A First Step Towards Nuance-Oriented Interfaces for Virtual Environments

Chadwick A. Wingrave, Doug A. Bowman and Naren Ramakrishnan
Department of Computer Science
Virginia Tech
Blacksburg, VA 24061 USA
{cwingrav,bowman,naren}@cs.vt.edu

Abstract

Designing usable interfaces for virtual environments (VEs) is not a trivial task. Much of the difficulty stems from the complexity and volume of the input data. Many VEs, in the creation of their interfaces, ignore much of the input data as a result of this. Using machine learning (ML), we introduce the notion of a *nuance* that can be used to increase the precision and power of a VE interface. An experiment verifying the existence of nuances using a neural network (NN) is discussed and a listing of guidelines to follow is given. We also review reasons why traditional ML techniques are difficult to apply to this problem.

1 Introduction

There are problems with existing interfaces for virtual environments (VEs) [Herndon94]. One problem is dealing with the noisy data exhibited by typical VE input devices such as magnetic trackers. Another is a lack of constraints on interaction techniques [Bowman95]. Third, interface designers typically impose a rigid model of interaction on the user that forces the user to work exactly as the designer, not the user, intends. Additionally, there is very little consistency in existing interfaces from one VE to another. Interface designers must also fill in gaps in VE input data in many cases because input devices do not always cover all the data points designers need. An example of this is torso-directed travel, when the torso direction might be inferred from the user's head and hands. Finally, many of the so-called "natural" interfaces do not feel natural in their implementation, which leads to shortcomings in environments that focus on training.

In this paper, we introduce the concept of nuance-oriented interfaces for VEs. Our hypothesis is that the user has a mental model of interaction in VEs trained at first by interaction in reality and followed by training in other VEs. This mental model will be their first intuition for performing an action. It will also dictate the user's methods of increasing performance of an interaction technique by performing *nuance* actions, which can be found in input device data. A nuance according to Webster has three definitions; 1. a subtle distinction or variation 2. a subtle quality 3. sensibility to, awareness of, or ability to express delicate shadings. A *nuance* will be defined for our purposes as 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. If these nuances could be identified and managed as a part of the interaction technique, users would have a more responsive environment to their actions. This could lead to improved efficiency and presence.

Consider the following scenario showing the power of nuance-oriented interaction:

> *Brad, the VE user, is wandering around a model of a building being planned. He raises his hand towards the door exiting to the hallway and gives a slight rotation of his wrist and*

he starts to accelerate towards the door. On his way, Brad hears a comment from the building architect, Dave, who is behind him. Not hearing what is said, he turns his head around, the acceleration towards the door pauses and Brad raises his head with a slight tilt causing Dave's remark to echo in his ears. "Does that electrical conduit run through here?" says Dave's automatically repeated response. Brad holds his hands in front of his face causing a finger-labeled menuing system to appear [Bowman01] *and he turns on the display of the electrical system. "Nope", Brad replies as he turns back towards the door causing the acceleration to immediately resume. Once in the hallway, Brad points down the hall and rolls his hand forward where he zooms down the hall. Before he turns, he notices two electrical conduits intersecting when they should not. 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.*

Our thesis is that nuances by users can be modeled and advantageously exploited through the use of machine learning (ML) and optimization techniques. By personalizing interaction techniques to specific users, we can produce robust interfaces designed to make use of user's mental model. Learning techniques can be used to deal with the errors in data and to help fill in the gaps. They can also reduce the rigidity of the interface and make the interaction feel more natural, if properly applied.

We first cover related work in ML applied to interfaces. Next, we discuss nuances in VE interfaces and then review a test system we created as a proof of concept. This is followed by a discussion of lessons learned and next steps in this research area.

2 Related work

The use of ML and artificial intelligence techniques in user interface research is not new. Machine 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.

We will show in Section 3.3 how the assumptions of these simplistic techniques render them inadequate for nuance-oriented VEs. We also propose new approaches based on recent developments in the ML arena.

We are trying to look at a larger problem in VE interaction. Research has up to this time been spent studying individual types of interaction and trying to compare and contrast them by observing the user's responses. This includes even those applications that have applied ML to optimize a technique. Our approach is to discover how the user wants to work in a system as a whole and what their mental models are of the interaction before them, not of a single particular interaction studied by itself. With this knowledge, we hope to build nuance-oriented VE interfaces that are tuned by the user and not imposed on them.

3 Nuances in VE interactions

Before we embark on the long and difficult task of creating a new type of interface built upon nuances, we have to ask ourselves why this is necessary. Jacob gives the major reasons why VE interfaces differ from traditional WIMP interfaces [Jacob99]. These are:

- single-thread input/output versus parallel, asynchronous, but interrelated dialogues
- discrete tokens versus continuous and discrete inputs and responses
- precise tokens versus probabilistic input, which may be difficult to tokenize
- sequence, not time, is meaningful versus real-time requirements, deadline-based computations
- explicit user commands versus passive ("non command-based") monitoring of the user

There are other less obvious reasons as well. As previously stated, users have evolved to operate in 3D spaces and are quite adept at it. Their methods of interaction will be based upon their existing knowledge of the world and this only becomes more important as the levels of presence rise in VEs. Also, many times actions in a VE are not discrete so creating an undo feature is difficult. Since the ability to reverse unwanted actions is one of the most important features of interface design, this could lead to severe difficulties, especially considering that VEs have difficulties interpreting user actions.

Therefore, we need nuance-oriented interfaces that can manage these differences. The nuances can work on parallel input data, they can form their own bounds for continuous data, they can be probabilistic and be based upon time. This type of a system has not fully been attempted in the past because of the work required in recognizing all the details and implementing them. A VE that discovered users' nuances by ML and acted on them would not require the interface designer to work out all the details of a robust interface but let the user flesh-out the environment with their mental model. Since users can process information from various sources and work on multiple tasks, we would expect that the created nuances would handle the difficulties of VEs well.

3.1 Categories of Nuances

One of the main research issues here will be to understand the fundamental processes by which nuances are created, employed, and refined as a user interacts with a VE. While some of this knowledge could come from an expert understanding of the particular interaction, other knowledge could be mined from experimental data, or learned by a system automatically as a result of experience.

We have identified four categories of nuances. These include *environmental nuances* (nuances that arise from an object existing in relation to the environment) and *object nuances* (nuances that arise from some affordance of the object). *Refinable nuances* adjust boundaries for existing techniques such as correcting for constant errors in interaction techniques like ray-casting and arm extension. 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.

For example, many users use body-centered references such as pointing to indicate objects or locations they want to remember later [Bowman99]. This is an example of a supplementary nuance culled from observations by experts. This could be modeled in a system as "rote knowledge." The second alternative, to mine nuances from experiments, will provide a phenomenological view of how nuances are exploited by users (alas, it may not explain them). This mode of investigation is prominent in knowledge discovery and data mining activities. The reader will be familiar with the beers-diapers discovery in commercial market basket data ("People who buy diapers in the afternoon are more likely to buy beer too") [Agrawal93], but the role of data mining in VE research is a larger and more complicated application. Finally, ML techniques such as reinforcement learning can be used to refine a (representation of a) nuance as more data or information is acquired.

Refinable nuances can be used to alter the existing behavior of an interface to make it more usable. 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. In this way, if the user is trying to select an object using arm extension, the refinable nuance will widen the acceptable depth error.

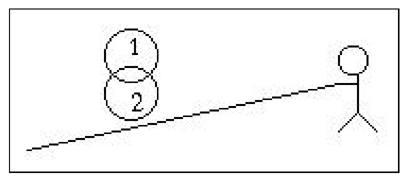

Figure 1: Using Ray Casting to select between objects 1 and 2, the user errs in the direction away from the object they do not want to select.

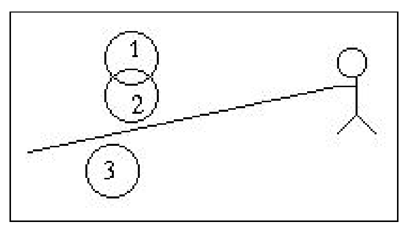

Figure 2: Because the user errs away from objects they don't want to select, we make the assumption that object 2 is the desired object even though objects 2 and 3 appear similar distances away from the user's Ray Casting selection.

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 1), 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 2), 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.

An example of an object nuance comes from cylinders. If an object is long and cylindrical, the user may feel the need to select it as they would a panhandle or a pipe; 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 could use the orientation of the hand to select the pot when they had to reach from across the room or a difficult angle.

3.2 Guidelines

As can be seen from the work that nuance-oriented VEs must do, there are certain requirements that are placed on such a system. These guidelines can be adapted from guidelines set forth in similar fields such as gesture recognition [Rubine92] and automated user interface design [Eisenstein00][Ruvini00]. They are easily remembered by the cumbersome acronym A FUR STAGE: accurate, fast, understandable, refining, sensitive, trainable, attribute-based, general, extensible.

3.2.1 Accurate

The VE needs to be accurate in its ability to recognize when a nuance has occurred. Incorrectly identifying a nuance may make an incorrect action occur for the user. This could increase frustration as well as produce incorrect results. As stated earlier, it can be difficult to implement an undo feature in VEs so incorrect results can be very painful to correct.

3.2.2 Fast

Response time greatly affects user satisfaction with an interface [Baecker87] and in VEs, most systems are already heavily taxed so any extra processing greatly affects performance. Many ML algorithms are not created to be run in real-time so the algorithms need to be specially handled or selected. Another solution might be distributing the computation but that is left up to the implementer.

3.2.3 Understandable

Users need to be able to perceive that a VE actually used their nuance in the interaction. If users feel that their nuance was ignored, they will stop making such actions and the nuance will go unused. This is not necessarily a conscious action on the part of the user as some refinement nuances simply expand on the boundary conditions of interactions.

3.2.4 Refining

Most interactions already have known affordances and users have existing knowledge. This should be supported by nuances and not rebuilt by them. The nuance interfaces should only increase the accuracy and robustness of the interface in addition to handling special cases.

3.2.5 Sensitive

Most nuances will be slight modifications to an existing interaction that a user makes so VEs will have to be very sensitive to recognize when a nuance has occurred. This is made difficult by the guidelines of being accurate and understandable because we must notice every detail and avoid all possible mistakes while accounting for each nuance whenever it first occurs. The upside is that in some cases the user may not even notice a change in the interface from a new nuance; they will only notice that they are struggling less with the interface.

3.2.6 Trainable

VEs need to recognize quickly that a nuance has occurred. If not, then users will stop making such actions and the possibility of supporting users with the nuance will be lost. Additionally, some

VEs might want to customize themselves to individual users or groups of users over time. Since nuances will hopefully be carried over from one environment to another, it will be important to have the transplanted nuance train for a specific environment's details in addition to just the general case.

3.2.7 Attribute-Based

The nuance itself should convey information from its size, duration, location, orientation, etc. Some nuances might purely be boolean but we should assume that most would apply some amount of numerical modification to an existing interaction and should be a function of user action. So, if the user does a nuance movement in a large sweeping motion, then we should increase the amount of change the nuance coveys.

3.2.8 General

Nuances should be identifiable in various sizes and orientations because users are of various sizes and orientations. A nuance that only works when the user is performing it along a certain axis is not general, as the chance that the user will be facing along that axis every time is not high (unless that is a specific characteristic of the environment). This may convey some attribute but it will also be due to the fact that users are not consistent and are error prone. So, if the nuance is to be useful, it needs to have a form general enough to be easily performed and structured enough to not be accidentally triggered.

3.2.9 Extensible

The nuance should be extensible to new interfaces both in the same and other VEs, and could be device independent. There might be a need to recognize a VE-specific nuance, but most likely an action in one VE will be performed by the user in another, especially if the user becomes accustomed to that nuance.

3.3 How a Nuance-Oriented VE Might Work

How does one go about creating a VE system that uses nuance-oriented interaction? At first glance, the problem of designing a nuance oriented VE looks suspiciously similar to programming-by-demonstration, with just a more complicated (and richer) demonstration sequence of interactions. This model of learning typically involves recording user scenarios, replaying them, and (in a limited way) generalizing the scenarios. A nuance is more than an enumerated (or captured) list of scenarios. A nuance implies an internal model that a user brings to the interaction task and employs (in the manner of a decision procedure) actively when interacting with the VE. In other words, a nuance is best modeled as a decision procedure, itself, imitating and mimicking the user's decision procedure.

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.

In many instances, the user might not be explicitly aware of their nuances or why/when they employ them. The observation-based approach to IRL will work even in these cases, as long as it is reasonable to assume that the user is systematic in his choices and chooses actions and interaction techniques in a consistent manner. Notice that the learned nuance may or may not be the same as the user's decision procedure but can serve as a meaningful model of the original nuance.

Once this problem is formulated, several challenges remain. The richness of the nuances employed (and the associated decision procedures) directly impact the choice of model. The model must be able to capture the full complexity of the interaction metaphor, and at the same time be computationally cheap to learn, update, and maintain. A good first step would be to qualify a design vocabulary of nuances and their various forms. A careful analysis will provide insights into model selection.

Once a preliminary representation of a nuance is available (either seeded directly, or by some basic data mining), integrating it into the control flow of a VE application is important for IRL. The system might be beset with judgment calls involving both false positives and false negatives. False positives are more serious (and irritating) than false negatives and this could be factored into the IRL training algorithm. When IRL is employed with sampled data (as it most likely will be in a VE), heuristics will become important to "shape" the nuances [Ng00][Kaelbling96], and steer them away from suboptimal solutions.

Another important distinction would be whether the training should be done off-line or on-the-fly. Off-line training will allow an interface designer to play with data and validate nuances before they are thrust upon the user. This could help weed-out the misjudgments by the learning system. Unfortunately, if a user is using a system and it does not respond to their nuance, the user will likely stop giving that nuance. Two possible types of systems are "train by example" systems and systems that mine user logs. The example learning systems have the problem of letting the user act naturally during the training periods such that they will repeat a nuance enough to be recognized. Mining systems can bring nuances to the attention of the UI designer and be less selective because the UI designer will ultimately decide if there is value in a nuance.

On-the-fly systems do not have to worry about users not repeating a nuance due to lack of recognition, because they can recognize the nuance and immediately change the VE. Such a system may confuse a user because of incorrectly created nuances. Also, it has the major disadvantage of being computationally expensive on a system that is taxed by the VE it is already running. This could be implemented as a local or remote agent, mining user data concurrently with program execution. This type of system would be more difficult to implement than off-line learning.

4 Proof of concept

We wanted to create an example system to test our theory that users employ nuances and that these nuances can be recognized and used to enhance interaction. We chose to focus on the task of selection because it is understandable and users have existing knowledge and mental models of this task. As a preliminary exploration of the modeling choices in nuance research, we employed a NN and the JIVE Toolkit [Wingrave] built on top of DIVERSE [Arsenault].

The environment consisted of three balls placed slightly out of reach of the user in a horizontal line in front of the user. There were two blue balls and one red ball and the user was to select the red ball using whatever they felt would distinguish that ball from the other two using their tracked hand (Figure 3). Fakespace Pinch Gloves were used to indicate selection at the hand's current location. There were seven (7) possible positions for the balls and neighboring positions had the balls partially overlapping to increase pressure on the user to select precisely. The data collected was the position of the user's head and hand, as well as the position of the balls and which was red, at the time of the pinch.

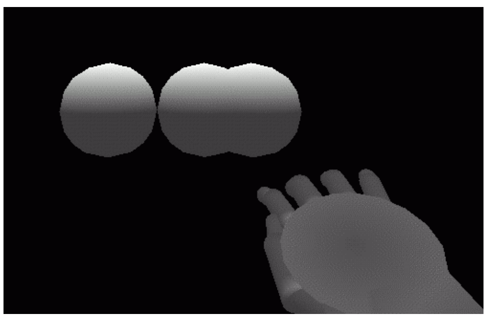

Figure 3: The proof of concept VE with three balls, two of which are overlapping, and the user's hand. When the user pinches his fingers, the VE will tell them which ball they were trying to select based upon the trained NN.

A NN was used for the ML method. A NN was chosen because NNs are easy to train and are fairly fast to run when plugged into the environment after being trained. They do not match our guidelines, so they are not suggested for full-scale nuance-oriented systems but their ability to handle a wide variety of situations in a proof-of-concept system was considered more important than their downside. The collected user data was normalized and split into a training and a test group. The data for the groups was fed into a feed-forward NN with one hidden layer and three outputs. Each output gave the level of selection of the corresponding ball. Training was performed with back propagation; then the net was loaded into the same environment with three balls. After each pinch, in the test environment, the NN printed which ball had been selected.

The results were both encouraging and frustrating. The NN was very hard to fool when used in exactly the same manner as the testing data was taken. It very easily distinguished the object the user was trying to select, even when all three of the objects were overlapping and the center object was to be selected. However, a small change, such as standing up for example, lowers the arm position and the manner of selection so the NN very quickly leaves its trained area. While some of these aspects could be addressed by using translation invariant testing, NNs have other disadvantages including (i) excessive dependence on the original network topology (and hence, experimental setup), (ii) the black-box nature of their function (rendering their results inscrutable), and (iii) their capacity to incorporate knowledge in only a limited form. Additionally, the training of the NN takes quite a bit of computation and requires much training data. Because of this, full user trials were not considered practical and no statistical results are reported here.

The conclusion was that NNs are not feasible for large-scale nuance-oriented interfaces, as simple changes to the VE would require a completely new dataset and more computation. Also, collecting data for all the probable situations is not practical or easy. The concept of nuance interaction has successfully been shown, however. Users do operate with nuances and an interface can make use of this to increase precision. The next step is discovering which ML methods hold the most promise.

5 Current Work

We contend that a VE based upon nuances has merit. Our immediate focus is on creating a system that recognizes and makes use of these nuances. Then, we will examine how users perform in these VEs.

Also, it would be interesting to see if they start to use nuances captured from other users and additionally, if they even know that they are making use of nuances. Future work may include the creation of nuance libraries for general VE interface design [Sutcliffe00].

One method of overcoming the difficulties of offline learning is to remove all forms of feedback from the user. We assume that the user will stop making a nuance when they realize that the system is not making use of it. By removing feedback, however, the user will not know to stop using the nuance and hopefully will not change their mental model. This can only be performed in an inverse reinforcement-learning environment where the system presents the desired result (e.g. “select the red ball”) and the user performs an action to achieve that result. We are currently investigating this method in the context of existing selection techniques. We wish to use refinement on existing techniques as opposed to discovering new ones because of the reduced learning required. We will then test the resulting VE in usability trials.

We are also looking into a series of experiments to discover examples of environmental nuances and object nuances working in collaboration with nuances we discover in the selection tasks. These experiments should lead to VE selection techniques that more accurately follow the user’s mental model of interaction and therefore will be more intuitive than techniques built by an interface designer.

6 Conclusions

This paper has introduced the concept of nuance-oriented interfaces applied to VEs. A review of ML techniques required to discover these nuances has been given as well as guidelines for systems that will use nuances. Finally, a proof of concept has shown that nuances do exist but require more research to become usable.

Currently, VEs are rigid and inflexible so as to minimize the unwanted actions that might be triggered accidentally by the interface. This makes applications that require fine-grained interaction such as assembly and surgery difficult but nuances are certainly performed that could easily increase the usability of the VE. In the case of rigidly defined interaction techniques, any nuance that the user performs will, in the best case, only be a waste of energy but in the worst case create errs. It is our hope that our work will remove the nuisance of the nuances from VEs and create more usable interfaces, designed after the user’s model of interaction, not the VE designer’s.

References

[Agrawal93] R. Agrawal, T. Imielinski and A. Swami. “Mining Associations Between Sets of Items in Large Databases”, *ACM SIGMOD International Conference on Management of Data,* 1993, pages 207-216.

[Arsenault] L. Arsenault, J. Kelso, R. Kriz, “DIVERSE: Device Independent Virtual Environments- Reconfigurable, Scalable, Extensible.”, http://www.diverse.vt.edu.

[Baecker87] R. Baecker and W.A.S. Buxton. Readings in Human-Computer Interaction – A Multidisciplinary Approach. Morgan Kaufmann Readings Series. Morgan Kaufmann, Los Altos, CA, 1987.

[Bowman99] D. Bowman, E. Davis, A. Badre and L. Hodges, “Maintaining Spatial Orientation during Travel in an Immersive Virtual Environment”. *Presence: Teleoperators and Virtual Environments*, vol. 8, no.6, 1999, pp. 618-631.

[Bowman95] D. A. Bowman and L. Hodges. “User Interface Constraints for Immersive Virtual Environments Applications", 1995 GIT-GVU-95-26 Atlanta: Graphics, Visualization and Usability Center.

[Bowman01] D. A. Bowman and C. A Wingrave. "Design and Evaluation of Menu Systems for Immersive Virtual Environments", to appear in the Proceedings of *IEEE VR 2001*.

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

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

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

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

[Herndon94] K. Herndon, A. van Dam and M. Gleicher. "The Challenges of 3D Interaction", 1994, SIGCHI Bulletin, 26(4), 36-43.

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

[Kaelbling96] L. Kaelbling, M. Littman and A. Moore. "Reinforcement Learning: A Survey", *Journal of Artificial Intelligence Research,* 1996, vol. 4, pages 237-285.

[Kramer89] J. Kramer and L. Leifer. "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, 1989.

[Mine95] M. Mine. "Virtual Environment Interaction Techniques", 1995 Technical Report TR95-018: UNC Chapel Hill CS Department.

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

[Rubine92] D. Rubine, R. Beale (Ed.) and J. Finlay (Ed.). "Criteria for Gesture Recognition Technologies", Neural Networks and Pattern Recognition in Human-Computer Interaction, West Sussex, England: Ellis Horwood Limited, 1992.

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

[Sutcliffe00] A. Sutcliffe. "On the Effective Use and Reuse of HCI Knowledge", *ACM Transactions on Computer-Human Interaction*, 2000, vol. 7, no. 2, pages 197-221.

[Werkhoven98] P.J. Werkhoven, J. Groen. "Manipulation performance in interactive virtual environments", Human Factors, 1998, vol. 40, no.3, pages 432-442.

[Wingrave] C. A. Wingrave, "JIVE: Just In a Virtual Environment", http://csgrad.cs.vt.edu/~cwingrav/Jive/home.html.