Embedding Speech into Virtual Realities

Christian-Arved Bohn and Wolfgang Krueger
German National Research Center for Computer Science (GMD)
Supercomputer Center (HLRZ)
Dept. *Scientific Visualization*
P.O. Box 1316
D-5205 Sankt Augustin 1, Germany
Tel. +49(2241)14-2230
Fax. +49(2241)14-2040
E-mail: {bohn, krueger}∂viswiz.gmd.de

Abstract

In this work a speaker-independent speech recognition system is presented, which is suitable for implementation in *Virtual Reality* applications.
The use of an artificial neural network in connection with a special compression of the acoustic input leads to a system, which is robust, fast, easy to use and needs no additional hardware, beside a common VR-equipment.

2 Introduction

Interaction has been proven to be one of the most important challenges for the future of nearly all computational tasks. Virtual Reality systems summarizes these under one subject. It may be a test for developed techniques, but also a mainspring for new research in this field.
Not leaving the user alone in a 'Virtual World' is the main intention and delivers the question: What are the best, the most intuitive, the most efficient interfaces between user and data-space ?
Thinking of the most often used 'real-world devices' automatically leads to speech recognition. To enable the human just to say, what he wants, is evidently a step away from the current user-interfaces, towards a more realistic man-machine interaction [14].
To get a practical speech recognition system, usable inside any VR-equipment, several conditions must be satisfied:

- Good recognition rate,
- Short delay time,
- Easy to use (easy adaptable to various speakers or configurable for any speaker),
- Avoidance of extensive resources (need to run in parallel with a VR hardware-software environment),
- Sufficient word capacity.

Therefore, emphasis was laid on *applicability* .
The descripted speech recognition system consists of the following moduls:

1. Built-in sound recording hardware of an *SiliconGraphics Indigo* workstation,
2. Fourier analysis module (software) for preprocessing speech input,
3. Compression module (software) for compression, generalization and for making the task robust,
4. *Artificial-neural-network* (software, *backpropagation*-network) for classification.

3 Speech Recognition with an Artificial Neural Network

3.1 Introducion

The ability to identify spoken words is desirable in a variety of application areas, such as manufacturing, telecommunication and medicine [15], but high-quality speech recognition systems are not easy to built. The challenging 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 artificial neural network (*ANN*) approaches to speech recognition tasks.

Most speech recognition systems are designed in a similar manner. Given is a continous data-flow, the sampled speech data, which is divided into parts, which stand for closed units of speech, like phonems, syllables, whole words or higher structures. This separation makes the data-flow suitable for classification, and can, in case of recognizing whole words for example, be done by detection of short pauses between the words.
The separation follows a preprocessing sequence, which converts the sampled acoustic data in a compressed form.

The classification then tries to assign the preprocessed input to certain classes, for example to a class of utterances of one certain word. The number of classes is limited, that means, increasing the number of recognizable words for example either decreases the recognition probability or needs additional structuring methods, which extends the classification.
So, speech recognition systems can be structured as follows.

- Preprocessing
  Mostly a Fourier transformation is used, in addition a cepstrum analysis can be used. The resulting spectrum is subsampled, leading to *melscale frequencies*. Finally, a certain kind of normalizing leads to so called *speech vectors*, which are of low dimension (10 - 20 elements).

- Classification
  The classification part determines from speech vectors certain symbols like phonems, words or other parts of the speech. There exist various neural methods, like multi-layer-perceptrons, recurrent-, time-delay- neural networks or feature-maps [8, 10, 20].

- Postprocessing
  In the simplest case, this may be just the assignment of the classification to a certain event, for example to words for word recognition. Some systems build higher levels of speech by the use of Hidden-Markov-Models or by combining several neural networks. This allows to recognize a arbitrary number of words by building higher structures of speech, like words from phonems or sentences from words or syllables.

In the most cases there is no need of a postprocessing part. Speech recognition systems are often used in very special applications, where only a limited number of words have to be recognized. The recognition of an arbitrary number of words is needed in *phonetic-typewriter* systems.

3.2. Realization

This paper presents a system that is one of the 'word-recognition-type'. It recognizes a limited vocabulary of spoken words. The implementation can be tuned to speaker independent speech-recognition or to one-speaker application. The system requires no additional hardware support for acoustic preprocessing.

3.2.1 Preprocessing

The sound were sampled with 16000 Hz in ambient noise conditions. Only the hardware of an standard *SGI-Indigo* were used.

In the first step, the preprocessing software automatically extracts the individual words from the speech signal under consideration. The developed extraction algorithm succeeds in finding the word boundaries with an error rate of less than 5%, which is quite satisfactory considering that no provisions have been taken to avoid environmental noise.

A 512-point fast Fourier transform analysis, computed every 10.9 milliseconds using a Hamming window [16] (2/3 overlap between successive windows), is then performed to obtain the short-time frequency spectrums of each extracted word.

Each spectrum is subsequently transformed into a 15-dimensional speech vector by integrating the (logarithmically scaled) spectral amplitudes centered at the following frequencies in Hz (taken from [6]; bandwith indicated in parentheses): 130 (30), 164 (38), 206 (48), 260 (60), 327 (76), 412 (96), 520 (121), 655 (152), 828 (192), 1040 (242, 1310 (305), 1650 (384).

n a final step, the sequence of speech vectors of each word is compressed in time by accumulating and averaging them until the sum of their distances exceeds a threshold, as proposed in [2]. When this threshold is reached, the sequence of speech vectors is replaced by the average value, leading to 10-17 of such 15-dimensional speech vectors for each word considered. These are used as the neural network input.

The use of a compression algorithm is one of the major differences of this approach to other word recognition endeavours [1, 4, 6, 11, 12, 19]. which seem to be based on the assumption that as much as possible of the speech information should be kept to achieve high recognition rates. However, compression does not only reduce the dimensionality of the neural network inputs, which enables the network to learn faster, but also provides a more uniform representation of the utterances, which seems to be beneficial for improving the generalization ability of the network.

3.2.2 Classification/Postprocessing

Used is an feed-forward, multi-layer perceptron. Several network architectures were studied with different numbers of hidden units/layers and parameter settings for the single speaker word recognition task. The net which gave the best performance results is used in all experiments. The resulting network architecture is a 3-layer feed-forward network with 240 input units, 18 hidden units and 45 output units. The input layer receives the speech vectors of one word ordered into a linear array and the output layer uses a simple 1-out-of-45 coding, where the output unit with the highest activation corresponds to the word recognized by the network.

The learning rule is the standard backpropagation algorithm with the following properties: a) the usual quadratic error function, as described in [8], is used; b) errors are accumulated after each input in the training set; c) the learning rate lies between 0.4 and 0.9; d) the momentum term is 0.7; e) the weights are initialized to random values in the range between -0.3-0.3; and f) the input vectors are always presented in the same order.

4 Artificial Neural Networks

4.1 Network-architecture

Artificial neural networks were developed to overcome the disadvantages of common algorithmical solutions of computational problems. Because of the excellent facilities of the human brain in solving almost all challenges, that the today's computers in a life time never could, people tried to simulate the functionality of human's neural network. The computer should be enabled to think more intuitively.

So, small programs raised from this thought, which simulates one neuron (*unit*) put together by a network of simulated connections.

The task, one neuron has to do, is to sum up a number (k) of weighted (w) inputs (x) and to deliver the result y to the input of connected neurons (1).

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

g is called *activation-* or *gain function* with its activation-threshold μ. The most implementations use (2).

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

It is a *sigmoid -function.* ß determines their steepness.
Putting m of such units together 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 ($t+1$).

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

So an ANN is defined by its topology (w)) and the kind and parameters of the activation function.

In this work a *multi-layer-perceptron* [8] is used. It is an feed-forward network with a certain number of input- and output-units.

4.2 Network-programming

The implementation of an ANN-simulation can be seen as program that simulates a stupid brain. To make it executing a wished function, it has to learn.

This learning is managed by presenting an event to the net and telling him what it is - especially, giving an input and the associated output, the network should try to adapt his own output (*supervised learning* [8]).

So, beside the net simulation program, there exists a second learning program (*Backpropagation*-algorithm (*BP*) ,[8, 5]). It does the learning by modifying the ANN's topology (w) and the parameters of the gain-function. A sequence of input-output pairs is presented to the ANN. BP calculates an error from the real and the wished output and the net parameters are corrected to a decreasing error.

This is done for the whole set of learning pairs several times, hoping that the net error sinks into a global minimum.

5 Application

5.1 Example

To give an impression of the problem let's see a short example. Figure 1 shows the sampled raw data of an utterance of the german word /quadrat/. This transformed in uncompressed speech vectors looks like figure 2. After compression there is just figure 3 left.

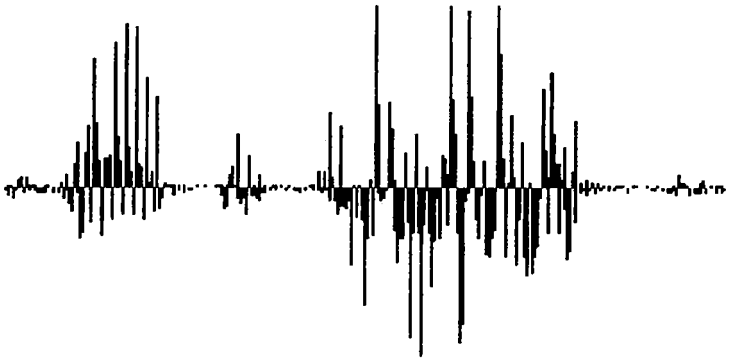

Figure 1 Sampled word /quadrat/.

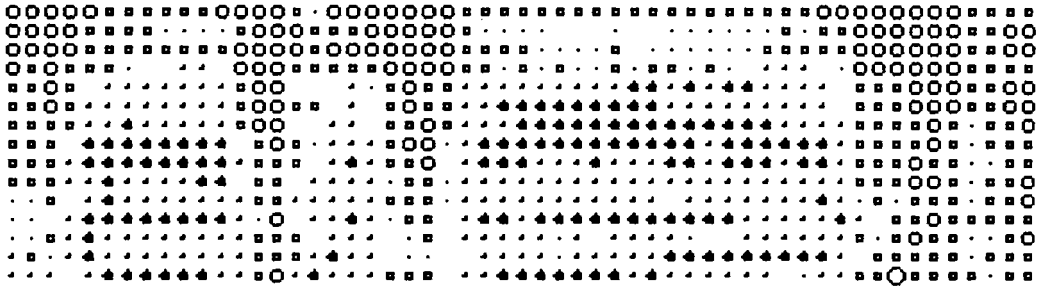

Figure 2 Word converted to speech vectors.

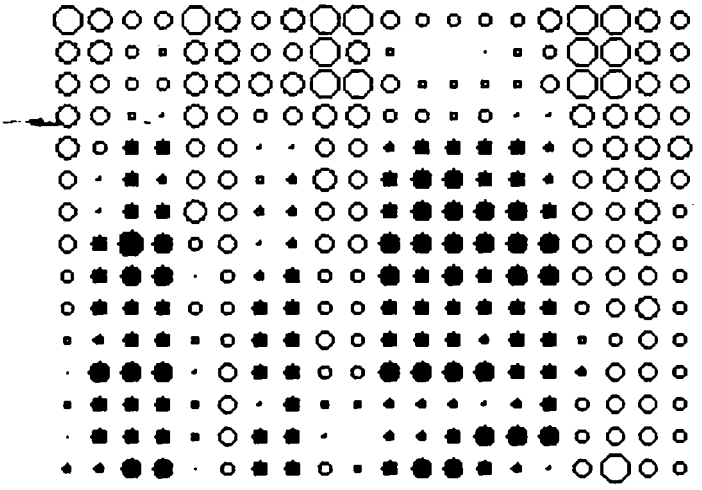

Figure 3 Speech vectors after compression.

5.2 Tests, Results

Training the net always happens in the same manner. There is a limited number of words, that the network is trained with, the *learn-set*. A second *test-set*, containing the same words but different utterances, is only applied, without any effects to the net parameters. The recognition rate of this foreign, unknown words is taken as a convergence criterion; it shows the generalization ability of the network, and may be seen similar to an application of the trained net after learning.

The net converges fast, at most in less than one minute. To learn a number of words, the speaker has to speak the application-specific vocabulary about five times - waits a few minutes, and then can use his trained network in an arbitrary implementation. Recognition time after learning is less then one second.
In contrast to other speech-recognition approaches, the system showed good robustness and applicability. Convergence appears in almost all cases, there is no need of support of specialists or people, familiar with the implementation. The following tests prooved usability of the system.

5.2.1 Single-Speaker Word Recognition

Goal was to train the network to 'understand' the vocabulary of one speaker, to have a higher recognition rate, than in the multi-speaker case. Each word was spoken four times. The net learned three different utteranes of 45 words. As test of generalization ability the last word-set were applied. The net learned all words from the learn-set correctly. The never-heard words of the test-set were correct identified in 96% of the cases.

5.2.2 Speaker-Independent Word Recognition

In contrast to trimming the net to the speech of one certain speaker, the training time when changing the user can be cancelled out by training the network to arbitrary users. The lower recognition rate can be neglected, looking to the advantages of a speaker(user)-independent application.

The utterances of 45 words of 16 speakers were used. 8 sets were defined as learn sets, the other 8 as test sets. The ANN learned again all utterances. The generalization ability converged to 72%. A similar test with only male speakers increased recognition-rate to 91%.

5.2.3 Speaker Recognition

The last test should show the typical flexibility of an ANN-system. It should learn speaker-recognition. Learn sets of each 30 words of five speakers were created. The test sets contained 10 other words of the same five speakers. After learning, hearing these unknown words, the system should say, which of the speakers this utterances did. The net could recognize the person with an exactness of 82%.

5.3 Observations

The typical behaviour, which are exspected if using ANN's could be seen.

- The hidden layer creates an internal coding of the net input, that means their output-values mostly have certain levels, similar to binary values.
- There is a better generalization-ability with *not exact learning* by either a) using less hidden units or b) stopping learning before final convergence. In most cases the best generalization is reached if 80% of the input words are learned.
- The learning behaviour is robuster for a smaller net. That means less input units and so less weights in the first layer accelerate learning. The convergence curve was more smooth and the convergence faster - a further reason for using the above mentioned compression-algorithm.

6 Conclusion

Our work demonstrates the practical feasibility of building a high quality speaker-independent speech recognition system, which can easily be trained to recognize the desired words with high accuracy and does not require the assistance of somebody, who is deeply familiar with the issues involved in speech processing.

The system has purely been implemented in software and is quite competitive to other approaches. Through the special design by using a certain compression method of the input data and an artificial neural network, a system has arised, which is well suited for applications where a limited vocabulary needs to be recognized and where using has to be fast, robust and uncomplicated, like in a Virtual Reality application. The time for running the preprocessing software and training the network with the backpropagation algorithm is pretty short (about one minute), and the recognition delay time lies under one second.

The system could also be used for speaker-recognition by simply training the network to learn the mapping between a set of words and a number of speakers. This shows a great flexibility, which is also an important feature for Virtual Reality applications, where one can imagine alternative noise-steering methods.

7 Discussion, Future

Actually the system has been implemented into several VR-applications, that includes a standard VR-system, but also an application with the *BOOM2* . Of course, it can be combined with every subset of VR-equipment. Almost steering tasks in addition or in case of gesture-recognition is tested and gives at most a better feeling in moving through the data-space.

It reveals that gestures, used for moving, are very far beyond the human intuitiveness. Simply saying 'back' for example seems to be easier than stretching or bending some middle (which ?) finger.

A current VR-project called *Responsive Environment* [9] will be a very attractive application. Subject of this project is a workbench, where many people can in parallel interact, work together. The project simulates the ordinary working-environment of architects designers and surgery crews. Such a cooperative work can greatly be assisted by tools like speech recognition. Even this case is, because of its overlapping acoustical events, a great challenge for a speech recognition system.

8 Bibliography

[1] Behme H: *A Neural Net for Recognition and Storing of Spoken Words*, In: Parallel Processing in Neural Systems and Computers, pp. 379-382, Elsevier Science Publishers, 1990.

[2] Bengio Y, Cardin R, and De Mori R: *Speaker Independent Speech Recognition with Neural Networks and Speech Knowledge* In: Advances in Neural Information Processing Systems , Vol. 2, pp. 218-225, Morgan Kaufman Publishers, 1990.

[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] Grajski K A, Witmer D P, and Chen C,: *A Preliminary Note on Static and Recurrent Neural Networks for Word-Level Speech Recognition*, In: Proceedings of the 1990 International Joint Conference on Neural Networks , Vol.~2, pp. 245-248, Lawrence Erlbaum Publishers, 1990.

[7] Hampshire II J B, and Waibel A: *Connectionist Architectures for Multi-Speaker Phoneme Recognition,* In: Advances in Neural Information Processing Systems , Vol. 2, pp. 203-210, Morgan Kaufman Publishers, 1990.

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

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

[10] Kohonen T: *The Neural Phonetic Typewriter,,* IEEE Computer , 3:11-22, 1988.

[11] Kowalewski F, and Strube H: *Word Recognition with a Recurrent Neural Network,* In: Parallel Processing in Neural Systems and Computers , pp. 390-394, Elsevier Publishers, 1990.

[12] 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.

[13] Lee Y, and Lippmann R P: *Practical Characteristics of Neural Network and Conventional Pattern Classifiers on Artificial and Speech Problems,* In: Advances in Neural Information Processing Systems , Vol.~2, pp. 168-177, Morgan Kaufman Publishers, 1990.

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

[15] Peacocke R D, and Graf D H: *An Introduction to Speech and Speaker Recognition',* IEEE Computer , 8:26-33, 1990.

[16] Rabiner L R, and Gold B: *Theory and Applications of Digital Signal Processing,* Prentice-Hall , 1975.

[17] Rigoll G: *Neural Network Based Continous Speech Recognition by Combining Self Organizing Maps and Hidden Markov Modelling,* In: Lecture Notes in Computer Science , Vol. 134, pp. 58-65, Springer-Verlag, Berlin, 1990.

[18] Rumelhart, D E, Hinton, G, and Williams, R E: *Learning Internal Representations by Error Propagation,* In: Parallel Distributed Processing: Explorations in the Microstructures of Cognition , Vol. 1, 318-362, MIT Press.

[19] 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.

[20] Waibel A, Sawai H, and Shikano K:, *Modularity and Scaling in Large Phonemic Neural Networks,* IEEE Transactions on Acoustics, Speech, and Signal Processing , 37(12):1888-1889, 1989.

[21] 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.