Multimodal Interfaces

ALEX WAIBEL[1,2], MINH TUE VO[1], PAUL DUCHNOWSKI[2] and
STEFAN MANKE[2]

[1] *School of Computer Science, Carnegie Mellon University, Pittsburgh,*
PA 15213-3890, U.S.A.; [2] *University of Karlsruhe, Computer Science Department,*
ILKD, Am Fasanengarten 5, 76128 Karlsruhe, Germany; Email: ahw@cs.cmu.edu,
tue@cs.cmu.edu, paul_d@ira.uka.de, manke@ira.uka.de

Abstract. In this paper, we present an overview of research in our laboratories on
Multimodal Human Computer Interfaces. The goal for such interfaces is to free
human computer interaction from the limitations and acceptance barriers due to rigid
operating commands and keyboards as the only/main I/O-device. Instead we move
to involve all available human communication modalities. These human modalities
include Speech, Gesture and Pointing, Eye-Gaze, Lip Motion and Facial Expression,
Handwriting, Face Recognition, Face Tracking, and Sound Localization.

Key words: multimodal interfaces, speech recognition, lip-reading, gesture recognition, handwriting recognition, neural networks.

1. INTRODUCTION

Recent developments in the computer and communication industries are accelerating the pace and variety of forms in which information is delivered to users
worldwide. This, in turn, multiplies the problems associated with managing and
interacting with the new wealth of data and information. The combination of
sound, images, and text is already available on Multimedia PC's and publishing
companies are advancing their goal of delivering multimedia information to
everyone as the "Information Super-Highway" unfolds. While the multiplicity
and amount of information expands, ways to access it or communicate with it
remain limited. Relatively primitive input devices and interfaces, such as keyboard
and mouse still dominate as interfaces. In contrast, human-to-human communication takes advantage of a wealth of hints and signals, explicit or implicit, that
are lost in human-computer interaction. Meeting "face-to-face", having "eye-contact", "reading one's lips", "handwaving", "pointing one's finger at someone
or something", as well as "plain talk" make human communication richer than
simple text transmittal. Speech and writing represent perhaps the most direct
expressions of language, but are routinely complemented by other mostly visual
modalities (e.g., "body-language"). These will need to be processed also if one
hopes to achieve truly natural human-computer interaction. To increase their
effectiveness, human-computer interfaces should, therefore, include and combine
visual as well as spoken or textual language to represent the full spectrum of
human communication.

In this paper, we present our efforts toward developing richer human-computer and computer mediated human-human interfaces that attempt to embrace and take advantage of *all* communication modalities. The INTERACT project involves a number of research projects in progress at our labs at Carnegie Mellon University in Pittsburgh, U.S.A., and University of Karlsruhe, Germany. They are aimed at interpreting the visual and acoustic instantiations of language as we use them to communicate day to day. The modalities of interest in our work at both labs are: speech understanding (and translation), sound source localization, gesture recognition, lipreading, handwriting recognition, eye- and face tracking. Our research involves improving recognition accuracies of the modality specific component processors, as well as development of optimal combination of multiple input signals to deduce user intent more reliably in cross-modal "speech"-acts. More specifically, we aim to combine visual, acoustic and textual cues, including:

- Speech recognition with lipreading for more robust recognition
- Gesture with speech for multimodal interpretation
- Speech with Handwriting for more flexible, redundant input
- Face- and eyetracking with sound source localization for robust speech extraction in adverse acoustic environments (the cocktail-party effect)
- Face- and eyetracking with speech recognition for focus of attention (who is the user talking to? What is he/she referring to?)

We have begun attacking these advanced goals along a rather broad front of activity. The present paper reviews where we currently stand in three of these subtopics: lipreading (as a complement to speech recognition), gesture recognition (as a complement to speech) and on-line handwriting character recognition. Wherever possible, we develop learning strategies (mostly connectionist and statistical), to ensure scalability and portability to larger and different application domains. In the following we discuss these three efforts and report on initial results of cross-modal integration.

2. AUTOMATIC LIP-READING AND SPEECH RECOGNITION

Lip movement is a visual information source tightly and synchronously coupled to the acoustic speech act and hence can be naturally viewed as an integral part of a speech recognition effort. This is in contrast to other communication modalities described later in this article, such as gestures or handwriting, which may be invoked independently of speech.

Most approaches to automated speech recognition (ASR) that consider solely acoustic information are very sensitive to background noise or fail totally when two or more voices are present simultaneously (cocktail-party effect), as often happens in offices, conference rooms, outdoors, and other real-world environments. Humans deal with these distortions by considering additional sources such as directional, contextual, and visual information, primarily lip movements. This latter source is subconsciously involved in the recognition process and is used extensively by hearing-impaired individuals, but also contributes significantly

to normal hearing recognition. The usefulness of lip movement stems in large part from its rough complementariness to the acoustic signal: the former is most reliable for distinguishing the place of articulation (ex. Jackson 1993), the latter conveys most robustly manner and voicing information (ex. Miller and Nicely 1955).

The task of exploiting lip-reading in an automatic speech recognition system requires the solution of two conceptually distinct but not independent problems: suitable representation and recognition of the visual signal and the integration of thus obtained visual evidence with the acoustic side. Clearly, a given type of visual pre-processing will constrain the options available for further combination of the two sources.

2.1. Related Research

The first significant attempt to supplement acoustic ASR with lip-reading was the system built by Petajan and applied to a speaker-dependent isolated-word (vocabulary of 100 words) recognition task (Petajan 1984). Four static features were extracted from each image frame and a linear time-warping procedure was used to identify the most probable word. By combining the output of the optical recognizer with that of a commercial ASR system the recognition rate was improved from 65 to 78%. In a follow-up effort (Petajan *et al.* 1988) simplified optical processing was used to achieve near real-time performance. The image of the speaker's mouth area was captured by a camera and lighting system installed in a head-mounted harness, circumventing some image-processing problems. The combination of optical and acoustic decisions was achieved via a set of 30 heuristic rules. The overall performance was similar to the earlier system.

Mase and Pentland (1991) chose to parameterize the oral cavity image by computing average optical flow vectors in four regions of the picture designed to capture the movement of particular facial muscles. The regions were selected manually by the experimenters. They used template matching (on optical data alone) to recognize strings of three to five digits from three speakers. Average word recognition rate was roughly 75%.

Neural networks were used by Yuhas *et al.* (1989) with both optical and acoustic input to distinguish among 9 vowel phonemes under varying acoustic signal-to-noise ratio (SNR). Only static images (not sequences) were used as the optical input. Furthermore, the relative contribution of visual and acoustic information was adjusted according to the SNR by an "omniscient controller" (i.e., the value of the SNR is explicitly given). The visual input was shown to compensate for noise-induced performance drop in purely acoustic recognition.

Stork *et al.* (1992) measured the position of ten reflective markers placed on the lips of the talker thus significantly simplifying the issue of optical data capture. From these measurements they derived five parameters as the visual input. Separate Time Delay Neural Networks (TDNN) processed acoustic and optical data to render a decision on one of 10 English letters spoken in isolation. Visual and acoustic-alone recognition was 51% and 64%, respectively. By combining

the outputs in a Bayesian framework, they achieved overall performance of 91%.

Goldschen (1993) used 13 visual features extracted from processed image frames acquired with a head-mounted camera as in (Petajan *et al.* 1988) to identify one out of 150 possible TIMIT sentences spoken by a single talker. It appears that the sentences were treated essentially as very long words in this setup. Vector quantization of the input allowed the use of discrete Hidden Markov Models (HMM) in the recognition process. The system using generalized "triseme" models achieved 25% recognition rate (visual information only).

2.2. *Initial System*

Our integrated acoustic/visual continuous-speech ASR system was first reported in (Bregler *et al.* 1993). It was developed for a spelling task using the German alphabet. Training and test utterances comprise spelled (without pauses) names and nonsense letter sequences of arbitrary and unknown to the recognizer lengths. The task is thus equivalent to continuous recognition with a small but highly confusable vocabulary.

2.2.1. *System Description*
We record acoustic and visual data in parallel and pre-process them as illustrated in Figure 1. The acoustic signal is sampled at 16 kHz with 12-bit resolution. A fairly standard front-end then divides the data stream into overlapping windows (Hamming windows) at 10-msec frame rate and transforms the data to frequency domain by computing 16 numbers (the so-called Melscale Fourier coefficients) for each frame.
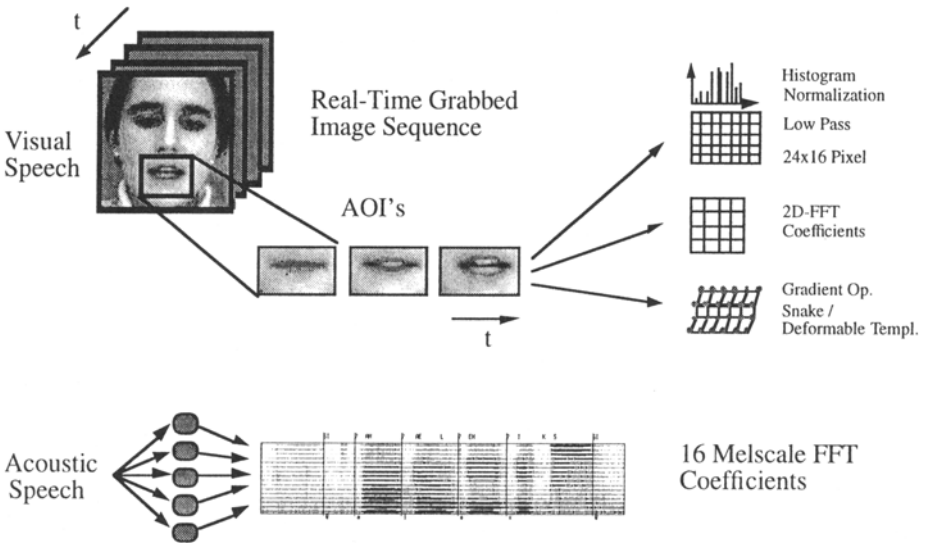

Fig. 1. Bimodal Data Acquisition for Speech Recognition and Lip-reading.

Visual data was initially captured as 256 × 256 pixel images with 8-bit gray level resolution per pixel. A smaller area of 144 × 80 pixels centered on the mouth was then manually extracted and, after low-pass filtering, downsampled to a 24 × 16 pixel image. The resulting 384 gray level values were then normalized for each frame to lie between −1.0 and 1.0 and constituted the visual evidence available to the classification algorithms.

We make use of a concept called "viseme" in our system. A "viseme" is the rough correlate of the phoneme, i.e., the smallest visually distinguishable unit of speech. In general, and for our purposes, visemes are defined by a many-to-one mapping from the set of phonemes. This reflects the fact that many phonemes (for instance: /p/, /b/, /m/) are essentially indistinguishable from visual information alone.

We use a modular Multi-State Time Delay Neural Network (MS-TDNN) (Hild and Waibel 1993) to perform the actual recognition. Figure 2 shows the architecture. Through the first three layers (input-hidden-phoneme/viseme) the acoustic and visual inputs are processed separately. The third of these layers produces activations for 65 phoneme states on the acoustic side and 42 viseme states on the visual side. In our system phoneme-to-phoneme (and thus viseme-to-viseme) transitions were included as separate phone (viseme) states.

The outputs of the phoneme and viseme layers are integrated in the units of the combined layer (this layer does not exist in the basic MS-TDNN). The activation of each combined phone-state is the weighted sum of the activations of the corresponding phoneme-state and viseme-state units. Some visemes will therefore influence more than one of the combined layer units. In the final layer (which copies the activations from the combined layer) a one stage Dynamic Time
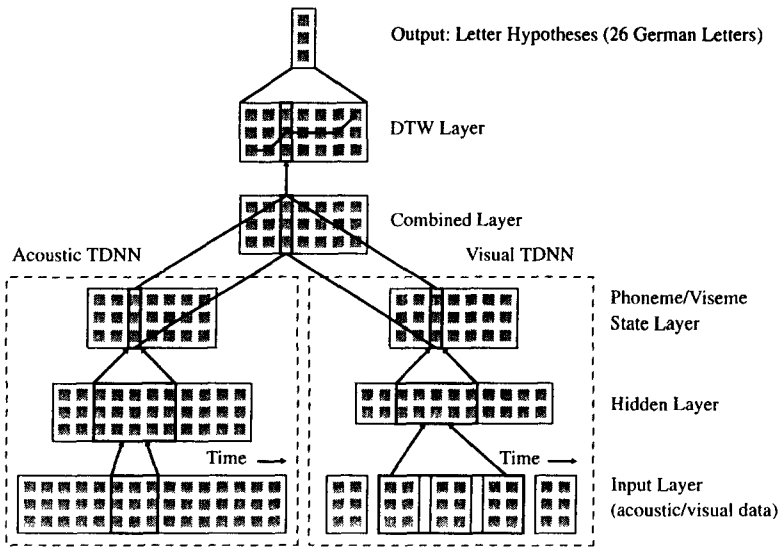

Fig. 2. Network Architecture for the Audio/Visual ASR System.

Warping algorithm (Ney 1984) is applied to find the optimal path through the phone-hypotheses that corresponds to a sequence of letter models.

Network training is done in two phases. First, the acoustic and visual sub-nets are trained separately to fit phoneme/viseme targets. Second, the complete network is trained to fit letter targets. Error backpropagation is used to find the connection weights resulting in optimal recognition of the training data. The weights determining the combination of the two sub-nets are not trained this way, rather they are computed dynamically during recognition to reflect the apparent reliability of the sub-net outputs. Specifically, the activations of the phoneme and viseme layers are normalized to represent probabilities and the entropy of each sub-net's output is computed. Low entropy signifies probability concentrated in a few units, i.e., relative confidence in the respective sub-net's identification. Conversely, high entropy corresponds to near equal probability of most phonemes or visemes. Accordingly, we symmetrically weight the acoustic and visual contributions to the combined layer in inverse proportion to their respective entropies.

2.2.2. Results

Table 1 shows recognition performance originally achieved on a speaker-dependent task (Bregler *et al.* 1993). Training data consisted of 75/200 training and 39/150 testing sequences for two speakers (identified by their initials msm and mcb in Table 1). Misclassified, omitted and inserted letters were counted as errors. The rows marked with "noisy" correspond to experiments in which the acoustic data was corrupted with broadband noise until the acoustic-alone performance was significantly reduced.

Table 1. Word accuracy (in %) of Speech/Lip System.

Speaker	Acoustic	Visual	Combined
msm/clean	89	32	93
msm/noisy	47	32	76
mcb/clean	97	47	97
mcb/noisy	59	47	70

The results show that adding visual information can significantly boost overall recognition rate despite the relatively poor performance on visual input alone. The improvement is naturally most evident when the acoustic speech is noisy.

Further experiments were carried out on a database of 6 speakers (2 female, 4 male) to test the performance on a multispeaker task (Bregler 1993). 80 sequences per speaker were used for training and 10 for testing. Visual-alone mode achieved only 12% word accuracy. Nonetheless, by adding it to the acoustic-mode we reduced the error rate by almost 9% for clean speech and 29% for 30 dB SNR.

2.3. *Current Development Directions*

At present we are seeking to improve the performance of the system on the letter spelling task, with the view of extending it to continuous speech recognition. We have been concentrating on the visual side of the system since the acoustic technology is much more mature at this point.

2.3.1. *Robustness*

In a practical system, manual extraction of the mouth region from the face image is not acceptable. As a first step away from this method we have recorded new data where the speaker is asked to position himself such that his lips are visible within a rectangle shown on the workstation screen. The image in that frame is then used directly.

We have been experimenting with this system to understand the principal weaknesses and sources of "fragility". Contrary to initial suspicion, the processing appears relatively insensitive to reasonable variation in lighting conditions. We have implemented a different version of the gray-level normalization procedure that further protects the performance under varying average image brightness. Severe illumination gradients would still pose a problem and might be alleviated through adaptive histogram equalization. However, this would significantly increase the computational load.

The shift of the lip image within the frame has been found to cause a more serious degradation as shown in the following experiment. We trained the network on 180 newly recorded sequences from one speaker. The images in the training sequences that the network recognizes perfectly were then diagonally shifted within the frame by three pixels. The direction of shift was chosen at random for each successive image (even in the same sequence). This shift is equivalent to the speaker moving his face by only about 2 millimeters. Yet the word accuracy dropped to 87%. With a six-pixel shift the recognition was 66%. Even more severe losses were observed when the shift was effected in a constant direction.

We are investigating several approaches to counteract this effect. First, we are designing a detector to automatically and precisely locate the lips within a picture. In addition to compensating for the likely shifts, this would obviate the need for the speaker to hold his head in a constant orientation. Initial, speaker-dependent results indicate that a neural network detector can reliably locate the lip region under varying image size, lighting and backgrounds.

To further increase robustness, we are training the visual TDNN on several copies of the training images artificially shifted and scaled. The idea is to let the network learn the patterns as they may occur in different locations and sizes within the frame. With 600 training sequences (created with artificial image translation but at constant size) the system already shows insensitivity to image shifts, approaching the performance levels observed on originally hand-excised frames. Finally, we are investigating different parameterizations of the input that might already exhibit shift invariance. The magnitude of the Fourier Transform of the image is one such representation.

2.3.2. *Parameterization*

There is almost certainly much irrelevant and/or redundant information in the 384 gray level values currently used as the visual input. Also, such a large parameter count increases significantly the number of network weights that need to be estimated. A smaller parameter set should lead to better generalization (particularly for speaker-independent recognition) and computational load reduction. We would like to reduce the number of visual parameters without invoking heuristics for image segmentation or feature extraction; the TDNN is expected to form its own internal representation of the relevant features.

Preliminary experiments show that we can represent the images by their first 32 principal components, thus reducing the data by a factor of 6, without visibly undermining performance. It should be noted that this representation (relying as it does on the correlations among the original data points) is sensitive to image shifts, as also found in other studies (ex. Turk and Pentland 1991). We are also investigating linear discriminant analysis, a related technique, which might prove better for image classification (as opposed to representation).

2.3.3. *Acoustic/Visual Integration*

There is evidence that humans combine acoustic and visual information before classification, i.e., without making separate decisions based on each modality (Summerfield 1983; Braida 1991). An automatic system should also benefit from integration at a low level, thanks to the availability of cross-modal features (for instance, temporal relationships between events in the two modalities). This is of course contingent on the availability of sufficient training data to robustly train the magnified network that results from increasing the size of the input vector. Preliminary experiments (Bregler 1993) suggest that this approach to modality integration is, in fact, not feasible without visual data reduction. This observation supplies more motivation still for the work described in Section 2.3.2.

Low level modality integration allows us also to avoid the tricky problem of viseme specification. While it is reasonably straightforward to specify the phoneme-to-viseme mapping in discrete syllables, the same is not true for continuous speech, especially when considering segmentation and coarticulation effects. However, if we are lead to maintain integration at the phoneme/viseme level, the combination scheme will be expanded. The units in the combined layer would likely benefit from drawing inputs from more than just the corresponding phoneme and viseme. For instance, the identity of the "second guess" of each sub-net should prove relevant.

3. ON-LINE CURSIVE HANDWRITING RECOGNITION

The recognition of cursive (or continuous) handwriting, as it is being written on a touch screen or graphics tablet, has not only scientific but also significant practical value, e.g. for note pad computers or for integration into multi-modal systems. Several different preprocessing and recognition approaches both for optical character recognition (OCR) and on-line character recognition (OLCR)

have been developed during the last decades. The main advantage of OLCR is the temporal information of handwriting, which can be recorded and used for recognition. In general this dynamic writing information (the coordinate sequence) is not available in OCR, where input consists of scanned text (bitmaps). By contrast, in OLCR systems, the spatial context and proximity of the strokes of characters are distorted or lost, when one merely retains and uses pen coordinates as a function of time.

We have developed an input representation for OLCR, which combines the advantages of bitmaps used in OCR with the dynamic writing information of OLCR. In this input representation characters and words are represented as a sequence of so called *context bitmaps*, which are basically low resolution descriptions of the coordinate's neighborhood.

This input representation is used with a connectionist recognizer, which is well suited for handling temporal sequences of patterns as provided by this kind of input representation. This recognizer, a Multi-State Time Delay Neural Network (MS-TDNN) (Haffner and Waibel 1991), integrates the segmentation and recognition of words into a single network architecture. The MS-TDNN, which was originally proposed for continuous speech recognition tasks (Hild and Waibel 1993; Bregler *et al.* 1993), combines shift invariant high accuracy pattern recognition capabilities of a TDNN (Waibel *et al.* 1989; Guyon *et al.* 1991) with a non-linear time alignment procedure (dynamic time warping) (Ney 1984) for aligning strokes into character sequences.

Figure 3a shows the basic architecture of our on-line handwriting recognition system. This recognition system is integrated into the example application, which is shown in Figure 3b. The following sections describe the preprocessing techniques, the MS-TDNN architecture, and present recognition results for
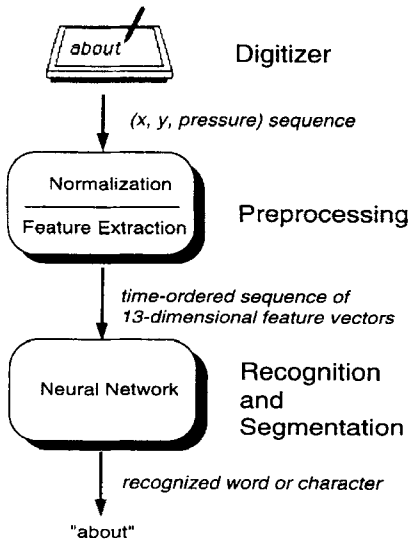

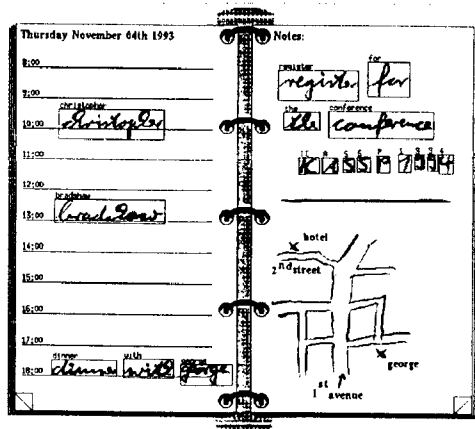Fig. 3a. System overview.   Fig. 3b. Example application.

writer-independent, single-character recognition tasks and large-vocabulary, writer-dependent, cursive handwriting recognition tasks with vocabulary sizes from 400 up to 20000 words.

3.1. *Preprocessing*

Preprocessing of the time-ordered coordinate sequence provided by the digitizer is performed in two successive steps: normalization and feature extraction.

3.1.1. *Normalization*
Normalization is performed to remove irrelevant variability occurring in the raw coordinate sequence. To compensate for varying writing speeds of different writers and of a single writer within a single word or character, the coordinate sequence is resampled, so that all successive coordinates are equally spaced. Then the resampled coordinate sequence is smoothed, using a moving average window, which mainly removes undesired sampling noise.

After that a baseline normalization of the word is performed. In a two stage process the word is first rotated according to the linear regression through all points, to get a rough correction of the word's orientation. Then, in a second fine adjustment the word is rotated according to the linear regression with respect to all local minima only. Finally, the word or character is linearly rescaled to a fixed height.

3.1.2. *Feature Extraction*
The second step of our preprocessing is the extraction of features along the pen trajectory yielding a sequence of time-ordered feature vectors. The basic idea of our feature extraction is to refer to low level topological features of the trajectory only and leave the extraction of high level features to the connectionist recognizer.

We started with a set of strictly local features similar to those in (Schenkel *et al.* 1994) and (Guyon *et al.* 1991). Each frame consisted of information on the pen position (x, y coordinates), directional features (Δx, Δy), curvature, speed and pen-up/pen-down indicator. But an inspection of the confusion matrices of networks trained on these features revealed significant problems in discriminating between cursive letters like "a" and "u" or "g" and "y", which look very similar and differ only in small regions of the characters (see Figure 4 for examples). These problems arise due to the fact that the features are strictly local, which

"a"   "u"   "g"   "y"

Fig. 4. Differences between cursive characters, which are hard to detect, if only local information is used.

means that they are local both in space and time. Therefore they are inadequate for modeling temporal long range context dependencies occurring in the pen trajectory.

The basis for the new set of features is a bitmap representation of the digitizer input. After normalization of the input we map the sequence of points (x_t, y_t) to a grey scale bitmap $B = \{b(i, j)\}$, where $b(i, j)$ indicates the number of points (x_t, y_t) falling into pixel (i, j).

In contrast to the limitations of optical character recognition, where bitmaps are the only source of information, we also know the time sequence of the points. This leads to the following method of combining spatial and temporal sources: Assume (x_t, y_t) falls into bitmap pixel (i, j). Consider a local $d \times d$ section of bitmap B centered around (i, j) (Figure 5b) and derive a 3×3 grey scale bitmap L_t by averaging this section (Figure 5c). That means, we derive a temporal sequence of low resolution bitmaps L_t centered around (x_t, y_t) (Figure 5a). These bitmaps plus directional information $(\Delta x, \Delta y)$ and the pen-up/pen-down feature form the new set of input features we use for recognition.

These features are still local in space but no longer local in time. Each point of the trajectory is visible from each other point of the trajectory in a small neighborhood. Therefore, we call the local bitmaps L_t *context bitmaps*. They seem to be appropriate for modeling temporal long range and spatial short range phenomena as observed in pen trajectories.
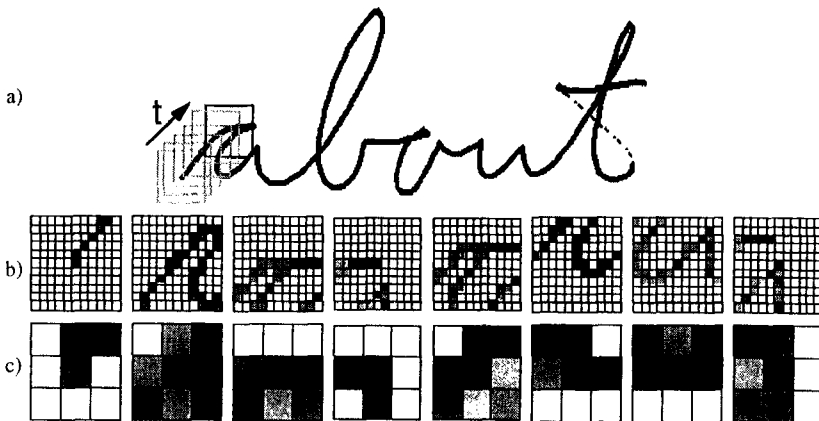

Fig. 5. Calculation of Context Bitmaps.

3.2. *The Multi-State Time-Delay Neural Network Architecture*

The input representation, which is described in the previous section, is used with a connectionist recognizer both for single character and cursive handwriting recognition tasks. This recognizer integrates the recognition and segmentation of words into a single network architecture, the Multi-State Time Delay Neural Network (MS-TDNN). Words are represented as a sequence of character, where each character is modeled by one or more states. For the results in

this paper, each character is modeled by 3 states, representing the initial, middle, and final sections of a character.

The basic MS-TDNN architecture is shown in Figure 6. Only a brief description of its operation with respect to handwriting recognition is given here; references (Haffner *et al.* 1991), (Haffner and Waibel 1992), and (Hild and Waibel 1993) describe in detail the MS-TDNN and its applications in continuous speech recognition.

The first three layers constitute a standard TDNN with sliding input windows of certain sizes. This TDNN computes scores for each state and for each time frame in the states layer. In the dynamic time warping layer (DTW layer) each word of the vocabulary is modeled by a sequence of states, the corresponding scores are simply copied from the states layer into the word models of the DTW layer. An optimal alignment path is found by the DTW algorithm for each word and the sum of all activations along this optimal path is taken as the score for the word output unit.
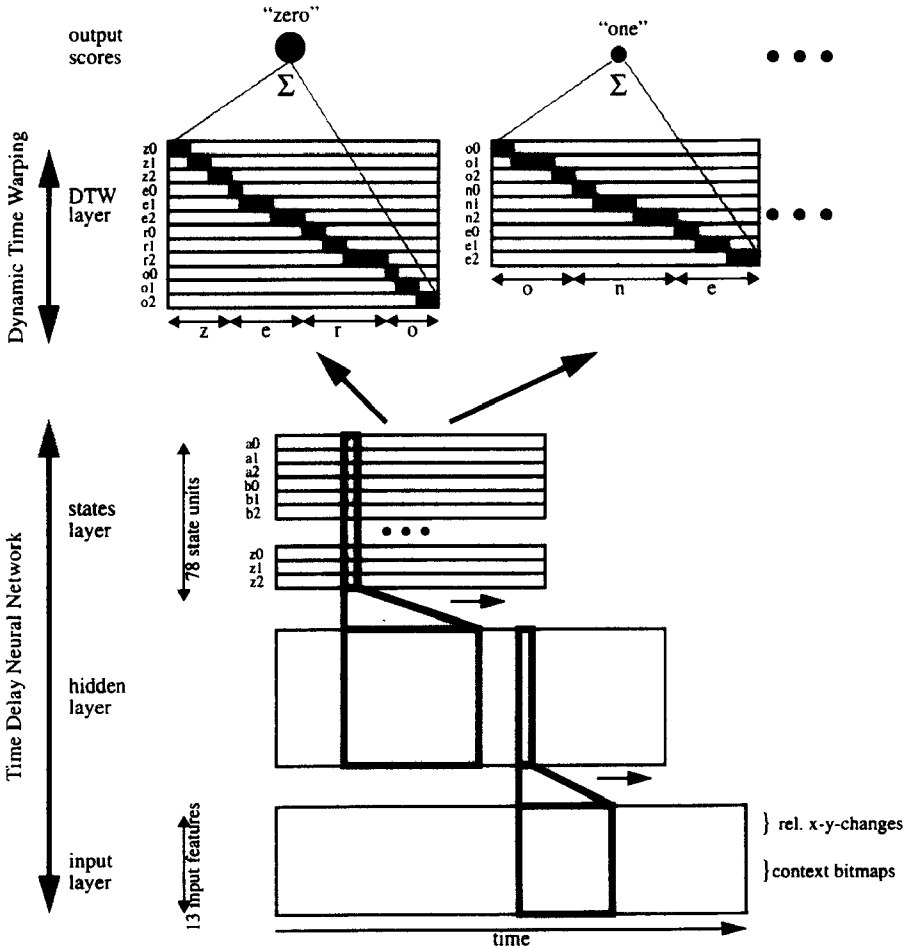
Fig. 6. The Multi-State Time Delay Neural Network architecture.

Once the network is trained on a particular vocabulary other vocabularies of varying sizes can be used without retraining, just by replacing the word models in the DTW layer.

All network parameters such as the number of states per character, the size of the input windows, or the number of hidden units are optimized manually for the results presented in this paper, but can also be optimized automatically by the Automatic Structure Optimization (ASO) algorithm that we have proposed elsewhere (Bodenhausen *et al.* 1993). By using the ASO algorithm, no time-consuming manual tuning of these network parameters for particular handwriting tasks and training set sizes is necessary while achieving optimal recognition performance.

3.3. *Experiments and Results*

We have tested the input representation together with the MS-TDNN architecture both on single character recognition tasks and cursive (continuous) handwriting recognition tasks. The handwriting databases used for training and testing of the MS-TDNN were collected at the University of Karlsruhe. All data is recorded on a pressure sensitive graphics tablet with a cordless stylus, which produces a sequence of time ordered 3-dimensional vectors (at a maximum report rate of 205 dots per second) consisting of the x-y-coordinates and a pressure value for each dot. All subjects had to write a set of single words from a 400 word vocabulary, covering all lower case letters, and at least one set of isolated lower case letters, upper case letters, and digit. For the continuous handwriting results presented in this paper only the data of one of the authors was used. All data is preprocessed as described before.

Table 2 shows results for different writer independent, single character recognition tasks (isolated characters). Writer dependent recognition results for cursive handwriting (isolated words) can be found in Table 3. The network used for the results in Table 3 is trained with aprox. 2000 training patterns from a 400

Table 2. Single character recognition results (writer independent).

Task	Training patterns	Test patterns	Recognition rate (%)
0_9	1600	200 (20 writers)	99
A_Z	2000	520 (20 writers)	95
a_z	2000	520 (20 writers)	93

Table 3. Results for different writer dependent cursive handwriting tasks.

Task	Vocabulary size (words)	Test patterns	Recognition rate (%)
msm_400_a	400	800	98
msm_400_b	400	800	97
msm_1000	1000	2000	95
msm_10000	10000	2000	87
msm_20000	20000	2000	83

word vocabulary (msm_400_a) and tested without any retraining on different vocabularies with sizes from 400 up to 20000 words. Vocabularies msm_400_b, msm_1000, msm_10000, and msm_20000 are completely different from the vocabulary on which the network was trained and were selected randomly from a 100000 word vocabulary (Wall Street Journal Vocabulary). First experiments on writer independent, cursive handwriting databases have shown recognition rates of more than 76% on a 400 word vocabulary.

These result show that the proposed input representation and MS-TDNN architecture can be used both for single character recognition and cursive handwriting recognition tasks with high recognition performance. The MS-TDNN performs well not only on the vocabulary it was trained for (see task *msm_400_a*), but also for other vocabularies it has never seen before (see task *msm_400_b*), even on much larger vocabularies (see tasks *msm_1000, msm_10000*, and *msm_20000*).

4. GESTURE RECOGNITION

We have been investigating pen-based gestures drawn using a stylus on a digitizing tablet. This kind of gesture is simpler to handle than hand gestures captured with a camera but still allows for rich and powerful expressions, as any editor who has to mark up manuscripts knows. Pen-based gestures are becoming popular on hand-held computers, but the focus of our research is mainly on how gestures can be effectively combined with other input modalities, because using gestures as the sole input channel seems to be a still clumsy way of issuing commands to computers. In order to pursue this direction of investigation, we developed a multimodal text editor capable of recognizing speech and gesture commands (Vo and Waibel 1993).

The initial multimodal editor we developed currently uses 8 editing gestures (see Table 4). Some of these were inspired by standard mark-up symbols used by human editors. Others, such as the "delete" symbols, are what most people would automatically use when correcting written text with normal pencil and paper.

Table 4. Text-editing gestures.

○	Select	(	Begin selection
✕	Delete	)	End selection
╱	Delete	∿	Transpose
∧	Paste	⌐	Split line

We use a temporal representation of gestures. A gesture is captured as a sequence of coordinates tracking the stylus as it moves over the tablet's surface, as opposed to a static bitmapped representation of the shape of the gesture. This dynamic representation was motivated by its successful use in handwritten

character recognition (Section 3 and Guyon *et al.* 1991). Results of experiments described in (Guyon *et al.* 1991) suggest that the time-sequential signal contains more information relevant to classification than the static image, leading to better recognition performance.

In our current implementation, the stream of data from the digitizing tablet goes through a preprocessing phase (Guyon *et al.* 1991). The coordinates are normalized and resampled at regular intervals to eliminate differences in size and drawing speed; from these resampled coordinates we extract local geometric information at each point, such as the direction of pen movement and the curvature of the trajectory. These features are believed to hold discriminatory information that could help in the recognition process and thus can give the neural network recognizer appropriate information to find temporal regularities in the input stream.

4.2. *Gesture Classification Using Neural Networks*

We use a TDNN (Waibel *et al.* 1989) (see Figure 7) to classify each preprocessed time-sequential signal as a gesture among the predefined set of 8 gestures. Each gesture in the set is represented by an output unit. Each data point in the input stream is represented by 8 input units corresponding to the 8 features extracted during the preprocessing phases; these include pen coordinates and pressure as well as local geometric information as mentioned above. The network is trained on a set of manually classified gestures using a modified backpropagation algorithm (Waibel *et al.* 1989).

During training, the 10 units in the first hidden layer essentially become "feature detectors" that extract low-level patterns from the input, and the 8 hidden units in the next layer learn to spot those features that contribute to the recognition of each of the 8 gestures. Each output unit integrates over time the
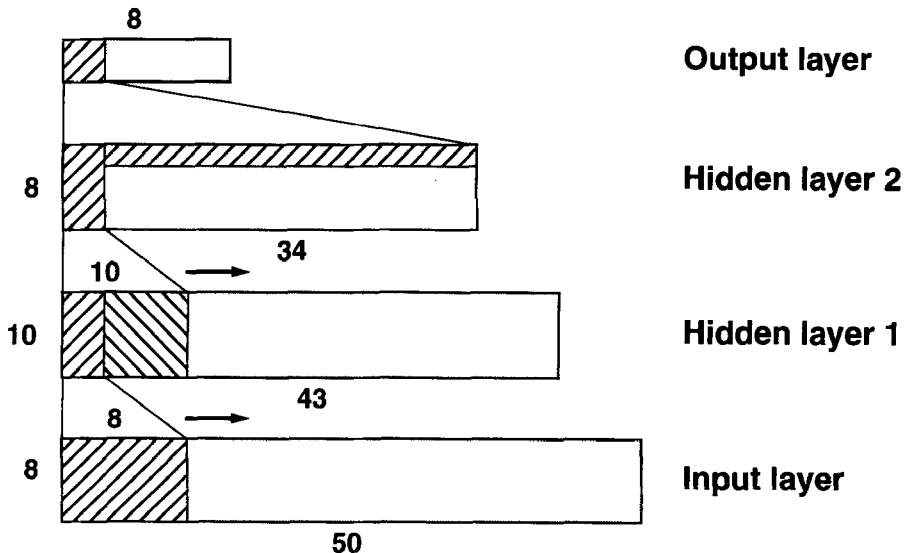

Fig. 7. TDNN architecture for gesture recognition.

evidence from the corresponding unit in the second hidden layer. The output unit with the highest activation level determines the recognized gesture.

The data samples used to train and evaluate the gesture recognizer were collected from a single "gesturer." Among the collected samples, 640 samples (80 per gesture) form the training set, and 160 samples (20 per gesture) form an independent test set which was never seen by the network during training. Our gesture recognizer achieves almost 99% recognition rate on both the training data and the test set.

4.3. *Learning in Gesture/Handwriting Recognition*

The usefulness of gesture and handwriting recognition depends largely on the ability to adapt to new users because of the great range of variability in the way individuals write or make gestures. No matter how many tokens we put in the training database to cover different gestures that mean "delete text", for example, there may always be totally different gestures that are not yet part of the gesture vocabulary. This is particularly troublesome for neural-network-based systems because usually the network has to be retrained using all the old training data mixed with a large number of new examples, in order to be able to recognize new patterns without catastrophically forgetting previously learned patterns. Because of the large number of examples needed and the long retraining time, this clearly cannot be done on-line in a way that would enable the user to continue to work productively. A good system should be able to query the user for correction and remember the particular input pattern that caused the error in order to make intelligent guesses when similar inputs occur. Such a fallback method would offer a reasonable level of performance until the network can be retrained off-line.

We have developed a method to accomplish this using an Incremental TDNN (ITDNN) architecture (Vo 1994). We start by training a regular TDNN using all the available data to obtain a base network. When a recognition error occurs during use, the system queries the user for the correct output and creates template-matching hidden units that influence the output units via excitatory or inhibitory connections (see Figure 8). Template matching is accomplished by making the weight matrix of the extra unit proportional to the activation matrix of the first hidden layer; this was deemed better than matching the input layer directly because during training by backpropagation the units of the first hidden layer have learned to spot input features relevant to classification.

In order to retain the time-shift invariant property that makes the TDNN such a powerful classifier of time-sequential patterns, we assemble the extra units out of subunits, each one having weights matching a different section of the activation template, that is, the activation matrix of the first hidden layer. Thus the extra "units" can in fact be thought of as extra hidden layers. The purpose of this is to enable these subunits to slide along the time dimension just like the regular TDNN units. Since consecutive subunits (within the same extra unit) will tend to have high activations in consecutive time slices, we employ a time-warping technique to compute match scores (see Figure 9). If a subsequent input pattern is similar to the template used to create an extra unit, the extra
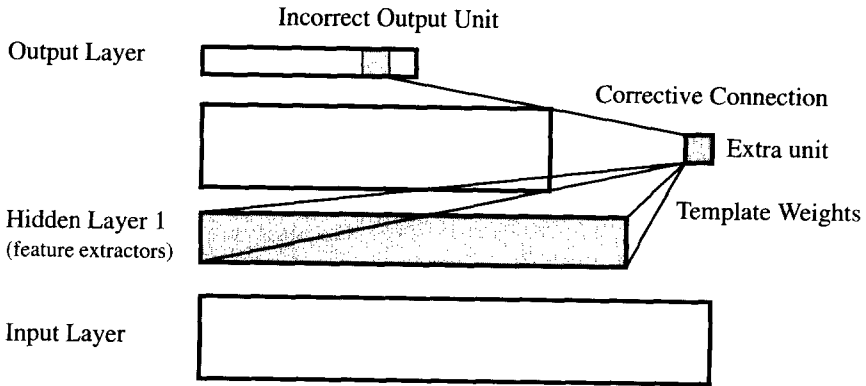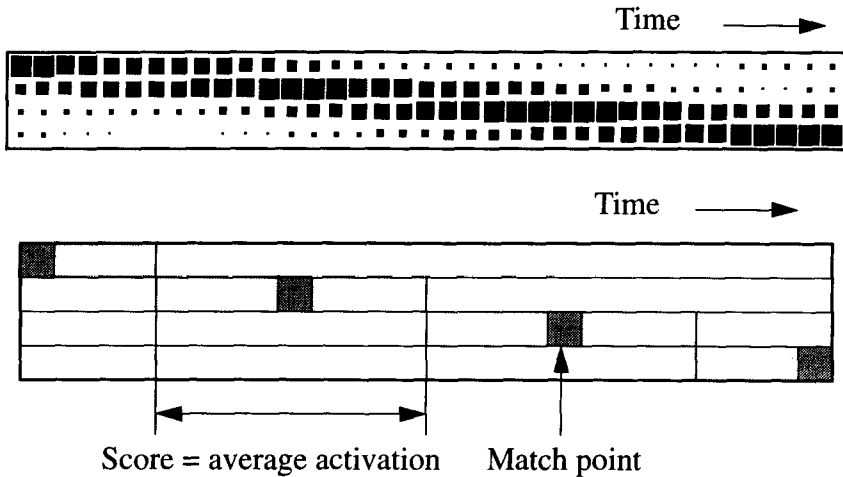
Fig. 8. The incremental TDNN architecture.

Fig. 9. Activation trace of an extra unit composed of four subunits.

unit is turned on and thus able to influence the corresponding output unit. We use extra units to fix recognition errors by lowering outputs that are incorrectly high via inhibitory (negative weight) connections, and by boosting outputs that are incorrectly low via excitatory (positive weight) connections.

We tested the incremental learning capability of the ITDNN in a series of experiments involving simple handwritten digit recognition (Vo 1994). This task was chosen because it is simple enough so that we can easily eliminate the influence of factors extraneous to what we want to measure: what is the degradation in performance on old input patterns when the ITDNN is trained on new input patterns. Although the development of the ITDNN was motivated by our gesture recognition research, handwriting recognition is very similar and poses the same problems as gesture recognition, hence the results of the experiments described here are still relevant.

We trained a base network with examples of handwritten digits, each written in a consistent way. We then tested the network on a different variation of one digit (namely the digit 6 written in a clockwise direction rather than counter-clockwise as in the training set). The base TDNN was unable to recognize any of the new examples. When a single extra unit was added, the resulting ITDNN was able to correctly classify 99% of the new examples while "forgetting" only 0.6% of the old training examples.

These experiments show that the ITDNN is capable of quickly adding coverage for a new input variation without forgetting previously learned information and thus is a good candidate for systems requiring on-line, immediate recognition improvements during use, such as gesture and handwriting recognizers for pen-based computers. Systems capable of incremental learning will be able to adapt quickly to a new user at a reasonable level of performance while allowing productive work to continue. During subsequent work sessions new data can be unobtrusively collected for off-line training of a full network with regular architecture. This presumably superior network can then replace the patched one.

4.4. *The Language of Speech and Gesture*

Figure 10 shows a block diagram of the multimodal interpreter module in our speech- and gesture- based text editor.

The TDNN-based gesture recognizer was described in Section 4.2. For the speech component we use many alternative speech recognition strategies; these include a keyword spotter developed by Zeppenfeld (1992, 1993) as well as full-scale continuous speech recognition modules such as Sphinx (Huang *et al.* 1993) and Janus (Wosczyna *et al.* 1993). The speech recognition module is coupled with a Recursive Transition Network parser (Ward 1991) using a semantic grammar developed for the editing task. For the keyword-spotting version, the word spotter was trained to spot 11 keywords representing editing commands such as *move, delete*, . . . and textual units such as *character, word*, . . . The effect is to let the user speak naturally without having to worry about grammar and vocabulary, as long as the utterance contains the relevant keywords. For example, an utterance such as "Please delete this word for me" is equivalent to "Delete word". In the case of continuous speech recognition, the semantic-fragment parser
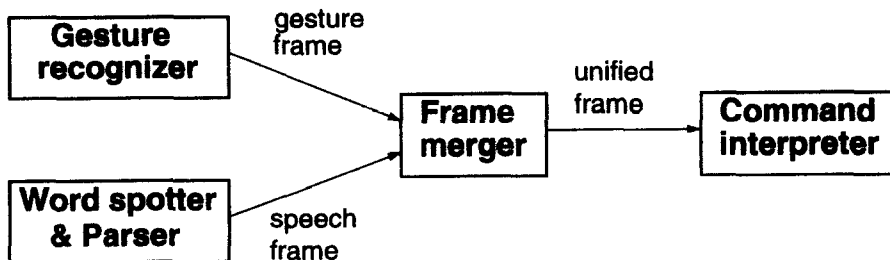

Fig. 10. Joint interpretation of gesture and speech.

achieves essentially the same effect by matching fragments of the recognized speech against predefined templates to find semantically meaningful parts of the text. It then creates a frame consisting of slots representing various components of a plausible semantic interpretation, and fills in any slot it can using semantic fragments found in the hypothesized sentence.

The interpretation of multimodal inputs was based on semantic frames and slots representing parts of an interpretation of user intent. The speech and gesture recognizers produce partial hypotheses in the form of partially filled frames. The output of the interpreter is obtained by unifying the information contained in the partial frames.

In the system each frame has slots named *action* and *scope* (what to operate on.) Within *scope* there are subslots named *type* and *textual-unit*. The possible scope types include "point" and "box"; the textual units include "character," "word," "line" . . .

Consider an example in which a user draws a circle and says "Please delete this word". The gesture-processing subsystem recognizes the circle and fills in the coordinates of the "box" scope in the gesture frame as specified by the position and size of the circle. The word spotter produces "delete word", from which the parser fills in the *action* and *textual unit* slot in the speech frame. The frame merger then outputs a unified frame in which *action* = delete, *scope.type* = box, and *scope.textual-unit* = word. From this the command interpreter constructs an editing command to delete the word circled by the user.

One important advantage of this frame-based approach is its flexibility, which will facilitate the integration of more than two modalities, and across acoustic, visual, and linguistic ones. All we have to do is define a general frame for interpretation and specify the ways in which slots can be filled by each input modality.

5. CONCLUSIONS

In this paper, we have presented research that is aimed at producing more natural, more robust (redundant) and more efficient human-computer interfaces, by exploring the combination of several different human communication modalities. Such combinations naturally involve acoustic but also visual and gestural expressions of human intent and form a multimodal "language" we seek to decode. We have shown that more robust recognition can indeed be achieved by combining speech with lipreading, i.e., visual and acoustic modalities. We have also shown an on-line handwritten character recognizer, that could be combined with speech and gesture. Finally, we have demonstrated that speech and gesture can be joined to provide more natural, robust interpretation of user intent, as speech and gesture both deliver complementary cues to complete the semantics of a multimodal "speech" act. Further research currently in progress includes exploring eye- and face- tracking and sound source localization, to deliver multimodal cues more accurately, even when a person is moving about the room, and to determine focus of attention and reference of human interaction.

ACKNOWLEDGEMENTS

REFERENCES

Baluja, S. & Pomerleau, D. (1994). Non-Intrusive Gaze Tracking Using Artificial Neural Networks. To appear in *Advances in Neural Information Processing Systems 6*, Morgan Kaufmann Publishers.

Bodenhausen, U., Manke, S. & Waibel, A. (1993). Connectionist Architectural Learning for High Performance Character and Speech Recognition. In Proceedings of *ICASSP'93*, Vol. 1, 625–628. Minneapolis, MN, U.S.A.

Braida, L. D. (1991). Crossmodal Integration in the Identification of Consonant Segments. *The Quarterly Journal of Experimental Psychology* 43A(3): 647–677.

Bregler, C., Hild, H., Manke, S. & Waibel, A. (1993). Improving Connected Letter Recognition by Lipreading. In Proceedings of *ICASSP'93*, Vol. 1, 557–560, Minneapolis, MN, U.S.A.

Bregler, C. (1993). *Lippenlesen als Unterstützung zur robusten automatischen Spracherkennung.* M.S. Thesis. Fakultaet für Informatik, Universität Karlsruhe.

Goldschen, A. J. (1993). *Continuous Automatic Speech Recognition by Lipreading.* Ph.D. Dissertation, George Washington University.

Guyon, I., Albrecht, P., LeCun, Y., Denker, J. & Hubbard, W. (1991). Design of a Neural Network Character Recognizer for a Touch Terminal. *Pattern Recognition* 24(2): 105–119.

Haffner, P., Franzini, M. & Waibel, A. (1991). Integrating Time Alignment and Neural Networks for High Performance Continuous Speech Recognition. In Proceedings of *ICASSP'91*, Vol. 1, 105–108. Toronto, Canada.

Haffner, P. & Waibel, A. (1991). Multi-State Time Delay Neural Networks for Continuous Speech Recognition. *Advances in Neural Network Information Processing Systems 4*, 135–142. Morgan Kaufmann Publishers.

Hauptmann, A. (1989). Speech and Gestures for Graphic Image Manipulation. In Proceedings of *CHI'89*, 241–245. Austin, TX, U.S.A.

Hild, H. & Waibel, A. (1993). Connected Letter Recognition with a Multi-State Time Delay Neural Network. *Advances in Neural Information Processing Systems 5*, 712–719. Morgan Kaufmann Publishers.

Huang, X., Alleva, F., Hon, H., Hwang, M., Lee, K. & Rosenfeld, R. (1993). The SPHINX-II Speech Recognition System: An Overview. *Computer Speech and Language* 7(2): 137–148.

Jackson, P. L. (1988). The Theoretical Minimal Unit for Visual Speech Perception: Visemes and Coarticulation. *The Volta Review* 90(5): 99–115.

Manke, S. & Bodenhausen, U. (1994). A Connectionist Recognizer for On-Line Cursive Handwriting Recognition. In Proceedings of *ICASSP'94*, Vol. 2, 633–636. Adelaide, Australia.

Miller, G. A. & Nicely, P. E. (1955). An Analysis of Perceptual Confusions Among Some English Consonants. *Journal of the Acoustical Society of America* 27(2): 338–352.

Ney, H. (1984). The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition. In *IEEE Transactions on Acoustics, Speech and Signal Processing* 32(2): 263–271.

Nodine, C., Kundel, H., Toto, L. & Krupinski, E. (1992). Recording and Analyzing Eye-position Data Using a Microcomputer Workstation. *Behavior Research Methods, Instruments & Computers* **24**(3): 475–584.

Mase, K. & Pentland, A. (1991). Automatic Lipreading by Optical-Flow Analysis. *Systems and Computers in Japan* **22**(6): 67–76.

Petajan, E. D. (1984). *Automatic Lipreading to Enhance Speech Recognition*. Ph.D. Thesis, University of Illinois.

Petajan, E. D., Bischoff, B. & Bodoff, D. (1988). An Improved Automatic Lipreading System to Enhance Speech Recognition. In Proceedings of *CHI'88*, 19–25. Washington, DC, U.S.A.

Pomerleau, D., (1992). *Neural Network Perception for Mobile Robot Guidance*. Ph.D. Thesis, Carnegie Mellon University, CMU-CS-92-115.

Rose, R. & Paul, D. (1990). A Hidden Markov Model Based Keyword Recognition Systems. In Proceedings of *ICASSP'90*, Vol. 1, 129–132. Albuquerque, NM, U.S.A.

Rubine, D., (1991). *The Automatic Recognition of Gestures*. Ph.D. Thesis, Carnegie Mellon University.

Rubine, D., (1991). Specifying Gestures by Examples. *Computer Graphics* **25**(4): 329–337.

Schwartz, R. & Austin, S. (1991). A Comparison of Several Approximate Algorithms for Finding N-best Hypotheses. In Proceedings of *ICASSP'91*, Vol. 1, 701–704. Toronto, Canada.

Schenkel, M., Guyon, I. & Henderson, D. (1994). On-Line Cursive Script Recognition Using Time Delay Neural Networks and Hidden Markov Models. In Proceedings of *ICASSP'94*, Vol. 2, 637–640. Adelaide, Australia.

Schmidbauer, O. & Tebelskis, J. (1992). An LVQ-based Reference Model for Speaker-Adaptive Speech Recognition. In Proceedings of *ICASSP'92*, Vol. 1, 441–444. San Francisco, CA, U.S.A.

Stork, D. G., Wolff, G. & Levine, E. (1992). Neural Network Lipreading System for Improved Speech Recognition. In Proceedings of *IJCNN'92*, Vol. 2, 289–295. Baltimore, MD, U.S.A.

Summerfield, Q. (1983). Audio-visual Speech Perception, Lipreading and Artificial Stimulation. In Lutman, M. E. & Haggard, M. P. (eds.) *Hearing Science and Hearing Disorders*, Academic Press: New York.

Tebelskis, J. & Waibel, A. (1993). Performance Through Consistency: MS-TDNNs for Large Vocabulary Continuous Speech Recognition. In *Advances in Neural Information Processing Systems 5*, 696–703. Morgan Kaufmann Publishers.

Turk, M. & Pentland, A. (1991). Eigenfaces for Recognition. *Journal of Cognitive Neuro-Science* **3**(1): 71–86.

Vo, M. T. & Waibel, A. (1993). A Multimodal Human-Computer Interface: Combination of Speech and Gesture Recognition. In *Adjunct Proc. InterCHI'93*. Amtersdam, The Netherlands.

Vo, M. T. (1994). Incremental Learning using the Time Delay Neural Network. In Proceedings of *ICASSP'94*, Vol. 2, 629–632. Adelaide. Australia.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K. & Lang, K. (1989). Phoneme Recognition Using Time-Delay Neural Networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing* **37**(3): 328–339.

Waibel, A., Jain, A., McNair, A., Saito, H., Hauptmann, A. & Tebelskis, J. (1991). JANUS: A Speech-to-speech Translation System Using Connectionist and Symbolic Processing Strategies. In Proceedings of *ICASSP'91*, Vol. 2, 793–796. Toronto, Canada.

Ward, W. (1991). Understanding Spontaneous Speech: The Phoenix System. In Proceedings of *ICASSP'91*, Vol. 1, 365–367. Toronto, Canada.

Ware, C. & Mikaelian, H. (1987). An Evaluation of an Eye Tracker as a Device for Computer Input. In *SIGCHI Bulletin, Spec. Issue, CHI+GI'87*, 183–188. Toronto, Canada.

Woszczyna, M. *et al.* (1993). Recent Advances in Janus: A Speech Translation System. In Proceedings of *EUROSPEECH'93*, Vol. 2, 1295–1298. Berlin, Germany.

Yuhas, B. P., Goldstein, M. H., Sejnowski, Jr. & T. J. (1989). Integration of Acoustic and Visual Speech Signals Using Neural Networks. *IEEE Communications Magazine* **27**(11): 65–71.

Zeppenfeld, T., & Waibel, A., (1992). A Hybrid Neural Network, Dynamic Programming Word Spotter. In Proceedings of *ICASSP'92*, Vol. 2, 77–80. San Francisco, CA, U.S.A.

Zeppenfeld, T., Houghton, R., & Waibel, A. (1993). Improving the MS-TSNN for Word Spotting. In Proceedings of *ICASSP'93*, Vol. 2, 475–478. Minneapolis, MN, U.S.A.