

ANALYSIS OF TETRODE RECORDINGS IN CAT VISUAL SYSTEM

Sergei Rebrik¹, Svilen Tzonev¹, and Ken Miller^{1,2}

¹Keck Center for Integrative Neuroscience and Department of Physiology,

²Department of Otolaryngology,

University of California at San Francisco, CA 94143

rebrik@phy.ucsf.edu, svilen@phy.ucsf.edu, ken@phy.ucsf.edu

<http://mccoy.ucsf.edu/>

INTRODUCTION

From simultaneous recording of closely located neurons one can obtain information about local inter-neuronal connectivity and acquire fine-scale neuronal response mappings. A multitrode (by which we mean a tight bundle of traditional extracellular electrodes) (McNaughton et al., 1983) is a natural choice for this kind of recording, because it allows discrimination of multiple neurons at a single recording site. Separation of spikes from different neurons is based on the assumption that spike amplitude decreases with the distance between the neuron and the tip of the electrode. Thus, the ratio of spike amplitudes measured on different wires provides reliable information for spike discrimination. The tetrode, which is made of four very thin wires (typical wire diameter with the insulation is 20 μm) is the minimal set of electrodes that provides "triangulation" in the three-dimensional space. If tetrode tips are not located in a plane, and under the idealization that neurons are point sources in a homogenous medium, the ratios of spike amplitudes will be unique for each neuron. In contrast to the tetrode, a single electrode is "spatially blind", and its use for multi-neuron recordings can easily lead to mixing of signals from different cells whose spike amplitudes as measured at the electrode site are equal.

The multi-unit nature of tetrode recordings imposes additional requirements on spike-sorting procedures. While in traditional extracellular recordings an experimentalist can position the electrode to enhance the signal from the neuron of interest, tetrodes are best positioned to record from as many cells as possible. This means that spike sorting can no longer be based on simple thresholding or on window discrimination (often implemented in hardware) - sorting of tetrode data is essentially software-based. As the tetrode technique is relatively new, and the number of labs using it is still small, the choice of software for spike-sorting is very limited. Existing packages are often system- and hardware-dependent, which makes it hard to port them to different lab environments.

In this paper we describe our ongoing development of spike sorting software for tetrode recordings. We discuss a natural spike-parameter representation that improves spike clustering, because noise common to all four electrodes is suppressed. We conclude by discussing future development of spike-sorting algorithms.

SPIKER: SOFTWARE FOR OFFLINE SORTING OF TETRODE DATA

The main goal behind the design of this software was to provide a user with a simple and universal program that could be used "out of the box". Program requirements included an intuitive graphical user interface, openness to modifications by the end user, and cross-platform portability.

Spiker is written in C using the commercially available "Galaxy C" environment. "Galaxy C" is essentially a C library that allows one to write system-independent programs that have a "native" graphical user interface.

The input file format of Spiker is a continuous binary stream of signed 16-bit ADC samples (any byte order). The total number of recording channels in the input file can be in the 1 to 16 range. Data samples are assumed to be in the following format: for every ADC time tick, samples for all channels are recorded in the channel order, and this structure is repeated for the duration of the recording. No information on number of channels, ADC sampling rate or timing is stored in the file. This allows use of Spiker with any data acquisition system that can continuously stream data to the disk (i.e. raw data format). An alternative to this format is to record only a preset number of points surrounding a spike peak, thus reducing the size of the data file. We chose not to use this approach because it severely limits the ability to reconstruct spike overlaps, an issue we intend to address in future work (see also Lewicki, 1994; Sahani et al., 1997). In addition, such extraction requires setting of thresholds at recording time. As we discuss later, threshold settings are very important for proper spike clustering, and it is easy to introduce irrecoverable losses of data if thresholds are set before data acquisition. Finally, in conditions in which many threshold-crossing events occur (many neurons and/or reasonable sustained firing rates), data compression obtained by extracting 1.5 msec spikes may not be substantial.

The spike sorting procedure implemented in Spiker is outlined below. First, spikes are detected in traces by a thresholding procedure. By default the threshold is set at 4 standard deviations from the channel mean, measured using data points taken from 10 random chunks. If needed, the threshold can be interactively adjusted by the user. After the spike detection, the amplitudes of spikes at all channels are measured. Peaks at different channels do not always precisely coincide in time, so the search for an extremum is performed on each channel individually. To reduce digitization error, data points are 10 times oversampled around the detected peak using Fourier interpolation over 80 neighboring points, and the amplitudes are corrected accordingly. This step is especially important for low ADC rates (20 kHz and below).

After the previous step, each spike (event) has a set of associated amplitudes; for a tetrode, there are four such amplitudes. Then each event can be treated as a point in this 4-dimensional space. The task of a clustering procedure is to determine the number of clusters and to assign events to clusters. Currently this task is performed manually. To visualize the distribution of points in four-dimensional space, we, like many others, project this space onto 6 two-dimensional planes (the number 6 is given by the number of different pairs of amplitudes). In other words, for the plane defined by channels X and Y, for each event a point is plotted at coordinates A_x and A_y , the amplitudes on the two channels (the remaining two amplitudes are disregarded for this plane). Thus, each event generates 6 points, one point per plane. Figure 1a shows a snapshot of a Spiker window used for

clustering. The user can draw an ellipse around the points and mark them as belonging to a particular cluster, and this marking will be seen simultaneously in all displayed planes.

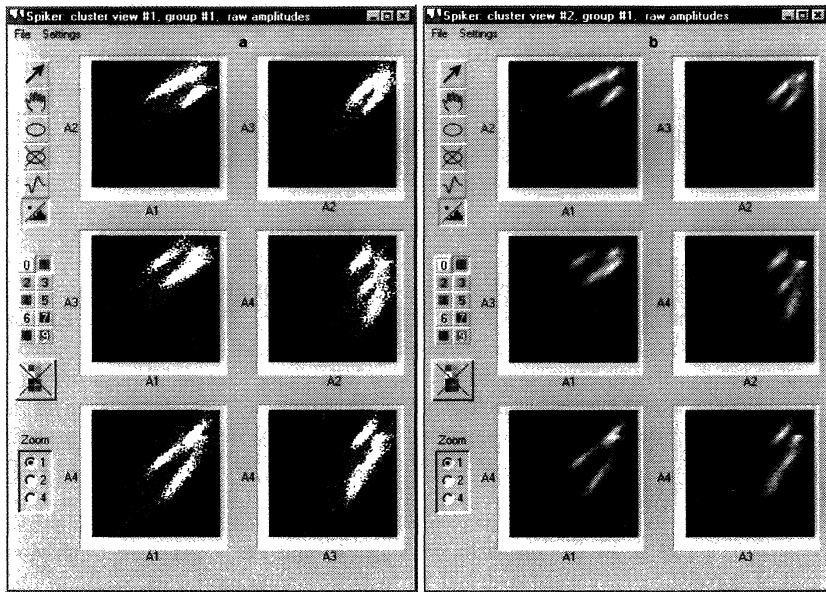


Figure 1. Clustering in raw amplitude coordinates. The point with (0,0) coordinates is located in the upper right corner of the quadrant. Spike amplitudes are negative and the absolute value of the spike amplitude grows in the down-left direction. Panel a shows events in the "point-per-event" representation, while panel b presents density of events coded by brightness. Note better cluster separation in panel b.

The total number of events in the recording affects perceived cluster size. As the total number of events grows, clusters increase in size because the number of outliers also grows. This can lead to overlap of clusters and make classification difficult or impossible. To avoid this kind of problem, a density distribution of events within the plane can be calculated and displayed. The portion of the plane containing spikes is divided by a rectangular grid into equal cells (in Spiker a grid of 128 by 128 is used) and the number of events in every cell is calculated. Each cell is displayed as a pixel, with brightness increasing with the number of events in the cell. To extend the dynamic range of displayed densities, a slower than linear dependency of brightness on the density is used. This prevents masking of low-density clusters by those with high-density. Clustering in the density representation is illustrated in Figure 1b.

In many multi-cell studies, neuronal interactions revealed by correlated firing patterns are of primary interest. This leads to an additional requirement on the "purity" of clustering. The thresholding procedure being used for spike detection can miss certain spikes in the clusters that lie in the low-amplitude zone. One can detect a bright-dark rectangular boundary in the upper-right corner of the quadrants in Figure 1, which is formed by this phenomenon. In other words, clusters crossing this boundary (representing the value of the threshold) can be missing a substantial number of spikes. Measures of reliability and synchronization based on such an incomplete cluster will be misleading,

even though the incomplete cluster may be "clean" in the sense that it contains spikes from one cell only. Furthermore, if the missing part of the cluster (smaller-amplitude spikes) represents a non-random sample of the cell's output, cross-correlations and other measures based on these spikes will also be tainted.

If a relatively small piece (10-20%) of the cluster is cut off, one can adjust the threshold to recover the missing points. However, it is generally impossible to recover clusters that are missing a larger number of spikes, since the low-amplitude part of the cluster is usually inseparable from the noise. Such partial clusters should be discarded. The ability to recover some clusters by adjusting threshold illustrates one advantage of keeping the full voltage traces, rather than extracting spikes online

CROSS-CHANNEL CORRELATION IN THE TETRODE RECORDINGS.

The clusters in Figure 1, which come from cat visual cortex, are elongated. This indicates that amplitudes on different channels tend to co-vary. Two different mechanisms can lead to this. If the observed covariance was caused by the intrinsic variation of the spike amplitude (e.g. due to bursting), the principal axis of the clusters should point to the origin of the coordinate system. Another possible source of the covariance is noise correlated between channels. In this case the principal axes would all have a common slope, i.e. be parallel to one another. Inspection of the clusters in Fig. 1 shows that the second source is probably dominant.

To test this hypothesis we have calculated cross-channel correlation in our recordings and found it to be surprisingly high. In the cat LGN, the cross-channel coefficient of correlation (averaged over pairs of wires) was around 0.94, while in the cat visual cortex it was about 0.78. These values were calculated as an average across five random chunks of data (each 10,000 samples long) taken from six different files of each type. To exclude possible influence of spikes on these calculations, all points exceeding the level of four standard deviations were removed from the dataset; however, this makes very little difference.

The common noise can be suppressed by the use of channels in a differential mode, in which the signal at one channel is subtracted from the signal at another one. We, however, use a slightly different approach. The Hadamard transformation of the measured amplitudes allows us to reveal the differential part of the signal and to suppress the common source. The Hadamard transformation is given by the equation:

$$\begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{pmatrix} = \begin{bmatrix} 1 & -1 & -1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix},$$

where a_1 - a_4 are the original spike amplitudes which are used to calculate a new vector of amplitudes shown on the left side of the equation. As can be seen from the matrix, the first three new components represent differences in the amplitudes, while the fourth is just the sum of all four amplitudes. The Hadamard transform is linear and can be viewed as a scaling and rotation in the four-dimensional space. Common noise is suppressed in coordinates h_1 through h_3 , but not in h_4 , the sum of all four amplitudes. The projections in the $h_1/h_2/h_3$ subspace produce less elongated clusters, which are in most cases better

separated than clusters in the raw amplitude coordinates. This is illustrated by Figure 2. Nevertheless, in certain cases raw amplitudes still provide better separation.

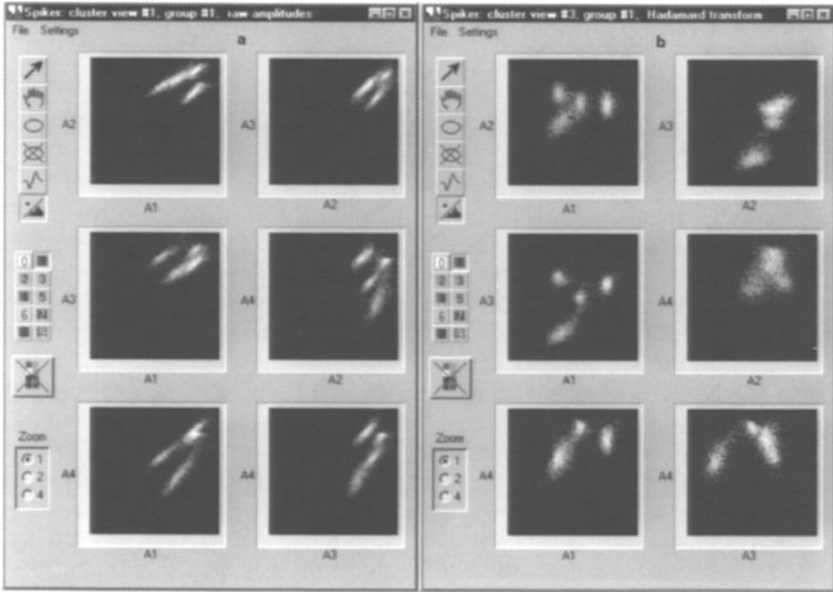


Figure 2. Comparison of clusters in the raw amplitude representation (a) and in the Hadamard representation (b). Note that in b clusters are less elongated and better separated, particularly in the three projections not involving h_4 .

Geometrically, the Hadamard transform is a local linear approximation (about the point $(1,1,1,1)$) to a transformation into polar coordinates. That is, axis h_4 is the radial $(1,1,1,1)$ coordinate, and the other three axes locally approximate the remaining 3 angular coordinates. It may seem tempting to ignore the h_4 coordinate and work in the remaining 3D space, on the assumption that cells from different spatial locations will have different angular coordinates (Zhang et al., 1997). However, it is not infrequently the case that the h_4 coordinate is critical to successful sorting. Geometrically, two clusters may have centers with angular coordinates sufficiently close that, given the finite size of clusters, they will be hopelessly entangled without the absolute amplitude information in h_4 . Near-threshold noise, which is distributed across all angular directions, also presents a problem. (Note, the Hadamard transformation handles these problems better than truly working in angular coordinates, because amplitude information is retained in the difference coordinates, but not in angular coordinates). In addition, it is always vital to keep the full amplitude information in order to check for clusters that have been cut off by the threshold, as discussed above; this checking must be done in the raw amplitude representation. Spiker allows simultaneous clustering in both representations.

The observed common source of noise is presumably due to the activity of distant neurons whose spikes are too small at the tetrode to be discriminated. One can probably significantly improve clustering if the common source suppression is performed before the

spike detection. This will reduce the number of missing spikes in the low-amplitude clusters.

CONCLUSIONS

Spiker is available in the public domain (<http://mccoy.ucsf.edu>). It allows sorting of data from multiple tetrodes and incorporates a number of features not found in existing commercial software. Most notably, it uses continuous voltage data, allowing user adjustment of threshold and other spike-defining parameters; and it allows simultaneous clustering in multiple projections, in particular projections onto axis pairs in both raw amplitude and Hadamard coordinates.

In addition to creating software, our modest contributions to existing clustering algorithms are to emphasize the importance of ensuring that clusters are not cut off by the threshold, and to point out the advantages of sorting in the Hadamard representation. We have shown that these advantages stem from the large component of common noise on the four wires in tetrode recording, at least in the neural structures we have examined (cat LGN and visual cortex). Future algorithm development may greatly benefit from taking into account this dominance of common noise.

In the future, we hope to improve methods of eliminating common noise. One obvious means is to replace the Hadamard matrix with one that uses the covariance between channels to optimize common noise suppression; hardware solutions might also be explored. We also plan to implement automatic spike-sorting algorithms in Spiker. We, like others (Sahani et al., 1997; Zhang et al., 1997) have been exploring the use of the Expectation-Maximization (EM) algorithm to automatically form Gaussian clusters, given a user specification of the number of clusters. More generally, we are pursuing approaches that make use of the full waveform information, analogous to Lewicki (1994) (see also Sahani et al., 1997) but (we hope) with a number of improvements as well as adaptation to the tetrode situation. As these algorithms develop to the point of reliability, we will implement them in Spiker.

References

- Lewicki, M.S. (1994). Bayesian modeling and classification of neural signals. *Neural Comput.* 6:1005-1030.
- McNaughton, B.L., J. O'Keefe, and C.A. Barnes, (1983). The stereotrode: A new technique for simultaneous isolation of several single units in the central nervous system from multiple unit records. *J. Neurosci. Methods*, 8:391-397.
- Sahani, M., J. S. Pezaris, and R. A. Andersen, (1997). Extracellular recording from multiple neighboring cells: A maximum-likelihood solution to the spike-separation problem. In Bower, J.M. (Ed.) *Computational Neuroscience: Trends in Research 1996*. Plenum Press. New York.
- Zhang, K., T.J. Sejnowski and B.L. McNaughton (1997). Automatic separation of spike parameter clusters from tetrode recordings. *Soc. Neuro. Abstr.* 23:504.