Acoustic Transient Classification with a Template Correlation Processor

R. Timothy Edwards
Johns Hopkins University Applied Physics Laboratory
11100 Johns Hopkins Road
Laurel, MD 20723-6099

April 9, 1999

Abstract

I present an architecture for acoustic pattern classification using trinary-trinary template correlation. In spite of its computational simplicity, the algorithm and architecture represent a method which greatly reduces bandwidth of the input, storage requirements of the classifier memory, and power consumption of the system without compromising classification accuracy. The linear system should be amenable to training using recently-developed methods such as Independent Component Analysis (ICA), and we predict that behavior will be qualitatively similar to that of structures in the auditory cortex.

1 Introduction

Our research in algorithms for acoustic pattern classification in micropower analog hardware [1, 2, 3] began with the simple premise that groups of acoustic transients (short, impulsive, and structurally simple sounds) could be classified reasonably well by a linear hyperplane separator. A template correlation in the time-frequency domain is a candidate linear classifier having the simple general form

$$c_z[t] = \sum_{m=1}^{M} \sum_{n=1}^{N} x[t-n,m]\, p_z[n,m] \tag{1}$$

where M is the number of frequency channels of the input array x, having been processed by a filterbank to retrieve a measure of time-averaged energy over the passband of the channel, N is the number of time bins chosen for the window of observable data, p_z is the matrix of template pattern values for pattern z, and t is the current time, in discrete units of the sampling period which is typically 1 ms. The formula produces a running correlation $c_z[t]$ of the input array with the template z. A signal may be interpreted as belonging to class z when the output $c_z[t]$ exceeds a threshold, and is the maximum correlation across all classes, $c_z[t] > c_j[t] \;\; \forall j \neq z$.

Choosing template correlation as a baseline algorithm, Pineda [1] searched for methods to reduce the computational load of the process without compromising accuracy on a classification task. One goal was to attack the problem with a massively parallel architecture; another was to reduce the numerical space of the problem to low resolution, preferably binary or bipolar, whereever possible. This research led to two possible paths for a micropower architecture: One involving a binary template and continuous-valued input [3], and another involving both a binary template and binary input. The slow rate at which correlations are computed in this architecture, along with the parallelism in the system and its qualitative relationship to proposed mechanisms for acoustic pattern recognition in the auditory cortex, lead us to consider the architecture a neuromorphic one, with potential insights into the neural mechanics of pattern recognition in the brain.

2 An architecture for trinary-trinary template correlation

A notable conclusion from the software trials of varieties of template correlation architectures was the observation that after appropriate manipulations to derive trinary values $\{-1, 0, +1\}$ from continuous-valued data, comparing trinary inputs against trinary templates yields results comparable to the continuous-valued baseline algorithm. The manipulations of input and the computation of templates which maintains the highest accuracy on correlation tests are described below. The trinary-trinary correlator is "digital," though not in the traditional sense of a DSP or microprocessor, with floating-point operations on wide data busses, but rather involves operations efficiently and simply implemented at the bit level with opportunities for significant amounts of parallelism. The frontend processor of such a system, shown in Figure 1, is analog (though not necessarily continuous-time), and has an architecture similar to models of the mammalian cochlea [5, 6, 7]. Unlike the cochlea models, it uses parallel channels rather than cascaded filters as a result of implementation decisions, and has many fewer channel outputs per octave. The filterbank architecture is constructed of constant-Q filters with band centers spaced evenly

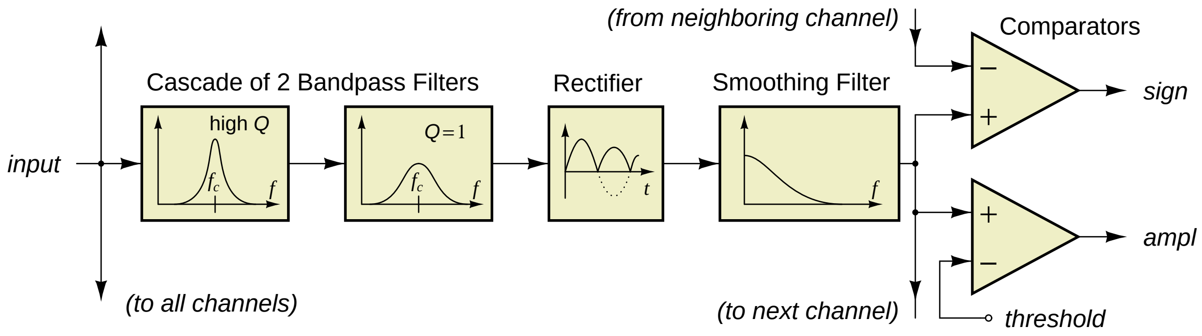

Figure 1: A single channel of the frontend parallel filterbank.

on a logarithmic frequency scale. An audio preamplifier simultaneously drives all filterbank inputs. Two comparators generate a two-bit output representing a trinary value as shown in Table 1.

ampl	sign	$x[t,m]$
0	X	0
1	0	-1
1	1	$+1$

Table 1: 2-bit Sign-amplitude trinary representation of input and template values in the digital ATP. "X" indicates a don't-care condition.

In a system of trinary values, signed-value computations reduce to simple logic operations. The multiplication $x[t-n,m]\,p_z[n,m]$ of Equation (1) takes the form of an exclusive-or and an "and" operation as shown in Figure 2.

A good strategy for the digital system is the pipelined, parallel column-wise configuration of Figure 2 with template values stored in SRAM-like cells incorporated as an integral part of the system. A first impression might be to perform the correlation as an addition using adder circuits. But the system may be rendered simpler by selecting each frequency channel serially and accumulating the result using counters, a comparatively low-level process. Each counter performs 2's-complement increment or decrement and shifts its value to the next column at the end of each counting cycle. At the end of each 1 ms timestep, the system performs a simple analysis over all template outputs, determining if any output exceeds a threshold and, if so, which output has the maximum value.

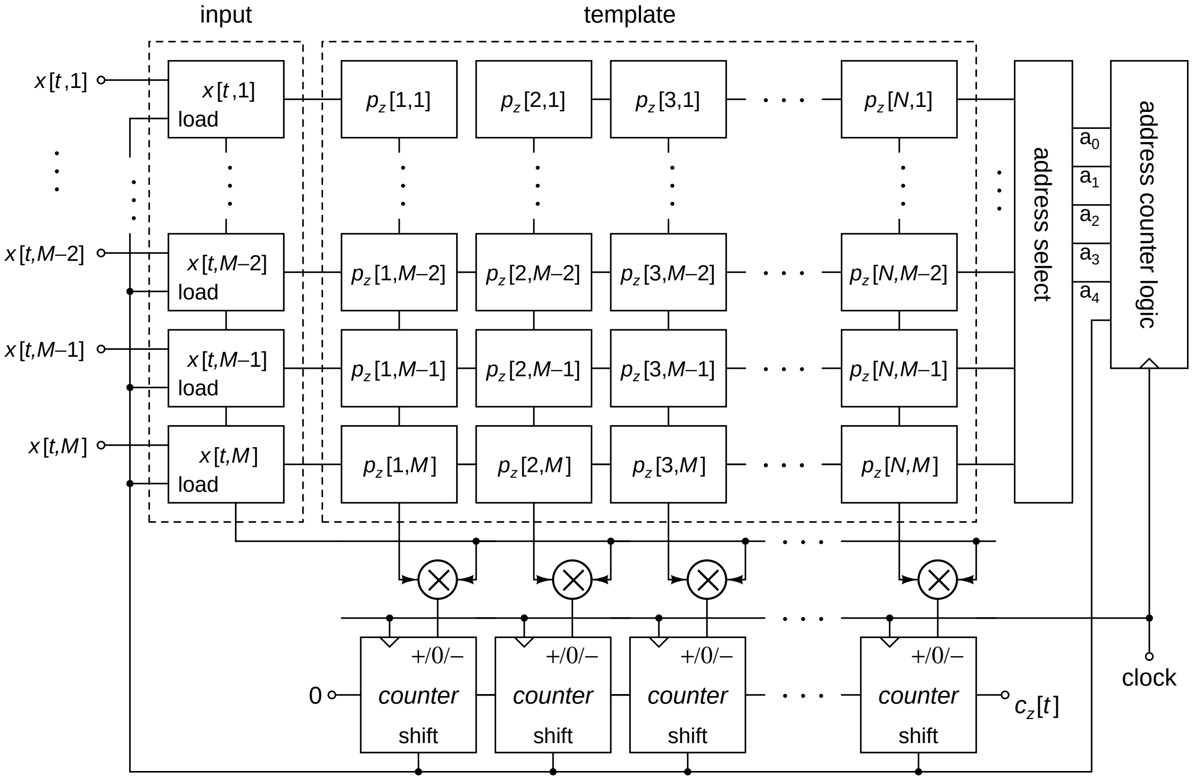

Figure 2: Architecture for the trinary-trinary correlator.

3 Experimental results of the trinary-trinary correlation hardware

Figure 3 is a photograph of the board-level system developed as a real-time implementation of the trinary-trinary correlation algorithm. For purposes of evaluating its performance on an acoustic transient classification test, input was supplied through the audio port of a computer. The dataset used for evaluation consists of approximately twenty examples each of ten target transient classes recorded in an office environment.

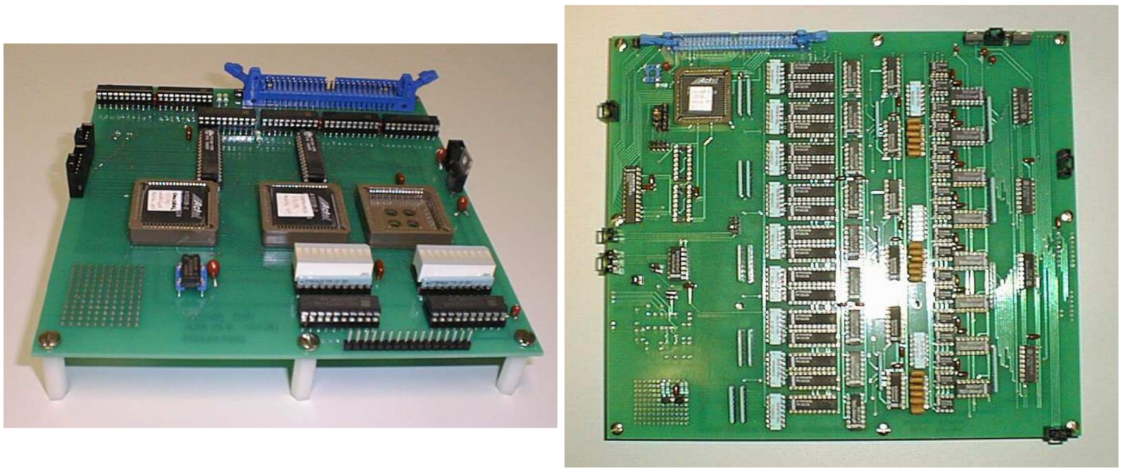

Figure 3: *Left:* Digital ATP correlator system, configurable for up to sixteen templates. *Right:* One board of the switched capacitor frontend filterbank, bandpass filtering and encoding the audio input into sixteen parallel channels.

3.1 Method of template generation

Templates are generated in the simplest manner by averaging aligned examples from a training dataset. All events in the training dataset have been "tagged" with their target class value. For each class z in the training dataset:

1. Set all values in template z to zero
2. Align each example transient on its tag
3. Truncate to the length of the template (100 samples)
4. Sum all examples using the relation

$$p_z[n,m] = p_z[n,m] + x[t-n,m]. \tag{2}$$

5. Threshold the template values using the relation

$$p_z[n,m] = \begin{cases} +1 & \text{if } p_z[n,m] > 0 \\ -1 & \text{if } p_z[n,m] < 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

3.2 Method of input segmentation

When training a template correlation system using the average value of the training set examples, the output can only be expected to produce a reliable classification when the input is perfectly aligned to the template, not shifted in time to the right or to the left. This is very clearly shown in the output response of the trinary-trinary correlator as shown in Figure 6, in which the output for the target class, though strong and persistant at and near the point of best alignment, is accompanied by spurious false classifications away from the alignment point. False classifications can be ignored only if the segmentation algorithm correctly identifies the presence of a transient event and predicts the correct time at which to accept the classification result. Making the correlation output temporally shift-invariant is the purpose of some of training methods such as Independent Component Analysis, Support Vector Machines, and Perceptron Neural Networks. Any system which assumes exact alignment of the input and template requires a robust segmentation algorithm to predict the best point of alignment. Pineda [1] used the scheme shown in Figure 4 as a segmenter for the baseline algorithm. The raw output of all channels of the frontend system were thresholded, summed together, and binarized by another threshold, resulting in a "noisy" segmentation. The noisy segmentation result was filtered in parallel by two lowpass functions with time constants of 1 ms and 10 ms, respectively, thresholded once more, and combined with an OR operation. The longer time constant ensured a clean segmentation for the duration of the input transient, while the shorter time constant ensured a quick reponse of the system to the onset of a transient event.

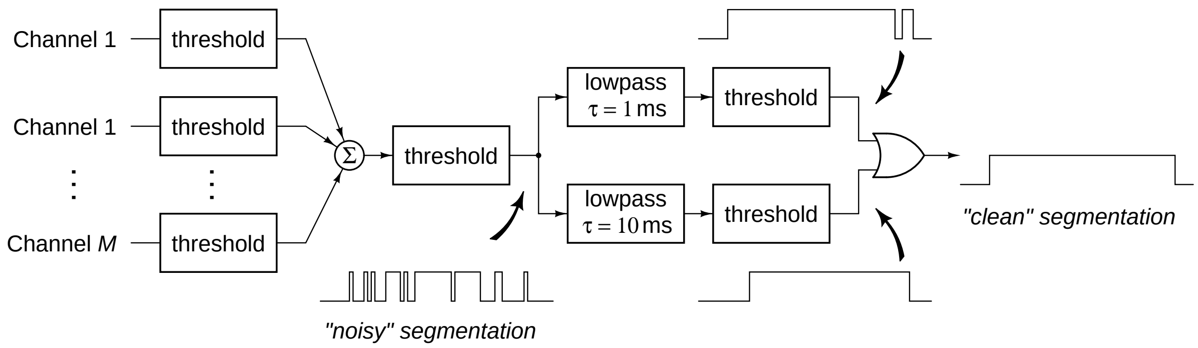

Figure 4: A mixed-signal method for detecting the onset of a transient event from the output of an analog frontend filterbank.

For the trinary-trinary correlator, I used a similar approach; the first threshold operation is already implemented in the frontend filterbank system to produce the “ampl” bit for each channel. The summation is performed by a counter, and the lowpass filter functions are approximated by a simple finite state machine using a few more counters.

3.3 Experimental Results

I evaluated the performance of the trinary-trinary correlation real-time hardware using a leave-one-out cross-validation method. Templates values were determined offline using data obtained from the frontend hardware. Figure 5 shows the templates derived for one step of the cross-validation test.

Figure 6 shows example outputs of the frontend system, the correlator, and the segmenter over a short period of time during the cross-validation test. For these two input transients, the correlator performed correct classification.

Table 2 shows the confusion matrix resulting from the cross-validation test. The system exhibits better than 95% accuracy, compared with a recognition rate of 96.4% for the baseline algorithm (1), which should be guaranteed virtually perfect results on linearly separable data. The cross-validation tests results confirm that the simple trinary-trinary correlation method approaches the baseline algorithm in accuracy, even when only trinary-valued time-frequency decompositions of the audio input are available for training, and when a simple finite state machine consisting of a few small binary counters and assorted logic is used to segment the input.

Event	Bar	Book	Can	Dink	Door	Finger	Hand	Mallet	Shelf	Tub	?	?
Bar	$\boxed{20}$	0	0	0	0	0	0	0	0	0	0	0
Book	0	$\boxed{20}$	0	0	0	0	0	0	0	0	0	0
Can	0	0	$\boxed{24}$	0	0	0	0	0	0	0	0	0
Dink	0	0	0	$\boxed{20}$	0	0	0	0	0	0	0	0
Door	0	0	0	0	$\boxed{20}$	0	0	0	0	0	0	0
Finger	0	0	0	0	0	$\boxed{17}$	1	0	0	0	0	2
Hand	0	0	0	0	0	1	$\boxed{19}$	0	0	0	0	0
Mallet	0	0	2	0	0	0	0	$\boxed{18}$	0	0	0	0
Shelf	0	0	0	0	0	0	0	0	$\boxed{20}$	0	0	0
Tub	0	0	0	0	0	0	0	0	0	$\boxed{20}$	0	0
?	0	0	0	0	0	3	0	0	0	0	$\boxed{0}$	0
?	0	0	0	0	0	0	0	0	0	0	0	$\boxed{7}$

Total instances presented:	215
Total events found:	214
Correct:	205
Incorrect:	9
Missed:	1
Correctly identified:	95.3%
(In set of known targets):	97.2%
Incorrectly identified:	4.2%
Missed:	0.5%

Table 2: Real-time recognition task: Cross-validation on the transient dataset using the simple finite state machine segmenter. In this transient set, spurious transient events were hand-tagged as “unknown” classes (denoted by a question mark).

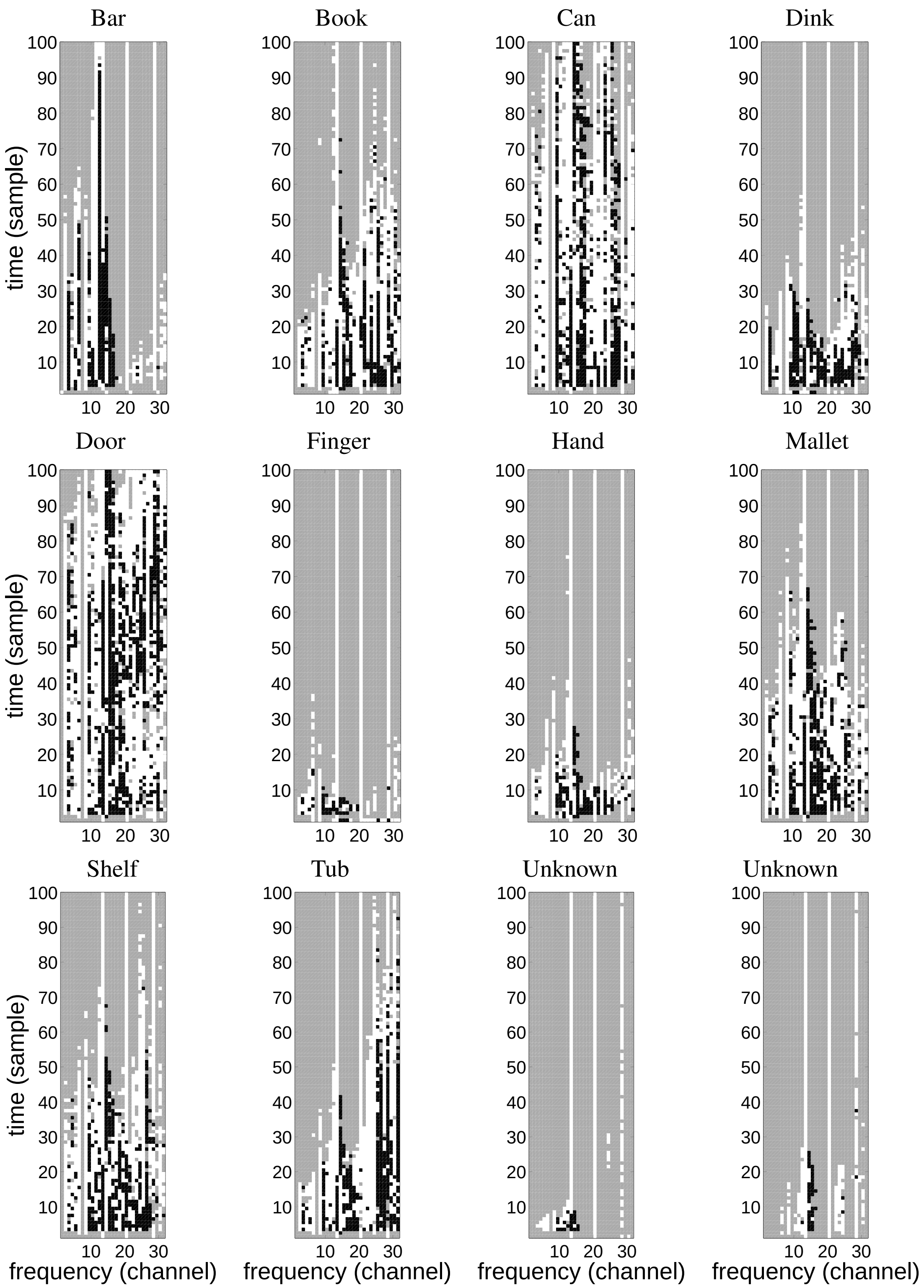

Figure 5: Templates for twelve transient classes, determined from data obtained from the frontend hardware. Frequency is on the x-axis, with the lowest frequency channel on the right, and time is on the y-axis, with the transient onset at the bottom of the template. Colors correspond to trinary values as follows: black = -1, white = $+1$, gray = 0.

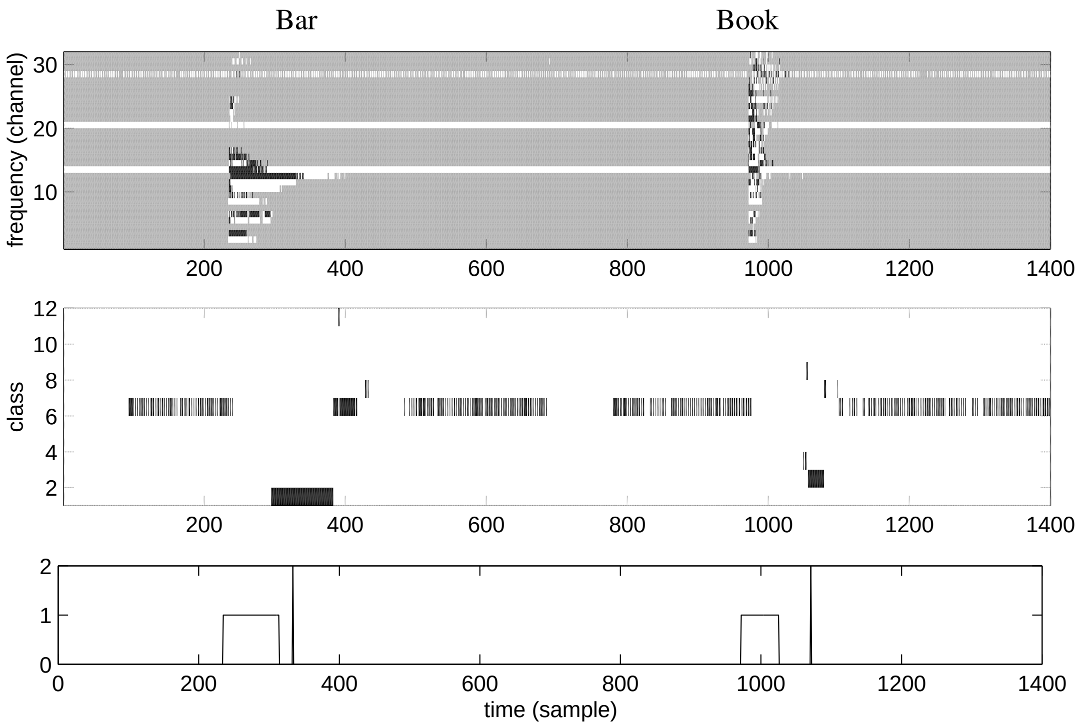

Figure 6: System input and output for sound “bar” *(left)* and “book” *(right)*. Top graph is the frontend system output (32 channels); middle graph is the correlation output over 12 classes; bottom graph is the output of the segmenter showing both the segmented transient and the optimal point of alignment.

4 The correlator as a model of the auditory cortex

A potential use of the acoustic transient processor is the investigation of acoustic processing in the brain. Recent research by Shamma [10] shows that certain areas of the mammalian auditory cortex encode time-frequency maps in neural “wetware.” Shamma subjected neural recordings to analysis and was able to produce the time-frequency mapping encoded by single neuron outputs. These maps are qualitatively similar to spatial mappings of the independent components of natural sounds as produced by Bell and Sejnowski [11]. A large proportion of both the neural and ICA outputs have Gabor logon-like characteristics (that is, sine- and cosine-modulated Gaussians in 2-dimensional space). This is not entirely surprising given the relationship of all three systems (wavelets, independent component analysis, and biological neural processing) to information theory and the efficient representation of signals.

A typical example of a map in the auditory cortex is shown in Figure 7. The figure represents the level of neural activity (spikes per second) in response to auditory signals over a range of frequency and change of frequency with time. Most of the map (gray) is unencoded—that is, the neuron does not respond at all, either with excitation or inhibition, to inputs in that range of time and frequency. Other areas produce an excitatory response (white) and others, an inhibitory response (black). In the figure, the mapping is tuned to a single falling tone of specific pitch and drop rate. A simple model of the distributions which covers most cases observed in the auditory cortex can be formed by one or two Gaussians encoding regions of either odd symmetry, as shown in the figure, or even symmetry consisting of a large excitatory response surrounded by inhibitory side lobes.

The acoustic transient architecture algorithm provides a way to actively test theories associated with the observations from the auditory cortex. In particular, the trinary encoding of the digital transient classifier hardware can encode regions of positive and negative response inside a region of zero response, as shown on the right-hand side of Figure 7. This is essentially an attempt to map the correlation algorithm onto a neuron, where each location of the template maps more or less directly to a synapse connection between the delayed output of the cochlear frequency mapping and the neuron under study

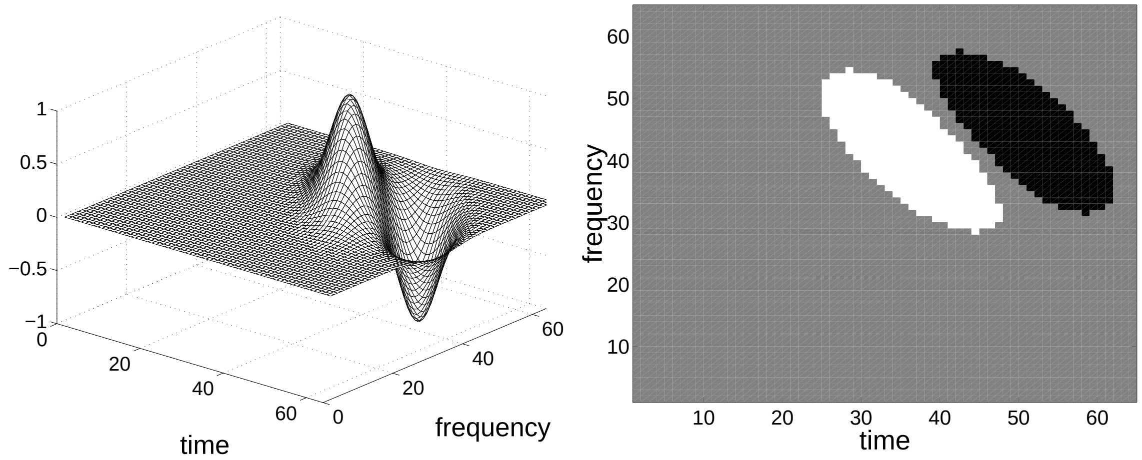

Figure 7: Example of a biologically-inspired template encoding a falling tone of specific frequency and rate. *Left:* Continuous-valued encoding. *Right:* Trinary encoding.

in the auditory cortex. The neuron is more efficient than the correlator because it presumably uses no more synaptic connections than necessary for each response. It also has a higher resolution than the correlator because the synaptic weights encode continuous-valued weights, a neat little trick of nature that is rather difficult to duplicate in silicon. Also, the neurons of the brain make optimally efficient use of the nonlinearities inherent to the system, whereas the correlator is a linear system (certain methods of training, such as independent component analysis, depend on this fact). Despite the differences between a real neuron and its model implemented as a linear, trinary correlation, we can expect that the system so implemented should remain reasonably robust under the same algorithmic manipulations.

References

[1] F. J. Pineda, K. Ryals, D. Steigerwald, and P. Furth, "Acoustic Transient Processing using the Hopkins Electronic Ear," *World Conference on Neural Networks*, Washington, D.C., 1995.

[2] F. J. Pineda, G. Cauwenberghs, R. T. Edwards, "Bangs, Clicks, Snaps, Thuds, and Whacks: An Architecture for Acoustic Transient Processing," *Neural Information Processing Systems (NIPS)*, Denver, 1996.

[3] R. T. Edwards, G. Cauwenberghs, and F. J. Pineda, "A Mixed-Signal Correlator for Acoustic Transient Classification," *Proc. ISCAS '97*, **1**, pp. 621–625, Jun. 1997.

[4] R. T. Edwards, G. Cauwenberghs, and F. J. Pineda "Optimizing Correlation Algorithms for Hardware-based Transient Classification," *Neural Information Processing Systems (NIPS)*, 1998.

[5] C. A. Mead, *Analog VLSI and Neural Systems*, Reading, MA, Addison-Wesley (1989).

[6] W. Liu, A. G. Andreou, and M. H. Goldstein, Jr., "Voiced-Speech Representation by an Analog Silicon Model of the Auditory Periphery," *IEEE Trans. Neural Networks*, **3** (3), May 1992, pp. 477–487.

[7] A. van Schaik, E. Fragnière, E. Vittoz, "A Silicon Model of Amplitude Modulation Detection in the Auditory Brainstem," *Neural Information Processing Systems (NIPS)*, Denver, 1996.

[8] K. P. Unnikrishnan, J. J. Hopfield, and D. W. Tank, "Connected-Digit Speaker-Dependent Speech Recognition Using a Neural Network with Time-Delayed Connections," *IEEE Transactions on Signal Processing*, **39**, pp. 698–713, 1991.

[9] K. P. Unnikrishnan, J. J. Hopfield, and D. W. Tank, “Speaker-Independent Digit Recognition Using a Neural Network with Time-Delayed Connections,” *Neural Computation*, **4**, pp. 108–119, 1992.

[10] K. Wang and S. Shamma, “Representation of Acoustic Signals in the Primary Auditory Cortex,” *IEEE Trans. Audio and Speech Processing*, **3**(5), pp. 382–395, 1995.

[11] A. J. Bell and T. J. Sejnowski, “Learning the higher-order structure of a natural sound,” *Network Computation in Neural Systems*, **7**(2), pp. 261–267, 1996.