LETTER Communicated by Nima Mesgarani

Noise-Robust Speech Recognition Through Auditory Feature Detection and Spike Sequence Decoding

Phillip B. Schafer
pbs130@psu.edu
Dezhe Z. Jin
djin@phys.psu.edu
Department of Physics and Center for Neural Engineering, The Pennsylvania State University, University Park, PA 16802, U.S.A.

Speech recognition in noisy conditions is a major challenge for computer systems, but the human brain performs it routinely and accurately. Automatic speech recognition (ASR) systems that are inspired by neuroscience can potentially bridge the performance gap between humans and machines. We present a system for noise-robust isolated word recognition that works by decoding sequences of spikes from a population of simulated auditory feature-detecting neurons. Each neuron is trained to respond selectively to a brief spectrotemporal pattern, or feature, drawn from the simulated auditory nerve response to speech. The neural population conveys the time-dependent structure of a sound by its sequence of spikes. We compare two methods for decoding the spike sequences—one using a hidden Markov model–based recognizer, the other using a novel template-based recognition scheme. In the latter case, words are recognized by comparing their spike sequences to template sequences obtained from clean training data, using a similarity measure based on the length of the longest common sub-sequence. Using isolated spoken digits from the AURORA-2 database, we show that our combined system outperforms a state-of-the-art robust speech recognizer at low signal-to-noise ratios. Both the spike-based encoding scheme and the template-based decoding offer gains in noise robustness over traditional speech recognition methods. Our system highlights potential advantages of spike-based acoustic coding and provides a biologically motivated framework for robust ASR development.

1 Introduction

Despite significant advances in automatic speech recognition (ASR) over the past several decades, humans still outperform the best artificial systems, especially on recognition tasks in noisy background conditions (Carey & Quang, 2005; Sroka & Braida, 2005; Cooke, 2006; Meyer, Wesker, Brand, Mertins, & Kollmeier, 2006; Barker, Vincent, Ma, Christensen, & Green,

Neural Computation **26, 523–556** (2014)
doi:10.1162/NECO_a_00557

2013). This performance shortfall is a major barrier to the widespread adoption of speech recognition technologies (Deng & Huang, 2004; Scharenborg, 2007). A growing number of ASR studies seek to better approximate human performance by emulating the speech processing performed by the brain. Such systems can improve on traditional methods by finding advantageous neuronal representations for sound and through new decoding paradigms designed to exploit them.

Experimental neuroscience has recently produced significant insights into auditory coding that can inform ASR development. Converging evidence shows that precisely timed and reproducible spike responses carry information in the inferior colliculus and auditory cortex, forming representations known as spike-timing codes (DeWeese, Wehr, & Zador, 2003; Escabí, Miller, Read, & Schreiner, 2003; Lu & Wang, 2004; Elhilali, Fritz, Klein, Simon, & Shamma, 2004; Heil, 2004). Such codes may be implicated in processing natural stimuli such as animal vocalizations (Schnupp, Hall, Kokelaar, & Ahmed, 2006; Huetz, Del Negro, Lebas, Tarroux, & Edeline, 2006; Huetz, Philibert, & Edeline, 2009; Kayser, Montemurro, Logothetis, & Panzeri, 2009; Kayser, Logothetis, & Panzeri, 2010) and speech (Steinschneider et al., 2005; Nourski et al., 2009) and may have a role in separating such signals from noise (Las, Stern, & Nelken, 2005; Bar-Yosef & Nelken, 2007). Although these spike representations appear to be a pervasive and important form of auditory coding, little is known about how they can be used to perform robust pattern recognition.

In this letter, we present a novel approach to ASR that pairs spike-based acoustic coding with a decoding scheme specifically designed to recognize speech patterns in noisy conditions. A major source of performance loss for traditional ASR systems is that they model the statistics of speech using hidden Markov models (HMMs) but use acoustic representations whose statistics are strongly influenced by the acoustic environment. This causes a mismatch between the models and the observed speech, particularly in noisy conditions not seen during model training. By contrast, in our system, we aimed to develop a scheme for producing spike representations of speech that are relatively invariant under additive noise and design a decoding scheme that avoids noise modeling but tolerates arbitrary corruptions of the spike code.

1.1 System Overview. Our encoding scheme is based on a model of acoustic feature detection in the auditory cortex. Cortical neurons in several species have been shown to respond preferentially to behaviorally relevant natural stimuli such as conspecific vocalizations (Rauschecker, Tian, & Hauser, 1995; Lewicki & Arthur, 1996; Sen, Theunissen, & Doupe, 2001; Wang & Kadia, 2001). A recent work further shows that such selectively tuned neurons can produce responses that are insensitive to background noise (Moore, Lee, & Theunissen, 2013). In the spirit of these findings, we propose that artificial neurons trained to selectively respond to specific

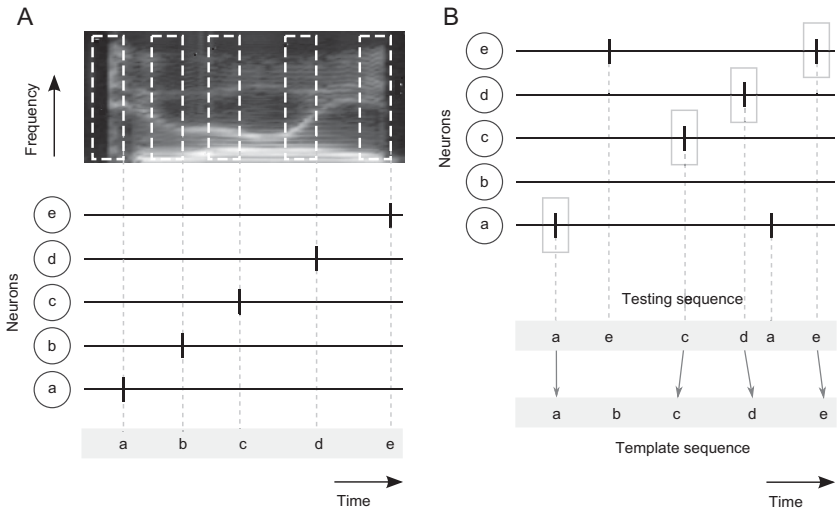

Figure 1: Illustration of concepts used in the design of our system. (A) A word (spectrogram shown at top) is represented by the spikes of a population of feature-detecting neurons (center), each of which is selectively tuned to a spectrotemporal speech feature. The spike code is translated into a sequence of neuron labels (bottom). (B) In noise, the code is corrupted by spike insertions and deletions (top), but a subset of spikes maintains its sequence from clean conditions (gray boxes). The invariant sub-sequence is identified by computing the longest common sub-sequence between the testing sequence and a template sequence stored from training data (arrows, bottom). Only five neurons are shown for illustrative purposes; in our experiments, 1100 are used.

speech features can generate robust spike codes by effectively ignoring signals that do not resemble their preferred stimuli. That is, in contrast with more traditional acoustic representations in which noise is faithfully conveyed along with speech, these neurons may represent only those elements of a sound that resemble speech in clean conditions.

To implement this idea, we trained each of a population of spiking artificial neurons ("feature detectors") to identify a brief spectrotemporal pattern ("feature") in the simulated auditory nerve response to speech. The features include onsets, offsets, and up-and-down frequency sweeps corresponding to speech formant structures (see Figure 1A, top). The spikes of the feature detectors represent speech in terms of the sequential appearance of its features through time. Due to our method of discriminatively training the neurons, the spikes often persist, with timing relatively unaltered, when the speech is mixed with noise.

In decoding the spikes of the feature detectors, we chose to disregard the precise length of the time intervals between spikes and treat the code as a

sequence of neuron labels, its spike sequence (see Figure 1A, bottom). This was motivated in part by recent biologically motivated spike decoding studies that operated on sequences of spikes (Loiselle, Rouat, Pressnitzer, & Thorpe, 2005; Jin, 2004, 2008). It also has the advantage of making our system robust to temporal warping due to speech tempo variations, which poses a major problem for some other spike decoders (Gütig & Sompolinsky, 2009). We found that the spike sequence code in itself offers improved noise robustness over more traditional representations when used with an HMM-based decoder. However, we obtained even further performance gains using a novel template-based scheme in which test samples are compared directly to speech exemplars stored from training data. The scheme centers on a novel speech similarity measure based on the length of the longest common sub-sequence (LCS) between feature detector spike sequences (see Figure 1B). We were motivated by the observation that in noise, although the spike code is in general corrupted by the insertion and deletion of spikes, a subset of spikes often maintains a sequence similar to that seen in clean conditions. This sub-sequence typically forms a good match with some sub-sequence within templates computed for the same word. The LCS algorithm locates the best sub-sequence match between a test sequence and template, allowing robust recognition based on a subset of spikes embedded in noisy code.

We tested our system's performance on isolated words from the AURORA-2 noisy digits database (Hirsch & Pearce, 2000) and found improved recognition results at low signal-to-noise ratios (SNRs) when compared with an HMM-based system with missing data imputation (Gemmeke & Cranen, 2008). This robust performance was attributable to both the noise invariance of the spike sequence code and the noise tolerance of the template-based decoding method. Significantly, unlike in many statistical ASR approaches, we achieved these results with a training scheme that used only clean speech, without any noise modeling or multicondition training.

1.2 Related Work. While we built our system largely from the ground up based on notions of feature detection in the auditory system, our approach has precedents in a number of other ASR studies. The state of the art in commercial software today lies in so-called connectionist ASR approaches, where artificial neural networks (ANNs) provide feature inputs to HMM-based decoders (Bourlard & Morgan, 1994; Trentin & Gori, 2001). In particular, systems using deep belief networks (DBNs) have recently produced significant improvements in recognition rates on large-vocabulary recognition tasks (Mohamed, Dahl, & Hinton, 2012; Dahl, Yu, Deng, & Acero, 2012; Hinton et al., 2012). In these systems, deep learning algorithms are applied to large speech data sets in order to better model the variations in speech due to noise and speaker variations. The ANNs can be trained either to directly model the posterior probabilities of HMM states (hybrid

systems) or to generate additional features within short time frames that are generatively modeled by the HMM (tandem systems) (Hermansky, Ellis, & Sharma, 2000). Tandem systems using DBNs were recently shown to offer some advantages for noise-robust recognition on a task similar to that used here (Vinyals & Ravuri, 2011).

Our approach is similar to these in its use of discriminatively trained artificial neurons, but the specific training algorithms and decoding methods used here differ significantly from most connectionist systems. In particular, we formulate our training criterion as a simple binary classification task (section 2), which permits relatively inexpensive training using a linear support vector machine (SVM) (Vapnik, 1998). Additional novel elements of our system are the variable rate of symbol generation by the feature detectors, which provides flexibility in representing speech separately from noise, and the discrete nature of the spike sequence code, which enables our robust template-based decoding using the LCS. The similarities and differences between our system and connectionist methods are further explored in section 6.

Our approach to acoustic coding aims to capture the physiologically measured response properties of auditory cortical neurons. In this respect, it recalls recent ASR studies that used efficient representation techniques such as independent component analysis (ICA) (Hyvärinen & Oja, 2000) and temporal linear generative models (TLGMs) (Smith & Lewicki, 2005) to model neural responses. These methods have successfully predicted receptive field properties in the peripheral auditory system (Smith & Lewicki, 2006) and in the midbrain and auditory cortex (Klein, König, & Körding, 2003; Carlson, Ming, & DeWeese, 2012), but so far they have offered limited performance improvements when used for speech recognition (Kwon & Lee, 2004; Rufiner, Martínez, Milone, & Goddard, 2007; Smit & Barnard, 2009; Sivaram, Nemala, Elhilali, Tran, & Hermansky, 2010). Our approach also echoes studies that took a bottom-up approach to auditory coding by modeling the spectrotemporal response characteristics of cortical neurons using linear filter methods, for example, with two-dimensional Gabor filter banks (Kleinschmidt, 2003; Mesgarani, Slaney, & Shamma, 2006). A method for finding more general, nonparametric receptive fields using discriminative training has also been explored (Mesgarani, Sivaram, Nemala, Elhilali, & Hermansky, 2009). These methods have yielded some ASR performance improvements when used with HMM decoders (Zhao & Morgan, 2008; Schädler, Meyer, & Kollmeier, 2012).

Our decoding method relates to another class of studies that used simpler spike-based encoding methods than ours while focusing efforts on new, biologically motivated decoders. For instance, spike codes generated by a bank of single-frequency band onset and offset detectors have been used as inputs to decoders that detected rapid sequences (Loiselle et al., 2005) or spike synchrony (Gütig & Sompolinsky, 2009). In the latter case, this yielded good results on a noiseless isolated digit recognition task. Another study

used spikes derived from a cochleagram as input to a liquid state machine for a digit recognition task, but the performance did not surpass the state of the art (Verstraeten, Schrauwen, Stroobandt, & Van Campenhout, 2005). While we do not attempt a network-based implementation of our decoding method here, other studies of neural sequence recognition may pave the way for such work in the future (Jin, 2004, 2008).

Template-based recognition has seen a resurgence of interest in recent years, with a number of studies finding improved performance over HMM-based methods (Axelrod & Maison, 2004; Aradilla, Vepa, & Bourlard, 2005; Deng & Strik, 2007; De Wachter et al., 2007; Ramasubramanian, Kulkarni, & Kämmerer, 2008; Seppi & Van Compernolle, 2010; Demuynck, Seppi, & Van Compernolle, 2011). Many of these studies have focused on extending the classical ASR method of dynamic time warping (DTW) (Bridle, Brown, & Chamberlain, 1983). Whereas in standard ASR systems a single HMM is taken to represent the statistical variations of each word or phone, in template-based systems, these variations are accounted for by storing many instances of each word and comparing these directly with test data. Template-based systems offer a number of advantages, including that they avoid the limiting statistical assumptions of HMMs and permit better incorporation of metadata such as speaker identity and gender. They are further supported by psycholinguistic studies of pattern storage by episodic memory (Maier & Moore, 2005; Strik, 2006). However, most template methods suffer from similar robustness problems to statistical methods, since noisy speech often does not form a good match to the templates. Our system aims to exploit the advantages of template-based methods while using a speech similarity measure that is more robust to noise.

1.3 Outline. This letter is organized as follows. In section 2 we describe our spike-based encoding scheme and our procedure for training the feature-detecting neurons. In section 3 we display receptive field characteristics for the feature detectors to facilitate comparison of our auditory model with physiological data. In section 4 we evaluate the robustness of the feature detector code and compare its performance to a more traditional feature encoding using an HMM decoder. In section 5 we describe our template-based sequence decoding scheme and demonstrate its improved noise robustness. In section 6 we explore our system's relationship to previous work in ASR and discuss the biological implications of our model.

2 Spike-Based Encoding Scheme and Training Procedure

Our approach to ASR centers on a method for representing speech by the spiking response of auditory feature-detecting neurons. This encoding scheme is realized using a two-stage model of acoustic processing in the auditory system (see Figure 2). In the first stage, a simulation of the auditory periphery transforms sounds into a firing rate representation on the

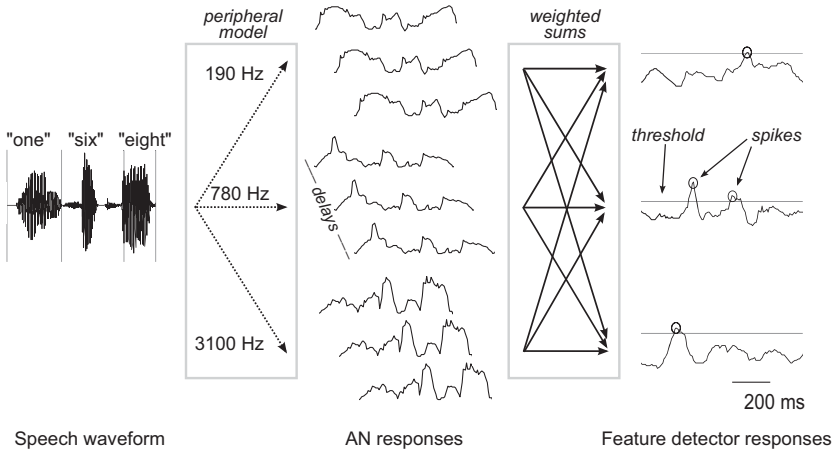

Figure 2: Schematic of the speech representation scheme. The displayed curves are real data from our experiments. The acoustic waveform for several digits (left) is transformed into simulated AN firing rates (center; 3 of 32 frequency channels shown). Several time-delayed copies of the AN response are input to the feature detectors, which take a weighted sum over their inputs (right; 3 of 1100 feature detectors shown). Spikes (shown as circles) are assigned at peaks in the response that surpass a fixed threshold (horizontal gray lines).

auditory nerve (AN). In the second stage, feature detector neurons integrate the AN response over short time intervals to produce spike responses indicating the presence of specific spectrotemporal features. In this section, we describe this encoding scheme in detail and define our method for discriminatively training the feature detectors.

2.1 Auditory Model. Acoustic processing begins in our system with a model of the auditory periphery. The peripheral model filters sounds into bandpassed signals that are conveyed as simulated AN firing rates to the feature detectors. This processing plays a similar role to the computation of the spectrogram used in many standard ASR front ends.

For the results presented here, we used a peripheral model based on a gammatone filter bank (Slaney, 1993), although roughly equivalent results could be obtained using a spectrogram method (see section 5). Our filter bank comprises 32 gammatone filters with center frequencies ranging from 100 to 4000 Hz on an equivalent rectangular bandwidth scale. To simulate hair cell transduction and the loss of phase locking in the ascending auditory pathway, the filtered signals are half-wave-rectified, smoothed and resampled at 8 ms intervals, and square-root-compressed. Adaptation

processes that lead to relative level invariance at the level of cortex (Dean, Harper, & McAlpine, 2005; Sadagopan & Wang, 2008) are accounted for by normalizing each channel to unit variance for each speech sample.

The central innovation of our encoding scheme is a model of feature detection in the central auditory system. We trained feature detector neurons to identify brief patterns in the AN signal across both the temporal and spatial (frequency) dimensions. This approach can be compared with physiological findings in which a neuron's ability to integrate over time and frequency is often described by a spectrotemporal receptive field (STRF; see section 3). Here, temporal integration is implemented by including several time-delayed copies of the AN response as the input to each feature detector (see Figure 2, center). Eight time-delayed copies of the AN response comprise a 256-dimensional input signal $\mathbf{s}(t)$, which gives the feature detectors a 64 ms "moving window" view of the sound. The window size was chosen to be long enough for the feature detectors to identify temporal structures such as up-and-down frequency sweeps, onsets, and offsets, but short enough that they responded only to mono- or biphonetic features that are relatively invariant under longer-scale temporal warping.

We modeled each feature detector as an artificial neuron that takes a weighted sum $\sigma(t) = \mathbf{w} \cdot \mathbf{s}(t)$ of its inputs, and spikes at peaks in $\sigma(t)$ that exceed a fixed threshold (see Figure 2, right). In cases where multiple peaks occur within a 100 ms interval, the largest peak is selected and the others are suppressed. The weights $\mathbf{w}$ are trained for each neuron to maximize its discriminative ability, as described below. Note that aside from the choice of peripheral representation (spectrogram versus gammatone filter bank), the integration over time and frequency performed in computing $\sigma(t)$ is similar to that performed in an STRF model. However, the assignment of spikes to peaks in the above-threshold summation signal introduces an additional nonlinearity that is not captured by an STRF.

2.2 Data Set. Training and testing of our system were carried out using speech recordings from the AURORA-2 database. AURORA-2 consists of connected digits ("zero" through "nine," plus "oh") from the TIDIGITS multiple-speaker database (Leonard, 1984) in clean and a variety of additive noise conditions. The training and testing utterances are spoken by distinct sets of speakers. To obtain isolated digit samples, we segmented the speech using HMM-based forced alignment (Yuan & Liberman, 2008). This gives speech samples that convey the full variability of continuous speech but allows us to focus on recognizing individual words independent of the segmentation problem. In the case of noisy speech, alignments were found for clean speech and then applied to the corresponding noise mixes.

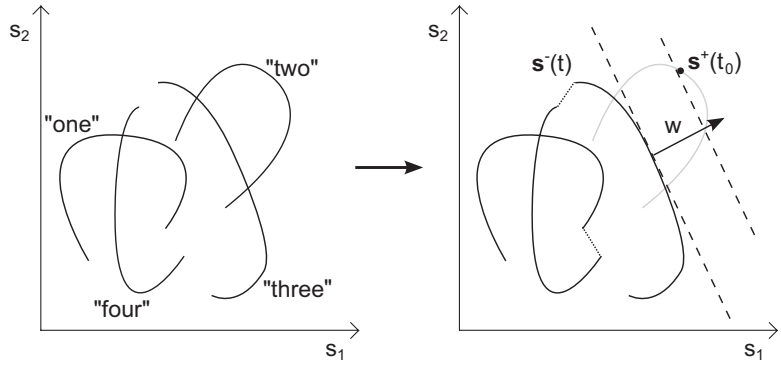

Figure 3: Schematic of SVM training of the weight vector $\mathbf{w}$ for a single feature detector. (Left) The AN responses to several words are visualized as trajectories in a 256-dimensional space. (Right) A single point $\mathbf{s}^+(t_0)$ in the trajectory for "two" is selected as the preferred feature; the remainder of the word (gray curve) is not used in the training. Other words are concatenated to form the background trajectory $\mathbf{s}^-(t)$. The surfaces $\mathbf{w} \cdot \mathbf{s} = b + 1$ and $\mathbf{w} \cdot \mathbf{s} = b - 1$ (dotted lines) define a margin separating the two training classes. The objective for SVM training is to maximize the size of this margin.

2.3 Training Scheme. The response properties of the feature detectors are set by training the weights $\mathbf{w}$. The goal of the training is to find a set of weights for each detector such that it will be unresponsive to most sounds but will respond reliably to a selected spectrotemporal feature. The procedure is as follows. For each feature detector, we select a single clean digit exemplar, known as the positive training exemplar, and compute its AN response with added delays $\mathbf{s}^+(t)$. From this word, we randomly select the response $\mathbf{s}^+(t_0)$ at a single time point to serve as the preferred feature for the neuron. We also select a set of background training data $\mathbf{s}^-(t)$ consisting of concatenated, word-length responses to five exemplars of each of the other digits, called negative training exemplars. For instance, if the preferred feature is selected from within an exemplar of "two," the background data consist of concatenated exemplars of each of the digits "one," "three," "four," and so on. Note that while the full time-varying responses of the negative exemplars are used in the training, only a single, randomly selected time point t_0 within the positive training exemplar is used.

The training aims to find a weight vector $\mathbf{w}$ such that the neuron spikes in response to the preferred feature but not in response to the background. This is accomplished using a linear support vector machine (SVM) (Vapnik, 1998). Intuitively, we can visualize $\mathbf{s}^+(t_0)$ as a single point along the trajectory $\mathbf{s}^+(t)$ taken by the positive training exemplar through the space of feature detector inputs (see Figure 3). The trajectories defined by the negative

training exemplars are combined to form a single trajectory $\mathbf{s}^-(t)$ through the space. After training, the weight vector defines a hyperplane that maximally separates the preferred feature from the background. Formally, the SVM finds a weight vector $\mathbf{w}$ and bias b such that $\mathbf{w} \cdot \mathbf{s}^+(t_0) = b + 1$ and $\mathbf{w} \cdot \mathbf{s}^-(t) < b - 1$ for all t, with the margin of separation $\frac{1}{||\mathbf{w}||^2}$ as large as possible. We interpret the value $b - 1$ as the unit's threshold, so that the response to the preferred feature is as far above threshold as possible while still keeping the background responses below.

Figure 4 shows the result of SVM training for a single feature detector. At the upper left is the word-length summation response $\sigma(t)$ to the positive training exemplar, that is, the word "two" from which the preferred feature was selected. The response has an above-threshold peak at the time of the preferred feature t_0, so the neuron produces a spike and successfully detects the feature. Shown at the lower left are the responses to the negative training exemplars that comprise the background. By design, these responses all have peaks below the threshold and therefore do not produce spikes.

To demonstrate the feature detector's ability to generalize to data not present in the SVM training, we also show in Figure 4 responses to randomly selected independent testing exemplars of each digit. Exemplars of "two" are termed positive testing exemplars. Because of variations in pronunciation, we expect to detect features similar to $\mathbf{s}^+(t_0)$ only in some of these exemplars. Indeed, only three of the five exemplars shown produce spikes, which we call hits. Exemplars of other digits are termed negative testing exemplars. Because the digits are mostly phonetically distinct, we expect to detect few features that are similar to $\mathbf{s}^+(t_0)$ among these exemplars. Most of the negative testing exemplars accordingly have subthreshold responses, but a few do produce spikes, which we call false hits.

2.4 Population Training. We created a population of feature detectors sensitive to a broad range of speech features by repeating the above training procedure with different preferred features and background sets. One preferred feature was randomly selected from an exemplar of each of the 11 digits spoken by each of 50 male and 50 female speakers in the AURORA-2 training set, for a total of 1100 features. Different background exemplars were selected for the SVM training with each preferred feature. Note that while the feature detector responses are temporally sparse due to their selectivity to specific speech features, there is no independence constraint on the responses, and in fact feature detectors that are tuned to similar features are very likely to fire together.

Because of pronunciation variations, we do not expect to detect precisely the same feature set in every exemplar of a given digit. Nevertheless, hit rates and false hit rates over the testing data provide a useful comparison of each feature detector's performance on testing data. These rates are defined

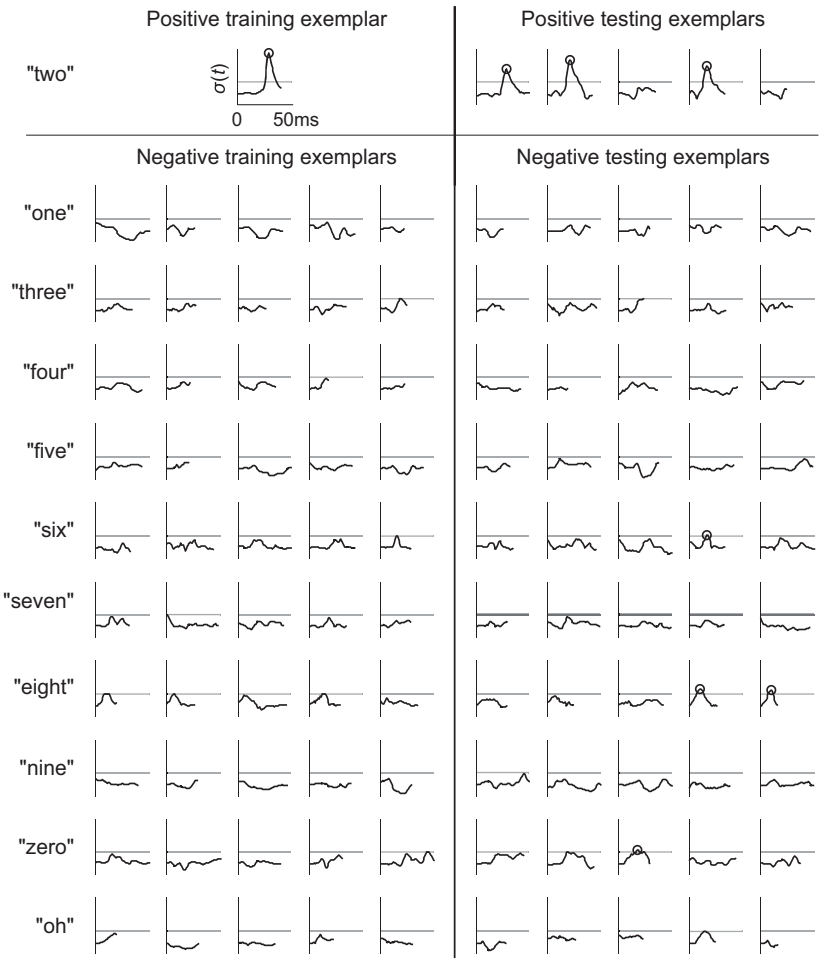

Figure 4: SVM training results for a single feature detector. Each plot shows the summation response $\sigma(t)$ of the neuron (arbitrary units) versus time for a single word. Gray horizontal lines show the neuron's threshold; circles indicate the location of spikes. (Top left) The response to the positive training exemplar of "two," with a peak at the time of selected preferred feature. (Top right) The responses to five testing exemplars of "two." (Bottom) Responses to five negative training (left) and testing (right) exemplars of each digit. Horizontal and vertical scales are the same for all plots.

as the fraction of positive or negative testing exemplars during which a detector produced a hit or false hit, respectively. Histograms of these rates for the feature detector population are shown in Figure 5.

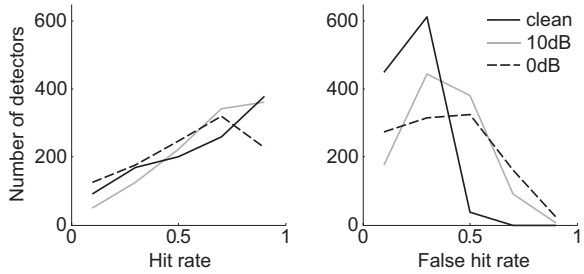

Figure 5: Histograms of hit rates and false hit rates for the feature detectors in clean conditions and in mixes with subway noise at 10 dB and 0 dB SNR. The hit rate (false hit rate) is the fraction of positive (negative) testing exemplars for each detector that produces a spike.

3 Spectrotemporal Receptive Fields

To facilitate comparison of our auditory coding model with physiological data, we computed STRFs for the feature detectors. The STRF characterizes the preferred stimuli of a neuron and is frequently used in experimental studies (Aertsen & Johannesma, 1981). Formally, it is a linear estimate of the input-output relationship of a neuron, where the input is a spectrographic representation of sound $s(f, t)$, usually the spectrogram, and the response function $r(t)$ is usually interpreted as a firing rate. The STRF is defined as the kernel function $h(f, t)$ in

$$r_{\text{est}}(t) = \int_{f_{\min}}^{f_{\max}} \int_{0}^{\tau_{\max}} h(f, \tau) s(f, t - \tau) df d\tau + \bar{r},$$

where $r_{\text{est}}(t)$ is the minimum mean square error estimate of $r(t)$ and $\bar{r}$ is $r(t)$'s mean.

We computed the STRFs using the software package STRFlab, which fits the kernel parameters using ridge regression (Theunissen, Sen, & Doupe, 2000; Moore et al., 2013). For the representation $s(f, t)$, we chose the logarithm of the spectrogram as computed by a short-time Fourier transform with frequency limits $f_{\min} = 100$ Hz and $f_{\max} = 4000$ Hz. The response $r(t)$ was taken as the feature detector spike response smoothed by a gaussian filter of width 10 ms. We used the spike responses to the clean speech of 10 male and 10 female speakers in the AURORA-2 training set to compute each STRF.

The STRFs display a variety of spectrotemporal modulations that in many cases resemble the modulation structure in the preferred features used for the training (see Figure 6). We characterized these modulations using temporal and spectral modulation scores as in (Rodríguez, Read, &

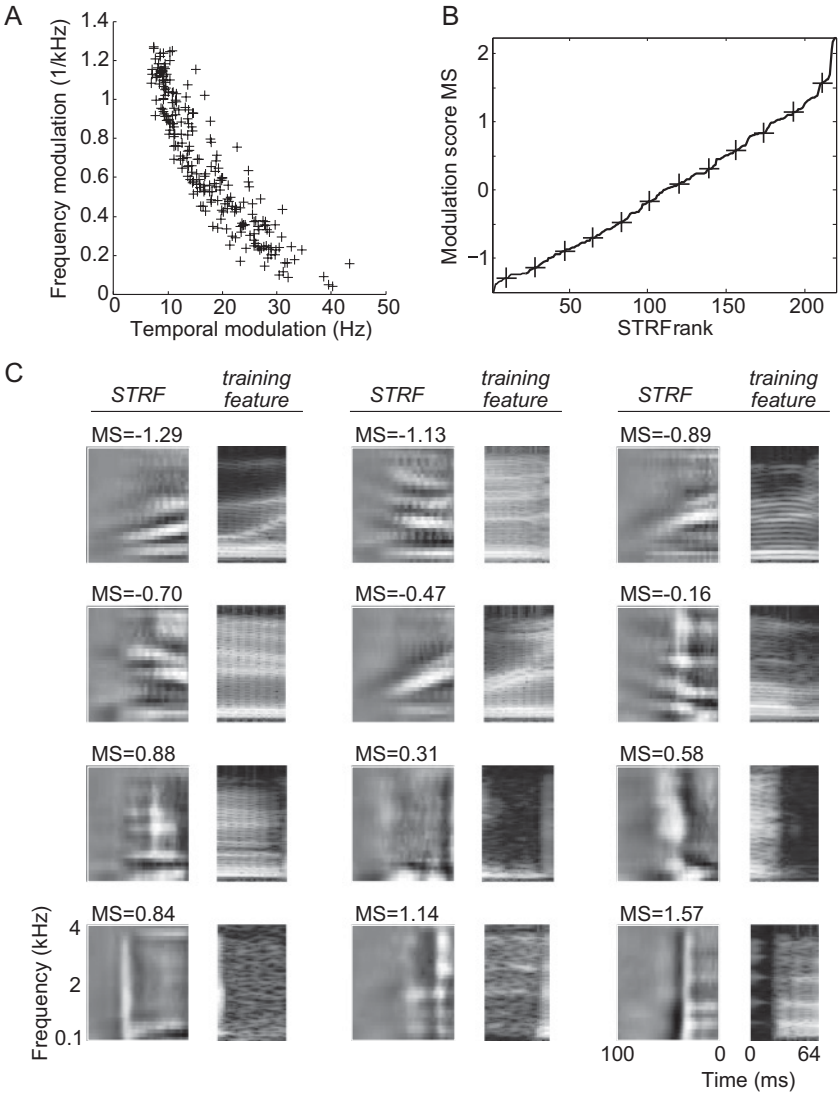

Figure 6: STRFs of the feature detectors. (A) Scatter plot of spectral versus temporal modulation scores for the STRFs showing a trade-off between the two types of structure. (B) The STRFs ranked by their combined modulation score (MS) describing the degree of spectral versus temporal modulation. Crosses indicate which STRFs are plotted below. (C) STRFs for selected feature detectors as well as spectrograms of the preferred features used in the feature detector training.

Escabí, 2010). These were calculated by taking the two-dimensional Fourier transform of each STRF and finding its power centroid along the time and frequency axes. As in Rodríguez et al. (2010), we found that the STRFs exhibited a trade-off between temporal and spectral modulations. This is because short speech segments used for training the feature detectors tend to have strong spectral modulations (vowels) or temporal modulations (consonants), but not both.

In order to display a set of STRFs representative of the full range of modulation types, we ranked the STRFs using an overall modulation score (see Figure 6B). This was defined as the temporal modulation score minus the spectral modulation score, where each score was first normalized by its mean over the feature detector population. A low modulation score indicates a STRF with greater spectral modulation, while a high modulation score indicates one with greater temporal modulation. We found that the full range of STRF modulation types was essential to our system's robust recognition performance; any attempt to select a subset of feature detectors based on its STRF characteristics yielded a decrease in performance (results not shown).

The STRFs of the feature detectors can be directly compared with physiological measurements to identify putative relationships between our simulated neurons and real neurons in the auditory system. The modulation structure seen in our STRFs is more complex than that typically found from neurons in the inferior colliculus (Lesica & Grothe, 2008; Versnel, Zwiers, & van Opstal, 2009; Rodríguez et al., 2010), but is comparable to the relatively complex shapes found in auditory cortical neurons using natural stimuli (Bitterman, Mukamel, Malach, Fried, & Nelken, 2008; Laudanski, Edeline, & Huetz, 2012; Moore et al., 2013). We explore further the relationship of our encoding scheme to biological coding in section 6.

4 Robustness of the Spike Sequence Code

In designing our encoding scheme, we hypothesized that selective tuning of the feature detectors could yield spike codes that are robust to acoustic noise. In particular, since the detectors are trained to respond to specific patterns derived from clean speech, we expect them to respond preferentially to speech-like elements in sound while responding poorly to noise. We evaluated this claim using speech from test set A of the AURORA-2 database, which includes mixes with babble, car, subway, and exhibition hall noise at SNRs of 20 dB through -5 dB.

To give a sense of why the feature detector training yields a robust code, in Figure 7 we show the summation responses $\sigma(t)$ for three different feature detectors in clean conditions and in mixes with subway noise. The noise affects the shape of the responses, but due to the large margin afforded by the SVM training, all of the response peaks shown here remain above threshold. Furthermore, the timing of the feature detector spikes, which is

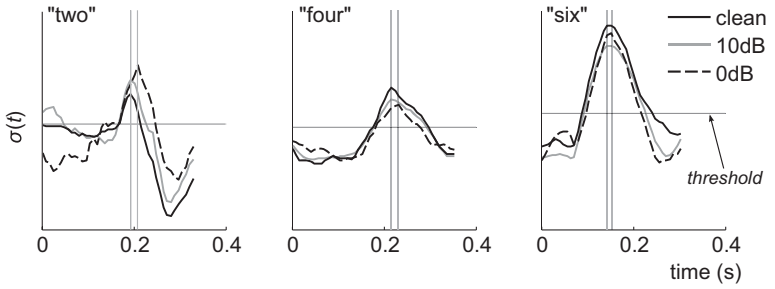

Figure 7: Responses of three different feature detectors to three word-length speech stimuli (one exemplar of "two," "four," and "six," respectively) in clean conditions and in mixes with subway noise at 10 dB and 0 dB SNR. Gray vertical lines show the change in spike timing (i.e., the change in response peak location) between clean conditions and 0 dB. Gray horizontal lines show the threshold for each neuron.

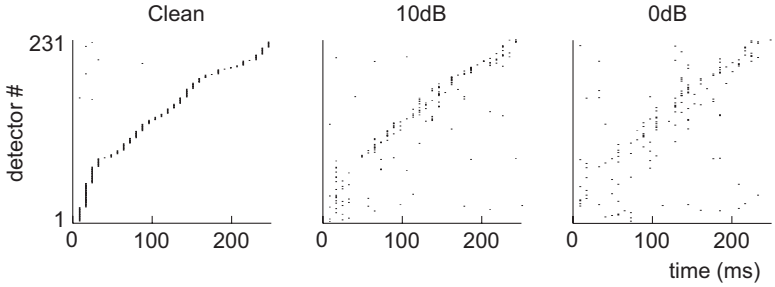

Figure 8: Spike code degradation in noise. Raster plots of feature detector spikes are shown for a single exemplar of the word "five" in clean conditions and in mixes with subway noise at 10 dB and 0 dB SNR. The feature detectors are ordered by their spike response to this exemplar in clean conditions. Only the 231 feature detectors that spike in clean conditions are shown.

determined by the positions of the response peaks, is shifted only slightly by the noise.

Figure 8 displays the effects of the subway noise at the level of the population spike code for a single word. Despite some changes in the spike timing and some extraneous spiking compared to clean conditions, a subset of well-ordered spikes is still visible at 0 dB. The effect of subway noise on the feature detector population as a whole can also be seen in the hit rate and false hit rate histograms in Figure 5. As the noise level increases, there is an increase in the number of false hits for some feature detectors, but only a modest decrease in the hit rate.

4.1 HMM Decoding of the Spike Sequence Code. To test the feature detector code's robustness in the context of a speech recognition task, we performed a comparison with a set of baseline frame-based features using an HMM-based decoder. This type of decoding forms the standard paradigm for ASR, in which words (or sometimes phones) are statistically modeled by HMMs, and unknown speech is recognized by comparison to the models. In our case, for each encoding, we trained whole-word, left-to-right HMMs with 16 states and no state skipping using the Hidden Markov Toolkit (HTK) (Young et al., 2005). Since isolated word samples were used, no silence model was needed.

For the baseline features, we used perceptual linear predictive (PLP) coefficients, which are cepstral coefficients with additional perceptually motivated spectral warping that increases their robustness to variations (Hermansky, 1990). We computed 39 PLP coefficients (13 cepstral coefficients plus delta and acceleration) within 25 ms frames located at 10 ms intervals. With the feature detector code, we considered only the sequential ordering of the spikes and not their precise timing, as described in section 1. That is, the spike code was converted to a sequence of discrete neuron labels ordered by their spike times (see Figure 1). In cases where multiple spikes occurred at the same 8 ms peripheral model time step, the corresponding neuron labels at that time step were placed in ascending order. Note that the assignment of neuron labels to neurons was arbitrary.

For the PLP system, the HMM state emission probabilities were modeled as a single gaussian mixture with diagonal covariance. For the spike sequence code, discrete emission probabilities were modeled.

Recognition results for the two encoding schemes were obtained by Viterbi alignment (HTK tool HVite) and are shown in Figure 9. The spike sequence code gives superior recognition results across all noise levels. Average recognition rates at each noise level can also be seen in the summary results in Figure 12.

5 Template-Based Decoding Scheme

While the feature detector code in itself offered improved performance when used with an HMM decoder, we sought an alternative decoding scheme that could better exploit the regularities of the spike code in noise. As previously noted, HMMs model the statistics of speech and therefore suffer performance losses in noisy conditions where speech acoustics no longer match the models. An alternative approach to HMMs that has gained increasing currency in recent years is template-based (or exemplar-based) recognition, in which statistical modeling is forgone altogether, and speech is recognized through direct comparison with other prelabeled speech data. However, most standard template-based systems also suffer robustness problems since, under typical speech distance measures, noise-corrupted

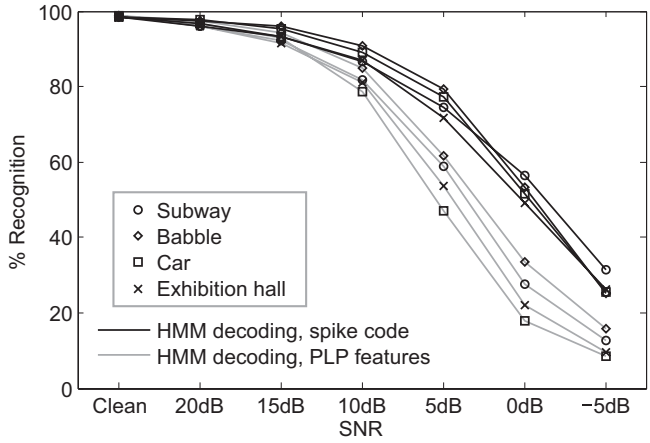

Figure 9: The spike sequence code gives improved recognition results over PLP coefficients when used with an HMM decoder. Recognition rates are shown for the four noise types in AURORA-2 test set A.

speech does not match any better to the templates than it would to a statistical model.

Here we developed a form of template-based recognition specifically designed to decode noise-corrupted speech, using a novel speech similarity measure based on the LCS between spike sequences. The motivation for our approach was illustrated in Figure 1. Although the feature detector code has some inherent noise robustness, corruptions in the form of spike insertions, deletions, and temporal shifting do occur in noise. On the other hand, we observed that a subset of spikes often occurs in the same sequence in noise as in clean conditions (see the spike raster plots in Figure 8). Indeed, as seen in Figure 5, noise typically leads to an increase in the false hit rate but minimally affects the hit rate, so that many of the spikes present in clean conditions persist embedded in otherwise noisy code. This invariant sub-sequence often closely matches some sub-sequence within template sequences computed for the same word in clean conditions. The LCS length quantifies the best sub-sequence match between a test sequence and template, allowing recognition based on only a subset of relatively uncorrupted spikes in the code.

5.1 Speech Similarity Measure. Given two sequences of symbols $X = \{x_1, \ldots, x_M\}$ and $Y = \{y_1, \ldots, y_N\}$, the LCS is defined as the longest sequence $Z = \{z_1, \ldots, z_K\}$ that can be composed by drawing elements in left to right order, but not necessarily contiguously, from either X or Y. For instance, in the example shown in Figure 1b, the sequences $\{a, b, c, d, e\}$

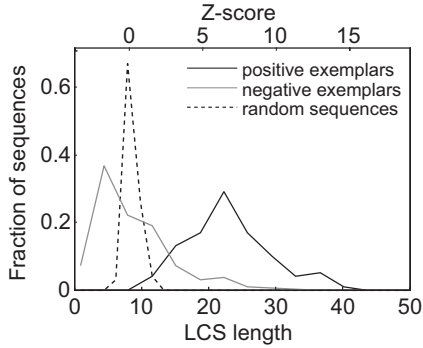

Figure 10: LCS length distribution for a single sequence template derived from an exemplar of "one." The solid curves show the distribution of LCS lengths with respect to testing exemplars of "one" (positive exemplars) and testing exemplars of other digits (negative exemplars). The dotted curve shows the distribution of LCS lengths with respect to a set of 100 randomly generated sequences of equal length (see the text). The mean and standard deviation of this random distribution are used to convert the LCS lengths of the testing data (lower axis) into Z-scores (upper axis) that were used as a similarity measure for template-based recognition.

and $\{a, e, c, d, a, e\}$ have an LCS $\{a, c, d, e\}$, with an LCS length of 4. The LCS length can be efficiently computed using a well-known algorithm in dynamic programming (Bergroth, Hakonen, & Raita, 2000).

Our goal in the decoding scheme was to quantify the similarity of a test utterance with a clean speech template using the LCS length of their spike sequences. Words could then be recognized by finding their best matches among a large bank of templates. However, we found that the raw LCS length alone did not constitute a good sequence similarity measure because some templates have longer average LCS lengths with test sequences than others. This is due in part to the varied lengths of the template sequences and in part to the varying average firing rates of the feature detectors: long templates with many commonly occurring neuron labels tend to have longer-than-average LCS lengths, leading to a bias toward recognition by these templates. To normalize out these differences, we computed a distribution of expected LCS lengths for each template using randomly generated spike sequences. For a given template, we generated 100 spike sequences of the same length, with each spike randomly assigned to a feature detector with probabilities proportional to the average feature detector firing rates on training data set. The LCS of these randomly generated sequences with the template were used to fill out the distribution of LCS lengths. We then expressed the LCS length of a test sequence with the template as a Z-score with respect to this distribution (see Figure 10) and defined this as the

sequence similarity measure for use in the template scheme. We summarize this process with the similarity formula

$$sim(S_{temp}, S_{test}) = Z_{temp}\big(|LCS(S_{temp}, S_{test})|\big),$$

where $|\cdot|$ denotes a sequence length and $Z_{temp}(\cdot)$ represents the transformation to a Z-score using the template's randomized distribution.

5.2 Recognition Scheme. Our decoding method uses the above similarity measure with a template-based recognition scheme that works as follows. We created a bank of sequence templates comprising spike sequence responses to 100 exemplars of each digit, which were randomly selected from the clean speech of 50 male and 50 female speakers in the AURORA-2 training set. The templates are grouped into 22 sets $\{\Omega_k\}_{k=1}^{22}$ corresponding to the 11 digits and 2 speaker genders. Because speaker gender is a major source of variability in the speech, grouping by gender gives template sets with more homogeneous composition than would grouping by digit alone.

To recognize a test word, we first find its spike sequence and compute its similarity score with each of the templates. Then for each template set Ω_k, we select its N_k best matches with the test sequence and average their similarity scores to find a mean similarity score for the set. The set with the highest score is taken to identify the digit. To find an optimal set of values $\{N_k\}_{k=1}^{22}$, we computed recognition rates using a set of validation data comprising 500 exemplars of each digit, drawn from the AURORA-2 training set. The values that gave the best recognition rates on this validation data were found by iteratively varying each value up and down and choosing the changes that improved the performance. The selected values were then applied to all subsequent recognition trials with testing data.

Due to the lack of statistical modeling in a template-based system, recognition performance depends on the number of template exemplars used; a greater number of templates covers more of the possible instantiations of a given digit, permitting matches to be found for a fuller range of test utterances. As shown in Figure 11, recognition rates on the task presented here begin to plateau after the inclusion of about 50 to 100 templates per digit. We chose to use only 100 for the sake of computational efficiency, but inclusion of more templates could yield further small gains in performance. We note, however, that even low numbers of templates still give robust performance exceeding that of traditional ASR systems, which we attribute to the noise robustness of our speech similarity measure.

5.3 Recognition Results. We applied our decoding scheme to the AURORA-2 test data and obtained the results shown in Figure 12. Results superior to the HMM decoder were obtained at all noise levels. Our results are further summarized in Figure 13. In Figure 13A, mean recognition

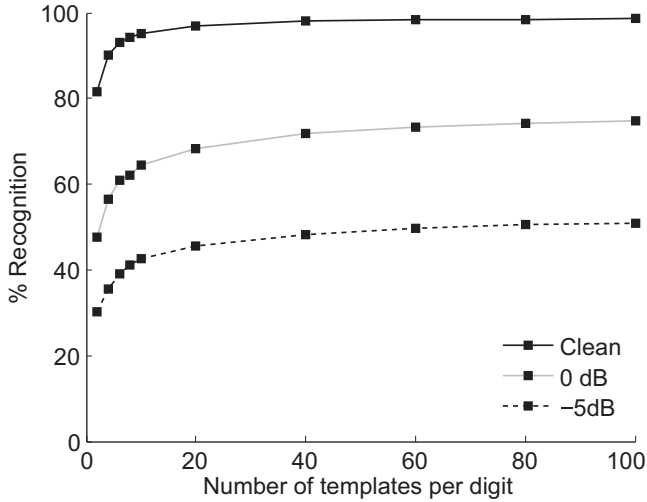

Figure 11: Recognition rates from the template-based scheme improve with the inclusion of more templates. Mean recognition rates are displayed for clean data and for the 0 dB and −5 dB conditions, averaged over the four noise types.

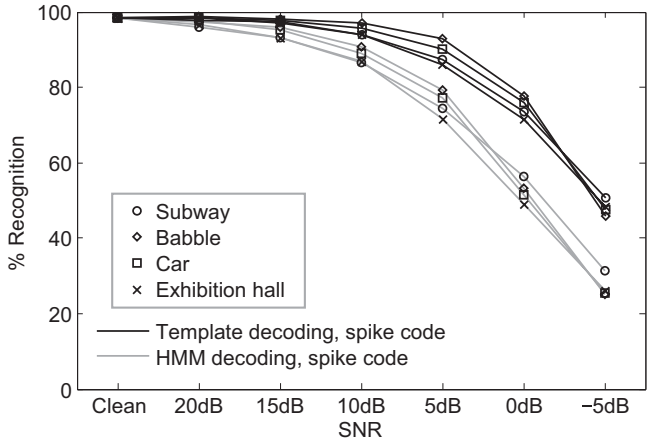

Figure 12: The template-based decoding scheme gives improved recognition results over an HMM decoder when used with the spike sequence code. Recognition rates are shown for the four noise types in AURORA-2 test set A. HMM results are duplicated from Figure 9 for comparison.

rates are displayed for the cases shown previously in Figures 9 and 12. As demonstrated, both the spike sequence coding and the template-based decoding contribute significantly to our system's robustness.

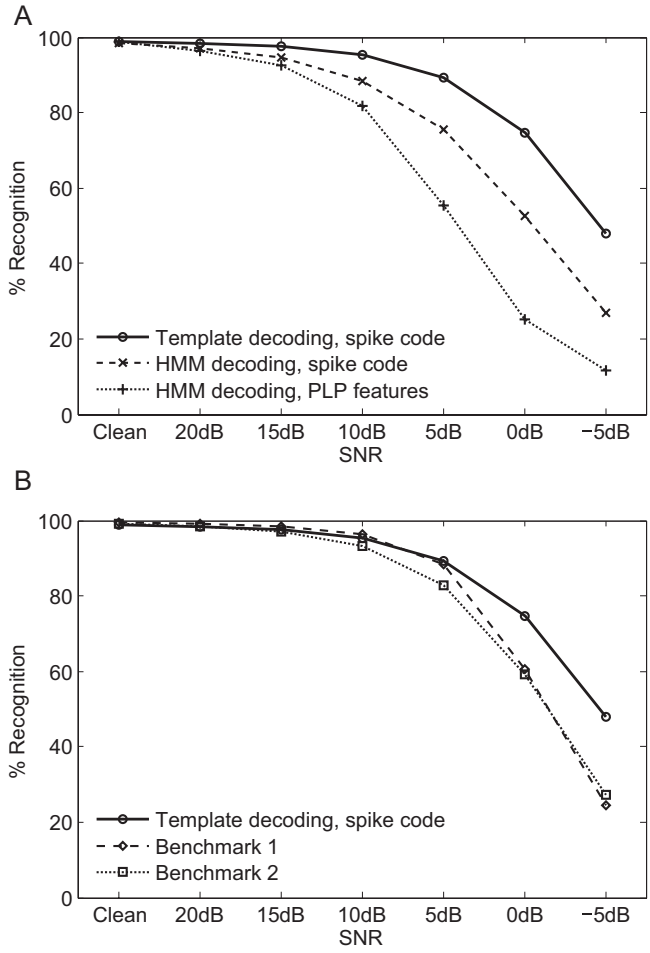

Figure 13: Summary of speech recognition results. Recognition rates are shown averaged over the four noise conditions. (A) Comparison of mean recognition rates for the conditions shown in Figures 9 and 12. Both the spike sequence coding and template-based decoding contribute to the system's noise robustness. (B) Comparison with two benchmark systems from Gemmeke and Cranen (2008). Our system gives improved noise robustness at low SNRs.

In Figure 13B, we compare our system's performance to benchmark results for isolated words from the AURORA-2 data set published in Gemmeke and Cranen (2008). These benchmarks were obtained using state-of-the-art HMM-based recognizers with missing data imputation, in which noise-corrupted spectrotemporal regions in the sound are identified,

removed, and filled in by inference from the remaining data. In these cases, the imputation was performed either by maximum likelihood inference using a harmonicity mask (benchmark 1) or through sparse decomposition of the remaining elements in a basis derived from clean data (benchmark 2). As with our system, the benchmark results were obtained by training only on clean data. Our model's performance is similar to the benchmarks in clean and low-noise conditions but gives a significant increase in performance at 0 dB and −5 dB, with mean recognition rates of 74.7% and 48.1%, respectively, representing relative word error rate (WER) reductions of 35.8% and 28.5%.

5.4 Effects of the Choice of Peripheral Model. While we have shown that our encoding method contributes to the noise robustness of our combined system, an additional question is to what extent our gammatone filter-bank-based peripheral model plays a role. To address this, we ran an additional set of experiments using an alternative, spectrogram-based peripheral model. To obtain the spectrogram, we computed the discrete Fourier transform of the sound waveform in 9 ms. Hamming windows at 1 ms intervals. For direct comparison with the gammatone filter bank method, the frequency axis was converted to an identical ERB frequency scale using triangular interpolation (Davis & Mermelstein, 1980). We observed in other experiments that this nonlinear frequency scaling contributed to improved recognition performance in our model (results not shown). The absolute value of the response in each frequency channel was square root compressed, resampled, and normalized just as in the gammatone method. The remainder of the training and testing procedure was the same as before. The resulting recognition performance was nearly identical to that obtained using the gammatone filter bank, except for a slight decrease in the noise-averaged recognition rates at low SNRs (0.9% decrease at 0 dB: 3.1% at −5 dB). We speculate that this small difference in performance may result from the gammatone filter bank's varied bandwidths, which permit better resolution of fine spectral patterns at low frequencies and sharp temporal modulations at high frequencies.

6 Discussion

Our aim in this work was to use biologically inspired auditory spike encoding to perform robust ASR. Our system achieved recognition performance exceeding that of a robust HMM-based system on a small-vocabulary, isolated-word task at low SNRs. We obtained these results using a model trained only on clean speech, offering a major advantage over statistical approaches to ASR, which typically must be trained in the specific acoustic environment of the end application (Hirsch & Pearce, 2000).

We chose the benchmarks in Figure 13 because they were obtained by a state-of-the-art system on an isolated word task identical to the one used

here. However, other studies that performed continuous speech recognition on the AURORA-2 data can provide further context for our results. Other state-of-the-art systems that, like ours, were trained only on clean data achieved up to 67% recognition at 0 dB and 35% at −5 dB (Barker, Cooke, & Green, 2001; Droppo, Deng, & Acero, 2001; Deng, Droppo, & Acero, 2005; Hirsch & Pearce, 2006), while those trained on data in the same noise conditions as the test data achieved up to 79% at 0 dB and 46% at −5 dB (Hirsch & Pearce, 2006; Chen & Bilmes, 2007; Kalinli, Seltzer, Droppo, & Acero, 2010; Seltzer & Acero, 2011). Our isolated digit results cannot be directly compared to these, but it is clear that the recognition rates that our system attained are quite good considering the severity of the noise conditions.

6.1 Relationship to Previous ASR Systems. As described in section 1, our system's use of feature detectors to represent speech is reminiscent of connectionist ASR systems that use ANNs in their front ends. These systems typically use the ANNs to produce additional frame-based features for input to an HMM decoder (tandem systems) or to directly compute the posterior probabilities of HMM states for each speech frame (hybrid systems). Although most research on these systems has focused on large vocabulary speech recognition tasks, not on noise robustness, one DBN-based tandem system yielded reasonably good results (50% at 0 dB, 25% at −5 dB) on the AURORA-2 connected digits task (Vinyals & Ravuri, 2011).

Our encoding scheme resembles these connectionist systems in its use of discriminatively trained artificial neurons, but a number of significant differences also distinguish our approach. First, our representation is not frame based; acoustic observations are not conveyed as vector features to the decoder at regular time intervals. Rather, the feature detector spikes form a point-process representation of the sound, which is subsequently converted to a sequence of discrete neuron labels corresponding to irregular time intervals. This has the advantage of allowing partial segregation of the representation of signal and noise. As discussed in sections 4 and 5, noise largely has the effect of adding extraneous spikes to the code, while leaving the representation of the speech relatively intact. Our decoding approach exploits this fact by identifying the spike sub-sequence that most closely resembles clean speech.

Second, in connectionist systems, the ANNs are discriminatively trained to produce HMM state alignment probabilities, while our feature detector neurons are trained to discriminate a single spectrotemporal pattern from a large collection of other sounds. This relatively straightforward binary classification task permits training with a simple linear SVM in our system, in contrast to the computationally intensive algorithms used, for example, in DBN training.

Third, our decoder differs from HMM and other statistically based decoders, and even from other template-based decoders, in that only selected

elements in the acoustic sequence are utilized in scoring a test utterance. In this sense, our system can be seen as providing a means of frame selection (Zhuang, Rui, Huang, & Mehrotra, 1998; Cooke, Green, Josifovski, & Vizinho, 2001), but with the selection built directly into the decoding process rather than requiring an additional stage of front-end processing. The selection of this subset of acoustic observations using the LCS is enabled by the discrete nature of the spike sequence code and allows recognition of speech even when statistical models would match the noisy data poorly.

Our treatment of the feature detector spike code as a sequence of discrete neuron labels is reminiscent of classic ASR studies in which vector quantization (VQ) was used to convert cepstral coefficients into sequences of discrete dictionary element labels (Jelinek, 1997). Accordingly, our HMM model training for the spike sequence code was procedurally similar to that used in VQ studies. However, the non-frame-based nature of the code distinguishes our approach. Our system's representation of speech in terms of the sequential appearance of features is also similar in spirit to other studies that represented speech as sequences of psychophysically relevant landmarks such as consonant-vowel transitions and points of highest sonority (Stevens, 2002). In our scheme, however, we make no attempt to parse out these differing types of landmarks, instead relying on the SVM training algorithm to select out the relevant discriminative features. ASR systems using detection of phonetic attributes (Bromberg et al., 2007) or whole words (Jansen & Niyogi, 2010) have also been explored previously, and in the latter case they gave modest gains in noise robustness.

6.2 Comparison of Decoding Methods. The gains in performance seen in our system were attributable in part to our novel template-based decoding method using the LCS. HMMs are the dominant decoding technique in ASR, but they have a number of known weaknesses that have inspired a search for alternative methods (Bourlard, Hermansky, & Morgan, 1996; Ostendorf, 1999; Deng & Huang, 2004). The past few years have seen a resurgence of interest in template-based techniques, particularly since advances in computing power have made them more viable for use on large data sets. A number of studies have revisited the classical template-based ASR method of dynamic time warping (DTW), in which a dynamic programming algorithm is used to compute the similarity of frame-based speech representations (Bridle et al., 1983). Our approach is similar to DTW in its template-based perspective and in its use of dynamic programming to compute the LCS, but significant differences exist as well. DTW has some of the same robustness problems as statistical methods, in that frame-based speech observations in noisy conditions may not match the stored templates any better than they would match a statistical model. By contrast, our system was designed to quantify the match of spike sequences to sequence templates in spite of arbitrary noise corruptions.

It is worthwhile to consider the formal similarities of the LCS algorithm with DTW, and also with the Viterbi algorithm, which is responsible for aligning speech frames with HMM states in HMM-based decoders (Rabiner & Juang, 1993). All three of these algorithms use dynamic programming to perform sequence alignment, but they differ in their alignment objectives and their choice of constraints. In DTW as in most other classical HMM-based methods, speech is represented as a sequence of frame-based vector features. Test utterances are compared directly to stored speech exemplars, in this case using a distance measure based on the pairwise alignment of frames. The dynamic programming objective is to minimize the sum of pairwise local distances between the speech frames, subject to some choice of constraints on the permissible degree of time warping.

Viterbi alignment operates similar to DTW, but with each speech frame aligned with an HMM state rather than a template frame. Historically, the shift from DTW to HMM-based systems held the advantage, in part, that a single HMM could serve as a "prototype" taking the place of a large number of templates. In the Viterbi algorithm, the alignment objective is formulated statistically, with state emission probabilities taking the place of DTW's local distance measure and state transition probabilities taking the place of the alignment constraints (De Wachter et al., 2007).

By contrast with these methods, the LCS algorithm operates exclusively on sequences of discrete symbols. The objective is to maximize the number of aligned symbols between two sequences. As such, it can be seen as a discrete version of the DTW algorithm where the local distance measure has been replaced by a binary local similarity measure—1 for a match between symbols, 0 for a nonmatch—which is to be maximized, rather than minimized, by the algorithm. The lack of a penalty for nonmatches is a key difference with the other methods. It effectively allows an unlimited amount of symbol skipping in the testing and template sequences, which is incompatible with both HMM systems, in which each speech frame must be assigned to exactly one HMM state, and DTW systems, in which typically at most one frame in the template sequence can be contiguously skipped (De Wachter et al., 2007). In our system, the allowance of nonmatches is essential in enabling recognition of heavily noise-corrupted sequences.

A further difference between our system and DTW or Viterbi is the normalization of the LCS score with respect to a distribution compiled from randomly generated spike sequences (see section 5). This is necessitated in part by the variable rate of spike generation by the feature detectors, which yields a wider array of template lengths than does a frame-based representation scheme.

An interesting question for further research is whether a more principled, prototype-based system in the spirit of an HMM could be constructed that preserves the benefits of our LCS-based decoding scheme. A key strength of our method is that it avoids explicit noise modeling, instead selecting out portions of the spike sequence code that together form a good match

for the templates. As such, a purely generative model such as an HMM would seem to be incompatible with our approach. Rather, a more heuristic model that captures the statistics of clean speech but does not penalize nonmatches with test data might be capable of reproducing the success of our template-based system. For our purposes here, we consider that both the general arguments in favor of a template-based approach and our system's effectiveness in the AURORA-2 task recommend our system as a viable alternative to HMMs.

6.3 Biological Significance. A major goal of theoretical neuroscience is to understand the coding principles underlying the brain's processing of sensory signals. Of particular interest is how sensory representations are specialized to process behaviorally relevant natural stimuli such as speech. In this context, speech recognition tasks represent a well-defined functional framework in which to test the advantages of neural coding schemes.

One approach to coding theory that has received significant attention is that of sparse coding, in which representations are adapted to natural stimuli in a way that minimizes the activity of a neural population (Olshausen & Field, 2004). Sparse auditory codes that are adapted to speech have been found to predict some receptive field properties of auditory neurons (Smith & Lewicki, 2006; Klein et al., 2003; Carlson et al., 2012). Recently an alternative training objective based on the sustained firing of cortical neurons was also proposed and produced good matches to physiological data (Carlin & Elhilali, 2013). These methods have been explored in the context of several speech recognition tasks and in some cases gave gains in performance (Kwon & Lee, 2004; Rufiner et al., 2007; Smit & Barnard, 2009; Sivaram et al., 2010; Carlin, Patil, Nemala, & Elhilali, 2012).

Our encoding method based on a discriminative training objective represents yet another alternative approach to the specialization of neural codes to speech. The feature detectors in our system produce responses that are temporally sparse, but here this behavior is a by-product of their selectivity and not an explicit training objective. Rather, the discriminative ability of the neurons is designed to yield noise-robust representations, the formation of which has been proposed as a major goal of auditory coding (Griffiths & Warren, 2004; Nahum, Nelken, & Ahissar, 2008). The formation of robust auditory codes has been explored to some extent experimentally (Las et al., 2005; Bar-Yosef & Nelken, 2007; Moore et al., 2013), but the theoretical principles underlying these codes have not been fully determined. Noise-robust recognition tasks such as the one used here can drive the development of new encoding schemes and enable evaluation of their effectiveness.

The STRFs computed for our feature detector neurons can help identify possible relationships between our model and physiological measurements in the auditory pathway. As discussed in section 3, the complex modulation structures seen in our STRFs putatively locate these neurons in

auditory cortex rather than in lower auditory areas, where simpler receptive field structures are usually observed. In particular, some of our STRFs display strong spectral modulations that, in addition to resembling the formant structure of speech, also resemble the STRFs of noise-invariant neurons recently observed in an avian secondary auditory cortical area (Moore et al., 2013).

The feature detectors in our system were implemented using an artificial neuron model based on weighted summation and thresholded peak search. This model was chosen because it was a minimally complex spike production scheme for which a well-known discriminative training method, SVM, could be used. The inclusion of delayed copies of the AN response to the feature detectors enables a linear summation over time and frequency that is similar to that performed in an STRF model. However, assignment of spikes to peaks in the summation response is an additional nonlinearity not accounted for by a STRF. While not strictly biological, this peak-finding element is essential for maintaining the approximate timing of spikes under changes to the overall level of the summation response in noise. Determining the relationship of our artificial neuron model to the dynamics of real cortical neurons is an additional problem that we did not explore here.

Other recent biologically motivated ASR studies (Loiselle et al., 2005; Verstraeten et al., 2005; Gütig & Sompolinsky, 2009) have used simpler spike encoding schemes than the one we have while focusing efforts on network decoding methods suited to spike representations. In this work, we did not attempt a network implementation of our decoding scheme, but we were motivated by approaches to spike decoding that operate on sequences of sensory spikes (Loiselle et al., 2005; Jin, 2004, 2008). In particular, a neural network using dendritic processing has been presented with dynamics equivalent to a finite state automaton (FSA) (Jin, 2008), which is a powerful general model for the processing of symbolic languages (Martin & Jurafsky, 2000). FSA networks can in principle be trained to recognize large classes of symbolic sequences and may provide a way forward for implementing the template recognition scheme used here in terms of episodic memory in a biologically realistic network. The combination of robust acoustic coding with such powerful sequence-based processing schemes could yield a unified understanding of humans' superior speech recognition performance and is a subject for future work.

7 Conclusion

We presented a template-based ASR system using biologically inspired acoustic coding that achieves highly robust performance on an isolated digit recognition task. Future work will evaluate the system's performance on larger data sets and investigate network implementations of the sequence recognition paradigm. Our work provides a framework for understanding

the selective tuning of auditory neurons and an avenue for further research on the use of neural coding in robust ASR.

Acknowledgments

This research was supported by NSF grant IIS-1116530. D.Z.J. thanks the Neural Systems Laboratory at the University of Maryland for its hospitality during his sabbatical visit. Jason Wittenbach and Sumithra Surendralal gave useful comments on the manuscript.

References

Aertsen, A., & Johannesma, P.I.M. (1981). The spectro-temporal receptive field. *Biological Cybernetics*, *42*(2), 133–143.

Aradilla, G., Vepa, J., & Bourlard, H. (2005). Improving speech recognition using a data-driven approach. In *Proceedings of INTERSPEECH* (Vol. 66, pp. 3333–3336). Red Hook, NY: Curran.

Axelrod, S., & Maison, B. (2004). Combination of hidden Markov models with dynamic time warping for speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004* (pp. 1–173). Piscataway, NJ: IEEE.

Bar-Yosef, O., & Nelken, I. (2007). The effects of background noise on the neural responses to natural sounds in cat primary auditory cortex. *Frontiers in Computational Neuroscience*, *1*.

Barker, J., Cooke, M., & Green, P. (2001). Robust ASR based on clean speech models: An evaluation of missing data techniques for connected digit recognition in noise. In *Proc. Eurospeech* (vol. 1, pp. 213–216). Aalborg, Denmark: Kornmunik Grafiske Løsninger A/S.

Barker, J., Vincent, E., Ma, N., Christensen, H., & Green, P. (2013). The PASCAL CHiME speech separation and recognition challenge. *Computer Speech and Language*, *27*(3), 621–633.

Bergroth, L., Hakonen, H., & Raita, T. (2000). A survey of longest common subsequence algorithms. In *Proceedings of the Seventh International Symposium on String Processing and Information Retrieval, 2000* (pp. 39–48). San Mateo, CA: IEEE Computer Society.

Bitterman, Y., Mukamel, R., Malach, R., Fried, I., & Nelken, I. (2008). Ultra-fine frequency tuning revealed in single neurons of human auditory cortex. *Nature*, *451*(7175), 197–201.

Bourlard, H., Hermansky, H., & Morgan, N. (1996). Towards increasing speech recognition error rates. *Speech Communication*, *18*(3), 205–231.

Bourlard, H., & Morgan, N. (1994). *Connectionist Speech Recognition: A Hybrid Approach*. New York: Springer.

Bridle, J. S., Brown, M. D., & Chamberlain, R. M. (1983). Continuous connected word recognition using whole word templates. *Radio and Electronic Engineer*, *53*(4), 167–175.

Bromberg, I., Qian, Q., Hou, J., Li, J., Ma, C., Matthews, B., . . . Tsao, Y. (2007). Detection-based ASR in the automatic speech attribute transcription project. In *Proceedings of INTERSPEECH* (pp. 1829–1832). Red Hook, NY: Curran.

Carey, M. J., & Quang, T. P. (2005). A speech similarity distance weighting for robust recognition. In *Proceedings of INTERSPEECH* (pp. 1257–1260). Red Hook, NY: Curran.

Carlin, M. A., & Elhilali, M. (2013). Sustained firing of model central auditory neurons yields a discriminative spectro-temporal representation for natural sounds. *PLoS Comput. Biol.*, *9*(3), e1002982.

Carlin, M. A., Patil, K., Nemala, S. K., & Elhilali, M. (2012). Robust phoneme recognition based on biomimetic speech contours. In *Proc. 13th Annu. Conf. Int. Speech Commun. Association*. Red Hook, NY: Curran.

Carlson, N. L., Ming, V. L., & DeWeese, M. R. (2012). Sparse codes for speech predict spectrotemporal receptive fields in the inferior colliculus. *PLoS Computational Biology*, *8*(7), e1002594.

Chen, C.-P., & Bilmes, J. A. (2007). MVA processing of speech features. *IEEE Transactions on Audio, Speech, and Language Processing*, *15*(1), 257–270.

Cooke, M. (2006). A glimpsing model of speech perception in noise. *Journal of the Acoustical Society of America*, *119*, 1562.

Cooke, M., Green, P., Josifovski, L., & Vizinho, A. (2001). Robust automatic speech recognition with missing and unreliable acoustic data. *Speech Communication*, *34*(3), 267–285.

Dahl, G. E., Yu, D., Deng, L., & Acero, A. (2012). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, *20*(1), 30–42.

Davis, S., & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, *28*(4), 357–366.

De Wachter, M., Matton, M., Demuynck, K., Wambacq, P., Cools, R., & Van Compernolle, D. (2007). Template-based continuous speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, *15*(4), 1377–1390.

Dean, I., Harper, N. S., & McAlpine, D. (2005). Neural population coding of sound level adapts to stimulus statistics. *Nature Neuroscience*, *8*(12), 1684–1689.

Demuynck, K., Seppi, D., & Van Compernolle, D. (2011). Progress in example based automatic speech recognition. In *Proceedings of 2011 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 4692–4695). Piscataway, NJ: IEEE.

Deng, L., Droppo, J., & Acero, A. (2005). Dynamic compensation of HMM variances using the feature enhancement uncertainty computed from a parametric model of speech distortion. *IEEE Transactions on Speech and Audio Processing*, *13*(3), 412–421.

Deng, L., & Huang, X. (2004). Challenges in adopting speech recognition. *Communications of the ACM*, *47*(1), 69–75.

Deng, L., & Strik, H. (2007). Structure-based and template-based automatic speech recognition-comparing parametric and nonparametric approaches. In *Proceedings of INTERSPEECH* (pp. 898–901). Red Hook, NY: Curran.

DeWeese, M. R., Wehr, M., & Zador, A. M. (2003). Binary spiking in auditory cortex. *Journal of Neuroscience*, *23*(21), 7940–7949.

Droppo, J., Deng, L., & Acero, A. (2001). Evaluation of the SPLICE algorithm on the Aurora2 database. In *Proceedings of INTERSPEECH* (vol. 1, pp. 217–220). Red Hook, NY: Curran.

Elhilali, M., Fritz, J. B., Klein, D. J., Simon, J. Z., & Shamma, S. A. (2004). Dynamics of precise spike timing in primary auditory cortex. *Journal of Neuroscience*, *24*(5), 1159–1172.

Escabí, M. A., Miller, L. M., Read, H. L., & Schreiner, C. E. (2003). Naturalistic auditory contrast improves spectrotemporal coding in the cat inferior colliculus. *Journal of Neuroscience*, *23*(37), 11489–11504.

Gemmeke, J. F., & Cranen, B. (2008). Using sparse representations for missing data imputation in noise robust speech recognition. *Proc. of EUSIPCO 2008*. Piscataway NJ: IEEE.

Griffiths, T. D., & Warren, J. D. (2004). What is an auditory object? *Nature Reviews Neuroscience*, *5*(11), 887–892.

Gütig, R., & Sompolinsky, H. (2009). Time-warpae invariant neuronal processing. *PLoS Biology*, *7*(7), e1000141.

Heil, P. (2004). First-spike latency of auditory neurons revisited. *Current Opinion in Neurobiology*, *14*(4), 461–467.

Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis of speech. *Journal of the Acoustical Society of America*, *87*, 1738–1752.

Hermansky, H., Ellis, D. P., & Sharma, S. (2000). Tandem connectionist feature extraction for conventional HMM systems. In *Proceedings of the 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing* (vol. 3, pp. 1635–1638). Piscataway, NJ: IEEE.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-R., Jaitly, N., . . . Sainath, T. N. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, *29*(6), 82–97.

Hirsch, H.-G., & Pearce, D. (2000). The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In *Proceedings of ASR2000—Automatic Speech Recognition: Challenges for the new Millenium ISCA Tutorial and Research Workshop*. Bonn: ISCA.

Hirsch, H.-G., & Pearce, D. (2006). *Applying the advanced ETSI frontend to the Aurora-2 task* (Tech. Rep.). http://aurora.hsnr.de/download/Aurora2_afe_v1_1.pdf

Huetz, C., Del Negro, C., Lebas, N., Tarroux, P., & Edeline, J.-M. (2006). Contribution of spike timing to the information transmitted by HVC neurons. *European Journal of Neuroscience*, *24*(4), 1091–1108.

Huetz, C., Philibert, B., & Edeline, J.-M. (2009). A spike-timing code for discriminating conspecific vocalizations in the thalamocortical system of anesthetized and awake guinea pigs. *Journal of Neuroscience*, *29*(2), 334–350.

Hyvarinen, A., & Oja, E. (2000). Independent component analysis: Algorithms and applications. *Neural Networks*, *13*(4), 411–430.

Jansen, A., & Niyogi, P. (2010). Detection-based speech recognition with sparse point process models. In *Proceedings of the 2010 IEEE International Conference on Acoustics Speech and Signal Processing* (pp. 4362–4365). Piscataway, NJ: IEEE.

Jelinek, F. (1997). *Statistical methods for speech recognition*. Cambridge, MA: MIT Press.

Jin, D. Z. (2004). Spiking neural network for recognizing spatiotemporal sequences of spikes. *Physical Review E*, *69*(2), 021905.

Jin, D. Z. (2008). Decoding spatiotemporal spike sequences via the finite state automata dynamics of spiking neural networks. *New Journal of Physics*, *10*(1), 015010.

Kalinli, O., Seltzer, M. L., Droppo, J., & Acero, A. (2010). Noise adaptive training for robust automatic speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, *18*(8), 1889–1901.

Kayser, C., Logothetis, N. K., & Panzeri, S. (2010). Millisecond encoding precision of auditory cortex neurons. *Proceedings of the National Academy of Sciences*, *107*(39), 16976–16981.

Kayser, C., Montemurro, M. A., Logothetis, N. K., & Panzeri, S. (2009). Spike-phase coding boosts and stabilizes information carried by spatial and temporal spike patterns. *Neuron*, *61*(4), 597–608.

Klein, D. J., König, P., & Körding, K. P. (2003). Sparse spectrotemporal coding of sounds. *EURASIP Journal on Advances in Signal Processing*, *2003*(7), 659–667.

Kleinschmidt, M. (2003). Localized spectro-temporal features for automatic speech recognition. In *Proceedings of INTERSPEECH*. Red Hook, NY: Curran.

Kwon, O.-W., & Lee, T.-W. (2004). Phoneme recognition using ICA-based feature extraction and transformation. *Signal Processing*, *84*(6), 1005–1019.

Las, L., Stern, E. A., & Nelken, I. (2005). Representation of tone in fluctuating maskers in the ascending auditory system. *Journal of Neuroscience*, *25*(6), 1503–1513.

Laudanski, J., Edeline, J.-M., & Huetz, C. (2012). Differences between spectrotemporal receptive fields derived from artificial and natural stimuli in the auditory cortex. *PLoS ONE*, *7*(11), e50539.

Leonard, R. (1984). A database for speaker-independent digit recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (vol. 9, pp. 328–331). Piscataway, NJ: IEEE.

Lesica, N. A., & Grothe, B. (2008). Dynamic spectrotemporal feature selectivity in the auditory midbrain. *Journal of Neuroscience*, *28*(21), 5412–5421.

Lewicki, M. S., & Arthur, B. J. (1996). Hierarchical organization of auditory temporal context sensitivity. *Journal of Neuroscience*, *16*(21), 6987–6998.

Loiselle, S., Rouat, J., Pressnitzer, D., & Thorpe, S. (2005). Exploration of rank order coding with spiking neural networks for speech recognition. In *Proceedings of the 2005 IEEE International Joint Conference on Neural Networks* (pp. 2076–2080). Piscataway, NJ: IEEE.

Lu, T., & Wang, X. (2004). Information content of auditory cortical responses to time-varying acoustic stimuli. *Journal of Neurophysiology*, *91*(1), 301–313.

Maier, V., & Moore, R. K. (2005). An investigation into a simulation of episodic memory for automatic speech recognition. In *Proceedings of INTERSPEECH* (pp. 1245–1248).

Martin, J. H., & Jurafsky, D. (2000). *Speech and language processing*. Upper Saddle River, NJ: Prentice Hall. Red Hook, NY: Curran.

Mesgarani, N., Sivaram, G., Nemala, S. K., Elhilali, M., & Hermansky, H. (2009). Discriminant spectrotemporal features for phoneme recognition. In *Proc. INTERSPEECH* (vol. 9, pp. 2983–2986). Red Hook, NY: Curran.

Mesgarani, N., Slaney, M., & Shamma, S. A. (2006). Discrimination of speech from nonspeech based on multiscale spectro-temporal modulations. *IEEE Transactions on Audio, Speech, and Language Processing*, *14*(3), 920–930.

Meyer, B., Wesker, T., Brand, T., Mertins, A., & Kollmeier, B. (2006). A human-machine comparison in speech recognition based on a logatome corpus. In *Speech Recognition and Intrinsic Variation Workshop*. N.p.

Mohamed, A.-r., Dahl, G. E., & Hinton, G. (2012). Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech, and Language Processing*, *20*(1), 14–22.

Moore, R. C., Lee, T., & Theunissen, F. E. (2013). Noise-invariant neurons in the avian auditory cortex: Hearing the song in noise. *PLoS Comput Biol.*, *9*(3), e1002942.

Nahum, M., Nelken, I., & Ahissar, M. (2008). Low-level information and high-level perception: The case of speech in noise. *PLoS Biology*, *6*(5), e126.

Nourski, K. V., Reale, R. A., Oya, H., Kawasaki, H., Kovach, C. K., Chen, H., . . . Brugge, J. F. (2009). Temporal envelope of time-compressed speech represented in the human auditory cortex. *Journal of Neuroscience*, *29*(49), 15564–15574.

Olshausen, B. A., & Field, D. J. (2004). Sparse coding of sensory inputs. *Current Opinion in Neurobiology*, *14*(4), 481–487.

Ostendorf, M. (1999). Moving beyond the "beads-on-a-string" model of speech. In *Proc. IEEE ASRU Workshop* (pp. 79–84). Piscataway, NJ: IEEE.

Rabiner, L., & Juang, B.-H. (1993). *Fundamentals of speech recognition*. Upper Saddle River, NJ: Prentice Hall.

Ramasubramanian, V., Kulkarni, K., & Kämmerer, B. (2008). Acoustic modeling by phoneme templates and modified one-pass DP decoding for continuous speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 4105–4108). Piscataway, NJ: IEEE.

Rauschecker, J. P., Tian, B., & Hauser, M. (1995). Processing of complex sounds in the macaque nonprimary auditory cortex. *Science*, *708*, 111–114.

Rodríguez, F. A., Read, H. L., & Escabí, M. A. (2010). Spectral and temporal modulation tradeoff in the inferior colliculus. *Journal of Neurophysiology*, *103*(2), 887–903.

Rufiner, H., Martínez, C., Milone, D., & Goddard, J. (2007). Auditory cortical representations of speech signals for phoneme classification. *MICAI 2007: Advances in Artificial Intelligence* (pp. 1004–1014). New York: Springer.

Sadagopan, S., & Wang, X. (2008). Level invariant representation of sounds by populations of neurons in primary auditory cortex. *Journal of Neuroscience*, *28*(13), 3415–3426.

Schädler, M. R., Meyer, B. T., & Kollmeier, B. (2012). Spectro-temporal modulation subspace-spanning filter bank features for robust automatic speech recognition. *Journal of the Acoustical Society of America*, *131*, 4132–4151.

Scharenborg, O. (2007). Reaching over the gap: A review of efforts to link human and automatic speech recognition research. *Speech Communication*, *49*(5), 336–347.

Schnupp, J. W., Hall, T. M., Kokelaar, R. F., & Ahmed, B. (2006). Plasticity of temporal pattern codes for vocalization stimuli in primary auditory cortex. *Journal of Neuroscience*, *26*(18), 4785–4795.

Seltzer, M. L., & Acero, A. (2011). Factored adaptation for separable compensation of speaker and environmental variability. In *IEEE Workshop on Automatic Speech Recognition and Understanding* (pp. 146–151). Piscataway, NJ: IEEE.

Sen, K., Theunissen, F. E., & Doupe, A. J. (2001). Feature analysis of natural sounds in the songbird auditory forebrain. *Journal of Neurophysiology*, *86*(3), 1445–1458.

Seppi, D., & Van Compernolle, D. (2010). Data pruning for template-based automatic speech recognition. In *Proceedings of INTERSPEECH* (pp. 901–904). Red Hook, NY: Curran.

Sivaram, G. S., Nemala, S. K., Elhilali, M., Tran, T. D., & Hermansky, H. (2010). Sparse coding for speech recognition. In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing* (pp. 4346–4349). Piscataway, NJ: IEEE.

Slaney, M. (1993). *An efficient implementation of the Patterson-Holdsworth auditory filter bank* (Tech. Rep). Cupertino, CA: Apple Computer, Perception Group.

Smit, W. J., & Barnard, E. (2009). Continuous speech recognition with sparse coding. *Computer Speech and Language*, *23*(2), 200–219.

Smith, E., & Lewicki, M. S. (2005). Efficient coding of time-relative structure using spikes. *Neural Computation*, *17*(1), 19–45.

Smith, E. C., & Lewicki, M. S. (2006). Efficient auditory coding. *Nature*, *439*(7079), 978–982.

Sroka, J. J., & Braida, L. D. (2005). Human and machine consonant recognition. *Speech Communication*, *45*(4), 401–423.

Steinschneider, M., Volkov, I. O., Fishman, Y. I., Oya, H., Arezzo, J. C., & Howard, M. A. (2005). Intracortical responses in human and monkey primary auditory cortex support a temporal processing mechanism for encoding of the voice onset time phonetic parameter. *Cerebral Cortex*, *15*(2), 170–186.

Stevens, K. N. (2002). Toward a model for lexical access based on acoustic landmarks and distinctive features. *Journal of the Acoustical Society of America*, *111*, 1872.

Strik, H. (2006). How to handle pronunciation variation in ASR: By storing episodes in memory? In *Speech Recognition and Intrinsic Variation Workshop*. N.p.

Theunissen, F. E., Sen, K., & Doupe, A. J. (2000). Spectral-temporal receptive fields of nonlinear auditory neurons obtained using natural sounds. *Journal of Neuroscience*, *20*(6), 2315–2331.

Trentin, E., & Gori, M. (2001). A survey of hybrid ANN/HMM models for automatic speech recognition. *Neurocomputing*, *37*(1), 91–126.

Vapnik, V. N. (1998). *Statistical learning theory*. New York: Wiley-Interscience.

Versnel, H., Zwiers, M. P., & van Opstal, A. J. (2009). Spectrotemporal response properties of inferior colliculus neurons in alert monkey. *Journal of Neuroscience*, *29*(31), 9725–9739.

Verstraeten, D., Schrauwen, B., Stroobandt, D., & Van Campenhout, J. (2005). Isolated word recognition with the liquid state machine: A case study. *Information Processing Letters*, *95*(6), 521–528.

Vinyals, O., & Ravuri, S. V. (2011). Comparing multilayer perceptron to deep belief network tandem features for robust ASR. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 4596–4599). Piscataway, NJ: IEEE.

Wang, X., & Kadia, S. C. (2001). Differential representation of species-specific primate vocalizations in the auditory cortices of marmoset and cat. *Journal of Neurophysiology*, *86*(5), 2616–2620.

Young, S. J., Evermann, G., Gales, M.J.F., Kershaw, D., Moore, G., Odell, J. J.,... Woodland, P. C. (2005). *The HTK book version 3.4*. Cambridge: Cambridge University English Department.

Yuan, J., & Liberman, M. (2008). Speaker identification on the SCOTUS corpus. *Journal of the Acoustical Society of America*, *123*(5), 5687–5690.

Zhao, S. Y., & Morgan, N. (2008). Multi-stream spectro-temporal features for robust speech recognition. In *Proc. INTERSPEECH* (pp. 898–901). Red Hook, NY: Curran.

Zhuang, Y., Rui, Y., Huang, T. S., & Mehrotra, S. (1998). Adaptive key frame extraction using unsupervised clustering. In *Proceedings of the 1998 International Conference on Image Processing* (vol. 1, pp. 866–870). Piscataway, NJ: IEEE.

Received April 2, 2013; accepted October 2, 2013.