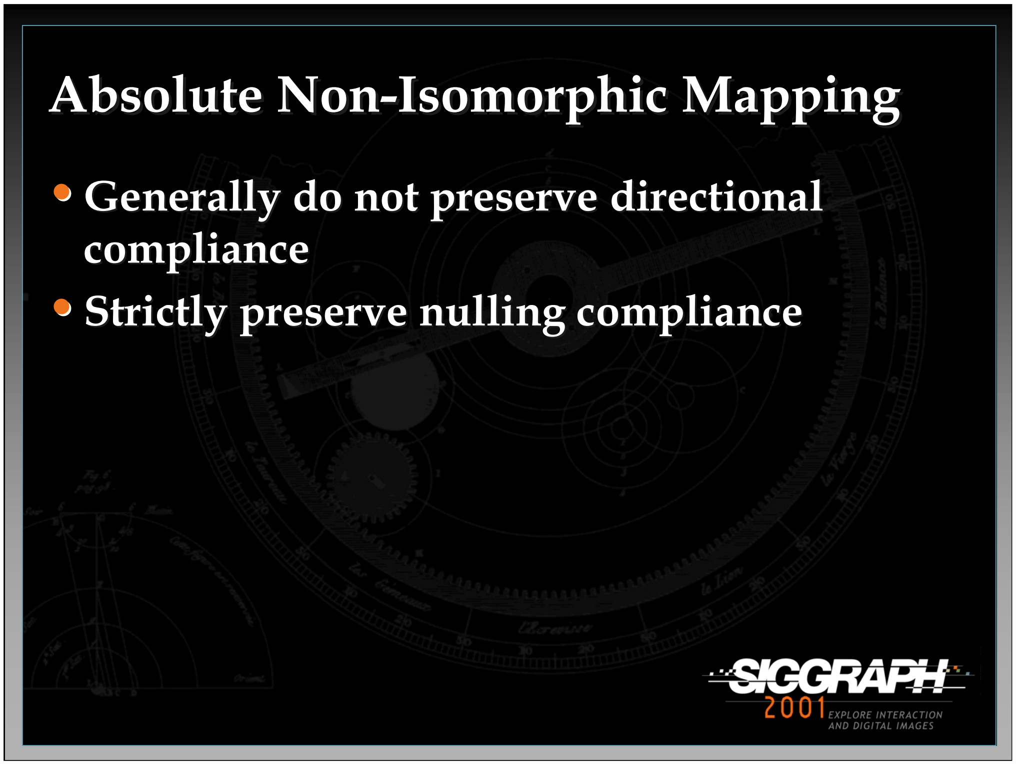
Equation 1 shows the quaternion, given an axis of rotation and angle, of a given multiple DOF input device. We can amplify the rotation angle by applying a constant coefficient *k* (the C-D gain) to it and leaving the rotation axis intact (equation 2). Therefore, the basic equation for a 0^{th} order linear CD gain is the power function shown as equation 3. Equation 3 specifies this rotation given the reference rotation is the identity quaternion. Sometimes we want to compute rotations given an explicitly specified reference rotation. This calculation can be done by connecting the reference quaternion and the input device quaternion, amplifying it, and combining it with the reference quaternion as shown in equation 4.



If we consider the linear 0th order rotation equations derived in the previous slide (equations 3 and 4) k is simply a constant C-D gain. We can use the same equations to create non-linear C-D mappings by letting k equal a non-linear function. In the example shown in the slide, F is a non-linear function based on the smallest angle between the reference quaternion and the input device's quaternion. If this angle is below a certain threshold the C-D gain has a constant ratio, otherwise the C-D gain grows in a non-linear fashion according to the distance between the reference and input device quaternion. Note that there are many other non-linear functions we could apply here and we will see another example later in the lecture.

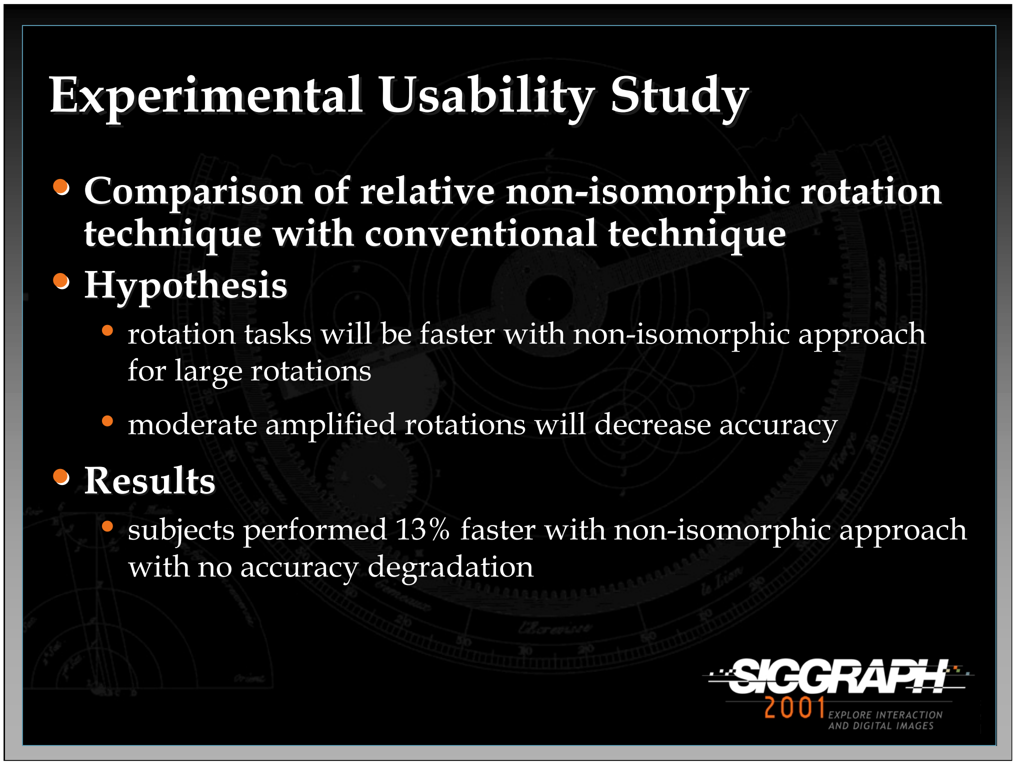


When using non-isomorphic spatial rotation techniques, we can use either an absolute or relative mapping scheme. With absolute mapping, we get the absolute orientation of the device, the absolute angular displacement relative to the initial, zero orientation, and use it on the *i-th* cycle of the simulation loop to compute the amplified, virtual object orientation. With relative mapping, we amplify only relative changes in the device orientation, i.e. we use the device's relative orientation between the *i-th* and *i-1th* cycle in the simulation loop to compute the virtual object rotation. Using one scheme or the other does make a difference. First, given the same rotation path about the device, these two mappings produce different rotation paths of the displayed object. Second, absolute and relative mappings are quite different from a usability point of view.



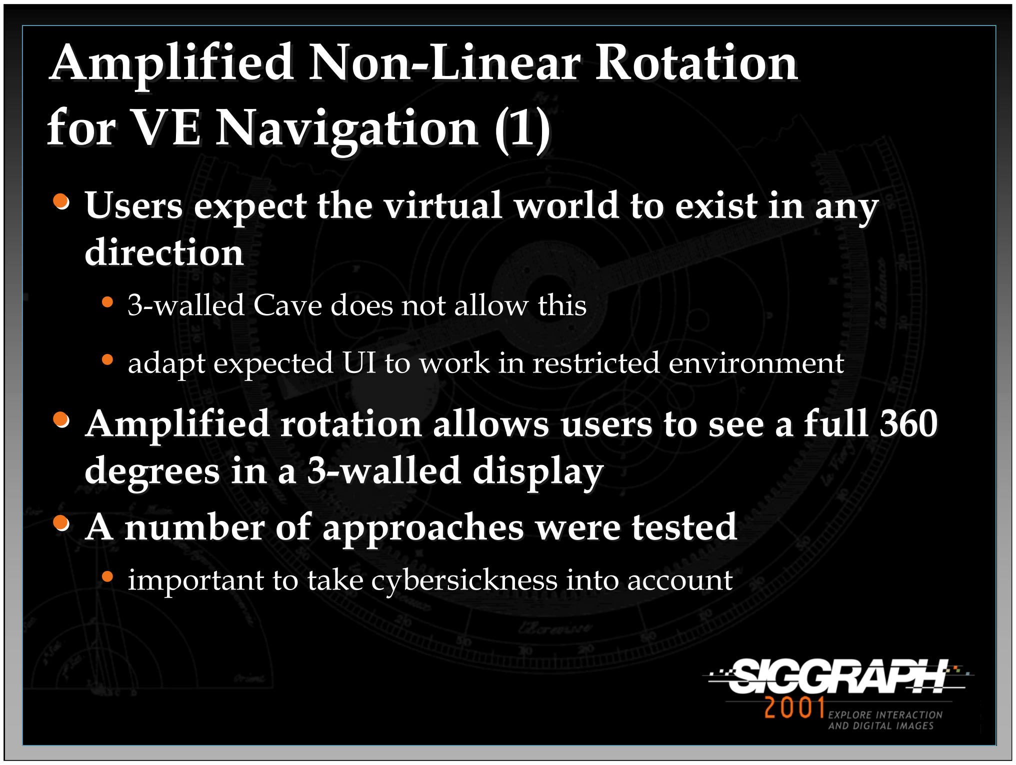
Absolute non-isomorphic mappings generally do not preserve directional compliance. What we mean by this statement is that virtual objects do not always rotate in the same direction as the device. On the other hand, absolute non-isomorphic mappings strictly preserve the nulling compliance, meaning that the input device always returns the virtual object to its initial orientation preserving a consistent correspondence between the origins of the coordinate systems in the physical and virtual spaces. How does this knowledge affect interface design? Absolute mappings have limited utility since users cannot consistently predict the response of the virtual object on the device rotations. However, these mappings can be useful when device rotations do not change the axis much as with viewpoint control using head rotations.

In contrast to absolute mappings, relative non-isomorphic mappings always maintain directional compliance and do not generally preserve nulling compliance. This means that relative non-isomorphic mappings can be very efficient in manual control tasks assuming the multiple DOF input device has the right form factor. The right form factor in this case is the device can be freely rotated in the fingers.

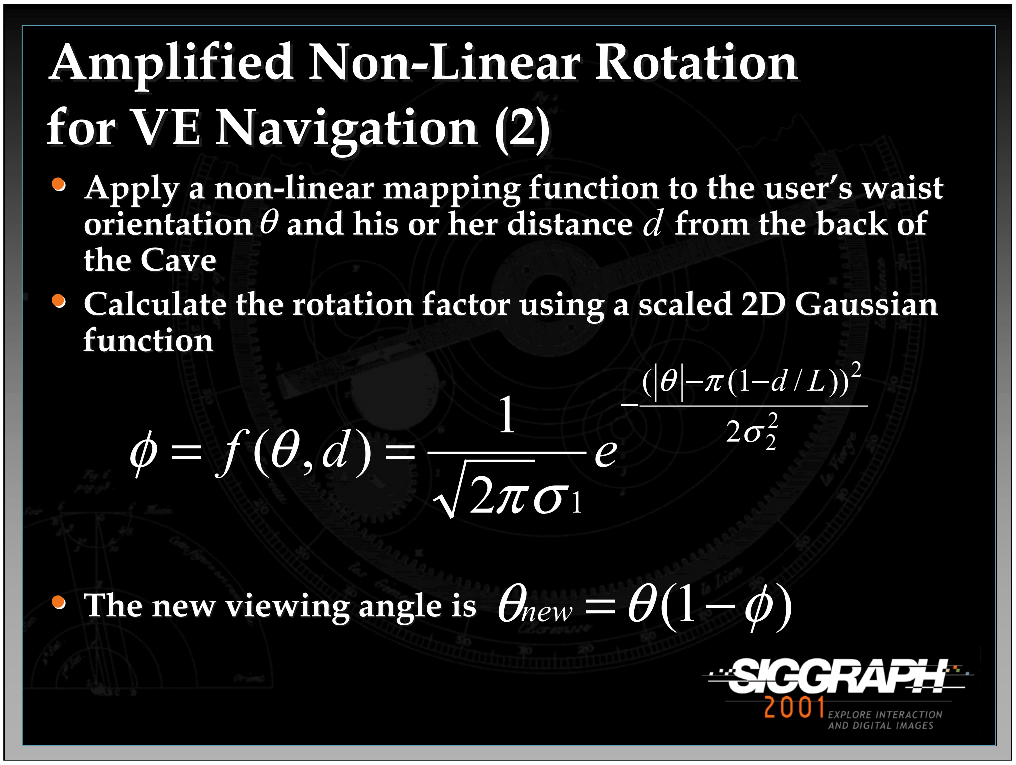


How effective are non-isomorphic rotation techniques in terms of speed and accuracy? Obviously, a complete series of usability evaluations would be required to determine the effectiveness of these techniques which is beyond the scope of this lecture. However, as an example, we can describe one such evaluation which compares a linear non-isomorphic rotation technique with a relative mapping versus a conventional one-to-one mapping scheme. There are two main hypotheses for this study. First, a relative amplification of multiple DOF input device rotations will allow users to perform a rotation task faster than a traditional approach assuming large range rotations are required. Second, non-isomorphic rotation techniques with moderate amplification will decrease rotational accuracy. Subjects in the study had to rotate a house model from a randomly generated initial orientation to a target orientation using both the isomorphic and non-isomorphic approaches. Results indicated that subject performed the rotation task 13% faster with the non-isomorphic approach with no accuracy degradation. These results show that in this instance, the non-isomorphic approach is the better one.

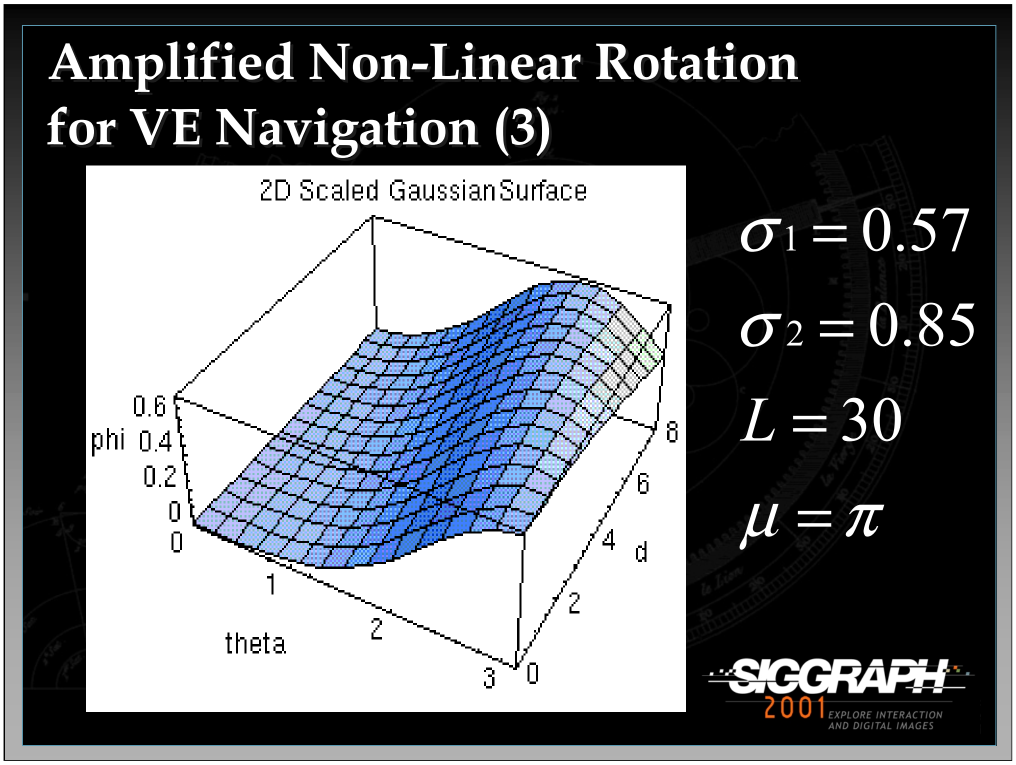
For more details on this experiment and on what we have discussed so far, refer to Poupyrev's paper, "Non-Isomorphic 3D Rotational Techniques", included in papers section of the course notes.



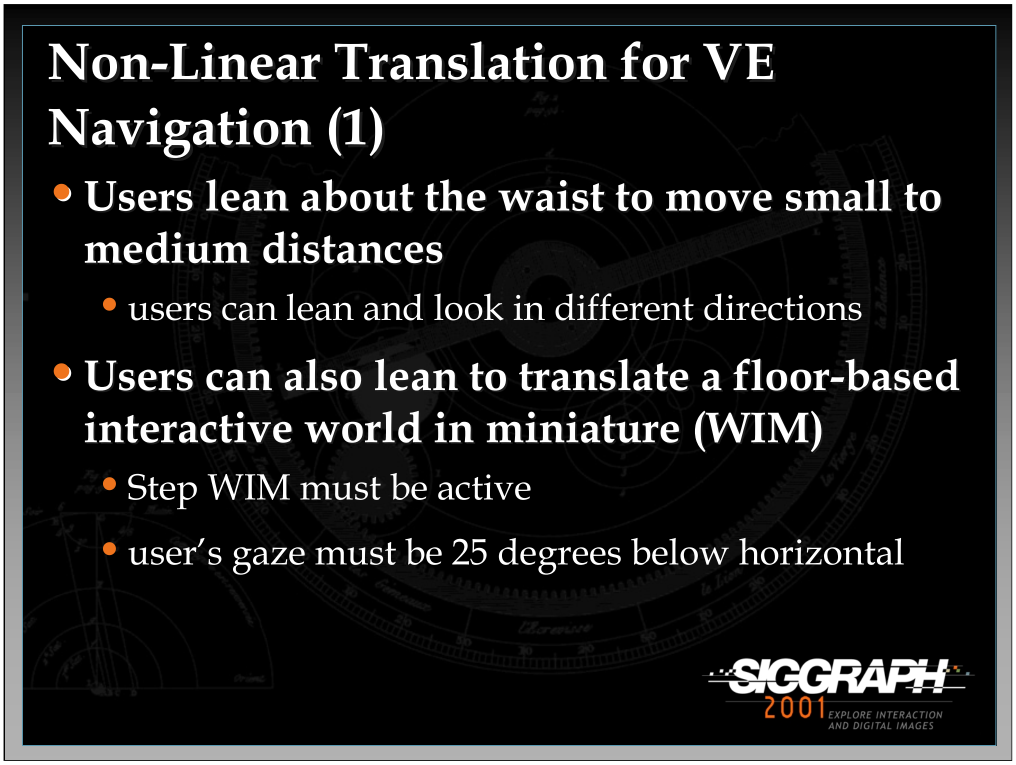
At this point in the lecture, we will describe two non-linear non-isomorphic interaction techniques for navigating through virtual environments. The first technique is an amplified non-linear rotation technique which lets user see a full 360 degrees in a semi-immersive display. In a fully immersive VE (such as an HMD), there is generally no need to provide any explicit control for rotating the virtual environment relative to the user, since the user can turn to face any direction. However, when dealing with a semi-immersive display such as a 3-walled Cave, the user only has approximately 270 degree view of the world and requires explicit controls to see what is behind him/her. A non-isomorphic approach can help to alleviate this problem and give the user the ability to see a full 360 degrees in a 3-walled display. A number of linear C-D mappings were initially tested with negative results. This lead to a subtle, non-linear scheme which provided smooth, less disturbing rotations.



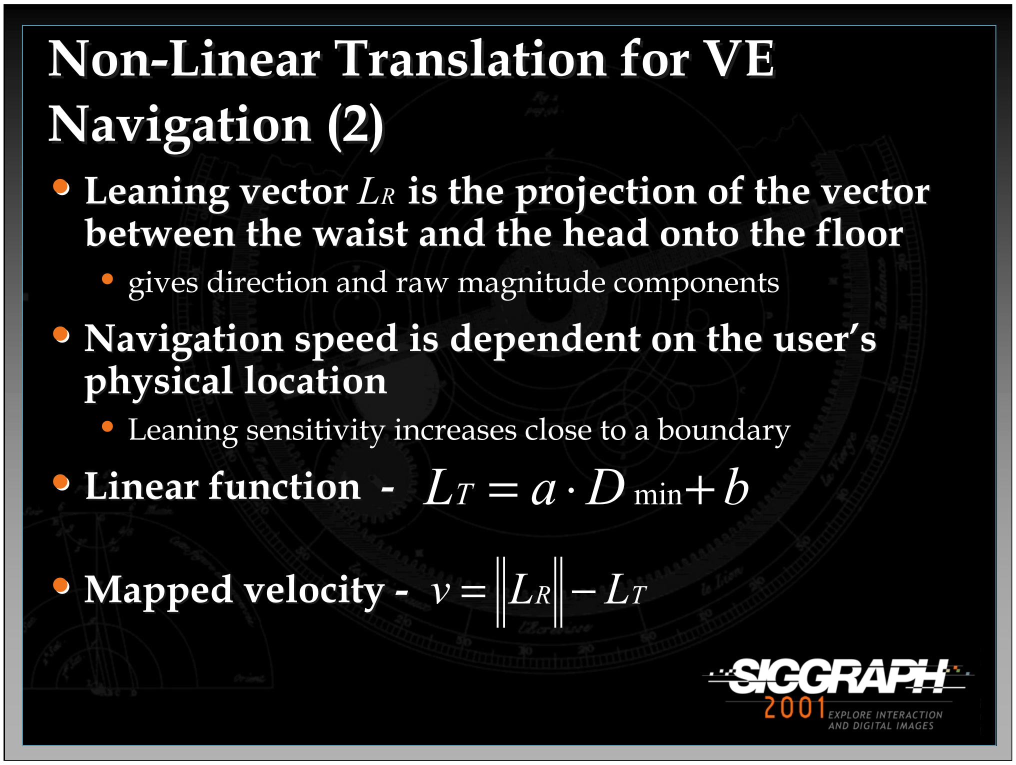
The particular non-linear mapping function is based on two parameters, the users waist orientation and his or her distance from the back (i.e. front wall) of the Cave. The distance parameter is used to determine how much screen real estate is available to the user. The closer to the front wall of the Cave, the more screen real estate the user has to work with. The mapping function is a scaled 2D Gaussian as shown in the slide where σ_1 is a height constant, σ_2 is a steepness constant, and L is a normalization constant. The new viewing angle is simply the amplified viewing angle subtracted from the old viewing angle.



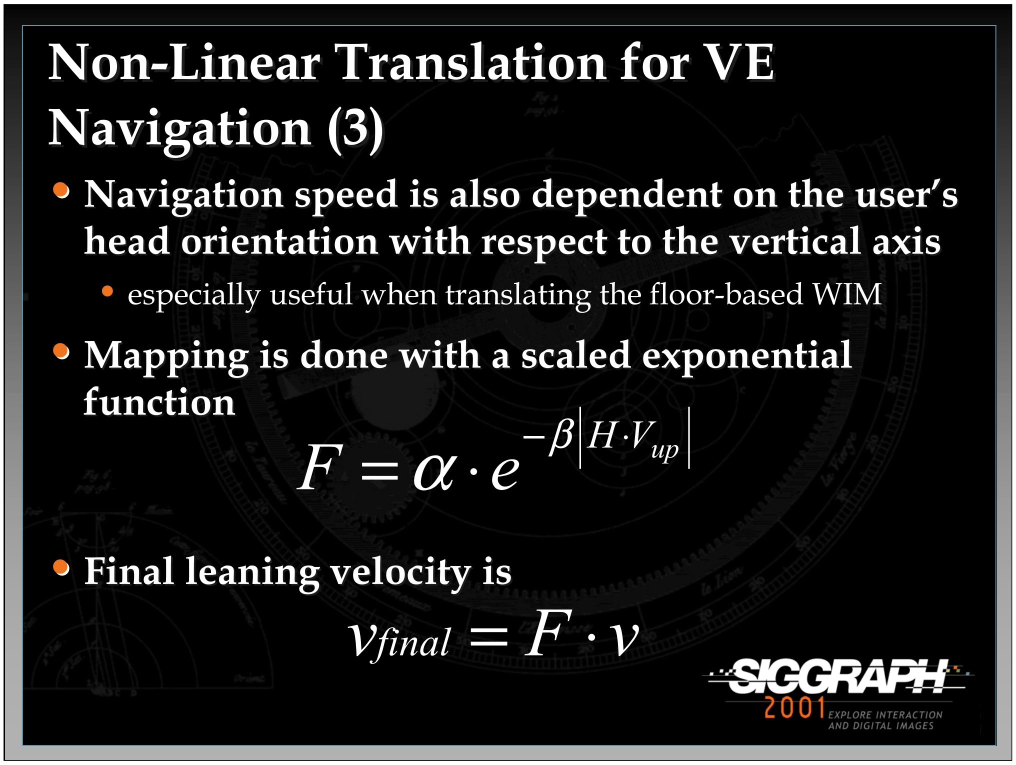
The figure in the slide shows a graphical depiction of what the scaled 2D Gaussian mapping function is doing given the constants shown on the right. As *d* increases the Gaussian bump shifts to the left indicating a higher degree of amplification is applied to the current viewing angle. Therefore, as the user moves farther away from the back of the Cave, virtual world rotation will increase since it will take more rotation amplification to see a full 360 degrees.



The second non-isomorphic technique we will discuss is a non-linear translation technique for navigating in virtual environments. The technique uses a leaning metaphor in which the user leans about the waist in order to navigate small to medium distances throughout the environment. In addition a user can translate a floor-based world in miniature (WIM) which allows for large scale navigation (more details on the floor-based world in miniature will be discussed in the Bringing 2D Interfaces to 3D Worlds lecture).

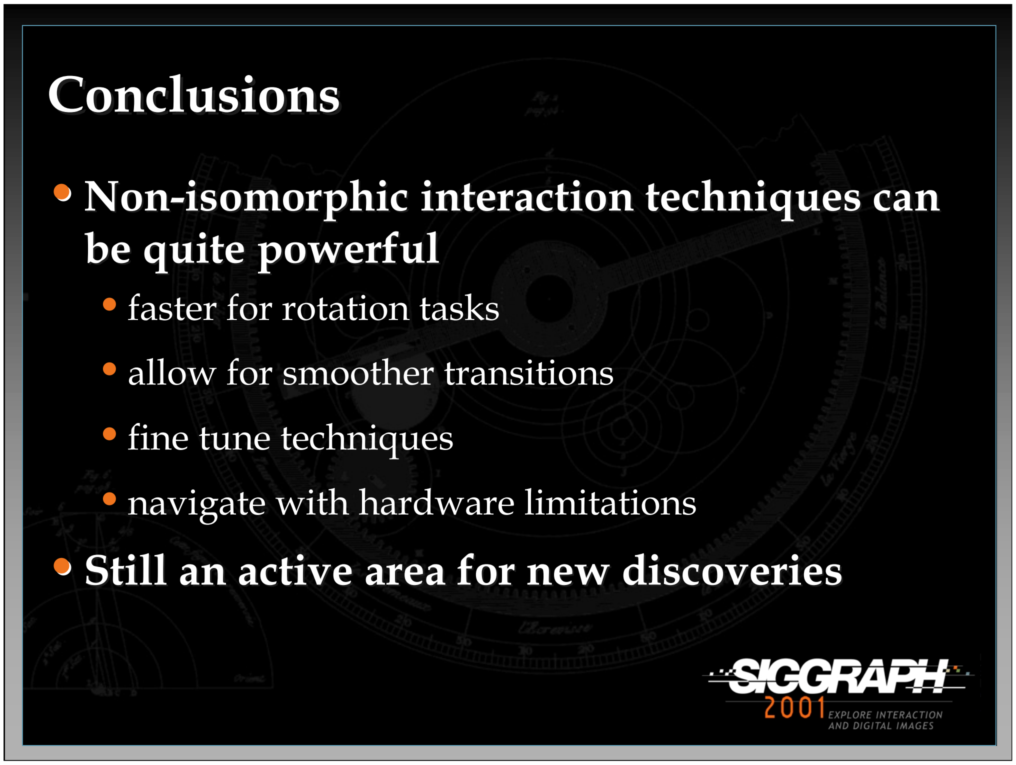


In order to calculate the user's translation at each cycle of the simulation loop, we first obtain the raw direction and magnitude components from the leaning vector which is the projection of the vector between the user's waist and head onto the floor. Second, we find the minimum amount of lean the user has to perform to obtain a translation, which is dependent on the minimum distance between the user's position and a physical boundary in the leaning direction. We choose this approach based on the observation that users need to lean less when they are closer to a physical boundary. Thus, leaning is most sensitive when users cannot physically move any farther in a given direction. The mapped velocity is then found using the magnitude of the leaning vector and the minimum leaning value.



Once we obtain the mapped velocity, a second non-linear mapping function is applied to the user's translation velocity based on the observation that users tend to focus their gaze on the place they wish to go even when this location is moving towards them. In cases where objects are generally lower than the user's head height, a correlation of the movement rate to the user's head orientation with respect to the vertical axis allows for smooth deceleration when the user reaches his/her destination. Therefore, a exponential mapping function (alpha is the maximum speed factor and beta defines the steepness of the exponential curve) is applied to the mapped velocity to obtain the final translation velocity. This double mapping scheme works especially well for navigating the floor-based world in miniature.

Note that more details on this technique and the amplified rotation for navigation can be found in LaViola's paper, "Hand-Free Multi-Scale Navigation in Virtual Environments", included in the papers section of the course notes.



Non-isomorphic interaction techniques can be quite powerful and are an important tool for interfaces designers to have at their disposal. They are still an active area of research and more work needs to be done to develop new mapping functions and evaluate how they affect user performance.

General References:

Poupyrev, I., S. Weghorst, and S. Fels. “Non-Ismporphic 3D Rotational Techniques”, ACM CHI’2000, 540-547, 2000.

LaViola, J., D. Acevedo, D. Keefe, and R. Zeleznik. “Hands-Free Multi-Scale Navigation in Virtual Environments”, In Proceedings of the 2001 Symposium on Interactive 3D Graphics, 9-15, 2001.