Through Artificial Neural Networks into Virtual Realities

Christian-A. Bohn Monika Fleischmann Wolfgang Krüger

German National Research Center for Computer Science
Dept. Scientific Visualization of HLRZ
Schloss Birlinghoven, D-53754 Sankt Augustin, Germany
E-mail: bohn@viswiz.gmd.de

Abstract

Implementations dealing with the term Virtual Reality have a certain goal in common: The interfaces between human and computer should not be perceived. On the one hand, generating images of 3d worlds should be as realistic as possible. Naturalistic views and high picture-refresh rates are demanded. On the other hand, the way back to the computer through its input devices should be robust against the natural, inaccurate actions of the user. The interfaces are wished to be "intuitive".

Laying emphasis on enhancing this understanding by the computer, leads to more human adapted input devices. This paper concentrates on two kinds of such interaction facilities, that fit more to human behaviour than to the "digital thinking".

Artificial neural networks are used to communicate to the computer by speech and gestures. Multi-layer Perceptrons and the backpropagation algorithm enable interfaces which show very good practicability.

The described modules are exemplary applied to two virtual reality projects, which were started to force new ideas in the field of interfaces.

1 Introduction

Interaction has been proven to be one of the most important challenges for the future of nearly all computational tasks. Above all, recent progress in the field of Virtual Reality (VR) and connected technologies has shown the huge lack in the abilities of common known interfaces between human and virtual, digital environments.

The connection of interfaces to the computer and the human should be planned carefully. The computer needs an exact definition of its commands, but the human user does not want to adapt in a machine-like manner. Different users as well as the same user at different times appear slightly different in their actions with the computer. The algorithms for interaction control need to have the ability to extract the essentials of the user's interaction and to suppress the meaningless parts of the input data. The interfaces must be adaptive and capable of training in a case/user dependent fashion. Beside of this ability, the user's inconvenience must be kept to a minimum. Learning must be fast, robust and automatic. Also the resulting functionality must be robust, with a small delay time, and there must be an estimate of the reliability of the found mapping between the user intention and the computer action.

To get a sufficiently good way into the computer several goals must be reached:

- adaptivity: trainable to different users or application fields,
- sufficient good recognition rate,
- small delay time,
- reliability: the capability of estimating the propability of a specific event,
- ease to use: ability to be retrained in minutes by the user without having knowledge about the internals of the program.

In this paper the incorporation of Artificial Neural Networks (ANN) into common virtual reality applications is described. Neural networks have been proposed to show very good facilities in applications where men interact with the computer, i.e. where the computer has to interprete human behaviour.

So, virtual reality, dealing with the subject 'men-machine interaction', is a huge application field of neural network algorithms.

Steering in the virtual space with speech and gestures are the examples which are successfully realized in two virtual environments, descripted below. They show functionality characterisitics, which are indispensible for common applications.

The developed ANN-based interfaces are

- a word-recognition system for natural (acoustical) speech,
- a gesture-recognizer for a VR data glove.

1.1 Virtual Reality

The standard metaphor for human-computer interaction is based on the daily experience of a white-collar office worker. For about 20 years, more and more enhanced desktop systems have been developed, providing the user with tools such as WIMP (Window-IconMousePointer), graphical user interfaces (UI), and most recently advanced multimedia extensions.

Raw computing power, huge storage capabilities, and broadband networks are becoming a commodity for computer simulations, visualization, data enhancement tasks, and graphics applications in general. From a user's perspective the relevant aspect is the problem solving process and not the details of the computation itself. The UIs and the algorithms supporting them, are the keys to an efficient data evaluation and manipulation process.

With the advent of immersive virtual environments 3D space can be made directly available to the user. Walkthrough experiences, manipulation of virtual objects, and meetings with synthesized collaborators have been proposed. Specific human-computer interfaces, originally developed for pilots and telepresence tasks, became available to the ordinary user [13]. So, interfaces between human and the computer has become more and more important, and virtual reality incorporates a short term for this development of techniques. It tries to give the user a better sight of digital data captured by a computer. Emphasis is laid on improving the intuitivity of interacting with the computer. The goal are interfaces that the human will not perceive as such any longer.

Because everybody "wants to play with", VR is the most critical testbed for various kinds of interfaces.

1.2 Artificial Neural Networks

In recent years, it appears that a lot of computational problems can be solved by neural algorithms much more efficiently.

Inspired by the brain's abilities, development of neural algorithms has been based on the imitation of biological neurons which are connected within a large connectionist model. Neural networks are "systems that are multiple simple uniform units whose main task is to communicate with each other, to exchange information" [8].

Simulating the functionality of the brain is known as simulating the neurons themselves, in addition to the flow of information in a large system. The neurons of such a neural network are called 'units', the artificial axons are 'connections'. The unit's task is to calculate a integration over its inputs, which arrives as outputs from other units, multiplied by the related connection weights. The result is propagated through the network as input to other units.

The task of a unit i is to sum up a number k of weighted (w_{ij}) inputs x_{ij}, and to deliver the result y to the input of connected units (1)

$$y_i = g\left(\sum_j w_{ij}x_j - \mu_i\right), \; j = 1..k \qquad (1)$$

with the activation- or gain-function g and its activation threshold μ. The sigmoid-function (2) is often used.

$$g(h) = \frac{1}{1+e^{-2\beta h}} \qquad (2)$$

β determines their steepness.

Combining m of these units leads to a network, whose overall function can be described by (3). The actual output values of the units (y) at a certain time (t) are used to accumlate the values at the next time step ($t+1$).

$$y_i(t+1) = g\left(\sum_j w_{ij}y_j(t) - \mu_i\right), \; i,j = 1..m \quad (3)$$

An ANN is defined by its topology (w_{ij}) and the kind and parameters of the activation function.

Working with neural networks starts on simulating it. At the beginning its functionality is at random and will be defined by fixing the network parameters, which are the unit's function parameters and the network topology (connection weights). Programming a neural network is characterized by tuning these network parameters until the network realizes a certain task. Here, supervised learning is used. Input/output pairs are presented to the network. According to them, the network parameters are changed, so that the network learns to reproduce the certain output later on if it gets only the input data. Programming is done by training.

Neural networks show several advantages. They are

- robust and fault tolerant,
- flexible, they can easily adjust to a new environment by 'learning' — they don't need to be programmed in Pascal, Fortran or C;

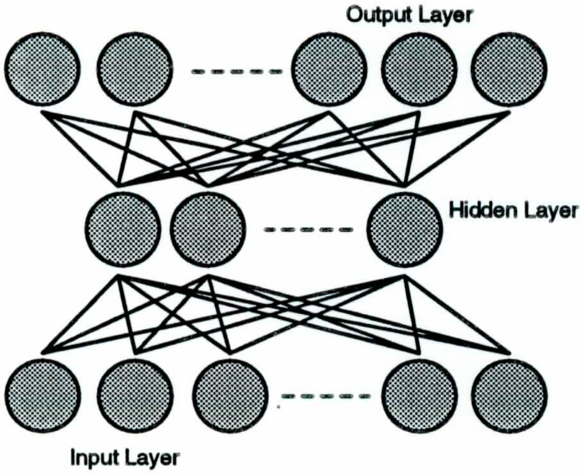

Figure 1: Backpropagation network

- they are highly parallel.

[6]

Nevertheless, one should be aware of the quality of a neural network simulation. The comparison with the biological model should point towards the question if it is generally possible to imitate such a complex system, because

- a neuron delivers continuous output — a unit does not,
- a neuron's output is a periodic signal defined by frequency and phase — the unit's output misses the phase,
- a network of units has a unique time synchronization — the brain has not,
- a (human) neural network has about 10^9 neurons with between 1000 and 10000 axons (connections) to others. It is evident that ANN's capacity ends up at a few hundred units with a similar number of connections.

In almost all cases, using a neural network is characterized by a trial-and-error process to get the right topology and/or to optimize the training process.

The neural network used in both applications is called multi-layer feed-forward network (multi-layer Perceptron[MLP]) with one hidden layer. The input layer is fully connected to the hidden layer, the same holds for the hidden to the output layer. The principle architecture can be seen in Figure 1. The network is trained by the backpropagation algorithm [6]. The wished function is a 1-out-of-n coding at the output layer. Each specific input should switch one output unit to a high level to determine the classified event. For training, a certain amount of input samples is presented to the network. According to the arised error at the output units, the connection weights are modified. After several iteration sequences the algorithm converges into a local — hopefully a global — minimum.

2 Realization

2.1 Speech Recognition

Computational problems, associated with speech recognition and the limited success of the conventional pattern matching techniques proposed to solve them, have fostered the development of neural network approaches to speech recognition tasks. The intention is that the generalization properties of neural network learning algorithms are useful to improve the recognition performance. [3, 4, 7, 10, 15]

Speech data, spoken by arbitrary speakers, arrives directly from a microphone through an A/D-converter. A separation into words (by detecting spoken pauses) follows a Fast-Fourier-Transform at 10 milliseconds each. The resulting short-time spectrums are averaged at certain frequencies, leading to a 15-dimensional vector and about 20 of them represent the whole word. The resulting 200 values are used for classification by the backpropagation network [5].

In our application, the recognition ability for a vocabulary size of 45 words lies at 96% for the speaker dependent, 91% for the speaker independent case, and seems to be an excellent result [14, 4].

Figure 2 shows a snapshot of running the speech recognition system. The states of the whole classification can be identified. On the left side of the screen the calculation of the short-time spectrums can be seen. Above, the raw speech data, in the middle the Fourier-transformed and finally the average frequency vectors are displayed in real-time. The right side shows the whole word as a time-frequency matrix. The dark matrix shows the original word, the transparent bright matrix shows the same after compression [5]. This is the result of the overall preprocessing parts of the speech recognition module, and finally is fed into the classifier. The neural network tries to judge about the characteristics of the "picture of the word", and delivers a number for the word which was identified.

2.2 Gesture Recognition

A CyberGlove (*Virtual Technologies*) is used. The user wears a glove which has up to four not visible bending-sensors at each finger, and two in addition for detecting the wrist's posture. The sensors are realized as fiber-optics wires, which change their optical resistance at different finger posture states. This information is D/A-converted and delivered through a serial line to the computer. 16 of them are used for classification by the MLP. The network was trained with 34 gestures. The number of hidden units is 6. First the network is trained user-dependent by one specific user, second by 6 users to get an user-independent functionality. Recognition rates in the first case were nearly 100% in the second 98%, which is quite competitive to other results [11]. Figure 3 shows a picture of learning/training the example gestures.

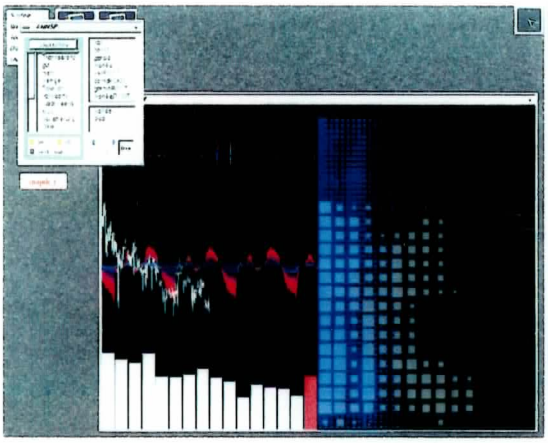

Figure 2: Speech-recognition module

2.3 Network Characteristics, Reliability

After recording the samples, learning lasts in both cases about two minutes. For user-independency the samples of 8 people were used. The networks converge in almost all cases to 100% accuracy for the training set. Otherwise a second iteration cycle fulfills the adaption.

Several additional tests with different networks were done. To realize one classification task, more than one network is used. In combining them as a network tree, which supports context specificity of the application dependent command grammar, small nets that have to distinguish between few words (for example 3), can be used. This idea enhances the robustness of the classification task greatly.

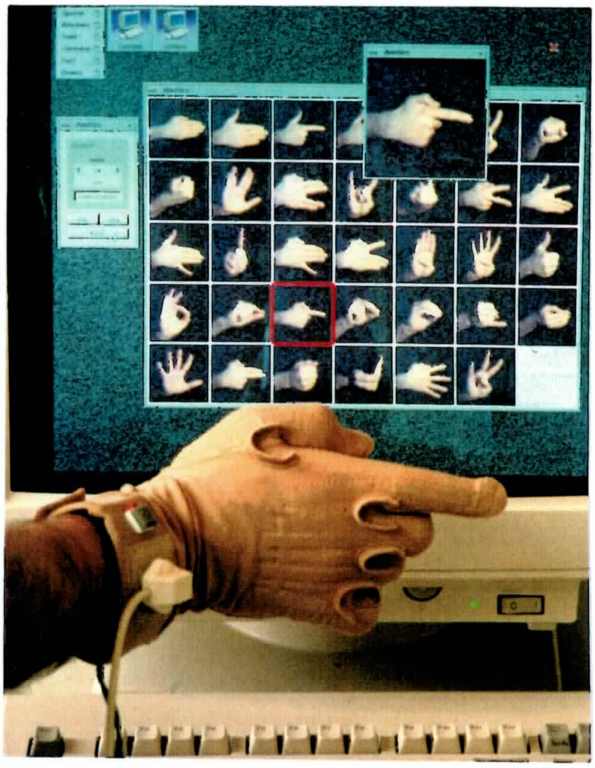

Figure 3: Gesture-recognition module

Also, additional nets are used that deal with the same samples, but are trained to different local minima. So, by "asking them all for their opinion", some sort of reliability can be measured, that a single network has not.

2.4 Hardware

The VR systems consist of two independent workstations. The main VR program, which manages rendering, the overall steering including the tracking devices, uses a SiliconGrapics-ONYX workstation. It is called the server and connected to a SGI-Indigo/R4000, the speech/gesture client.

A CyberGlove is connected to the serial interface of the client. The speech is recorded by the standard inbuilt microphone and A/D-converter. Both classification parts work in parallel on the client and contain a third module which is engaged in the voice recording, storing and sound output.

To get an interactive update rate, a R4000 microprocessor should be used. The results of the word/gesture classification are delivered as integer numbers through a serial line or by Ethernet via the socket mechanism to the server. Of course, graphics (Figure 2, 3) are switched off in a real application.

3 Applications

3.0.1 The Responsive Workbench

The first project realized is called Responsive Workbench (Figures 4, 5). It is an alternative to the multi-media and VR models of the past decade. The computer display is embedded in the user's environment. It is seen as a workbench surface. Objects are displayed in 3d-stereo-technique [1]. The user's eye off-axis is tracked by a 3d-tracking system, wearing LCD-shutter-glasses. The computer generates the 3d-sight of the virtual model in real-time. The user sees the objects nearly as they would appear in reality. So, the final result gives the imagination of perceiving a real object similar to a 3d-hologramme.

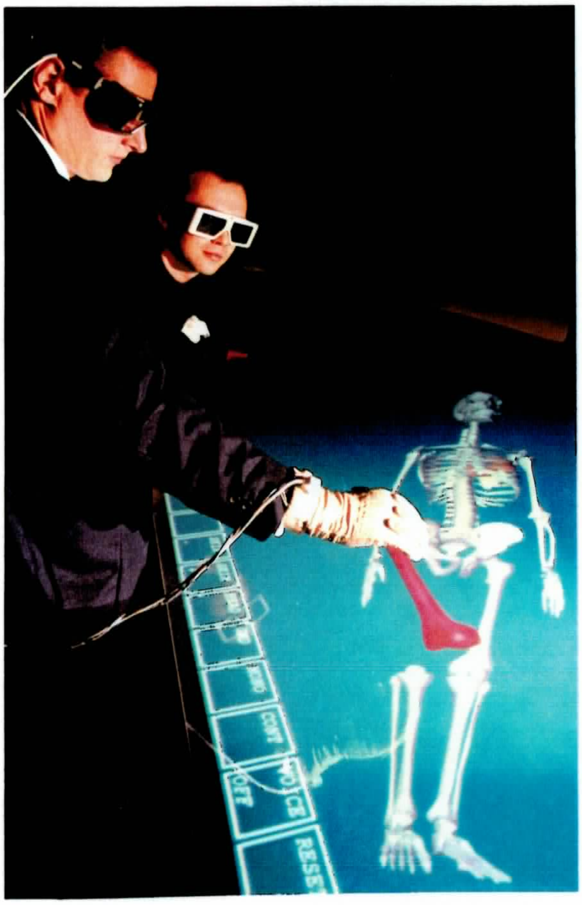

Figure 4: Surgery planning

The speech interface is used to steer the overall system. Moving commands and certain switch-like functions are recognized by the computer. Figure 4 shows an example from surgery planning. The virtual skeleton can be moved by speech. Even by gestures, it can be pushed and transformed. Bones can

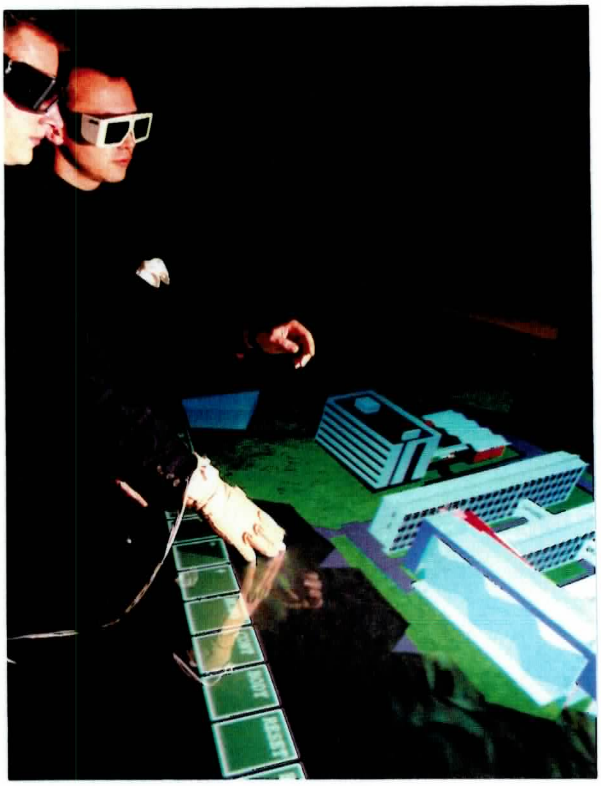

Figure 5: Architectural design

Figure 6: Walking through a virtual castle

be taken away from the body by hand.

Also architecture gives a huge field for applications with the responsive workbench (Figure 5). In this a case, a virtual model of a landscape can be moved. The walls of the building can be made transparent by speech commands.

3.0.2 Virtual Castle Tour

The second application is called Virtual Castle Tour Through Medial Rooms (Figure 6). With the aid of a Spatial Navigator, users can experience a complex, stereoscopically computer-generated 3d scenario in real-time.

The user moves with a motion simulator through the virtual simulation of Birlinghoven Castle — telling the computer what he intends to do. The guide gives commands in natural speech for intuitive navigation.

Objects can be grapped and certain events can be switched on by pointing towards them. Furthermore real-time video enclosed in the 3d scene can be displayed. By the use of speech- and gesture-recognition whose functionality partly overlap, the user gets a feeling of doing things intuitively. The computer reacts flexible to the actually kind of interaction.

4 Conclusion, Future Work

Both integrations of neural networks show that they can be used efficiently by carefully determining its functionality. They can be made robust and easy to handle, supported by recognition rates up to 100% for gesture-recognition and learning time of about two minutes.

For adapting the machine to the human, neural networks seem to be a competitive way. They are easy to handle and need less computational resources than the most known alternative methods. Because of their generality, they already have been implemented highly optimized or integrated as hardware modules.

A new way of getting immersed in a virtual environment has been shown. Speech- and gesture-recognition could easily be added to two arbitrary virtual environments without the common disadvantages of an awkward adaption to users or to the application.

Laying emphasis on advanced I/O algorithms has been proven as very profitable. This may be an idea for getting another view on new VR implementations by primarily looking from the interfaces on it. Neural networks, with its flexibility, robustness and their easy handling facilities, seem to be the right approach to extract the non-constancy of human's action, to get knowledge about the useful information in user commands, and finally to achieve a behaviour which is intuitive from the human's point of view.

A challenge may be to give one neural network a better view on the overall human behaviour, that means to combine several interfaces into one ANN approach to get information from their correlation. An example could be the combination of gesture and speech, that is even a natural way of communication between humans (supporting speech by gestures).

Generally, thinking about VR must be converted to thinking about interfaces and even about applying results from other fields like, for example, connectionism.

References

[1] Akka R, *'Writing Stereoscopic 3D Graphics Software using Silicon Graphics GL'*, IRIS Universe magazine, No. 16, Silicon Graphics, April 1991.

[2] Bohn C-A, Krueger W, *'Embedding Speech into Virtual Realities'*, Proceedings of the 'Intelligent Computer Aided Training and Virtual Environment Technology'-Conference (ICAT/VET-93), Houston, TX, 1993.

[3] Bourlard H, and Morgan N, *'A Continuous Speech Recognition System Embedding MLP into HMM'*, In: Advances in Neural Information Processing Systems, Vol. 2, pp. 186–193, Morgan Kaufman Publishers, 1990.

[4] Franzini M A, *'Learning to Recognize Spoken Words: A study in Connectionist Speech Recognition'*, In: Proceedings of the 1988 Connectionist Models Summer School, pp. 407–416, Morgan Kaufman Publishers, 1988.

[5] Freisleben B, Bohn C-A, *'Speaker - Independent Word Recognition with Backpropagation Networks'*, In: Artificial Neural Nets and Genetic Algorithms, pp. 243–248, Springer-Verlag Wien New York, 1993.

[6] Hertz J A, Krogh A, and Palmer R, *'Introduction to the Theory of Neural Computation'*, Addison-Wesley, Reading, Massachusetts, 1991.

[7] Kohonen T, *'The Neural Phonetic Typewriter'*, IEEE Computer, 3:11–22, 1988.

[8] Kemke C, *'Der neuere Konnektionismus'*, Informatik Spektrum, Springer Verlag, 1988.

[9] Krueger M, *'Artificial Reality II'* Addison-Wesley, Reading, Massachusetts, 1991.

[10] Lee K, *'Context-Dependent Phonetic Hidden Markov Models for Speaker-Independent Continuous Speech Recognition'*, IEEE Transactions on Acoustics, Speech, and Signal Processing, 38(4), 1990.

[11] Murakami K, Taguchi H. *'Gesture Recognition using Recurrent Neural Networks', ACM* pp. 237–242, 1991.

[12] Nielson J, *'Noncommand User Interfacer'*, In: Communications of the ACM, Vol. 36, No. 4, pp. 82–99, ACM, New York, 1993.

[13] Rheingold H, *Virtual Reality*, Summit, New York, 1991.

[14] Sung C, and Jones W C, *'A Speech Recognition System Featuring Neural Network Processing of Global Lexical Features'*, In: Proceedings of the 1990 International Joint Conference on Neural Networks, Vol. 2, pp. 437–440, Lawrence Erlbaum Publishers, 1990.

[15] Waibel A, Hanazawa T, Hinton G, Shikano K, and Lang K, *'Phoneme Recognition Using Time-Delay Neural Networks'*, IEEE Transactions on Acoustics, Speech, and Signal Processing, 37(3):328–339, 1989.