A FAST METHOD OF PRODUCING TALKING HEAD MOUTH SHAPES FROM REAL SPEECH

A. P. Breen[1], O. Gloaguen[2], P. Stern[2]

BT Labs.[1], Ecole Nationale Superieure des Telecommunications de Bretagne[2]

ABSTRACT

The subject of computer generated virtual characters is a diverse and rapidly developing field, with a wide variety of applications in industries as varied as entertainment, education and advertising. Many of these applications require or would be greatly enhanced by having the virtual characters speak with the recorded voice of a real person. Such an ability is particularly useful in applications where users are interacting via avatars in real time in a virtual world.

There are three basic problems which need to be addressed when developing an interface which has this functionality:

- The process must be capable of animating mouth shapes in real time.
- The process should not mouth extraneous sounds such as music, doors slamming etc. To do so would diminish the effectiveness of the illusion.
- The mouth shapes produced by the avatar should approximate that of the speaker.

This paper describes a series of experiments which attempt to address each of the points outlined above. The experimental procedures are based around a real time low computation approach which relies on a particular variety of neural network known as the Single Layer Look Up Perceptron (SLLUP).

1. OVERVIEW

The work described in this paper is based on the results of two six month student projects conducted at BT Laboratories. The aim of the projects were to investigate a means of developing a low cost, near real time method of extracting parameters from a speech signal which could then be used to drive a real time talking head [3]. The method based on an the concept of intermediate representations [1]. Huckvale [1] demonstrated that an MLP could be used to transform a speech signal into a phonetically motivated Intermediate Representation IR. The IR consisted of a set of binary features, (e.g. absence of speech signal, presence of frication, presence of voicing). It was suggested that from this intermediate representation it would be possible to perform either speech synthesis or recognition. Figure 1 shows his "Y-model" of speech representation relations.

The concept of an IR was retained in this work, as was the use of neural networks (SLLUP) for feature extraction (analysis). However, rather than using the phonetically motivated features described by in [1], an alternative set of features more appropriate to the generation of visemes was devised.

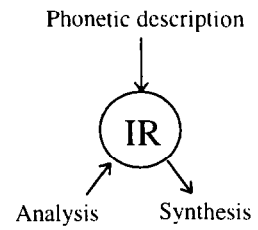

Figure 1: The "Y-model" of Speech Representation Relations.

2. THE SLLUP

Like all neural networks, the SLLUP can be seen as a vector transformer, in which the transformation is learnt, as shown in figure 2. A training sample X is applied to the system and a target output vector T is shown to the system at the same time. The difference between the actual output Y and the target is used to modify the internal parameters of the system so that the actual becomes more like the target. This is done using a gradient descent procedure.

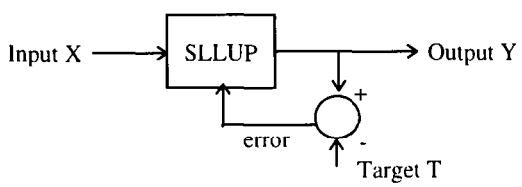

Figure 2: The learning procedure of the SLLUP.

The vector transform performed by the SLLUP has the form shown in figure 3. The input vector is encoded as an image of black and white pixels formed by bits of the code representing the scalar elements of the input. The code may be bar chart, Gray code, binary code or some other code. Random connections are made onto the pixels of the image and groups of 'n'connections are formed into n-tuples which are used to address a large number of memories, the RAMs. These RAMs themselves are grouped into neuron blocks and the outputs of

all the RAMs in the i^{th} block are added to form the value of the i^{th} element of the output vector.

The system is trained by applying a vector to its inputs. The connections onto the coded image of the input produces a specific set of addresses. As an example, in figure 4, only one 3-tuple is used to address a RAM. The input presented to the system generates the address "011" where each bit represents the value of the pixel on which the connection has been made. We can say that the address "011" has been selected.

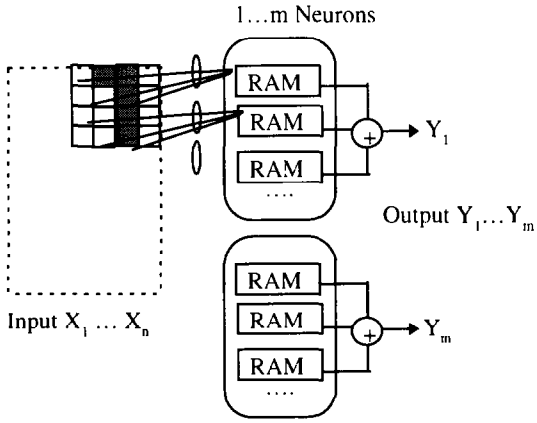

Figure 3: The vector transform performed by the SLLUP.

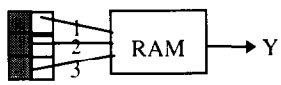

Figure 4: The "Selection" of an address in a RAM. Active cells (shadded) are signified by a '1', while inactive cells are signified by '0'. Hence in this example the 3-tuple connections result in the address '011'.

The summation of the contents of each group of RAMs produces the elements of the output vector according to the principle depicted in figure 3. The output is then compared with the desired output (the target) and the error vector is used to modify the values of the currently addressed RAM locations so that next time the same input vector is applied, the output is nearer to the desired output.

Repeated applications of different training vectors allows the system to learn the required input-output mapping function. It is important to notice that with appropriate choice of n-tuple order and number of RAMs in each neuron block, the system can estimate the best function to fit a rather sparse training set. I.e. it is not necessary to expose the machine to all possible input-output vector pairs because it is able to interpolate the required function between training points.

The SLLUP neural network was chosen because it has been demonstrated [4] to have similar properties to multilayer perceptrons, but learns much faster and is guaranteed to converge. It also has the advantage of being computationally simple. However, the specific nature of the discriminator is not important, any algorithm with similar properties to those stated above would suffice.

3. THE EXPERIMENTS

Two sets of experiments were conducted, both used the same general experimental procedure but differed in the training and test set used, the type of parameterised speech data presented to the SLLUP and the complexity of the IR produced by the SLLUP.

3.1. Experimental 1

The first experiment attempted to derive a complex IR from a database of clean speech. In this experiment, the SLLUP was presented with a vector of MFCC coefficients. MFCCs were considered because of their success as a feature vector within HMM based recognition systems.

The speech database consisted of 200 phonetically rich phrases recorded in a studio. This data was digitally sampled at 8kHz and stored as separate files on a computer. Each file had an associated annotation file which contained a time aligned transcription of the utterance. The transcription system was strictly phonemic and consisted of 42 phonemic symbols.

A nine element MFCC input vector was generated from the speech data at a frame rate of 32ms with an overlap of 16ms. The procedure was as follows. The sampled speech data was divided into overlapping blocks of 256 samples. Each block was pre-emphasized using a filter coefficient of α equal to 0.95. An FFT was then used to calculate the power spectrum which was then organized into frequency bands according to a series of 19 MEL scale filters. Finally, a Discrete Cosine Transform (DCT) was performed to produced 8 MFCC coefficients and the spectral energy.

A five feature intermediate representation was devised based on a model of coarticulation. The five features differed from those proposed in[1] in that they were not binary features, but could take on a range of values as shown in table 1.

Feature labels	Range
Lip (mouth shape)	rounded → neutral → open; 0 → 0.5 → 1
Tongue 1	close → open; 0 → 1
Tongue 2	front → central →back; 0 → 0.5 → 1
Pursing	neutral → pursed; 0 → 1
Jaw opening	closed → open; 0 → 1

Table 1: List of the five features used to represent the speech signal in the IR.

Each of the five features represented an output from the SLLUP. When presented with MFCC vectors, the SLLUP would produce a constant stream of output. These outputs were then transformed via a lookup table into a set of visemes which were then used to control a talking head.

For the SLLUP to train correctly, a set of target data was needed which accurately represented the expected output from the system. This data was generated from the annotation data associated with each file. Label and timing information was used as input to a model of coarticulation derived from a method proposed by Cohen and Massaro[6]. The output from this model was a time varying set of articulatory parameters. These parameters were simply mapped onto the five features presented in table 1. The SLLUP was trained and tested using the parameter values shown in table 2.

Parameter Label	Value
Length of the context window	3
Coefficient of step decent	0.003
Number of RAM connections	500
Precision in the image quantification	10
Dimension of tuple	8
Dimension of the 3 window frame	27
Percentage of frame overlap	50%
Encoding	Gray

Table 2: Complete list of parameter values used during the training and testing of the SLLUP in experiment 1.

Using the training data described above, a SLLUP with the specifications outlined in table 2, converged after only 20 iterations. Further iterations produced no significant reduction in the RMS values.

The results suggested that while the SLLUP was able to "remember" the training data, it was not able to generalize from it. Worse still, the ability of the SLLUP to remember the training data was related to the amount of memory used by the SLLUP. In other words the SLLUP was acting as a table.

It was considered that a possible explanation for the poor performance of the SLLUP was the type of input vector used. MFFC's while an appropriate transform for HMM's may not be the best representation for a neural network. An alternative input vector was designed which used the 19 channel output from the MEL scale filters in preference to the MFCC output.

Initial results using this feature vector demonstrated that the SLLUP was capable of learning, and that a plausible set of mouth movements could be produced.

At this point, it was decided to take a step back and perform a more rigorous set of experiments on the SLLUP using more realistic speech data and a simpler intermediate representation.

3.2. Experiment 2

As mentioned in the preceding section, the results of driving a SLLUP with the output from a 19 channel Mel scale filter bank produced promising results. This was in line with the observations reported in [1]. However, these results were based on experiments using speech recorded in a studio, produced by a high quality microphone and with no significant extraneous noise. In any realistic application, the performance of the algorithm would be judged as much on its ability to ignore extraneous noises as it would on the accuracy of the mouth shapes produced.

A new, harsh, database was selected which contained a high degree of background noise. This database consisted of 4800 recordings of speakers using the UK telephone trunk network. The data contained continuous speech (sentences), non-speech (i.e. background noise, coughing, sneezing, etc.) and silence. A portion of the database was set aside for testing. Due to the size of the database the data was divided into five sub-sets labeled (a) to (e).

The speech data was stored on a computer at 8kHz and split into a number of data files. Each data file had an associated annotation file, which contained a time aligned phonetic transcription extended to include non-speech sounds. For the purposes of these experiments, the annotation data was converted into greatly simplified descriptive set as shown in table 3. The simplified labels also serve as the features used in the intermediate representation.

Original Database Labels	Simplified Labels
LIN: Silence (or noise)	Silence
HIS: Periodic impulse noise	Non-speech
OTN: Other noise	Non-speech
PSN: Pre/post speech noises (lip-smacks, coughs, etc.)	Non-speech
BRT: Breath filled pause	Non-speech
IMP: Impulsive noise	Non-speech
EXS: Extra speech (e.g. background speaker)	Speech
All other labels are consider as speech	Speech

Table 3: Table describing simplified label set.

The SLLUP was trained and tested using the parameter set shown in table 4 unless otherwise stated.

Parameter Label	Value
Length of the context window	3
Coefficient of step decent	0.003
Number of RAM connections	200
Precision in the image quantification	16

Dimension of tuple	12
Dimension of the 3 window frame	57
Percentage of frame overlap	50%
Encoding	Bar Chart

Table 4: Complete list of parameter values used during the training and testing of the SLLUP in experiment 2.

The result of training of on a sub-set of the database is shown in table 5:

Number of iterations	%error (sub-set 'a')
5	20.81
10	14.38
20	8.31
30	4.72
40	3.43
50	2.42
60	2.28
120	1.92

Table 5: Overall % error of files used during training, (*step descent 0.005, 10_tuple*).

Table 5 shows that, as with experiment 1, the majority of the training had been completed by the 20th iteration. However, further training produced a noticeable reduction in error but with an increase risk of over training.

The results of this experiment are shown in figure 5 as a 3x3 confusion matrix.

	Silence	Non-speech	Speech
Silence	13199	73	1636
Non-speech	1380	145	1394
Speech	1678	105	32725

Figure 5: 3x3 confusion matrix showing the classification of silence, non-speech, and speech sounds produced by SLLUP.

4. CONCLUSIONS

This paper has briefly presented the results of two experiments designed to investigate the practicality of using a neural network, in this case a SLLUP, as a method of extracting features from a speech signal. It was further suggested that these features could be used as an intermediate representation which could be further transformed into a set of visemes needed to drive a real time talking head. Results from the first experiment suggested that, provided an appropriate input representation was used, near real time generation of approximate mouth movements was possible. However, this experiment was conducted on a limited set of training and test data, which would not necessarily represent the type of data observed in real applications. In addition, further research is needed into the accuracy and acceptability of the mouth shapes produced. The second experiment attempted to perform a simpler discrimination task on a much larger body of training data which better represented the type of speech data present in real applications. This experiment showed that the SLLUP was capable of learning in a very short number of iterations, with an overall error of less than 10%. However, while the SLLUP was able to discriminate silence and speech or non-speech sounds, there was strong confusion when non-speech data had a long term power spectral density and periodicity similar to speech. Initial studies suggest that a reduction in overlap would improve this performance. A finer classification of non-speech sounds may also produce better results. However, a high degree of confusion over "speech like sounds" was expected.

5. REFERENCES

1. Huckvale, M. "Speech Analysis/Synthesis Using a Trained Intermediate Representation," 7th FASE Symposium, Edinburgh, p 867-874, 1988.
2. Tattersall, G.D., Johnston, R.D., and Foster, S.. *Speech and Language Processing: Single Layer Look-Up Perceptrons (SLLUPs)*, Edited by C. Wheddon and R. Linggard, Chapman and Hall Publishers, 1990.
3. Breen, A. P, Bowers, E., Welsh, W.," An Investigation into the Generation of mouth Shapes for a Talking Head," ICSLP '96, 1996.
4. Gloaguen O., "Synthetic Persona: Real Time Animation of a Talking Head Using a Neural Network," Final Year Project Report for Ecole Nationale Superieure des Telecommunications de Bretagne, 1996.
5. Stern, P., "Real Time Extraction of Speech Parameters Using a Neural network ," Final Year Project Report for Ecole Nationale Superieure des Telecommunications de Bretagne, 1997.
6. Cohen, M., Massaro, D., *Modelling Coarticulation in Synthetic Visual Speech*, Tokyo:Springer, In N. M. Thalmann & D. Thalmann (eds.), *Models and Techniques in Computer Animation*, p 139 - 156,1993.