

Tracking neurons recorded from tetrodes across time

A.A. Emondi^{a,b,c,d}, S.P. Rebrik^{c,d,*}, A.V. Kurgansky^{c,d}, K.D. Miller^{c,d,e}

^a Institute for Sensory Research, Syracuse University, Syracuse, NY 13244-5290, USA

^b Space and Naval Warfare Systems Center, P.O. Box 190022, North Charleston, SC 29419, USA

^c Department of Physiology, University of California, San Francisco, CA 94143-0444, USA

^d Keck Center for Integrative Neuroscience, University of California, San Francisco, CA 94143-0444, USA

^e Sloan Center for Theoretical Neurobiology, University of California, San Francisco, CA 94143-0444, USA

Received 23 July 2003; received in revised form 9 December 2003; accepted 12 December 2003

Abstract

Tetrodes allow isolation of multiple neurons at a single recording site by clustering spikes. Due to electrode drift and perhaps due to time-varying neuronal properties, positions and shapes of clusters change in time. As data is typically collected in sequential files, to track neurons across files one has to decide which clusters from different files belong to the same neuron. We report on a semi-automated neuron tracking procedure that uses computed similarities between the mean spike waveforms of the clusters. The clusters with the most similar waveforms are assigned to the same neuron, provided their similarity exceeds a threshold. To set this threshold, we calculate two distributions: of within-file similarities, and of best matches in the across adjacent file similarities. The threshold is set to the value that optimally separates the two distributions. We compare different measures of similarity (metrics) by their ability to separate these distributions. We find that these metrics do not differ drastically in their performance, but that taking into account the cross-channel noise correlation significantly improves performance of all metrics. We also demonstrate the method on an independent dataset and show that neurons, as assigned by the procedure, have consistent physiological properties across files.

© 2004 Elsevier B.V. All rights reserved.

Keywords: Tracking neurons; Tetrode; Electrode drift; Extracellular recording

1. Introduction

A tetrode is a set of four closely located single electrodes made of very thin wire, each about 10–15 μm in diameter. The typical distance between the centers of the tips is about 20–30 μm . In comparison with traditional extracellular electrodes, tetrodes provide more information per spike and thus allow isolation of more neurons with higher reliability (Gray et al., 1995). To sort spikes from different neurons, spike features such as first peak amplitude, width, peak-to-peak amplitude, principal components, etc., are measured on each of the four channels. Consider a case in which only amplitudes of the first peak at each channel are used. Then every tetrode spike can be described by four values and thus corresponds to a point in a four-dimensional space. One of the natural ways to visualize this space is to use its projections on different planes. A standard choice

of the projections is six planes given by all possible pairs of the axes: (A_1, A_2) , (A_1, A_3) , (A_1, A_4) , (A_2, A_3) , (A_2, A_4) , (A_3, A_4) . Fig. 1 shows the typical scatter plot of spike amplitudes, with points colored according to their assigned clusters.

As recordings at a single site can be quite long (several hours and more), drift in the amplitudes and shapes of spikes recorded at a site is often observed (Lewicki, 1998; Snider and Bonds, 1998). This drift makes spike clustering much harder since it “smears” the clusters and effectively increases their size, which can lead to significant cluster overlap and thus make the clusters inseparable. This effect is much less pronounced at shorter time-scales of 10–20 min or less. So even in the case when the clusters obtained from a long recording are completely inseparable, smaller chunks of the same recording might produce quite distinct clusters. For a number of practical reasons, recordings at a site in acute cortical recording are typically done in chunks of less than 20 min duration (which correspond to computer data files), and the clusters can be considered stationary within each file. Fig. 2 illustrates the effect of electrode drift at a single

* Corresponding author.

E-mail address: rebrik@phy.ucsf.edu (S.P. Rebrik).

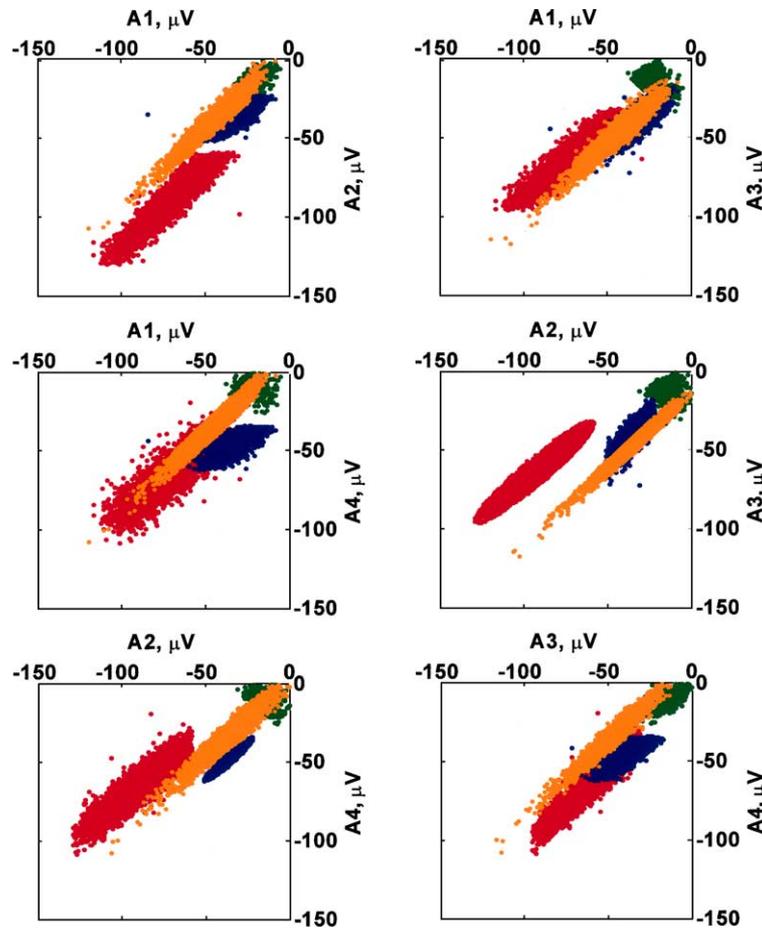


Fig. 1. An example of tetrode spike clustering. For each tetrode spike, four amplitudes of the first peak (A_1 – A_4 , one per channel) were measured and each spike was represented by a point in the four-dimensional space corresponding to the measured amplitudes. The projections of points in this space onto the six planes defined by the coordinate axes are shown. The points are colored according to the assigned cluster. Spikes that are not clusterable (mostly low-amplitude) are not shown.

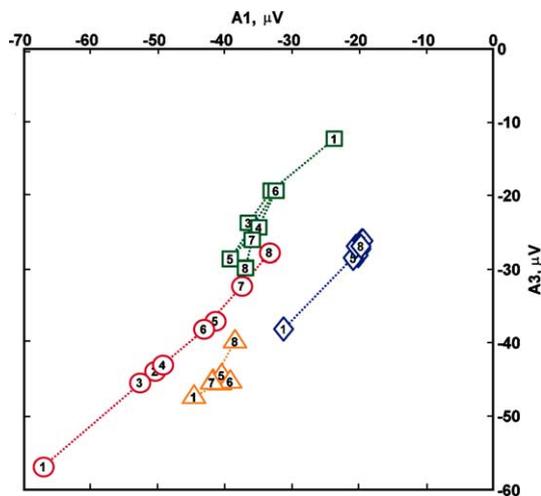


Fig. 2. An example of cluster drift. The centers of the clusters are shown for projection A_1 – A_3 (see Fig. 1) as they were changing in time. Different clusters are labeled by different shapes and the numbers inside the shapes correspond to the consecutive file number at the recording site (total eight files, of 2–10 min duration). Total time at the site was 1 h and 40 min.

site over an interval of 1 h and 40 min. During this time eight files were recorded, with the duration of each file varying between 2 and 10 min. The positions of cluster centers obtained from each file are labeled with the file number, and the corresponding clusters across files (as assigned by the methods to be described in this paper) are coded with symbols and colors.

As a result of “per file” clustering one gets information about cell activity in each file, but the information on the correspondence of clusters across files is missing. A researcher can try to match the clusters across files visually, e.g. by comparing cluster shapes and positions across files that are adjacent in time. This method is highly subjective, unreliable, and time-consuming. Another problem is that the number of clusters may change from file to file. A cluster can disappear due to electrode drift or because it corresponds to a neuron that did not respond to the stimuli presented in a given file. The same effects can also lead to appearance of new clusters. Strictly speaking, recording in the presence of electrode drift cannot be called “single site recording”, but what really matters is the ability to record from the same set of neurons for a long enough period. The procedure sug-

gested in this paper makes it possible to track neurons across files recorded at the same site in a systematic and objective way.

2. Methods

2.1. Surgery and anesthesia

We recorded from area 17 of 9 anesthetized adult cats. In the beginning of the experiment a cat was anesthetized using isoflurane carried by oxygen (typical concentration 1.5–5% for 1 l/min flow). After reaching the proper level of anesthesia and for the remainder of the experiment the animal was anesthetized with pentobarbital (to effect, 25 mg/kg surgical dose, IV), and isoflurane was discontinued. The animal was mounted in a stereotax, a craniotomy was made above the visual cortex, and the dura in that area was removed. In some cases the brain surface was covered with agar to reduce tissue pulsation. After the surgery the animal was paralyzed with a continuous infusion of gallamine (10 mg/(kg h) IV) or pancuronium bromide (0.088 mg/(kg h) IV) diluted in Ringer solution with 2.5% dextrose (5–10 ml/(kg h)), and was actively ventilated with a 1:1 mixture of oxygen and nitrous oxide. Every 6–12 h atropine sulfate (0.04 mg/kg²) was given to reduce secretions, and dexamethasone (2 mg/kg² or IV) was given to control edema. To prevent onset of infection, cefizox (10 mg/kg) was given at 12 h intervals. Heart rate, EKG, EEG, respiratory rate, rectal body temperature, O₂ saturation, expiratory CO₂, and lung pressure were monitored and logged on a per minute basis using an automated system of our design.

Contact lenses were chosen to provide the best focus at a distance of 35–40 cm by reflecting the optic disk onto a white background using a fiber optic light source.

2.2. Tetrode fabrication

In this study we used both home-made and commercially available (Thomas Recording, Germany) tetrodes. In-house tetrodes were made with a technique similar to Wilson and McNaughton (1993) and Gray et al. (1995). We used either NiCr (15 μm wire diameter) or tungsten (7.5–12.5 μm wire). The four wires were twisted together, then quickly heated by a heat gun to slightly melt the wire insulation and so attach the wires to one another. The four wire ends at one side were then manually stripped of their insulation. The exposed wire ends were then spot welded to an electronic connector or gold-plated and soldered to the connector. The twisted strand of four wires was threaded through a small tubing (approximately 28 gauge) and the connector was attached to the tubing with an epoxy glue. The twisted part of the wires was cut with fine scissors at the other end of the tubing leaving 5–7 mm of the wire protruding from the tubing. The tetrode tips were gold-plated to bring the tip impedance in the range of 0.7–1.2 MOhm at 1 kHz.

For the physiological test of the method (described below) we used modified tetrodes developed in our lab. They had five wires: a central one which served as a spacer and four thinner wires twisted around it. This geometry provided a better tetrode “stereo effect” by increasing the measuring base. The central wire was “free-floating” while the peripheral wires were connected to the amplifier. All five wires were made of insulated tungsten with core diameters of the central and peripheral wires of 25 and 10 μm correspondingly. The impedance of the peripheral wires was in the range of 0.7–1.4 MOhm at 1 kHz.

2.3. Data acquisition

The tetrode was connected to a custom-made head stage amplifier (based on the INA110 chip by Burr-Brown) providing a gain of 10, dc coupled. The signal was further amplified and filtered by a CyberAmp 380 (Axon Instruments) with the following settings: gain of 1000, ac coupling at 300 Hz, a fourth-order Bessel type low-pass filter at 3000 Hz, and a notch filter at 60 Hz. The voltages were sampled at a rate of 15 or 20 kHz with 12 bit resolution and the data were continuously streamed to disk. Recording to a file was initiated approximately 1 s before the stimulus onset and was terminated approximately 1 s after the stimulus presentation was completed.

2.4. Visual stimulation

Once a stable active site was located, various visual stimulus sets were shown for periods of 2–20 min each; the response to each such set constituted one file. Stimulus sets varied from experiment to experiment—the most “typical” sequence being as follows. First, moving sinusoidal gratings of different orientations (spatial frequency: 0.5 cycles/degree, temporal frequency: 3 cycles/s, contrast: 80–100%), were shown for 4 s each, separated by a 1 s blank period. The set of orientations covered 360° with a 5° step and was presented in a pseudo-random order, with each orientation repeated twice. The response to this “orientation tuning batch” constituted one file and was used to find the orientation tuning curves (OTCs) of the neurons. Next a checkerboard noise stimulus was presented. Typical size of the stimulus was 90 × 90 blocks, and each block could take at random one of three levels of luminance. As the noise stimulus did not cover the whole screen, its position and the block size were chosen to maximize the multiunit response. The response to this noise presentation constituted a second file. During the noise presentation, the orientation tuning dataset was analyzed to determine the preferred orientations of the neurons at the site. Finally, a set of gratings with spatial frequencies of 0.1, 0.14, 0.2, 0.28, 0.4, 0.57, 0.8, 1.13, 1.6, 2.26, 3.2, and 4.0 cycles/degree, and temporal frequency fixed at 3 cycles/s were presented at the found preferred orientations. There were 4–8 repetitions of each set randomly intermixed. The response to this “spatial

frequency batch” constituted a third file. Thereafter other noise or grating stimulus sets were presented.

2.5. Datasets

In this study we used two non-overlapping datasets: the first one was used to compare different waveform metrics, while the second dataset was used to verify the method on the physiological properties of recorded neurons.

The first set included 33 sites recorded from four cats, with 5–39 files recorded at each site, a total of 350 files. Typical duration of a file was from 2 to 20 min, and total time at a site varied between 40 min and 13 h. Out of 33 sites, four were recorded using commercial tetrodes, four using tungsten gold-plated tetrodes, and in the remaining 25 sites NiCr gold-plated tetrodes were used. A 15 kHz sampling rate was used in nine sites, and a 20 kHz rate was used in the rest. The number of distinguishable clusters per file varied between 3 and 9 with a median of five clusters. In this set we did not use sites that had less than five files recorded or that had files with less than three clusters.

The second set included 10 sites recorded from five cats (different from the first set). At each site from 4 to 8 files were recorded, a total of 66 files. File durations varied from 5 to 19 min, and total time at a site varied between 1 h and 2 h 10 min. The number of distinguishable clusters per file was between 2 and 9 with a median of 7 clusters. Sampling rate was 20 kHz for all files and only tungsten five wire tetrodes were used. In recording of this dataset we used bilateral pneumothorax and lumbar suspension in an attempt to stabilize the electrode. Recordings at all sites had orientation tuning batches in the first and in the last files at the site, but the batches were somewhat different than previously described: each drifting grating orientation was presented for 1 s and repeated four times.

Spikes were clustered with a manual clustering program “spiker” developed in our lab and available for downloading from <http://millerlab.ucsf.edu>. To cluster we used the amplitudes of the first peak (four per event), which were extracted after a 1:10 Fourier interpolation of the spike. After clustering, all individual spike waveforms were 1:10 Fourier-interpolated, aligned at the first peak, 10:1 decimated, and then averaged per assigned cluster. Averaged spike waveforms containing 0.9 ms before the first peak and 1.2 ms after the peak were saved to disk and were used to link the neurons across files; the length of each waveform was 33 or 43 points, depending on the sampling rate. An example of average waveforms of four neurons recorded in the same-file is shown in Fig. 3.

2.6. Computing preferred orientation and width of the orientation tuning curve

We used orientation tuning of V1 neurons to test the method (details of the test are explained later). For every neuron from the second dataset, an orientation tuning curve

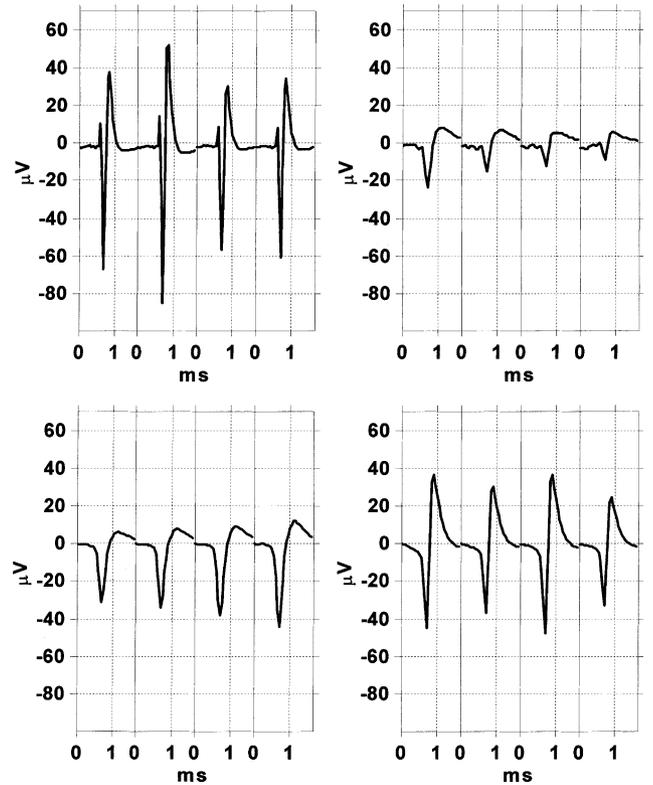


Fig. 3. An example of average spike waveforms. Each subplot corresponds to a different neuron. Within each subplot the four waveforms are shown—one per tetrode channel.

was computed. Those neurons that did not show orientation selectivity and/or did not demonstrate a reproducible response to repetitive presentation of the same visual stimulus were excluded.

To check if the response was orientation selective, we first computed the neuron’s preferred orientation as

$$\varphi_{\text{pref}} = \frac{1}{2} \arctan \left\{ \frac{\sum_{k=1}^{72} r_k \cdot \sin(2\varphi_k)}{\sum_{k=1}^{72} r_k \cdot \cos(2\varphi_k)} \right\}. \quad (1)$$

Here, $r_k = r(\varphi_k)$ represents the average spike rate r_k elicited by the grating of orientation φ_k .

Then we determined the component of the response along the preferred orientation as

$$R^{ij} = \sum_{k=1}^{72} n_k^{ij} \cdot \cos[2(\varphi_k - \varphi_{\text{pref}})], \quad (2)$$

where n_k^{ij} indicate the number of spikes elicited by the j th cycle of the i th repetition of the grating of orientation φ_k .

A neuron was considered to be orientation selective if the set of $\{R^{ij}\}$ had mean value significantly different from zero. Significance was estimated using a t -test, with significance level 0.01.

Linear regression analysis was performed to assess OTC reproducibility. If the regression coefficient of the sum of the first two stimuli repetitions onto the sum of the third

and forth repetitions was significantly different from 0 ($P < 0.01$), then the OTC was considered as reproducible and accepted for further analysis.

Given that the orientation tuning curve, $r_k = r(\varphi_k)$ does not take negative values, r_k can be viewed as a distribution, and the width of the OTC can be computed as the standard deviation of this distribution:

$$w = \sqrt{\frac{\sum_{k=1}^{72} (\varphi_k - \varphi_{\text{pref}})^2 \cdot r_k}{\sum_{k=1}^{72} r_k}}, \quad (3)$$

where $\varphi_k - \varphi_{\text{pref}}$ is computed as the shortest distance around a circle of 180° , e.g. 5° and 175° have a difference of 10° .

3. Measures of similarity between spike waveforms

To trace neurons across files a measure of spike similarity (a metric) is required along with some criteria that allow one to decide whether certain clusters from different files have been recorded from the same neuron at the site. It is natural to assume that if the spikes of two clusters recorded in different files appear to be similar, they were recorded from the same neuron, and if the spikes are not similar, they were generated by two different neurons. To measure spike similarity we used average waveforms described above. Four average spike waveforms (each 33 or 43 points long) can be represented as a matrix $[x_{ct}]$, where index c represents four channels and t is the time index. As we are considering metrics that take into account only per-element differences of the two waveform matrices, the matrices can be transformed into 132- or 172-dimensional vectors \vec{x} by concatenating the rows of the original matrix. These vectors correspond to points in a multidimensional space. An obvious metric in this space is the Euclidean distance, however certain properties of the tetrode recordings suggest that other metrics can be beneficial for linking the neurons.

Waveform vectors can be represented as:

$$\vec{x} = \|\vec{x}\| \cdot \vec{u}, \quad (4)$$

where $\|\vec{x}\|$ is the length of the vector, while $\vec{u} = \vec{x}/\|\vec{x}\|$ can be interpreted as the “shape” of the spike vector. Obviously the shape is very important in differentiating between spikes coming from different neurons, but it is not clear how strongly one should weight the length in trying to distinguish mean waveforms of different neurons. The basic idea of the tetrode is that the ratios of spike amplitudes on the four electrodes should stay roughly the same for spikes from a given neuron, even as absolute amplitudes may change, e.g. the amplitude can decrease over a burst. For the metric to ignore such variations, it should be scale-invariant, i.e. insensitive to the differences due to scaling of the spike waveforms.

Two metrics that are at the opposite extremes in the sensitivity to scaling are the correlation coefficient:¹

$$C = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \|\vec{y}\|}, \quad (5)$$

and the Euclidean distance:

$$\text{ED} = \|\vec{x} - \vec{y}\|, \quad (6)$$

where \vec{x} and \vec{y} are the waveform vectors of two neurons. If \vec{x} or/and \vec{y} are scaled, the correlation coefficient does not change, while the Euclidean distance is quite sensitive to scaling. There is a relationship between these two metrics which can be written as:

$$\begin{aligned} -(\text{ED})^2 &= -\|\vec{x} - \vec{y}\|^2 \\ &= 2\|\vec{x}\| \|\vec{y}\| \left(C - \frac{1}{2} \left[\frac{\|\vec{x}\|}{\|\vec{y}\|} + \frac{\|\vec{y}\|}{\|\vec{x}\|} \right] \right), \end{aligned} \quad (7)$$

where C is the correlation coefficient introduced in (5). Note that the term $[(\|\vec{x}\|/\|\vec{y}\|) + (\|\vec{y}\|/\|\vec{x}\|)]$ is insensitive to equal scaling of the two spike vectors, but is sensitive to the differential scaling of the two. The influence of the differential scaling can be titrated with a set of metrics parameterized by γ

$$D\gamma = C - \frac{\gamma}{2} \left[\frac{\|\vec{x}\|}{\|\vec{y}\|} + \frac{\|\vec{y}\|}{\|\vec{x}\|} \right]. \quad (8)$$

For $\gamma = 0$, $D_0 = C$ which is not sensitive to scaling, while for $\gamma = 1$, D_1 is sensitive to differential scaling only; $D_{1/2}$ will be somewhere in between regarding the sensitivity to differential scaling. In this study we consider similarity metrics D_0 , $D_{1/2}$, D_1 and ED, which have different sensitivity to different types of vector scaling. Note that we are using negative values of ED as a similarity measure.

4. Cross-channel whitening (CCW)

The noise in the tetrode recordings is strongly correlated across channels, and it is dominated by the firing of the surrounding, “background” neurons (Rebrik et al., 1997, 1999). The high degree of the noise correlation between the tetrode channels means that some directions in the four-dimensional channel space are “noisier” than the others. This is illustrated in Fig. 4A where the signal measurements made at two tetrode channels are plotted against each other (spikes make a negligible statistical contribution to this distribution). This plot can also be viewed as a projection of a four-dimensional scatter plot of the four-dimensional sample points onto the plane defined by the two coordinate axes. In this space the distances between points of spike waveforms should be corrected for the direction in which they were measured: distances measured along the diagonal (the noisiest direction)

¹ Strictly speaking, C is the cosine of the angle between the vectors x and y , but since the dc is removed from the voltage traces, the means of x and y are close to 0, so we refer to C as a correlation coefficient.

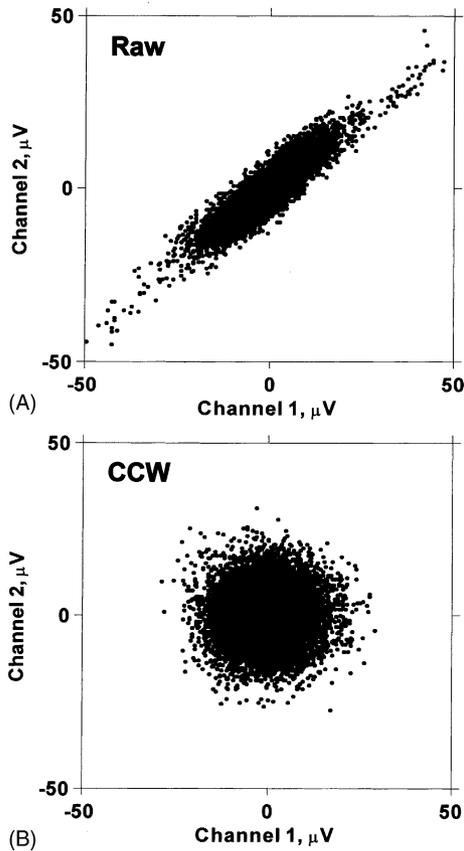


Fig. 4. Demonstration of the cross-channel whitening procedure. Here each sample of a continuous tetrode recording is plotted as a point with the coordinates given by the voltages at tetrode channels 1 and 2 (total of 10 000 points taken from 10 random chunks). Panel (A) shows the original (raw) data, panel (B) shows the same data after cross-channel whitening.

should be less significant than distances measured in the perpendicular direction. This correction can be achieved by the standard procedure of “cross-channel whitening” (Bishop, 1995). Firstly one finds the eigenvectors \vec{e}_a of the 4×4 cross-channel correlation matrix. These vectors give four orthogonal directions in the “channel space”, and along each direction, the noise is uncorrelated with the noise along other directions (note that we are not considering time-correlations here). The corresponding eigenvalues λ_a give the variance of the noise along each direction. Then the shift from the raw voltage basis $[x_{ct}]$, to the CCW basis, $[x'_{ct}]$ is made:

$$[x'_{ct}] = [e_{cc}][x_{ct}], \quad (9)$$

where the rows of the 4×4 transform matrix $[e_{cc}]$ are equal to: $\vec{e}_a / \sqrt{\lambda_a}$. In this basis the noise distribution is circular, and all directions are equivalent in their “noisiness” as shown in Fig. 4B.

Alternatively one can use Mahalanobis distance (Bishop, 1995) as a measure of the spike similarity, i.e. use the complete covariance matrices describing each cluster’s waveforms, and express the distance between two clusters as the Euclidean distance between their means divided by the geometric mean of their standard deviations. In this case scaling

of distance in the waveform space would take into account the shape and the size of both clusters. We did not take this approach since it is more computationally expensive, and because the major source of the spike variance—common background noise—is accounted for in the simpler approach described above.

5. Tracking neurons across files

Tracking of neurons across files is equivalent to arranging clusters from different files into a number of groups corresponding to neurons recorded at this site. A straightforward way to assign clusters to a neuron is to find the best match between clusters of the two neighboring files and to assign both of them to the same neuron. One could repeat this procedure by excluding the matched clusters and then finding the best match among remaining clusters, and so on. It is obvious however that members of some poorly matching pairs could actually belong to different neurons. Hence, we need a similarity threshold, below which clusters are never assigned to the same neuron. To get such a threshold, we assume that for most of the recording we are observing the same set of neurons, and we also assume that the changes of the spike shapes from file to file are relatively small. In this case it is natural to expect that the similarity between clusters recorded in the same-file, which correspond to different neurons, should be smaller than the similarity of the best matches across consecutive files, which are taken to correspond to the same neuron. We can calculate the distribution of the “same-file” similarities and the “best across adjacent files” similarities at a given site. Note that in the “best match” similarities each cluster is represented in one pair only, while all possible same-file pairs are included in the calculation of the same-file similarity distribution. An example of these distributions is shown in Fig. 5A. The distribution of the same-file similarities represents the similarities of waveforms of different neurons, while the distribution of the across files similarities for the most part represents the similarity of the same neuron waveforms recorded in different files. So a threshold that separates these two distributions can be used to separate pairs of clusters coming from different neurons from pairs of clusters coming from the same neuron. Since these distributions can overlap, finding an optimal position of the threshold is not trivial, and requires an error function that weights two different types of errors: a false-positive, i.e. assigning clusters from two different neurons to the same neuron, and a false-negative, i.e. failing to detect a pair of clusters originating from the same neuron. We weight both types of errors equally, so the error function is simply the number of the same-file similarities above the threshold plus the number of the across-file similarities below the threshold. Note that we have a slight bias to overestimate the number of false-negative errors since in the error function calculations we assume that each neuron is always represented in both files, while in reality we “drop” neurons

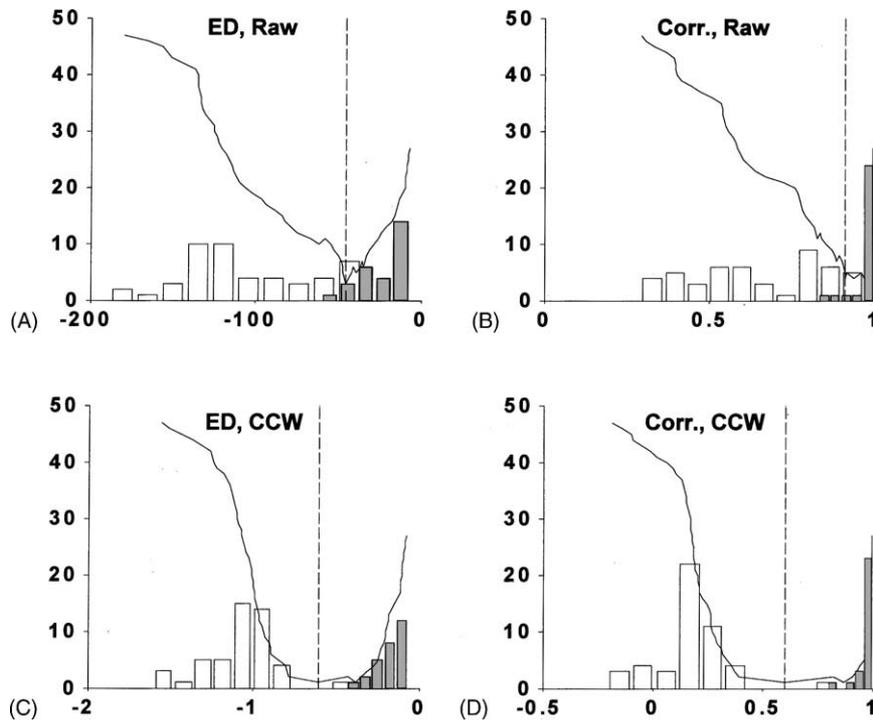


Fig. 5. Distributions of within-file similarities (white bars) and across-file similarities (grey bars) for two particular adjacent files at a single site. The solid line shows the error as a function of the threshold position. The dashed line shows the optimal threshold position. Results in (A) and (B) were calculated for the raw waveforms, and (C) and (D) for the whitened waveforms. In (A) and (C) the Euclidean distance was used, and in (B) and (D) the correlation coefficient. The difference between raw and whitened results shown here is typical.

once in a while. Fig. 5A shows an example of the number of misclassifications as a function of the threshold position.

Based on these considerations we suggest the following algorithm to assign clusters to neurons:

- Step 1. Allocate a number of neurons equal to the number of clusters recorded in the first file at a site.
- Step 2. From the list of all pairs of clusters taken between the first and the second files exclude the pairs that have similarity below the threshold.
- Step 3. Among the remaining pairs find the one with the highest similarity and assign the cluster from the second file to the same neuron as the other cluster in the pair. Exclude from the list of pairs all members that contain either cluster from the best matching pair.
- Repeat step 3 until the list of pairs is exhausted.
- Step 4. If there are clusters in the second file that have not been assigned to a neuron, add the corresponding number of neurons and assign the unmatched clusters from the second file to the new neurons.

Steps 2–4 are repeated for the remaining consecutive pairs of files recorded at the site, i.e. files 1–2, 2–3, 3–4, etc. (files are assumed to be ordered by the time of the recording). As a result of this procedure we get a number of neurons which are represented as lists of clusters assigned to a given neuron.

Obviously this algorithm can be extended to include pairs of more distant-in-time files, like 1–3, 2–4, and so on, but for the purposes of this paper we decided not to include distant matches since they are progressively more sensitive to the electrode drift and the assumption of recording from the same set of neurons in the two files becomes unreliable. In the case when distant-in-time links are absolutely necessary one has to design the error function accordingly.

6. Results

6.1. Comparison of different metrics

As illustrated in Fig. 5, different metrics produce different distributions of across- and within-file similarities of waveforms. The better the separation of the distributions, the more reliable is the neuron tracking. We used several approaches to estimate the impact of CCW and to rate the metrics by their ability to separate the distributions. The obvious choice is to use the value of the error function taken at the optimal threshold, or the error rate which we define as

$$\frac{\text{Err}_{\text{across}} + \text{Err}_{\text{within}}}{N_{\text{across}} + N_{\text{within}}},$$

where $\text{Err}_{\text{across}}$ is the number of across-file best matches with similarity below the threshold, $\text{Err}_{\text{within}}$ the number

Table 1
Performance of the similarity metrics, see explanation in the text

Similarity metric	Error rate			Drop rate (%)	MD	N Best		
	Average (%)	Across fraction (%)	Within fraction (%)			Error rate	Drop rate	MD
Raw								
$\gamma = 1$	7.88	81	19	6.36	2.11	3	4	2
$\gamma = 0.5$	7.49	78	22	5.88	2.22	8	3	1
$\gamma = 0$	8.42	64	36	5.39	2.28	4	6	0
ED	8.48	70	30	5.96	2.20	1	3	2
Whitened								
$\gamma = 1$	5.50	74	26	4.05	3.31	15	12	5
$\gamma = 0.5$	5.55	71	29	3.95	3.41	18	15	5
$\gamma = 0$	6.35	61	39	3.89	3.47	8	15	14
ED	6.08	65	35	3.95	2.83	16	15	4

of within-file similarities above the threshold, and N_{across} and N_{within} are the number of across-file and within-file pairs contributing to the distributions at the site. Note that not all the contributions to $\text{Err}_{\text{across}}$ are truly errors; some are mismatched pairs that the algorithm correctly excludes from being assigned to the same neuron. Thus, the error rate is an overestimate of the true number of errors. The “average error rate” column in Table 1 gives the average of this error rate across all sites. It is clear that the metrics based on the whitened data have a better performance than the metrics based on the raw data. On the other hand the distinction between different metrics within these two groups is small. A more detailed comparison of the error rate is shown in Fig. 6A where the averages across specific groups (raw/whitened) are plotted against each other for all recording sites. It is again obvious that CCW leads to a lower error rate. Interestingly, the fraction of errors that are across-file (see Table 1) is quite substantial: over 70%. Again, these errors could reflect the success of the algorithm in excluding false matches, as neurons are lost or gained across files due to electrode drift or changes in the stimulus, rather than a failure of the algorithm.

Another way of rating the metrics is to compare them by their ability to link neurons, so that the best metric would be the one with the smallest number of neurons that could not be linked across the files (dropped). We defined the “drop rate” as $\text{Err}_{\text{across}}/N_{\text{across}}$. Again the group of metrics based on CCW appears to be superior to the raw data group, while the variations within the groups are not substantial (see Table 1 and Fig. 6B). The drop rate estimates the probability of dropping a neuron while recording at a site. Given that, we can estimate a “neuron half-lifetime”, i.e. the number of files after which there is a 50% chance of dropping a neuron. Neuron half-lifetime can also be interpreted as the number of files after which we lose 50% of the original neurons. For the value of the drop rate of 3.89%, the estimated half-lifetime is 17.5 files, which for the purposes of our experiments was quite sufficient. At the same time this value varies significantly from site to site (see Fig. 6B) and one should use this value more as a metaphor rather than a strict law analogous to exponential decay law in the nuclear physics.

Finally, to assess the separation of the two distributions we calculated the Mahalanobis distance (MD) which in our case is defined as:

$$\text{MD} = \frac{|\mu_{\text{across}} - \mu_{\text{within}}|}{\sqrt{\sigma_{\text{across}} \cdot \sigma_{\text{within}}}}, \quad (10)$$

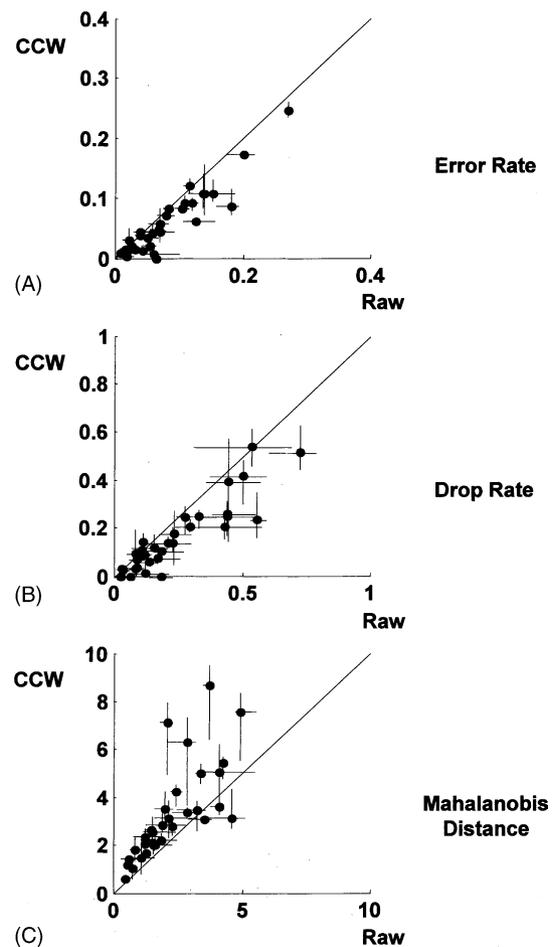


Fig. 6. Comparison of the results for the whitened and for the raw waveforms: (A) error rate, (B) drop rate, (C) Mahalanobis distance. Points represent different sites, bar lengths show the range of the values calculated for the four different distance metrics: D_0 , $D_{1/2}$, D_1 and ED. Vertical axis corresponds to the whitened data, horizontal axis corresponds to the raw data.

where μ_{across} and μ_{within} are the averages of the across and within similarities, and σ_{across} and σ_{within} are their standard deviations. This measure can be interpreted as the distance between the centers of the two distributions measured in units of the geometric mean of their standard deviations. As can be seen in Table 1 and Fig. 6C, according to this metric, the channel-whitened group better separates the distributions.

For each measure of performance we can calculate the number of sites at which a given metric produced the best (or tied for best) separation. These values are shown in the “N Best” part of Table 1. Note that the error rate and the drop rate measures of performance are discrete, so that at any given site there could be a number of metrics that provide equally good separation. For this reason, the columns in the “N Best” part of the table do not add up to 33 (the number of sites). As can be seen from Table 1, the distinction between the metrics based on the raw and CCW data is substantial, while the differences within the CCW group are not strong. Overall there is a certain bias to favor the metric with $\gamma = 0$ since it has the lowest drop rate and the biggest MD. This metric (the correlation coefficient) does not take into account the amplitudes of the spikes, but uses only their shapes. For practical reasons however it seems that the choice of a particular metric within the CCW group is less important than insuring the quality of the dataset itself by reducing the noise, stabilizing the electrode, etc.

6.2. Demonstration of the method

The fact that the clusters can be linked across files still does not guarantee that the clusters are assigned correctly to the underlying neurons. One can indirectly verify the tracking by comparing physiological properties derived from the clusters linked to the same neuron. One would expect that if the linkage is correct, the properties will be the same for the linked clusters while the properties for the non-linked clusters should show some scatter from file to file. Since most of the neurons in V1 show orientation tuning, we used preferred orientation (defined in Section 2) as the neuronal property against which we checked our method. We did the tests for the sites where orientation tuning batches were recorded in the first and in the last files at that site (the second dataset in Section 2). This way we could compare orientation tuning of neurons as measured from the first and the last file. Ideally they should be identical.

To link the clusters from the second dataset we used the correlation metric ($\gamma = 0$), with CCW. The procedure found 34 matched pairs out of 53 and 66 clusters in respectively the first and the last files recorded at the 10 sites. Twenty seven out of 34 pairs passed the test on orientation tuning curve selectivity and reproducibility (see Section 2). A histogram of pairwise orientation tuning differences in linked pairs is shown in Fig. 7A. As expected, this distribution tends to cluster around 0° . This is in contrast with the overall distribution of pairwise differences for all possible pairs (97

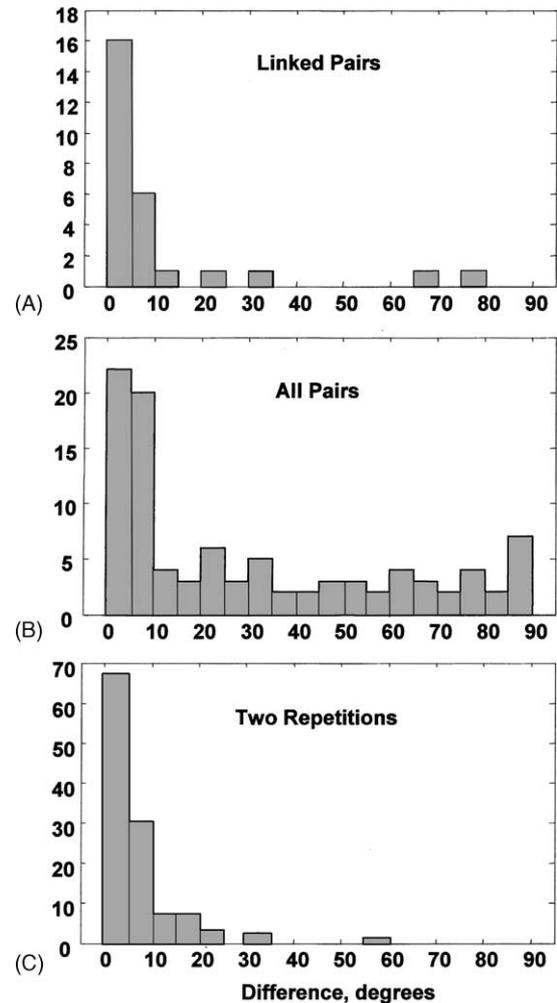


Fig. 7. Distributions of the pairwise differences in the preferred orientations for: (A) linked cluster pairs, (B) all cluster pairs, (C) two repetitions of the orientation batch at the same site.

total) shown in Fig. 7B. Note that only the pairs across the clusters that were linked in the first and in the last files at the same site are included.

The outliers in the distribution of pairwise differences (Fig. 7A) can come from two different sources: (a) “false-positive” link errors, i.e. clusters from two different neurons incorrectly linked together, (b) errors in the measurements of the preferred orientation. We tried to estimate the contribution of the second source by the following procedure. We divided our orientation batches in which we had four repetitions of the same orientation into two batches of two repetitions each. The distribution of differences in the preferred orientations as measured from the first and from the second parts of the orientation batch (shown in Fig. 7C) can be used as an estimate of the measurement error. It is clear that the distributions shown in Fig. 7A and C are very close, but still the relative number of outliers in 7A seems to be bigger: given the total of 27 links in 7A, one expects to see 2.7 cases of over 15° difference based on the distribution in 7C, but in fact there are four such cases in 7A.

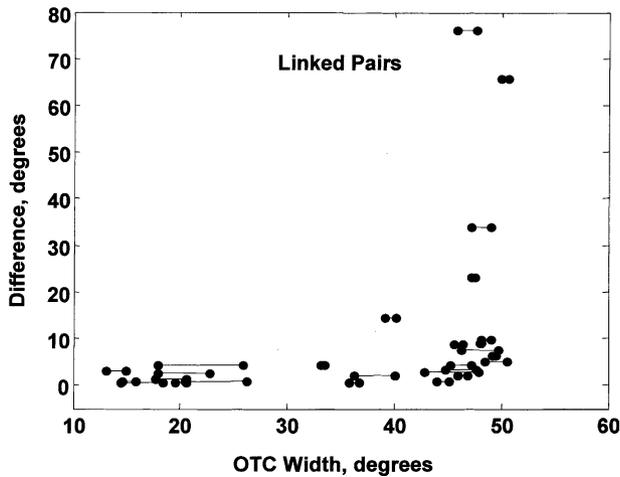


Fig. 8. Pairwise differences in the preferred orientations of the linked cluster pairs vs. the OTC width. Each horizontal line connects two points representing the OTC widths in the pair.

However, a scatter plot of the differences in preferred orientations of linked clusters versus the widths of their OTCs (Fig. 8) shows that all of the outliers are coming from clusters with wide OTCs whose preferred orientation is presumably poorly defined. Furthermore both clusters in each outlier pair have wide OTCs. Since the clusters were linked simply by waveform shape without regard for functional properties, this matching of OTC width is suggestive that the outliers indeed represent single neurons that have been correctly linked, and that their large differences in preferred orientations are simply due to their preferred orientations being poorly defined.

It is possible that the probability of incorrect linking increases for clusters with low-amplitude spikes which are harder to isolate. If this were the case one would expect to find a negative correlation between the differences in preferred orientations of linked clusters and the spike amplitudes. However a scatter plot of orientation difference versus spike amplitude (not shown) does not demonstrate such a tendency: the linear regression explains only 6% of the total variance ($F(1, 26) = 1.58, P > 0.22$). Also, the spike amplitudes of three out of the four outliers are above the median.

From the available data we cannot determine the actual number of errors in the linkage since incorrectly linked clusters can have similar preferred orientations just by chance and will not stand out in the distribution. On the other hand we cannot exclude a possibility that the outliers in the distribution are due to non-stationarity of the OTCs over a period of 1 h or more: small changes in the shape of wide OTCs can lead to significant changes in preferred orientation. In any case we find the results of this test quite promising given that we did not optimize the error function against the false-positive errors and that the stimuli were not optimized to drive all neurons in all files.

We also tried to link the clusters without tracing them from file to file, just by making a direct link between the

first and the last files at a site (using our linking algorithm as though those two files were adjacent). Surprisingly, the results of this “direct” linkage were not much different from the “chain” linkage: 27 pairs were linked and 23 out of 27 pairwise differences were within 15° . Examination of the linked pairs revealed that 23 pairs were identical to those obtained through the chain linkage (four pairs being different), and 21 out of these 23 pairwise differences were within 15° .

7. Discussion

Suppose we call a cluster “static” if it was formed in the absence of electrode drift. In this case its size and shape are defined by the background noise and by intrinsic spike variability. If the electrode drifts these “static” clusters move in the parameter space producing “dynamic” clusters that are bigger in size than the “static” clusters. The drift also leads to jumps in the cluster position from file to file as shown in Fig. 2. One can think of the resulting cluster picture as a painting made with spray paint: the paint spot is the “static” cluster, continuous trajectories correspond to the within-file drift, and the jumps from one trajectory to another correspond to the between-file interruptions. It is obvious that for a successful linkage of clusters across files it is sufficient to keep both across and within-file drifts much smaller than the “static” cluster size, which can also be viewed as the smallest possible distance between the same-file clusters. However, the requirement is not absolute: e.g. if clusters drift along sufficiently separated trajectories, it may be possible to link the clusters in spite of drift that is significantly larger than the static cluster size.

As we reformulate the problem as a dynamic one it becomes clear that to deal with cluster drift one can use continuous tracking of clusters (Snider and Bonds, 1998). We could not implement this approach with the available dataset: it had inter-file gaps in the recording. Continuous tracking however is not sufficient by itself for reliable tracking: the neurons should not have long interruptions in activity which will again result in cluster jumps. This requirement has some implications for experimental design: stimuli that are very specific in the neurons they excite should be intermixed with less specific stimuli to avoid long periods of silence of any given neuron. In the case of discrete tracking and matching limited to adjacent files (as presented in this paper), this means that all neurons should be sufficiently active in all files, which is sometimes hard to achieve. Our tests with either chained or direct linkage of clusters in files separated by many intermediate files illustrate this point: some neurons were lost in the chained linkage due to neuron inactivity in the intermediate files, while the direct linkage lost some pairs due to electrode drift. To provide better neuron tracking one can periodically repeat special runs that sufficiently excite all neurons at a site and use these runs as reference points for the linkage.

Another possible way to improve the method is to add a probabilistic measure of the cluster-linkage confidence, i.e.

to build a model that will give an estimate of the probability that a given pair of clusters is linked. To deal with confidence issues in the current implementation of the method one can use the error measures (described in this paper), so at least some obviously bad datasets can be avoided.

In the second dataset (see [Section 2](#)) we used electrode stabilization procedures, but we did not see any apparent difference from the first dataset in regard to either “static” or “dynamic” cluster size. If relatively long sessions at a site are required (well above 1 h), one should consider usage of chronically implanted electrodes which are known to have a much smaller rate of electrode drift. One can even imagine an electrode manipulator that will automatically adjust the position of the electrode to compensate for the cluster drift.

Although we used data acquired in an acute experiment, the method we suggest may also be useful for chronic experiments: e.g. for linking clusters between recording sessions separated by days or weeks. Essentially the same rules apply to chronic recordings, but the time-scale of the drift is very different.

In conclusion: we have demonstrated a simple, automated procedure for tracking neurons across time that seems to perform well, as assessed both by estimates of errors based on the distributions of cluster similarities and by the commonality of functional properties of clusters that the procedure links together as representing a single neuron. The procedure as formulated relies on time being discretized into a set of files: the procedure links clusters across files, while cluster drift within a file is ignored (it simply adds to the variability of the cluster). The most important technical point to emerge is that it is important to take account of cross-channel correlations by measuring waveform similarities in the

cross-channel-whitened (CCW) space, rather than in the space of raw voltages; this greatly improves performance.

Acknowledgements

The authors would like to thank Hiroki Sugihara for making the tetrodes and for help with the experiments, and Brian D. Wright for useful discussions in the early stages of the work. The authors are also grateful to Michael P. Stryker for comments and suggestions on improving this paper. Supported by RO1-EY13595.

References

- Bishop CM. Neural networks for pattern recognition. Oxford/New York: Clarendon Press/Oxford University Press, 1995.
- Gray C, Maldonado P, Wilson M, McNaughton B. Tetrodes markedly improve the reliability and yield of multiple single-unit isolation from multi-unit recordings in cat striate cortex. *J Neurosci Methods* 1995;63:43–54.
- Lewicki MS. A review of methods for spike sorting: the detection and classification of neural action potentials. *Network: Comput Neural Syst* 1998;9:R53–78.
- Rebrik S, Tzonev S, Miller K. Analysis of Tetrode Recordings in Cat Visual System, Computational Neuroscience Meeting Poster, 1997. <http://millerlab.ucsf.edu/Papers/CNS97/CNS97-paper.htm>.
- Rebrik S, Wright BD, Emondi AA, Miller KD. Cross-channel correlations in tetrode recordings: implications for spike-sorting. *Neurocomputing* 1999;26–27:1033–8.
- Snider RK, Bonds AB. Classification of non-stationary neural signals. *J Neurosci Methods* 1998;84:155–66.
- Wilson MA, McNaughton BL. Dynamics of the hippocampal ensemble code for space. *Science* 1993;261:1055–8.