Personalized Nuance-Oriented Interaction in Virtual Environments

Chadwick A. Wingrave, Doug A. Bowman, Naren Ramakrishnan
Department of Computer Science, Virginia Tech
660 McBryde Hall (0106)
Blacksburg, VA 24061
cwingrav@vt.edu,{bowman,naren}@cs.vt.edu

Abstract

Virtual Environment (VE) user interfaces do not generally support personalized interaction or the ability to adapt to the user. Our research is developing nuance-oriented interfaces – interfaces that take advantage of subtle clues that users give in their actions. To that end, we have performed five experiments with the goal of recognizing and applying nuances for the task of selection. We found evidence that users adapt their behavior to the feedback given by the system, rather than using a preconceived mental model of the environment. In addition, users' behavior can be modeled by a reward function. Lastly, users interact by trading off exploration with exploitation. We propose a method of modeling this behavior using machine learning as future work.

1 Introduction

Consider the following scenario:

> *Brad, the architect, is wandering around a model of a building being planned. Brad holds his hand to his ear and says, "Dave, I'm in the hallway here and two conduits are overlapping, should I fix it?" "Yes, move the north-south conduit up" came a voice from nowhere. Brad reaches midway towards the pipes with his hand aligned with the north-south conduit, pinches his fingers together and his hand shoots-out and grabs the conduit. Moving his hand up a little and releasing his pinch, the pipe moves up and above the other conduit. Dave arrives shaking his head and says, "No, you are getting too close to the PA system, see those field lines there?" Dave reaches up with two hands, and begins to position both ends of the conduit at once, avoiding the problem.*

These two users are both performing similar tasks, but they prefer to do them in different ways. Current interfaces for virtual environments (VEs) do not support personalized interaction. So, to handle this, we need a new view of the situation. Our approach is called nuance-oriented interfaces.

A *nuance* is, "A repeatable action the user makes in their interactions in an environment, intentional or not, that are highly correlated with an intended action but not implicit in the interaction metaphor." [Wingrave01] The can be treated as a function that operates on parameters in the environment. In this paper we use

the nuance-oriented framework to investigate selection techniques under the assumption that users have a mental model of selection in VEs. To do this we need to discover the parameters affecting the user and parameters of existing selection techniques.

Our thesis is this, that by isolating a few parameters of the user's mental model of selection through experimentation, we can use those values to find solutions for other parameters and build an expanding view of interaction. We started with the task of selection, something common to nearly all interactive VEs.

2. Categories of Nuances

We have identified four categories of nuances. These include *object nuances* (nuances that arise from some affordance of the object) and *environmental nuances* (nuances that arise from an object existing in relation to the environment). *Refinable nuances* adjust boundaries for existing techniques such as correcting for constant errors in interaction techniques. There are also *supplementary nuances* that are not intuitive but exist and can be mined from interaction data. These will take time and research to discover.

2.1 Object Nuances

Object nuances are better described as affordances that the object possesses. An affordance is, "the perceived and actual properties of the thing, primarily those fundamental properties that determine just how the thing could possibly be used" [Norman88]. These object nuances will change the manner in which the user chooses to interact with the object.

For example, we might intuitively want to grab a stone column and a glass of water in the same manner because of their similar cylindrical shape (see Figure 1). Objects of this shape may make the user feel the need to select it with the orientation of the hand matching the orientation of the object. So, a VE that had the user select between various pots on a stove with handles at various angles could use the

Figure 1. Object nuances of similar shaped objects such as a cup (above) and a column (below) afford the same selection principle.

orientation of the hand as a nuance in determining the correct pot. If the user were trying to select a glass on the stove, the orientation of the hand would be key to differentiating between the pots and the glass because the hand would be at a 90-degree rotational difference to the panhandles. These both work off of the same nuance, "cylinders tend to be selected in the same orientation of the hand."

If we were to select a flower and a golf tee however, we would grab each in a different manner even though they are both cylindrical. In the case of the flower, we would grab along the stem and in the case of the tee, at the tee's head. This is because the flower offers affordances that say the petals are fragile so grab at the base while the tee's affordances state that the head would offer the best grip. These two nuances are different in that they are based upon the properties of what the objects are and not the shape of the object.

2.2 Environmental Nuances

An example of an environmental nuance can be seen in the ray-casting selection technique [Mine95]. In the selection of objects that are close together, users should be able to produce small nuances and the VE should interpret the extra data. For example (Figure 2), the user might try to err in the direction away from object 1 when they are trying to select object 2. This could help differentiate between objects 1 and 2. In another case (Figure 3), there is an object 3 that appears to be fairly similar in distance from the ray as object 2. Since we know that the user consistently errs to distinguish between two close objects, we would assume that the intended object is object 2 and not 3.

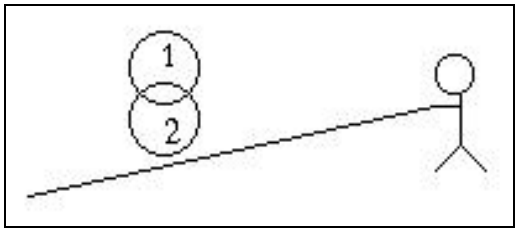

Figure 2: Using Ray Casting to select between objects 1 and 2, the user errs in the direction away from the object they do not want to select.

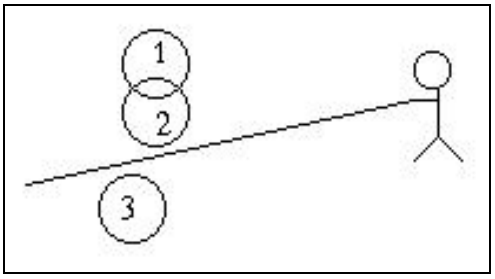

Figure 3: Because the user errs away from objects they don't want to select, we make the assumption that object 2 is the desired object even though objects 2 and 3 appear similar distances away from the user's Ray Casting selection.

Another example would be grabbing the same stone column as mentioned before but at different distances; try this yourself. Imagine the stone column a few inches in front of your nose and try to grab it. Then, imagine it further out and grab it. If the object appeared to be in front of your nose, you tend to come at it from the side. If you imagine grabbing it at a distance, you tend to come at it with your wrist pointing away from you. This is a simple nuances but helpful if the user were standing on the steps of the Athenian Acropolis and the interface had to decide if the user was attempting to grab the inner or outer columns. Such an interface created by a designer would be daunting and costly, if possible at all, to

create. Additionally, it would most likely be limited in scope to selecting columns of Greek temples or glasses on a table.

2.3 Refinable Nuances

Refinable nuances can be used to alter the existing behavior of an interface to make it more usable. This can be in order to correct for errors in input data or the user. It can also be used to increase the precision or increase the usability of boundary conditions for environments that require it.

It has been shown that users err more in depth than in the horizontal and vertical dimensions [Werkhoven98]. A refinable nuance would reduce the emphasis placed on accuracy in the depth dimension or maybe even model how the user errors in that dimension. In this way, if the user is trying to select an object using arm extension, the refinable nuance will widen the acceptable depth error.

Also, ray casting implementations use a ray extending from the tip of the user's finger and have the user align that ray with the object they wish to select. A refinable nuance would discover that a user prefers the ray to extend at a slightly adjusted angle to reduce their stress and increase accuracy. Therefore, a more general nuance-oriented system, like the one we propose, is necessary.

2.4 Supplemental Nuances

Not based upon object affordances or environmental effects, these nuances can better be grouped into two subclasses of strategies and discoverables. These nuances take advantage of user mental models. Also, these could be crafted from nonintuitive means to create expert interfaces or simply to extend an existing interface.

Strategies are means in which users tend to work within an interaction framework to increase the usability of it. Strategies can be small improvements such as locking their wrists to improve pointing accuracy, up to large improvements such as rotating their head to move occluding objects out of their way [Bowman ??]. When users have the option of signaling a selection by pinching either their index and thumb or their middle and thumb during ray casting selection, they eventually start to use their middle and thumb. This allows the index finger to stay pointing and the middle finger and thumb to move closer together so when a pinch occurs, moving the finger a smaller distance upsets the tracker less. Larger strategies are learned too. For example, many users use body-centered references such as pointing to indicate objects or locations they want to remember later [Bowman99]. Then, as they move through the environment, they keep to the previous position to serve as a physical mnemonic.

Discoverables are methods of interaction that users develop on their own inside an interaction framework. Users tend to develop methods dealing with difficult to handle situations within the existing interface in ways that were not originally intended. In our implementation of ray casting, the object selected when the user pinches their fingers is the object closest to the ray no matter what the angle. Once users realized this, some of them would select the object by pointing hastily at an angle that is not necessarily accurate but easily seen to be closer to the object they wish to select than any other object in the environment. The selection was much faster and required less thought because of this strategy.

3 Personalizing Selection Techniques

Selection tasks are what the user does when singling out a specific object or point in a VE. Most metaphors for this can be broken down into the following categories; ray casting, occlusion and arm extension. Ray casting is where a ray, going to infinity, is projected from the user's finger and object's it intersects can be selected. Usually a button is pressed at that point to verify that the user truly did intend to select that object. Its feedback is the ray coming from the fingertip and can be implemented such that objects are highlighted when the ray falls on top of them. Occlusion [Pierce97] is similar to ray casting in that a ray is draw and falls on an object but that ray originates from the user's eye and continues through a point, usually the fingertip, to infinity. Arm extension has several implementations that vary only slightly in each case. The concept is that of the hand being the point of selection and it moving in the environment according to some function of the distance it is away from the user. A simple case would be a one-to-one linear scaling so if the user moves their tracked hand one foot forward, the virtual hand moves one foot forward. The linear function can also be scaled for example such that one real inch forward is ten virtual inches, one foot is ten virtual feet, etc. This helps in selecting objects at a distance. Another implementation, the Go-Go [Poupyrev96] technique uses a linear scaled hand up close but as the hand extends, it scales exponentially. This increases the range of the hand. Each of these tasks has their advantages but no single selection metaphor is optimal in all cases. It has been shown that pointing is a more efficient metaphor for distant objects but arm extension metaphors are more optimal for near objects [Poupyrev97].

The equipment used for these phases was an SGI Indigo 2 with Max Impact graphics with the user inside a Virtual Reality V8 Head Mounted Display (HMD). They had their hands and head tracked using a Polhemus 3 Space Fastrak™ magnetic tracker and finger pinches were recorded using Fakespace PinchGloves™. A selection was considered to have taken place when the user pinched either their index and thumb or middle and thumb fingers together.

All eight of the users of these phases were taken from a graduate level course on virtual environments though not all were computer science students. All had some familiarity and interest in the field but not necessarily experience. There

were five males and three females between the ages of 24 and 54. Their compensation was receiving extra credit in the course.

3.1 Plan

Our plan was to develop a simple nuance model of selection in VEs using existing selection techniques. To that end, we started off by personalizing each selection technique for a user in an environment, the trick being isolating the technique from other nuances that affect that personalization. We did this in Phase 1. From there, we determined what technique users prefer based upon where they are selecting in the VE in effect creating a 3D preference map. The outcome we intended was a set of parameters explaining users and the selection techniques. This was phase 2. Using the results from both phases, we can start applying nuances to the parameters as the beginning of our nuance-oriented environment.

3.2 Phase 1: Tuning Selection Techniques

We performed phase 1 two different times in an attempt to tune the selection techniques. We will call them Phase 1 A and Phase 1 B.

Phase 1 A: Personalizing Through Feedback Removal

Phase 1 A to discover refinable nuances for three VE selection techniques: arm extension, ray casting and occlusion. Users were told how each technique worked in wording vague enough to not guide their actions but informative enough to let them know how it works and is implemented. The concept is that users have a mental model of how they wish to interact with the environment given a selection technique description and if we discover that underlying model, then we can use that knowledge to predict their actions, reducing the Gulf of Execution [Norman90].

Environment:

The environment (see Figure 4) in all three experiments had the user standing on a platform overlooking a floor with one orange sphere that they were told to select using the selection technique that was currently being tested. To account for a lack of depth cues, the users were told the sphere was the same size throughout the experiment and that the floor was a grid of one-meter squares. There was also a shadow, properly scaled for depth and approximately scaled for height, placed below the sphere on the ground. The sphere was moved through 30 different locations in each environment with the first three being the sphere at its furthest distance, middle distance and closest

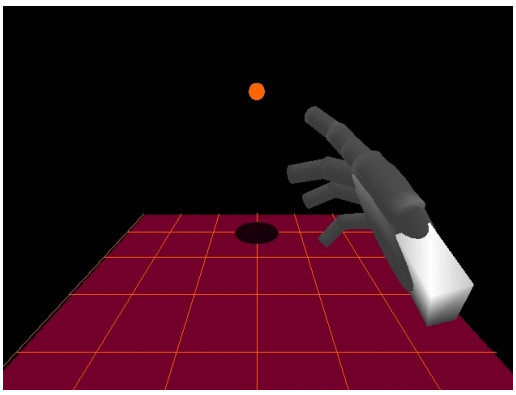

Figure 4. Phase 1 A with the ball at a distant position. Notice the shadow of the sphere and the gridded floor.

distance to the user. The other 27 locations of the sphere existed in a 3x3x3 array of near, mid and far, low, level and high and left, center and right. The sphere was randomly moved through each location during an experiment.

Isolating Parameters

Our difficulty was in trying to make an interface in which users would act naturally and not adapt to. To this end, we attempted to remove all forms of feedback from the environment relevant to each selection technique. For this reason we did not implement ray casting with a ray extending from the user's finger or even a hand for arm extension. We then assumed that since the user was operating according to their definition of optimality, and we knew their goal to be the selection of the orange sphere, then each time they conclude a selection with a pinch they were correct in their selection since they are operating by their own definition of optimality. The side effect is that users are able to cycle quickly through selections so we added a three second pause between selections. An audible sound was given when the orange sphere reappeared.

Data Collected:

The data from this phase was the position of the user head and hands as well as the position of the sphere at the time of pinch. This data would then be used to discover if users made consistent errors based upon the location of the intended object of selection and the selection technique. Clustering techniques such as k-means were used on the data to group user actions.

Results:

With users free from feedback of the environment, we expected them to revert to their most natural form of interaction built off of natural or proprioceptive intuition. What occurred was an amazing display of adaptation on the part of the user, completely unnatural and inefficient but incredibly adept at making use of the scarce feedback that was left in the system. As an example, one subject spent the entire occlusion technique making selections with their palm facing out. This is a very uncomfortable position even for short periods of time and especially for objects elevated in the environment. Most importantly, the palm being turned out **completely** occludes the environment thus reducing accuracy.

There were some supplemental nuance strategies observed. One was that users tended to rotate their wrists out when selecting objects that were high or close to them. Another was that since the objects being selected were spherical and since the palm of the hand was

Figure 5. The "Heisenberg" effect of making a selection induces errors.

circular, the users attempted to position their hand such that the sphere was perfectly occluded by the palm of the hand achieving an eclipse. Another strategy was to switch to pinching the middle finger to make a selection thus avoiding what we have termed the "Heisenberg effect" (see Figure 5) of selection [BowmanHCI01]. This is when performing the action to signal an event induces errors.

Users of arm extension were found to not have a concept of depth. We expected users to scale the extension of their arm to the objects being selected but found that users only divided space into "far" and "near" with far being a fully extended arm and near being a half-way extension. We were hoping that since objects appeared at different depths over time, the user would learn this and scale accordingly. The lack of other objects in the scene at the time of pinch could have made indicating depth unnecessary so the result may be inconclusive. Users found it very unsatisfactory not being able to see their hand. Also, users were very slow and hesitant compared to the other two selection techniques. Because of this, the usefulness of arm extension was questioned since it requires specifying a parameter that users do not seem to have a good grasp of except through proprioception [Mine97].

Occlusion selection contained the most interesting results. Since we did not remove the hand from the scene, users had almost all the feedback of the full implementation. Because of this, we expected nearly optimal usage. The users however choose unusual points on the hand as the occluding points. The two most common were actually the palm of the hand and the knuckle where the thumb meets the hand (see Figure 6). The palm of the hand occlusion technique occludes most of the scene making its accuracy very low but could possibly be what users interpreted the selection technique to be. The thumb knuckle technique is inaccurate and again occluding. It does however leave the hand in a natural and thus non-fatiguing state.

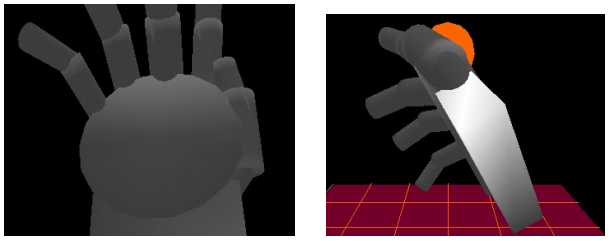

Figure 6. Two occlusion selections used most commonly in phase 1 implementation A. Left is the palm occlude and right is the thumb knuckle occlude. Both are inaccurate and highly occlude the scene but for some reason users converged to them.

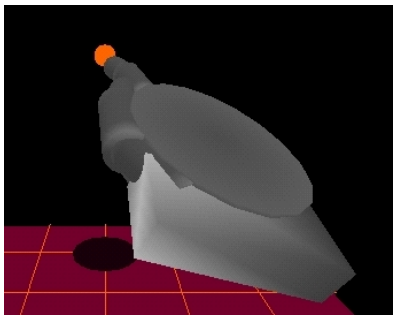

Figure 7. All but one user considered ray casting to be like an occlusion technique.

For ray casting selection, only one user did true ray casting. All the other users occluded the object with the tip of their finger and

considered that pointing at the object (see Figure 7). This completely voided the concept of "shooting-from-the-hip" to reduce fatigue.

Our original intent was to build personalized selection techniques for the users. After reviewing the results, it was not considered possible to use the data since the users were so inefficient with their interaction in virtual environments without feedback to guide them. Clustering was performed to see if trends existed in user data but the trends that were observed were themselves so inefficient that they were not used either.

What We Learned

The results lead to the following conclusion:

> *Users largely do not have a mental model of the environment but a mental model of how to respond to feedback the environments affords.*

Stated another way, users attempt to align their actions with feedback and not their senses. The effect of user experience with VEs may play an important role in this conclusion.

Phase 1 B: Personalizing by Pressure to Improve

If users do not have a mental model of an environment but of responding to the feedback it produces, then the parameters we should be personalizing are those of the feedback. So, how do we go about dealing with the problem of the adaptability of the user since even a poorly tuned interface they will consider fair and stop their optimization? To do this, we need some sort of pressure to improve as simply asking the users to do so is not enough. Another problem was making the system friendly and usable such that users will not mind staying in the environment long enough to personalize the system to their needs. There was also the notion of getting users to search for alternative possibilities of selection within a technique and not just a tuning of the current one. Lastly, what types of feedback should the selection techniques have and what parameters should be tunable?

To provide pressure to improve, we framed the selections into trials of selections where the user was told to select as quickly and accurately as possible. At the end of each trial, a qualitative ranking was returned to the user in the hope that the competitiveness of the user would make them want to achieve better and better rankings. To provide pressure to search, we asked the users to use several configurations of the selection technique with each configuration using different values for its set of parameters. Included in the configurations were the typical methods of selecting from Phase 1 A to see how they compared against other configurations now that feedback was enabled.

To make the environment user-friendly and non-threatening, we did several things. The first thing we did was have the researcher direct the user through the

experiment answering questions along the way. We also allowed the user to work at their own pace and change the predefined paths of the experiment by telling the researcher that they were going to do another trial with the current configuration or go back to a previously tried configuration. We then implemented the TULIP menuing system [BowmanIEEEVR01] because it has low fatigue, allows the occluding hand to be removed from the scene easily and is fairly fast. These properties of TULIP outweigh its non-intuitive nature and with the researcher able to answer questions, users quickly understood its use. Lastly, personalization of a selection technique was done by selecting a parameter of that technique to tune using TULIP and then rotating the left hand to change the value.

Environment:

The environment is a 3x3x3 array of 27 light blue cubes placed in front of the user. The head and hands are tracked and the user is wearing PinchGloves™. In the occlusion selection environment, there is a bullseye on the hand and in the ray casting environment there is a ray extending from the fingertip. The left hand is labeled with the TULIP menuing system at all times and its submenus are displayed on the right hand when the choices need to be displayed. The three menu types are: "Configure", "Trials" and "Personalize". The "Configure" menu has seven configurations for occlusion selection and six for ray casting. The "Trials" menu allows the user to select a trial of 2, 4, 10 and 27 selections as well as stop the current trial. The options of the "Personalize" menu will be explained. The environment displays text to the user such as the ranking they receive for each trial, if a selection was correct or not and when trials start.

There is one environment for each selection technique. In each environment, the user is asked to do at least one trial of 10 selections for each configuration. They are then asked to rate that configuration on a scale of 1 to 5. After going through all the configurations, they are introduced to the methods of personalizing the interface. The researcher encourages them to do several small trials with each personalized technique and then at least two full trials (27 selections) with their final settings.

Selection Techniques: Their Feedback and Tunable Attributes

There were two selection techniques in Phase 1 B; ray casting and occlusion selection. Arm extension was dropped from further phases because it is not good for selecting at a distance due to the need to specify a distance-from-user parameter and also because users in the first study only had the concept of selecting "near" and "far". Again, all selection tasks are triggered by either a middle and thumb or index and thumb pinch.

The passive feedback for ray casting is the ray extending from the fingertip. This guides users in their selection as when the ray passes over objects, it is easy to see where the ray is in relation to it. Active feedback was also added such that when the user's ray gets close to an object, the ray snaps to that object changing the color of the ray and object. The properties that were tunable with this

implementation were the yaw and pitch values of how the ray extends off the fingertip with yaw being the angle across the fingers and pitch being the direction the fingers move when the hand closes. When the user tunes the values by rotating their left hand, the angles change immediately so they have an immediate feedback as to what the newly tuned position of the ray is. The user also has control over the snap-to angle and when they are tuning it, a cone representing the snap-to angle is drawn.

In occlusion selection, a user-facing, passive feedback, bullseye was attached to the hand representing the point where the ray from the eye passes through and extends into the environment. Additionally, there was an active feedback snap-to angle but instead of a ray snapping-to, the bullseye snapped-to and changed color. The tunable properties were the x (across fingertips), y (along fingers) and z (out from palm) positions of the bullseye in relation to the hand as well as the snap-to angle.

The configurations for each selection technique are as follows:

Ray Casting Selection
Config 1: The ray extends straight from the fingertip with a 10-degree snap-to angle.
Config 2: The ray has negative pitch with a 10-degree snap-to angle.
Config 3: The ray has positive pitch with a 10-degree snap-to angle.
Config 4: The ray is straight but with a 40-degree snap-to angle.
Config 5: The ray is straight but with a 3-degree snap-to angle.
Config 6: The ray has positive pitch and heading with a 10-degree snap-to angle.

Occlusion Selection
Config 1: The bullseye is on the index finger and has a 10-degree snap-to angle.
Config 2: The bullseye is on the middle finger and has a 10-degree snap-to angle.
Config 3: The bullseye is on the thumb's knuckle with a 10-degree snap-to angle. This was a configuration that was used by almost every user in the first implementation.
Config 4: The bullseye is on the palm of the hand with a 10-degree snap-to angle. This was a configuration that was used by almost every user in the first implementation.
Config 5: The bullseye is placed a few centimeters off of the palm with a 10-degree snap-to angle.
Config 6: The bullseye is placed on the index finger and has a 45-degree snap-to angle.
Config 7: The bullseye is placed on the index finger and has a 3-degree snap-to angle.

Data Collected:

Data is collected from several sources in this experiment. The system logs data on user trials, accuracy and preference. The user uses the speak-aloud method for qualitative data and give their rating of each configuration which is recorded by the researcher along with the researcher's own observations. There is also a comfort ratings form and a post experiment questionnaire.

Results

After this implementation, we obtained acceptable results for parameters to the selection techniques. Users were able to modify the selection techniques easily with an average rating of about two out of five (one being the best) for the usability of the personalization task. They also seemed eager enough to try extra

trials with the average number of trials being 15 when the required amount was 8 for occlusion selection and 7 for ray casting.

Quantitative ratings were taken from user surveys and log files of the user experiments. Figure 8 gives the user's ranking of four criteria for each selection technique. Occlusion selection was preferred to ray casting in all but the comfort category. The configurations for occlusion selection that were ranked highest were generally those where the bullseye was on one of the fingertips. Configurations 3 and 4 of occlusion selection that were preferred in Phase 1 A were ranked very low as compared to the other configurations as should be expected. This adds support to the notion that users need feedback to determine the value of a configuration. In the ray casting configurations, there was little variance among the different configurations. Also, various snap-to angles made little difference. In general, snap-to angles were set low in occlusion configurations, possibly because there was no need for feedback to try and refine a selection since the proprioceptive sense helps guide the user. In ray casting, snap to angles were larger though users complained about the ray flickering to other objects. This snap-to could be more useful in sparser environments where the flickering would not be such a problem.

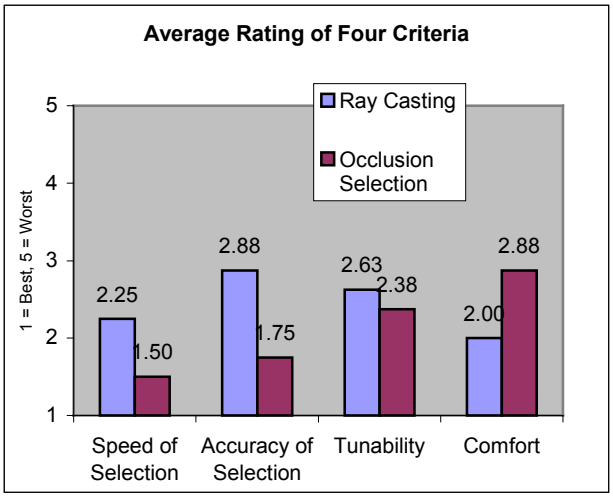

Figure 8. Users rated occlusion superior to ray casting in all criteria except comfort

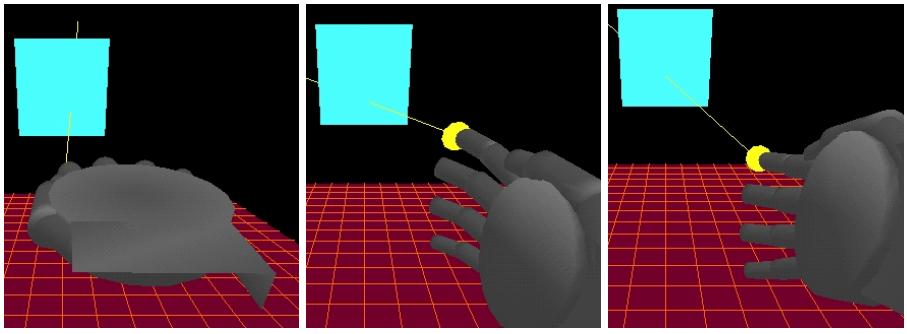

Figure 9. Unusual ray casting configurations: (left) the ray extends up and to the right (center) the ray extends down and to the left (right) the ray extends up

There were a few different configurations that users personalized towards (see Figure 9). For the most part, there was very little tinkering with parameters other

than the snap-to as most users knew what they wanted in their configuration from the start. Users preferred occlusion selection performed with either their index or middle finger. The major difference between these personalizations being the snap-to angle which was set to (in degrees) 20.60, 16.02, 0.0, 10.0, 4.47, 1.83 and 3.0 twice. Ray casting has four general categories with the first being the ray extending straight from the hand with normal to larger snap-to angles. A second personalized ray casting configuration has the ray extending up and to the right when the hand is parallel with the ground. This lets the user hold the hand in front of themselves in a natural 45-degree angle and make selections with a relatively small snap-to angle of 4.54 degrees. This configuration was not anticipated. Another configuration not anticipated had the ray extending down and to the left when the hand was placed flat. This allowed them to keep the hand in a natural position of close to a 45-degree angle and low, which reduced stress. Both users also liked large snap-to angles that might be explained by a difficulty to aim the ray when the hand is so far from the head. In the final configuration, the hand was held angled upwards and also at a natural yaw angle. The fact that the hand was close to the head could explain the relatively small snap-to angle.

What We Learned

Many users tended to select in the environment rather quickly until they encountered a selection that was difficult to make, due to the object being at a distance, near other objects or general user sloppiness. This difficulty caused them to slow down and select with more accuracy. Other times this happened were when the feedback was not matching what they were expecting such as when the bullseye was placed off-the palm or when the snap-to angle flickers between objects. This "bad vibes" behavior tended to last for a few selections until the user ramped back up to speed or discovered the source of their confusion.

There were six general properties that users demonstrated. The first was *spatial awareness*. Some users had the ability to understand where they existed in the space and this greatly improved their ability to use ray casting and most likely extends to other proprioceptive tasks. The next property is *feedback alignment*. Users responded to feedback with varying degrees of acceptance. Those with high feedback alignment considered the feedback to be infallible such that any indication of correctness, such as a snap-to event occurring, immediately triggered some sort of confirmation action. Users with these tendencies preferred smaller snap to angles. A third property is *exploration* or the ability to turn lemons into lemonade. Those with high search were able to create strategies to adapt to bad configurations quickly. This includes becoming more accurate faster and orientating the hand to reduce occlusion and fatigue. Those with little or no search basically never changed the way they selected throughout a configuration no matter the fatigue or difficulty. The forth property of a user was *resilience*. A resilient user will not mind fatiguing selection techniques. This can be an advantage because they can perform tasks faster using techniques that are not confined by fatigue, though long-term VE exposure could be a problem. The fifth

property is *precision*. This affected how much they tested the error boundary conditions of a selection technique, expanding their sloppiness to fit the bounds. The last property and near antithesis of precision is *speed*. This affects how fast they move and how long they must receive positive feedback before they go forward with an action. These parameters are most likely part of a set of parameters that users optimize when performing an action.

The implications are this:

> *There exists an innate set of parameters of the user and the interface that can be used to explain the mental model of users.*

So, a reward function can be created based on these parameters of the user that explains how a user will prefer to behave.

3.4 Phase 2: Maps of Selection Technique Preferences

Under the assumption that we have tuned selection tasks appropriately, then the next understanding we need is which types of selection tasks are preferred for various locations around the user. In effect, a 3D map of selection preferences between selection techniques. This incorporates user's personalized settings of the selection techniques from the previous phase.

Before we consider a map of selection space, one trouble we must overcome is handling the situation of two concurrently running selection techniques selecting two different objects at the same time. A nuance-based system would be able to handle such a situation because it would treat a technique as a function and the output would have an error value. The selection technique with the lowest error value is the technique to use. In our experiment, ray casting and occlusion selection both have error values in angular units. The measure of how comparable they are will be a topic of future research but for now they seem similar enough to consider each angular unit of error equivalent. Once a decent selection map is created, it too could be used to scale the error value for a technique.

Isolating Factors Affecting Selection Technique Preference

To correctly isolate the map of selection, we must identify the nuances that affect the user and minimize those effects. Those identified nuances are the hand's previous location, the selection technique previously used and other minor factors.

The previous location of the hand will affect which technique users prefer. As an example, if the user is pointing at an object across their view and the next object that appears is directly behind their hand, occlusion selection will be highly emphasized. To handle this, we added a delay of one second to the environment. This was enough time for the user to instinctively bring their hand to a more comfortable state and in many cases, to remove it as an occluding object from the scene.

The selection technique previously used will be an indicator of what they might use in the next step. This will be due to the hand already being in a position that is used for a selection technique such as pointing forward for ray casting. Also, the user will be in the mindset of that technique and switching mindsets will be a factor that needs to be considered.

Some of the other factors that affect the selection between selection techniques are object nuances, the existence of other objects in the scene, fatigue and "bad vibes". Object nuances can tell the user how to select the object or at least how to act around it. If an object seems to have a pan handle on it, selection techniques should account for a number of users wanting to select the object by its handle. We used grey cubes so this should not be a factor. Other objects in the scene can interfere with a selection technique by occluding an object, affecting the active feedback, etc. We removed all objects in the scene except for the hands, the cube and the floor with only the cube being selectable. Fatigue will affect how users work in an environment and only by users conserving energy by using less fatiguing selection techniques will they be able to avoid this. Since the selection of elevated objects is more fatiguing for occlusion selection than ray casting, the map should reflect this. "Bad vibes" also affects what people do. A user who uses a selection technique incorrectly a few times will shy away from it in future trials.

Selection Maps as Predictors

We created selection maps using examples of previous selection under the assumption that past experience will be a good indicator of future actions. To that end, data for the map is a Cartesian point and the type of selection technique used at that point. Our definition of a map will be the space in front of the user 120 degrees in the yaw and 80 degrees in the pitch. This roughly conforms to the viewable area of human vision. The user could select off screen or turn their body off of the world point of view which could make locating the cube difficult. The need to be efficient and locate objects quickly keeps users in-line with how objects appear and thus, they remain oriented with the world. Also, users do not interact out if their vision without at least some sort of proprioceptive sense.

Once data is collected, it can be used in two general methods; model-based and model free. Model-based representations take data and generalize it into a predictive model of the data. Model-free keeps the data around and use the data itself to predict. In our experiment, we used two model-free predictors and anticipated using one model-based but declined for reasons to be mentioned. The benefits of model-based predictions are a reduced amount of computation since the model is a generalization of the data it represents. The downside is that with any generalization, data is lost. Also, the ability to incorporate additional data varies among models.

The two model-free predictors used are sphere and cone based. The sphere-based predictor took the input point and drew a sphere of a given radius around the point

identifying all the sampled points contained. The cone-based predictor draws a ray from the user's eye through the input point and creates a cone around that ray using a certain angle again identifying all the sampled points contained. These identified sample points are then split among which selection technique was used when they were created and the selection technique with the most data points is the predicted technique of a certain probability. The radius and angle used is dependent upon the number of points you desire to be used and the density of your sampled points that you have collected. In our experiment, we used a radius of four meters and an angular error of twenty degrees.

Alternatively, model-base techniques induce a high-level representation that can be used as predictors for future interaction scenarios. A promising class of such techniques involves finding optimal and "admissible" regions [Fukuda96]. These algorithms exploit spatial continuity to postulate areas and regions of the 3D space that have high confidences for one selection technique to be preferred over all others. A union of such areas for all techniques constitutes what we can call a "selection map."

Environments Used

There were two environments used for this phase (see Figure 10). The first environment is similar to Phase 1 A's environment. The user stood above a gridded floor and saw one grey cube floating in space before them. The head and gloves were both tracked with models of hands attached to the gloves. TULIP was used to let the user choose when to start a trial, in this case a Demo trial of four selections or a Full trial of 100 selections. The second environment is much like Phase 1 B's environment. There was an array of 27 objects layed out 3x3x3 in front of the user. Again, the head and gloves were shown and again models of hands were attached to the gloves. In both environments, both selection techniques could have been used at any time with active and passive feedback operating. In order to give the users a break and keep the pressure to perform up, there was a pause every 10 selections where the environment told the user how many selections are left in the current trial and their qualitative ranking so far.

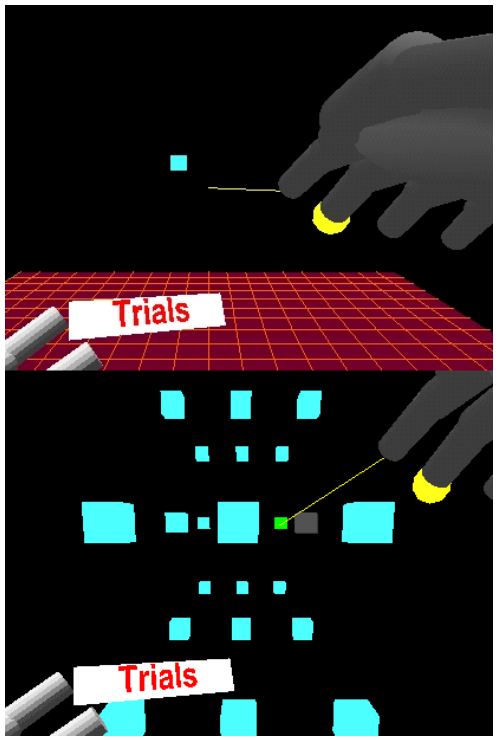

Figure 10. Phase 2 environments 1 (top) and 2 (bottom).

Experimental Design

There were three experiments in this phase. The first two experiments used environment 1 and the last used environment 2. Experiment 1 was designed to collect data points and experiment 2 was designed to test how predictive the environment could be of the users actions in the same environment using the sampled data. Experiment 3 used the sampled data and predicted user action inside environment 2 to test how applicable the map was to another environment. To collect the points in environment 1 and 2, the gray cube would disappear after a selection and reappear somewhere else in the map region. In experiment 3, one object was colored dark blue and that coloration would move randomly from object to object after each selection. The selection techniques were tuned to the same settings as the user preferred in Phase 1. Additionally, users were asked to perform a few selections of both techniques to help them regain familiarity before the experiment started.

Data Collected

Experiment 1 sampled 100 Cartesian coordinates of the cube when selected and also the selection technique used at that point. Additionally, the researcher recorded qualitative information and any comments the users made during the experiment. Since the users were trying to behave optimally, they were not explicitly told to think-aloud. Experiment 2 sampled 50 Cartesian coordinates of the cube when selected and additionally the results of the two prediction techniques with their corresponding supporting data. Experiment 3 recorded the same data as experiment 2.

Results

Predicting users using data from the same environment was nearly always correct, the reason being that users do not like to change between selection techniques. Users tended to pick their favorite selection technique and use it throughout the trial. When the environment switched to having multiple objects (experiment 3), the users still tended to use their favorite selection technique but when selections became difficult, they had a tendency to switch. Usually, they would switch back to their favorite selection technique immediately but some user would continue for a selection or two, then switch back to their favorite selection technique. This seemed to suggest a high user dependent cognitive cost of switching techniques.

The questionnaire returned a few interesting results the first being that in the first two experiments, the selection map was not a major factor in the prediction of the selection technique. In experiment 3 however, a selection map based upon user preference was considered by the users to be slightly a factor where ray casting was liked at a distance and to the left and occlusion selection in the other locations. Other results showed that user felt switching between selection techniques in the second environment was less difficult than in the first even though they claimed the utility to be about the same. In all environments, the users liked their configuration for the selection techniques.

What We Have Learned

When comparing selection techniques, there needs to be some sort of distractor task to remove the users from their current mindset. Without this, selection maps are overshadowed by the cost of switching so the extent to which selection maps are useful is still a topic of research though it appears to have impact.

Our experience can be summarized as this:

> *When given a choice between interaction techniques, users have a perceived utility of each and an associated penalty for switching from their current technique to another; in a sense exploration versus exploitation.*

Additionally, incorrect selections, frustration or the "bad vibes" concept reduces the perceived utility of a selection technique.

4 Why Traditional Approaches will not work

The three aspects identified above --- (i) users' mental models of how to respond to feedback, (ii) parameters that users employ in their mental models, and (iii) the explicit modeling of "exploration vs. exploitation" --- have been seen to characterize a nuance oriented VE. An adaptive selection system, for example, can model these aspects to provide a more responsive and personalized interface. Such adaptable behavior is typically achieved by machine learning and pattern recognition techniques, drawn from the AI literature.

For example, ML learning has been used in handwriting recognition [Garris98], sign language recognition [Kramer89], automatically adapting interfaces to users as they work in environments [Brown90] and to support programming-by-demonstration [Cypher93].

One of the primary advantages of using ML techniques is their ability to generalize to situations not encountered before. This generalization ability is aided by model-based techniques such as neural networks, decision trees, production systems, rules, and navigation maps. Such techniques require a reasonable amount of both "training data" and "training time" in order to construct a model. Evaluation of such techniques thus involves a distinct training phase followed by a test phase to validate the models. The techniques differ in their complexity of learning the representations (models), amount of training data required, the nature of their induced representations, and their ability (or lack of) to incorporate new data on a continual basis.

While ML techniques are prevalent in many desktop user interfaces, VE interfaces constitute a relatively nascent field of application. Slater et al. describe the use of neural networks to learn when users are walking in place to create a VE travel technique [Slater95]. Neural networks can approximate any function to any required level of accuracy (perhaps with exponential increase in complexity). They use one or more layers of intermediate functional elements to model the

dependence of output signal(s) on given input parameters. The general problem of learning NNs is NP-complete, but that has not dissuaded engineers and scientists from employing them as a tool to solve functional modeling problems, particularly noisy ones.

Similarly, models such as decision trees [Ruvini00] and version spaces [Eisenstein00] have been employed in VE research. In this thread of research, the choice of the model has been driven by the characteristics of the dataset, real-time constraints, and the explainability of the induced representations.

Neural networks have also been tried but were found to not model the data in an acceptable manner due to the excessive amount of data collection and training that would need to take place.

Such simple approaches do not adequately model the nuance interface problem presented earlier. In other words, a nuance is best modeled as a decision procedure, itself, imitating and mimicking the user's decision procedure.

5 Inverse Reinforcement Learning

This problem is formally referred to in machine learning as "inverse reinforcement learning (IRL)" [Ng00]. The assumption in IRL is that an agent's behavior (which can be observed) is the result of a deliberative process of choosing and weighting actions. If the agent (a VE user) can be assumed to be behaving "optimally" (based on his or her own notion of what this means), then the IRL problem can be formulated as one of (i) uncovering the user's "reward function," (ii) finding a policy (a representation of a nuance) that works as well as the user's nuance, or (iii) both. For example, perhaps a user employs a nuance to minimize hand fatigue but is otherwise unconcerned with the strain on his eye. The user's notion of optimality then would correspond to a weighted linear combination of these response variables with hand fatigue having a higher additive contribution than eyestrain. Using IRL, we can uncover this nuance and attempt to model the decision procedure that optimizes the user's reward function.

Employing IRL to create a nuance oriented VE requires the following steps:

a. Identify a suitable representation for mapping precepts (signals, movements, recordings, etc.) to selection goals and tasks. This could involve logic-based models (rules), attribute-value representations (neural networks), and/or approaches that explicitly model uncertainty (bayesian networks).

The primary constraints are that the representation be rich enough to model the user's mental procedure and simple enough to learn and maintain, computationally. Notice that if the representation is "overly rich," it could "over-fit" the data, meaning that its generalization could be affected.

b. Capture sequences of user interaction that serve as demonstration examples. Analysis could be done per user-group, per user, or per session. For each set of examples, IRL is applied to uncover a potential reward function and a value function (that models the tradeoff between how much the user explores selection techniques and how much the user is satisfied with what he/she knows).

c. Ensure the validity of the mined reward and value functions by (i) characterizing consistency with prior knowledge (e.g., "does it subscribe to known and tested models of user feedback?"), (ii) isolating extraneous factors ("do the learned functions apply equally in "similar" VEs? how much are they sensitive to the initial configuration?"), (iii) conformance with people's perceived costs of switching techniques ("do the learned functions help identify when some techniques get `etched' in interaction sequences?"), (iv) level of personalization observed in the sample group ("do the learned functions sufficiently characterize this variation"), and (v) sensitivity of the learned functions to the representation.

d. Depending on the causes and factors identified thus, the previous two steps can be iterated.

The end product is an understanding and modeling of the mental model of a task, as amenable to inverse reinforcement learning (IRL). For example if successful in uncovering users' reward and value functions, IRL can be used to suggest hitherto unnoticed selection forms and techniques. The first constraint is easy to justify, but notice that having a `too rich' representation can actually

6 Conclusions and Future Work

In this paper, we have introduced the concept of nuance-oriented interaction - personalized user interfaces based on subtle but crucial differences in users' behavior and mental models. We have contributed a classification of nuances into four categories, and have shown the existence of such nuances through our experiments. One key finding was that users adapt their behavior to the feedback presented by the system, rather than performing actions based solely on an internal model of the environment. We also found that although we could identify users' nuances observationally, current ML techniques are not well suited to this task, and we propose IRL as a possible solution.

We have laid the groundwork for the future development of nuance-oriented interfaces for VEs, but there is much work still to be done. First, we need to test our hypothesis that IRL will be able to recognize user nuances and predict user behavior. Second, we plan to gather more concrete data about user nuances based on properties of the environment and objects within it. Third, we need to extend this work beyond the relatively simple task of object selection and show its applicability to other, more complex, tasks such as 3D navigation and object manipulation.

Our long-term goal is a fully automated nuance-oriented system that will learn users' patterns of behavior and allow them to interact in an intuitive and natural manner. With such a system, users could train the interface to recognize personal "shortcuts" for common tasks, to modify the direction of motion based on one's preferred hand posture, or even to provide support for a personal style of design. Recall the scenario from the introduction. An interface designer could never anticipate all of the complexity required for this type of interaction, but nuance-oriented interfaces have the potential to allow this type of personalized interaction.

References

[BowmanHCI01] D. Bowman, C. Wingrave, J. Campbell and V. Ly, 2001: "Using PinchGloves™ for both Natural and Abstract Interaction Techniques in Virtual Environments", HCI International. (to appear)

[BowmanIEEEVR01] D. Bowman, C. Wingrave, 2001: "Design and Evaluation of Menu Systems for Immersive Virtual Environments", Proceedings of IEEE Virtual Reality 2001, pg 149-156.

[Brown90] D. Brown, P. Totterdell and M. Norman, 1990: Adaptive User Interfaces. London: Academic Press.

[Cypher93] A. Cypher (Ed.), 1993: Watch What I Do: Programming by Demonstration, Cambridge, MA: MIT Press.

[Eisenstein00] J. Eisenstein and A. Puerta, 2000: "Adaption in Automated User-Interface Design", *Intelligent User Interfaces*, pages 74-81.

[Fukuda] T. Fukuda, Y. Morimoto, S. Morishita, T. Tokuyama, 1996: "Data Mining Using Two-Dimensional Optimized Association Rules: Schema, Algorithms and Visualization", Proceedings of the ACM SIGMOD Conference on Management of Data, pg 13-23.

[Garris98] M. D. Garris, 1998: "Intelligent System for Reading Handwriting on Forms", *International Conference on System Science*, vol. 3, pages 233-242.

[Jacob99] R. J. K. Jacob, L. Deligiannidis and S. Morrison, 1999: "A Software Model and Specification Language for Non-WIMP User Interfaces", *ACM Transactions on Computer-Human Interaction*, vol. 6, no. 1, pages 1-46.

[Kramer89] J. Kramer and L. Leifer, 1989, "The Talking Glove: An Expressive and Receptive 'Verbal' Communication Aid for the Deaf, Deaf-Blind and Nonvocal", tech. Report, Stanford University, Dept of Electrical Engineering, Stanford, CA.

[Mine97] M. Mine, F. Brooks and C. Sequin, 1997: "Moving Objects in Space: Exploiting Proprioception in Virtual Environment Interaction", Proceedings of SIGGRAPH, pg 19-26.

[Ng00] A.Y. Ng and S. Russell, 2000: "Algorithms for inverse reinforcement learning." *International Conference on Machine Learning*, Stanford, California: Morgan Kaufmann.

[Norman90] Donald A. Norman, 1990: The Design of Everyday Things. Doubleday.

[Pierce97] J. Pierce, A. Forsberg, M. Conway, S. Hong, R. Zeleznik and M. Mine, 1997: "Image plane interaction techniques in 3D Immersive environments", Proceedings of the 1997 Symposium on Interactive 3D Graphics, pg 39-44.

[Poupyrev96] I. Poupyrev, M. Billinghurst, S. Weghorst and T. Ichikawa, 1996: "The Go-Go Interaction Technique: Non-Linear Mapping for Direct Manipulation in VR", Proceedings of the ACM Symposium on User Interface Software and Technology, pg. 79-80.

[Poupyrev97] I. Poupyrev, S. Weghorst, M. Billinghurst, and T. Ichikawa, 1997: "A Framework and Testbed for Studying Manipulation Techniques for Immersive VR", Proceedings of the ACM Symposium on Virtual Reality Software and Technology, pg. 21-28.

[Ruvini00] J.-D. Ruvini and C. Dony. "APE: Learning User's Habits to Automate Repetitive Tasks", *Intelligent User Interfaces 2000*, pages 229-232.

[Slater95] M. Slater, M. Usoh and A. Steed, 1995: "Taking Steps: The Influence of a Walking Technique on Presence in Virtual Reality", *ACM Transactions on Computer-Human Interaction*, vol. 2, no. 3, pages 201-219.

[Wingrave01] C. Wingrave, D. Bowman, N. Ramakrishnan, 2001: "A First Step Towards Nuance-Oriented Interfaces for Virtual Environments", Laval Virtual.