Neural Signal Classification Using a Simplified Feature Set with Nonparametric Clustering

Zhi Yang*, Qi Zhao* and Wentai Liu

School of Engineering, University of California at Santa Cruz, 1156 High Street, Santa Cruz, CA 95064

Abstract

This paper presents a spike sorting method using a simplified feature set with a nonparametric clustering algorithm. The proposed feature extraction algorithm is efficient and has been implemented with a custom integrated circuit chip interfaced with the PC. The proposed clustering algorithm performs nonparametric clustering. It defines an energy function to characterize the compactness of the data and proves that the clustering procedure converges. Through iterations, the data points collapse into well formed clusters and the associated energy approaches zero. By claiming these isolated clusters, neural spikes are classified.

Key words: Spike sorting, spike feature extraction, clustering, action potential

1. Introduction

Neurons in the brain communicate by firing of the action potential, which induces transient voltage fluctuation in the surrounding tissue environment. The brief voltage fluctuation can be recorded with extracellular electrodes and the signal take the shape of spike. Very often one electrode is surrounded by multiple firing neurons, and their recorded activities become superimposed. To correctly understand the information in the biological neural network, it is critical to resolve spikes to individual neuronal sources [1–6], which is referred to as spike sorting.

Spike sorting is a high dimensional clustering problem. Directly classifying the recorded waveforms in high dimensional space is challenging partially because data points would be sparse, and clustering algorithms tend to be imprecise [7]. Feature extraction algorithms that represent spike waveforms using a few significant features are normally applied before clustering. A spike feature extraction algorithm is preferably to consume low computation and small storage space, and feasible for microchip implementation. A miniaturized microchip further equipped with inductive power and data links [8–10] can provide ultra-portability and real-time performance targeting neural prosthetic devices. Important examples of spike feature extraction algorithms include principal components analysis (PCA) [11–15], template matching [16–18], and wavelet [19,20], which involve significant computation. Tremendous efforts to improve the efficiency of traditional spike feature algorithms have been reported. The reported studies, however, either have an over simplified functionality, e.g. implementing spike detection [21–23] only, or have a hardware system consuming too much power and space.

After spike feature extraction, the extracted features become input to the clustering algorithm, which further attributes spike events to individual neuronal sources. The literature on clustering is vast [19, 24–27], where the most commonly used clustering algorithms can broadly fall into two categories, the hierarchical clustering and the partitional clustering. A hierarchical algorithm yields a structure representing the nested groupings of patterns and similarity levels at which groupings change. On the other hand, a partitional clustering algorithm obtains a single partition of the data instead of a clustering structure. As an example, the k-means family [11, 28, 29], which belongs to the partitional category, is by far a most widely used clustering method for spike sorting. The popularity is largely due to its low computation. The performance, however, is not always satisfactory due to the following reasons. First, the k-means is sensitive to initial seed selection and outliers. Second, it tends to produce hyperspherical clusters and it would lead to erroneous results with irregularly shaped clusters. Third, the k-means algorithm is parametric, which requires the number of clusters known as a priori.

This paper has two emphases. First, a feature extraction algorithm targeting integrated circuit implementation is

[1] *Equally contribute to the work

Preprint submitted to Elsevier 24 July 2009

presented. Second, an energy based evolving mean shift (EMS) algorithm with kernel scope obtained through N nearest neighbor search (NNS) is formulated.

This rest of the paper is organized as follows. Section 2 describes spike feature extraction. Section 3 presents EMS clustering algorithm. Section 4 shows experimental results and Section 5 concludes the work. Additional explanations of the spike detection algorithm and noise shaping theory are included in the Appendix .

2. Spike Feature Extraction

Figure 1 shows our feature extraction engine, which includes spike detector, noise shaping filter, feature extractor and controlling and storing units. It was initially designed to process one channel spike data [30] and later expanded to simultaneously process multiple channel data [31]. Efforts are also made to integrate the feature extraction engine into neural recording and stimulating chips [32–34].

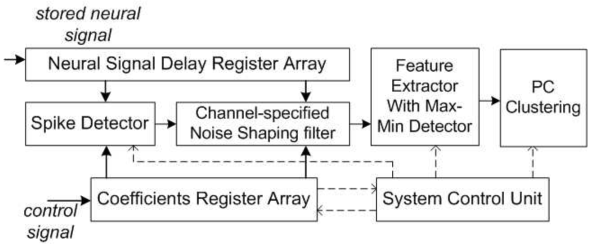

stored neural signal
Neural Signal Delay Register Array
Spike Detector
Channel-specified Noise Shaping filter
Feature Extractor With Max-Min Detector
PC Clustering
control signal
Coefficients Register Array
System Control Unit

Fig. 1. Block diagrams of the digital hardware of the proposed spike detection and sorting algorithms. Except the PC clustering block, the rest have been implemented on chip.

2.1. *Spike Detection*

Spike detector is a critical component for spike sorting. In our work, we choose nonlinear energy operator (NEO) [23, 35] to implement due to its efficiency of separating spikes from background activities, which are reported to exhibit a low frequency fashion through power spectrum measurement of data recorded from in-vivo experiments [36]. NEO was originally invented by Teager [37] and is used for the amplitude frequency demodulation and speech analysis. With a discrete time signal, NEO is

$$\psi[x(n)] = x^2(n) - x(n+1)x(n-1). \qquad (1)$$

where $\psi[\cdot]$ and $x(n)$ represent the NEO score and input signal, respectively. In Appendix 1, a simplified formulation is included to show the usefulness of NEO as a spike detection method.

Receiver Operating Characteristic (ROC) curves are used to quantify the performance of detection algorithms, and comparative results are plotted in Figure 2. The "Probability of Correct Detection" used in Figure 2 is defined as the ratio of the number of correctly detected spikes over the total number of neural spikes. The "Probability of False Detection" is defined as the ratio of the number of detected noise events over the total number of detected spikes. With NEO based spike detection, the detection threshold is set to be three times of the root-mean-square (RMS) score, which corresponds to 1.4% error detection and detects 99.5% spikes as a worst case in our test [38] on spike data from waveclus [19].

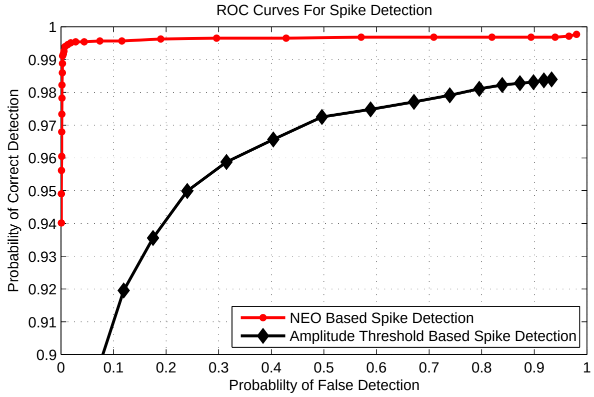


Fig. 2. ROC curves for spike detection algorithms. The red dotted curve is the result from NEO based spike detection. The black diamond curve is the result from amplitude based spike detection

2.2. *A Simplified Feature Set*

According to [36], a derivative based frequency-shaping filter significantly attenuates the low frequency noise (derivations related to noise shaping are included in Appendix 2) and could help differentiate similar spikes from different neurons. As a complementary approach to PCA, spike feature extraction algorithm based on informative sample set was first reported in [36] to identify uncorrelated local features. This concept requires only a subset of samples containing the necessary information to cluster the data. Intuitively, a sample is considered to be informative if the superimposed spikes can be classified into multiple clusters by evaluating the sample alone. Combining derivative operation and sample selection, improved sorting performance are observed. As a preliminary implementation to our feature extraction algorithm using informative samples, the height of the original spike waveforms and maximum and minimum values of its first derivatives are used as the features to classify spikes. The choice of this simplified sample set for implementation is based on three reasons. First, it requires small computation and little memory [31]. Second, samples during the fast transition period frequently exhibit high information. Third, obtaining these three features requires no training.

The digital filter in Figure 1 serves a two-fold purpose. First, it sets the low pass and high pass corner frequencies f_{c1} and f_{c2}. Second, the filter outputs the derivative of the spike waveforms. To handle a variety of noise profiles and spike widths, the filter coefficients are programmable

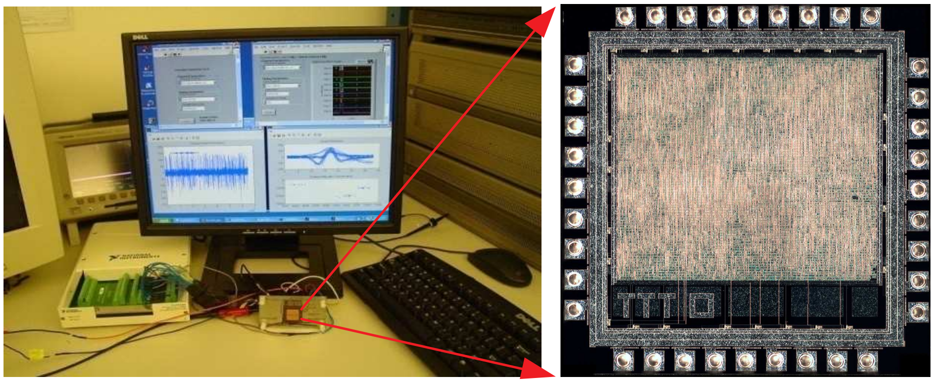


Fig. 3. Spike feature extraction IC and its test setup.

through the coefficient register array, and a matched filter profile to a specific biologic neural network can be precisely tuned. To achieve the minimal phase distortion, Bessel filter is used. The out-of-band rejection is achieved at more than 60 dB, which is generally enough to reject low frequency noise.

A NEO based spike detector, noise shaping filter, feature extractor, the corresponding storing device and control units described in Figure 1 are implemented with a custom digital IC with .35 μm CMOS process, which consumes 93 μW at 40 KHz clock frequency and occupies $1.62 \times 1.62 mm^2$. The test setup and chip photo is shown in Figure 3 [30, 31].

3. Evolving Mean Shift Spike Clustering

The mean shift algorithm is a mathematical tool proposed in the 1970s [39]. It offers an iterative scheme to identify the peaks of the kernel density estimate of the target data set. Surprisingly, the mean shift algorithm did not reach people's attention until several publications [40, 41] reestablished its theoretical foundation. Recently, the algorithm has been successfully applied in the areas related to visual tracking, image segmentation and clustering. For clustering, the mean shift based algorithm is nonparametric, which does not require prior knowledge of the number of clusters, and does not constrain the shape of the clusters.

The reported clustering algorithm in this work is related to mean shift techniques. Compared with the current mean shift algorithm and its variants [40–42], the used one has improved efficiency. To avoid confusion, the used clustering algorithm is named as evolving mean shift (EMS). The main novelty of our algorithm and the advantages are described as follows. First, EMS clustering algorithm inherits the advantages from the mean shift algorithm, e.g. it is nonparametric and robust to various cluster geometry and density variation. Compared with other mean shift based algorithms, EMS is faster, since it adopts a highly selective and adaptive iterative scheme. Second, EMS is designed to be insensitive to noisy events. The recorded spike data are usually contaminated by noise, sampling distortion and events overlapping. It is also possible that some detected spike waveforms are actually attributed to elements of noise. Compared with k-means, mean shift or other clustering algorithms where noisy spike events tend to corrupt the classification, EMS automatically handles those noisy events at an early stage and they are less likely to mislead the classification of other events. Third, EMS takes less number of iterations to converge, compared with traditional mean shift algorithms.

3.1. *Principles of Mean Shift Clustering*

The mean shift algorithm deals with the density estimate using radially symmetric kernel satisfying

$$K(x) = k(|x|^2), \tag{2}$$

where $K(x)$ integrates to one for normalization. Attaching data points to peaks (modes) of the kernel density estimate is an intuitive method to claim individual clusters. The mean shift procedure elegantly locates these modes without estimating the density, therefore, has significant speed improvement compared with directly searching.

Consider a general density estimate of data set $\{a_i\}$ with kernel $K(x)$ and unform scope h

$$p(x) = \frac{1}{nh^d} \sum_{i=1}^{N} K(\frac{x - a_i}{h}), \tag{3}$$

where d is the dimension of the feature space, h is the scope with positive values and N is the number of data points. $p(x)$, by definition, is the estimated density distribution of data set $\{a_i\}$ at location x. The focus of mean shift is to find the modes of $p(x)$ and therefore, claim individual clusters.

At the modes, the gradient of density estimate is zero

$$\nabla p(x) = 0. \tag{4}$$

Combining the definition of the density estimator $p(x)$ and the gradient yields

$$\nabla p(x) = \tag{5}$$

$$\frac{1}{Nh^{d+2}}[\sum_{i=1}^{N} k'(|\frac{a_i - x}{h}|^2)][\frac{\sum_{i=1}^{N} a_i k'(|\frac{a_i - x}{h}|^2)}{\sum_{i=1}^{N} k'(|\frac{a_i - x}{h}|^2)} - x].$$

The last term in Eq. 5 is recognized as the mean shift vector $M(x)$. At the modes, the corresponding mean shift vector has zero magnitude, therefore, zero gradient. For any input data point x_i, the iterative scheme $x^{i+1} = f(x^i)$ with $f(x^i) = x^i + M(x^i)$ would eventually converge to the modes. The mean shift procedure is an iterative scheme, and any data point at location x_i is assigned to the mode it converges to.

If the iteration $x^{i+1} = f(x^i)$ applies to a replica of $\{a_i\}$ and the original data set $\{a_i\}$ is always kept intact, it is the typically used mean shift. If the iteration directly applies to $\{a_i\}$ rather than its replica, the process is referred to as blurring mean shift [40, 42].

3.2. *Evolving Mean Shift Clustering*

In this section, we describe an energy based evolving mean shift clustering algorithm for spike sorting.

3.2.1. *Energy Definition*

Here we describe an objective function for EMS, whose score is referred to as "energy" in this paper. The EMS iterative scheme is essentially an energy reduction procedure, and the energy is minimized to zero when the data points are fully converged. Specifically, we denote the set of data points to be clustered by X, each data point by x_i and the set of n data points that are closest to x_i by $N(x_i)$. The energy of the data points X is defined as

$$E(X) = \sum\nolimits_{x_i \in X} E(x_i) \tag{6}$$

where

$$E(x_i) = \sum\nolimits_{x_j \in N(x_i)} f(h_{x_i})(K(0) - K(\frac{x_i - x_j}{h_{x_i}})) \tag{7}$$

where $K(x)$ is an arbitrary isotropic kernel with a convex profile $k(x)$, i.e., it satisfies $K(x) = k(|x|^2)$ and $k(x_1) - k(x_2) \geq k'(x_2)(x_1 - x_2)$. Without loss of generality, we set $k(0) = 1$. h_{x_i} in Eq 7 is the kernel bandwidth and $f(h_{x_i})$ is a shrinking factor that is designed to be a monotonically increasing function of bandwidth h_{x_i}. For popular kernels, e.g. Gaussian kernel, Epanechnikov kernel, $h_{x_i} \sim O(h_{x_i}^{\kappa})$ with $\kappa \geq 2$ satisfying the requirement. In this paper, we use $\kappa = 2$. For the case of $\kappa > 2$, derivations can be similarly applied.

3.2.2. *Formulating the Energy based Clustering Algorithm*

The problem of clustering the data points can be cast as an energy minimization problem. EMS is an iterative scheme, which reduces the total energy $E(X)$ by moving each data point. The goal of EMS is to create a well clustered status where the data points stop moving and converge.

In EMS, the scope h_{x_i} with each data point x_i is adaptively calculated according to the surrounding environment. This manipulation serves the purpose of resolving the potential conflict from the scope and the minimal distance between two modes. By the linearity of Eq. 6, the gradient of the total energy $E(X)$ is

$$\bigtriangledown E(X) = \sum\nolimits_{i \neq j} (\frac{\partial E_{i,j}}{\partial x_i} + \frac{\partial E_{i,j}}{\partial h_{x_i}} \frac{\partial h_{x_i}}{\partial x_i} + \frac{\partial E_{i,j}}{\partial h_j} \frac{h_j}{\partial x_i}). \tag{8}$$

At each iteration, EMS intends to move a data point x_i with large energy gradient. Selecting a point with the largest energy reduction for moving has several important benefits. First, it avoids to operate data that lead to small energy reduction (e.g. data points in plateau regions); therefore, requires less iterations compared with the mean shift or blurring mean shift algorithm. Second, it efficiently pushes loosely distributed points towards a localized peak, which prevents them from being absorbed into nearby clusters with larger densities.

An example of feature space evolution is presented in Figure 4. In Figure 4 (a), the original feature space is displayed, which includes 10000 data points. Figure 4 (b) - (h) display an evolved feature space after 5000, 10000, ... 35000 iterations that correspond to 0.5, 1, ..., 3.5 iterations per point. Through iterations, data points move according to the EMS vector defined in Section 3.2.4 and eventually converge to theirs corresponding cluster modes. The partition of the feature shown in Figure 4 (a) can be performed by simply claim the two isolated cluster modes.

3.2.3. *Selection of Kernel and Kernel Scope*

Two kernels are commonly used in the mean shift algorithm. One is the Epanechnikov kernel defined as

$$K_E(x) = \begin{cases} c_e(1 - |\frac{x}{h}|^2) & \text{if } |x| \leq h \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

where c_e is merely a normalization constant.

The Epanechnikov kernel based energy function has partial derivative to scope as

$$\frac{\partial E_{i,j}}{\partial h_{x_i}} = \begin{cases} 2c_e h_{x_i} \quad \text{if} & |x_i - x_j| = h_{x_i} \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

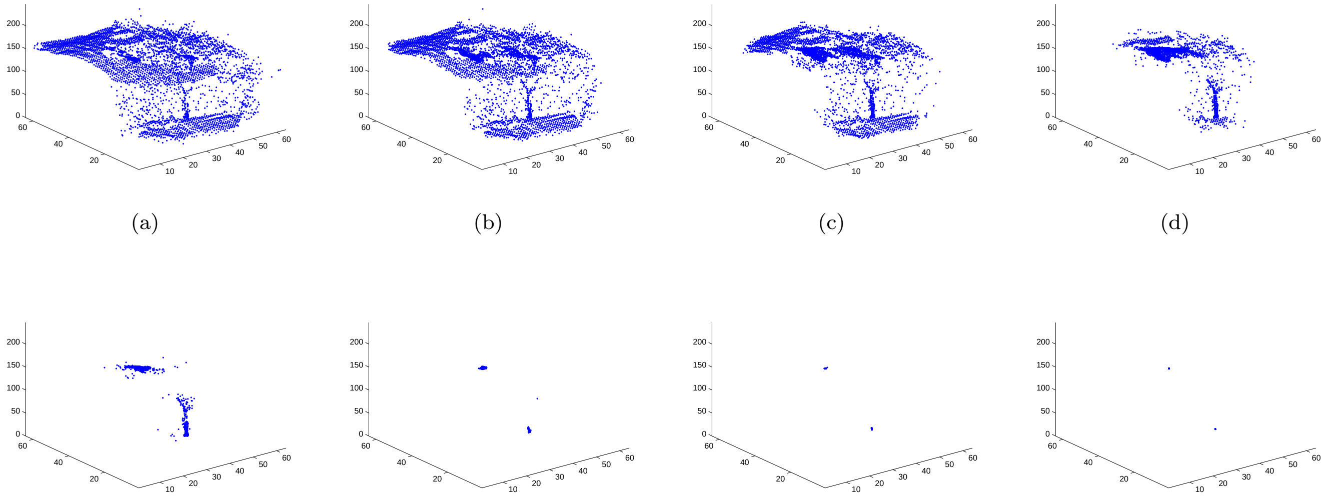
The other commonly used kernel is Gaussian kernel defined as

$$K_G(x) = c_g exp(-\frac{1}{2}|\frac{x}{h}|^2), \tag{11}$$

where c_g is a normalization constant for the Gaussian kernel.

The Gaussian kernel based energy function has partial derivative to scope as

$$\frac{\partial E_{i,j}}{\partial h_{x_i}} = 2c_g[h_{x_i} + |x_i - x_j|(1 - |\frac{x_i - x_j}{h_{x_i}}|)] \tag{12}$$



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 4. Snapshots of the data evolution, 10000 data points. (a) - (h) Snapshots are captured at 0, 5000, 10000, ..., 30000, 35000 iterations (or 0, 0.5, 1, ...,3, 3.5 iterations per point).

$$\cdot exp(-\frac{1}{2}|1-\frac{x_j-x_i}{h_{x_i}}|^2).$$

Both Epanechnikov and Gaussian kernels are applicable to EMS. Compared with Gaussian kernel, Epanechnikov requires less computation since $\frac{\partial E_{i,j}}{\partial h_{x_i}}$ is 0 except that the moving point x_j is the n^{th} nearest point to x_i. To accelerate the clustering speed, Epanechnikov is applied in this work.

h_{x_i}, the scope of x_i, is a sensitive parameter which influences the performance of the algorithm. Many works [43,44] have recognized the sensitivity of the mean shift and blurring mean shift algorithms to the kernel bandwidth. When the local characteristics of the feature space differ across data, it is difficult to find an optimal global bandwidth [41]. [44] calculates the bandwidth through a sample point estimate, and the algorithm works well with moderate training procedures. More sophisticated bandwidth estimation method incorporating the input data is reported in [45], with an increased computational complexity and manual efforts from domain experts.

For EMS algorithm, h_{x_i} is adaptively calculated using one of the two following methods. A simple way is specifying h_{x_i} to be the distance from the n^{th} nearest point to x_i, which is a NNS problem with complexity of $O(logN)$. A more complicated way is using a pilot density estimate to train and individually assign "n" to each point. The derivations presented in this paper hold for both the cases of a global "n" and individually assigned "n" to each point.

3.2.4. *The EMS Clustering with Epanechnikov Kernel*

The iterative scheme with Epanechnikov kernel is described as follows.

Algorithm 1 The EMS Clustering Procedure

```
Input: A set of data points $X^k$, where $k$ is initialized as 0
Output: A well clustered set of data points $X_{EMS}$
– Find the data point $x_i^k \in X^k$ where $|\overrightarrow{v_{EMS_i}^k}| = max(|\overrightarrow{v_{EMS_p}^k}|, x_p^k \in X^k)$. Here, $\overrightarrow{v_{EMS_i}^k}$ represents the EMS vector for the data point $x_i^k$, where the EMS vector is defined in the $4^{th}$ step.
– Obtain $U(x_i^k)$, the set of $n$ nearest neighbors of $x_i^k$.
– Obtain the set of $m$ data points $V(x_i^k)$ where for each point $x_l^k$ in the set, its $n$ nearest neighbors contain $x_i^k$, i.e., $x_i^k \in U(x_l^k)$.
– Move $x_i^k$ to the centroid of $U(x_i^k)\bigcup V(x_i^k)$. The vector from the original $x_i^k$ to the centroid of $U(x_i^k)\bigcup V(x_i^k)$ is the EMS vector under an Epanechnikov kernel.
– If $E(X^k)$ satisfies Eq. 24, stop; otherwise, set $k \leftarrow k+1$ and go to the $1^{st}$ step.
```

3.2.5. *Convergence of the EMS Algorithm*

In this and the next subsections, theoretical validations that support our clustering method are presented.

Theorem *The Evolving Mean Shift Clustering Algorithm converges.*

Proof. Since $E(X)$ is lower bounded, it is sufficient to show that $E(X)$ is strictly monotonic decreasing, i.e., if $X^k \neq X^{k+1}$ then $E(X^k) > E(X^{k+1})$ for all $k = 1, 2....$

Following the procedure described in Algorithm 1, for the k^{th} iteration (in the following, the superscript k for the data points is omitted for clarity), the data point $x_i \in X$ with the largest EMS vector is selected and moved to the centroid of $U(x_i)\bigcup V(x_i)$. In this proof, we denote the energy related to x_i as $Er(x_i)$, which refers to the energy that could change with the movement of x_i. By assuming without loss of generality that the movement of x_i does

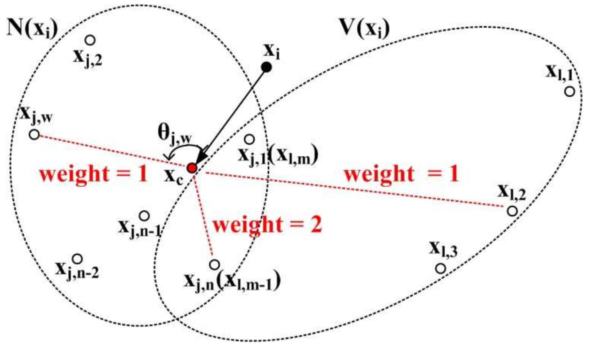


Fig. 5. Evolving mean shift movement

not change the local data configuration, or, in other words, the points in $U(x_i)$ and $V(x_i)$ do not change (cases with configuration changes are explained later), we can write

$$Er(x_i) = \sum_{x_j \in U(x_i)} |x_i - x_j|^2 + \sum_{x_l \in V(x_i)} |x_i - x_l|^2, \qquad (13)$$

where the first term on the right hand side is the energy of x_i and the second term corresponds to the energy of points in $V(x_i)$ that is related to x_i.

As illustrated in Figure 5, denoting x_c as the centroid of $U(x_i) \bigcup V(x_i)$, θ_j as the angle between each $\overrightarrow{x_c x_j}$, $(x_j \in U(x_i))$ and $\overrightarrow{x_c x_i}$, and θ_l as the angle between $\overrightarrow{x_c x_l}$, $(x_l \in V(x_i))$ and $\overrightarrow{x_c x_i}$, Eq. 13 can be expanded as

$$\begin{aligned} & Er(x_i) = \\ & \sum_{x_j \in U(x_i)} (|x_i - x_c|^2 + |x_c - x_j|^2 - 2|x_i - x_c||x_c - x_j| \cos\theta_j) \\ & + \sum_{x_l \in V(x_i)} (|x_i - x_c|^2 + |x_c - x_l|^2 - 2|x_i - x_c||x_c - x_l| \cos\theta_l) \\ & = (n+m)|x_i - x_c|^2 + \sum_{x_j \in U(x_i)} |x_c - x_j|^2 + \sum_{x_l \in V(x_i)} |x_c - x_l|^2 \\ & -2|x_i - x_c|(\sum\nolimits_{x_j \in U(x_i)} P_j + \sum\nolimits_{x_l \in V(x_i)} P_l), \qquad (14) \end{aligned}$$

where P_j is the projected vector from x_j to the vector $\overrightarrow{x_c x_i}$ and P_l the projected vector from x_l to $\overrightarrow{x_c x_i}$. n and m are the numbers of data points in $U(x_i)$ and $V(x_i)$. Since x_c is the centroid of the data points $U(x_i) \bigcup V(x_i)$, we have

$$\sum\nolimits_{x_j \in U(x_i)} P_j + \sum\nolimits_{x_l \in V(x_i)} P_l = 0. \qquad (15)$$

Substituting Eq. 15 into Eq. 14 results

$$\begin{aligned} & Er(x_i) = (n+m)|x_i - x_c|^2 \\ & + \sum_{x_j \in U(x_i)} |x_c - x_j)|^2 + \sum_{x_l \in V(x_i)} |x_c - x_l|^2 \qquad (16) \end{aligned}$$

After the movement of x_i to the centroid x_c, the first term of the right hand side of Eq. 16 becomes zero and the second term remains the same as only x_i moves in this particular iteration. Recall that the goal here is to prove that after the movement, the resulting energy $E(X^{k+1})$ is smaller than the that of the previous iteration $E(X^k)$. Therefore, we write

$$E(X^{k+1}) - E(X^k) = \Delta Er(x_i) = -(n+m)|x_i - x_c|^2. (17)$$

Since $(n+m)(x_c - x_i)^2 \geq 0$, $E(X^{k+1}) - E(X^k)$ is nonpositive. Further, as long as $X^k \neq X^{k+1}$, the right hand side of Eq. 17 is strictly negative, therefore $E(X^{k+1}) - E(X^k) < 0$. Consequently, the sequence $E(X^k)$ is convergent.

To prove the convergence of the sequences $E(X^k)|_{k=1,2...}$ without assuming that the data configuration after the movement of x_i is preserved, we rewrite Eq. 13. The configuration changes are decomposed into two basic categories to make this proof clear.

(1) Elements in $U(x_i)$ change.

This case describes the situation that a certain amount of points in $U(x_i)$ may become too far to the moved x_i thereby their positions in $U(x_i)$ be replaced by another set of points. Denote the set of points changed out of $U(x_i)$ as $U_{out}(x_i)$ and the set of points changed in as $U_{in}(x_i)$, Eq. 13 can be expressed as

$$\begin{aligned} & Er'(x_i) = \sum\nolimits_{x_j \in U(x_i)} |x_i - x_j|^2 + \sum\nolimits_{x_l \in V(x_i)} |x_i - x_l|^2 \\ & - \sum\nolimits_{x_j \in U_{out}(x_i)} |x_i - x_j|^2 + \sum\nolimits_{x_j \in U_{in}(x_i)} |x_i - x_j|^2. \quad (18) \end{aligned}$$

Recalling that $U(x_i)$ contains n nearest neighbors to x_i, it is now straightforward to see that the distances between the x_i with the points in $U_{in}(x_i)$ are smaller than those with points in $U_{out}(x_i)$. Therefore, after the movement

$$Er'(x_i) < Er(x_i) \qquad (19)$$

so that the resulting $E(X^{k+1})$ is even smaller and $E(X^{k+1}) < E(X^k)$ holds.

(2) Elements in $V(x_i)$ change.

(2-1) After the movement, x_i may be too far to some points in $V(x_i)$ so that these points do not belong to $V(x_i)$ any longer. Denoting $V_{out}(x_i)$ as the set of points that leave $V(x_i)$ and x_q as the new point that replaces the x_i for points in $V_{out}(x_i)$, we can write Eq. 13 as

$$\begin{aligned} & Er''(x_i) = \sum\nolimits_{x_j \in U(x_i)} |x_i - x_j|^2 + \sum\nolimits_{x_l \in V(x_i)} |x_i - x_l|^2 \\ & - \sum\nolimits_{x_l \in V_{out}(x_i)} (|x_l - x_i|^2 - |x_l - x_q|^2). \qquad (20) \end{aligned}$$

For each point in $V_{out}(x_i)$, their distance to the corresponding x_q is smaller than that to the moved x_i, thereby

$$Er''(x_i) < Er(x_i). \qquad (21)$$

(2-2) The other possibility in this category is that some points that previously did not belong to $V(x_i)$ find the moved x_i close enough so that they become elements in $V(x_i)$. Denoting $V_{in}(x_i)$ as the set of points that enter $V(x_i)$

and x_r the point replaced by the x_i for points in $V_{in}(x_i)$ yields

$$Er'''(x_i) = \sum\nolimits_{x_j \in U(x_i)} |x_i - x_j|^2 + \sum\nolimits_{x_l \in V(x_i)} |x_i - x_l|^2$$

$$- \sum\nolimits_{x_l \in V_{in}(x_i)} (|x_l - x_r|^2 - |x_l - x_i|^2). \quad (22)$$

Similar to (2-1), we have

$$Er'''(x_i) < Er(x_i). \quad (23)$$

From Eq. 19, 21 and 23, we prove that the data configuration changes after each movement can only reduce the total energy. This completes the proof.

3.2.6. *EMS Stopping Criteria*

A most intuitive stopping criteria for EMS is the convergence of the data set. Quantitatively, for the each kth iteration, a new set of data points are constructed and denoted as $Y^k = \{y_i^k|_{i=1,...,N}\}$ where $y_i^k = x_i^k + \overrightarrow{v_{EMS_i}^k}$. Then, the stopping criteria is: given a small positive number of ϵ, if

$$E(X^k) - E(Y^k) \leq \epsilon, \quad (24)$$

the iterative procedure stops. The required amount of iterations increases as ϵ decreases. In our implementation, ϵ is fixed to be a fraction of the ADC's resolution which is normally designed according to the SNR referred at the input of the pre-amplifier.

3.3. *Discussion*

When applying our algorithm to the spike sorting application, there are a few practical challenges.

A first challenge is that neurons may fire at a stochastic pace: sometimes it fires at high frequency and later becomes quiet. A data collection procedure is typically a few minutes and harvests thousands of spikes, thus, the constructed feature space maybe spare and with large density variation. Kernel bandwidths dictated by a global "n" could be a bias to large cluster, and contaminate high firing neurons with low firing ones. An alleviation that is also mentioned in Section 3.2.3 is to assign "n" locally, where data points at different density region can have different "n". Sample density estimator techniques, which are used to train the kernel bandwidth for mean shift algorithms [43, 44], can be similarly applied for EMS.

Specifically, a pilot density estimate is first calculated as

$$p(x_i) = \frac{1}{h_0^d} \sum_{j=1, j\neq i}^{N} K(\frac{x_i - x_j}{h_0}), \quad (25)$$

where h_0 is a manually specified global bandwidth and d is the dimension of the data space. Based on the pilot density estimate, local bandwidths are updated as

$$h_{x_i} = h_0[\frac{\lambda}{p(x_i)}]^{0.5}, \quad (26)$$

where $p(x_i)$ is the estimated density at point x_i, λ is a constant which is by default assigned to be geometric mean of $\{p(x_i)\}|_{i=1...N}$. "n" can be assigned locally in accordance with h_{x_i}. An example that includes both fast and slow firing neurons is presented in Figure 6, which shows that EMS algorithm equipped with locally trained "n" using sample point estimator can handle feature space with large density variation.

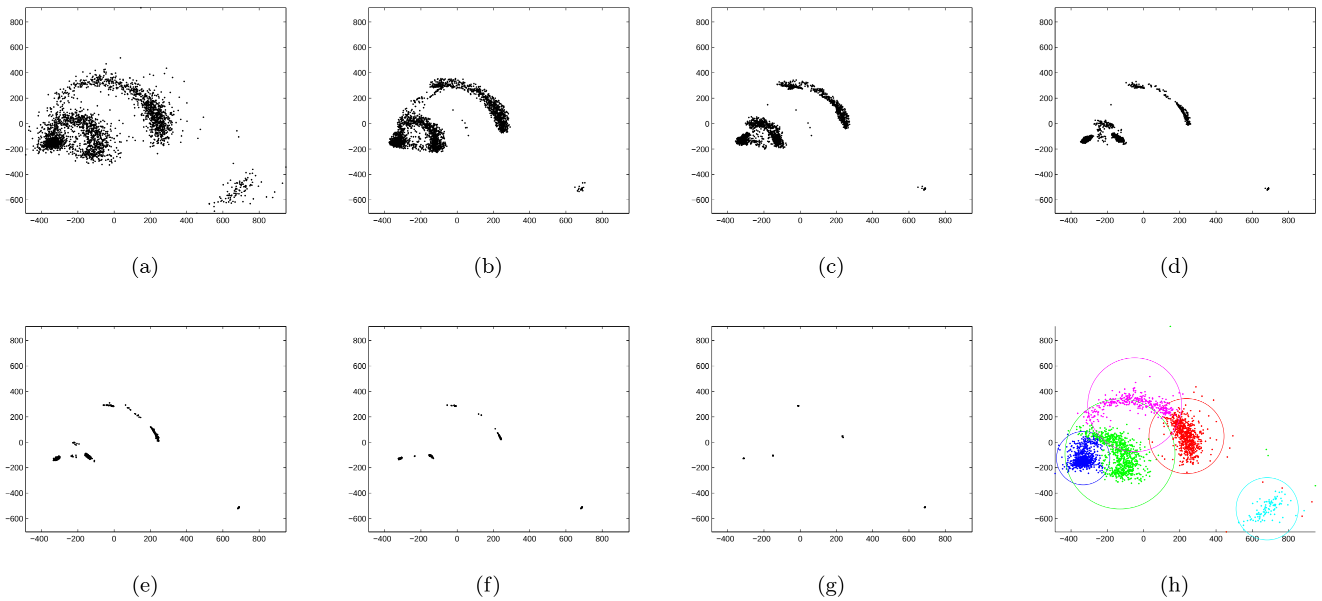
A second challenge is the involved computation and storage space cost for the proposed spike sorting algorithm, and the feasibility for microchip implementation. Using the simplified feature set as spike features, the required computation and storage space for feature extraction are negligible compared with clustering part. Regarding the EMS clustering algorithm, it typically converges a spike feature space with $2 \sim 6$ iterations per point and each iteration demands $O(logN)$ operations to find associated data points based on which calculate to the EMS vector. Modern semiconductor technology can operate at very high frequency(>GHz), which are sufficient to support clustering algorithms with moderate computational load. One of the difficulties of implementing clustering algorithm on chip is the required memory size. According to a recent work implemented under $.35\mu m$ process [33], 1Mbit on-chip memory alone can consume about $50mm^2$ die area. Many clustering algorithms are usually designed for software implementation, which may require large memory size that is expensive for microchip implementation. EMS algorithm, for example, consumes ~1Mbits total memory for handling a few thousands spikes. A corresponding implementation of EMS using $.35\mu m$ process, especially for processing multiple-channel spike data, would be too large in size to pursue. However, efforts have been made to implement EMS algorithm using a smaller feature size technology. For example, we have pursued to implement one dimension EMS clustering algorithm that has relaxed requirements on computation and storage space using $90nm$ process [46].

4. EXPERIMENT

4.1. *Experiments on Synthesized Spike Data*

Synthesized spike sequence from waveclus are used to compare the performance of different feature extraction approaches. Figure 7 (a) - (h), spikes with ground truth are grouped by colors. Feature extraction using the pre-specified subset consists of the peaks of the spike derivative as well as the height of the original spike is shown in Figure 7 (i) - (p). Comparative feature extraction results using PCA are also shown in Figure 7. The extracted spike features are clustered on a PC. About 5% overlapping spikes are ignored to clearly quantify the performance of different spike feature extraction algorithms. The sorting accuracy comparisons are listed in Table 1.

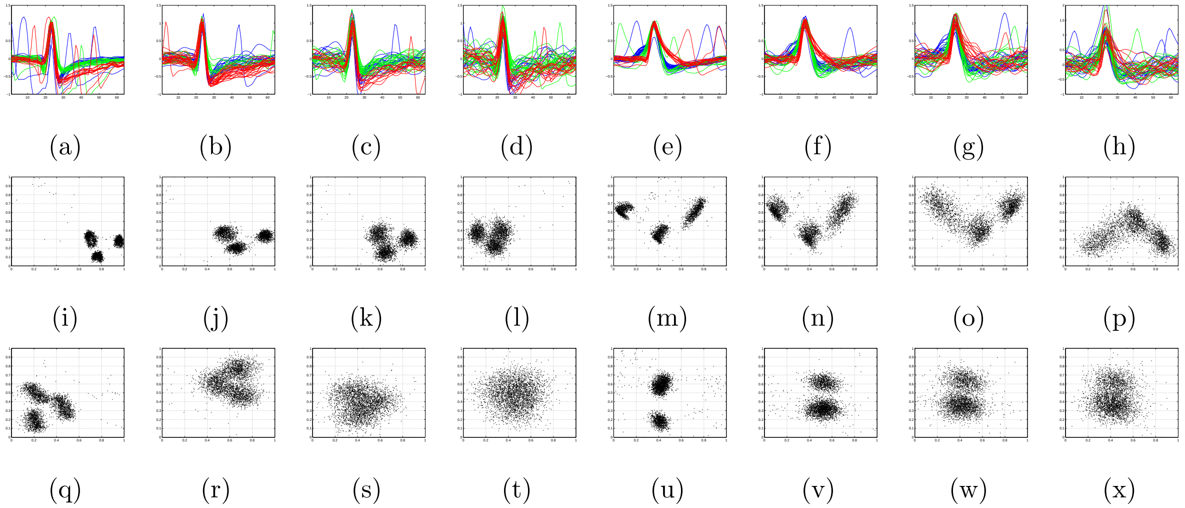
After the features have been extracted, clustering is done by the EMS clustering algorithm, as introduced in Section



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 6. spike clustering on a feature space that includes both fast and slow neurons. (a) - (g) display the feature space at 0, 0.5, ... 3 EMS iterations per point. (h) display the clustering partition results based on (g) without post processing.

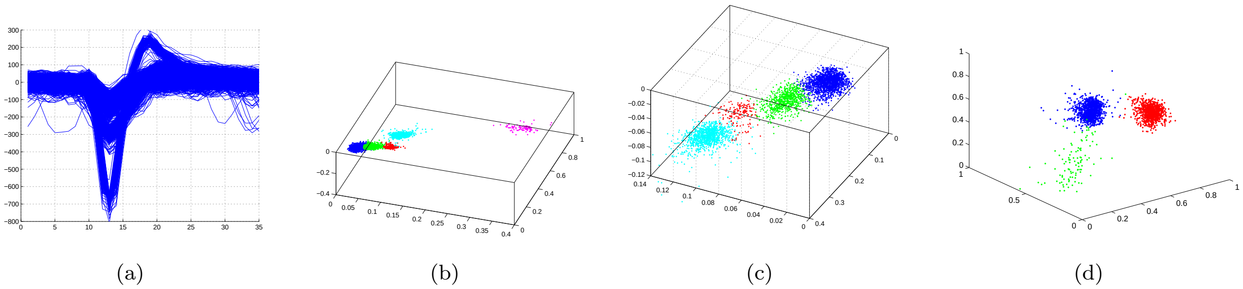


(a) (b) (c) (d) (e) (f) (g) (h)

(i) (j) (k) (l) (m) (n) (o) (p)

(q) (r) (s) (t) (u) (v) (w) (x)

Fig. 7. spike sequences from waveclus are tested using both the proposed feature extraction method and PCA. (a) - (h), spikes with ground truth grouped by colors. (i) - (p) display the extraction results using hardware. (q) - (x) display the extraction results using PCA.



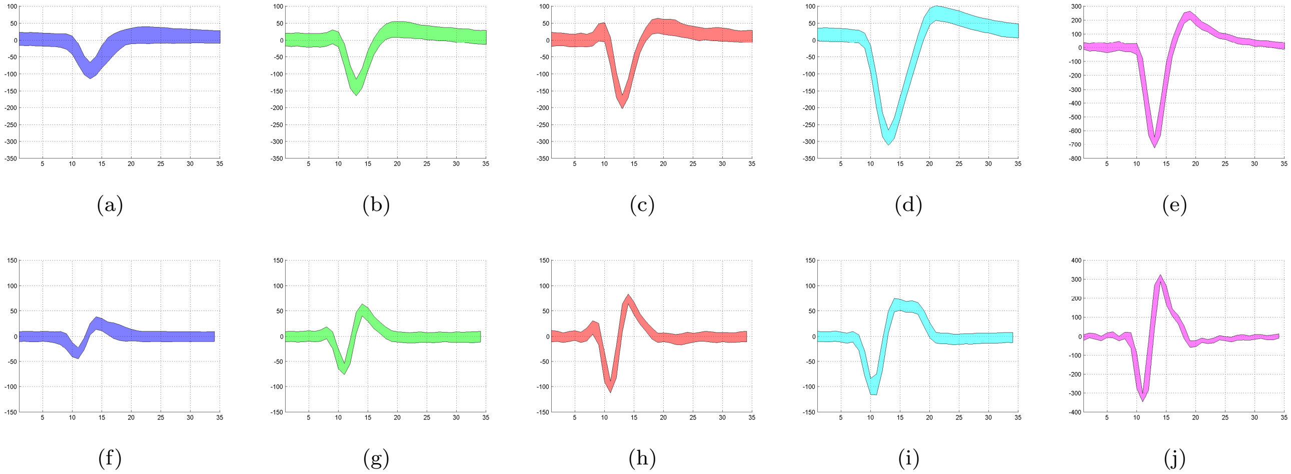
(a) (b) (c) (d)

Fig. 8. (a) detected spikes from an in-vivo preparation, (b) extracted spike features using a subset of samples, (c) zoom in of (b) for better visualization; (d) extracted features using PCA.

3. Only a spike cluster has a size larger than the threshold, it is recognized as a neuronal source.

4.2. *Experiments on Real Data*

An example containing more than 4000 spikes recorded from an in-vivo preparation is shown in Figure 8. In Fig-



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Fig. 9. (a) - (e) the classified 5 clusters of the in-vivo preparation shown in Figure 8, (f) - (j) the derivative of the classified 5 clusters. The identity is indicated by color.

Table 1
Accuracy comparison of using different spike feature extraction algorithms

Sequence	1	2	3	4	5	6	7	8
Hardware	97.6%	97.6%	97.4%	95.4%	98.2%	98.4%	93.2%	91.0%
PCA	97.8%	89.0%	60.4%	55.2%	97.6%	77.8%	80.2%	68.8%

ure 8 (a), detected spikes are superimposed. Extracted features using the proposed method are shown in Figure 8 (b). A zoom in of Figure 8 (b) is plotted in Figure 8 (c) to display the isolation quality of clusters in feature space. The corresponding PCA based feature extraction is shown in Figure 8 (d) as a comparison. The classified spike clusters using the proposed method are plotted in Figure 9 (a) - (e). Spike clusters plotted in Figure 9 (b), (c) and (d) resemble each other in shape and magnitude. To demonstrate that the informative samples based sorting does not over partitioning the data set, the derivatives of spike clusters plotted in Figure 9 (a) - (e) are also plotted in Figure 9 (f)-(j) with the same color indication. Clearly, Figure 9 (g), (h) and (i) present three well-differentiated waveform patterns in either peak-to-peak magnitude or shape.

Unlike synthesized spikes where the “ground truth” is given for comparison, sorting results on animal data are difficult to evaluate. Spike correlogram that is defined as a neuron’s conditional firing possibility could reveal certain temporal statistics of the classified spike clusters. Estimated correlograms of spike clusters in Figure 8 are displayed in Figure 10 and 11 at different time scales.

5. Conclusion

This paper presents a spike sorting method using a simplified feature set with nonparametric EMS clustering algorithm. The proposed spike feature extraction algorithm requires very low computation and has been implemented with a custom integrated circuit chip. The EMS clustering algorithm performs nonparametric clustering. Through iterations, the data points collapse into well formed clusters. By claiming these isolated clusters, neural spikes are classified.

6. ACKNOWLEDGMENT

The authors want to thank Moo Sung Chae, Tung Chien Chen and Linh Hoang for designing neural recording and processing chip, Kuan Fu Chen and Linh Hoang for working on the interface between the chip and PC, Eric Basham for providing technical suggestions, Natalia Tchemodanov for proofreading. The authors want to thank Dr. Victor Pikov for cat experiments and helpful technical suggestions, Dr. Rodrigo Quian Quiroga for sharing the spike database and Prof. Krishna Shenoy for providing a monkey sequence. The authors also want to thank the anonymous reviewers for very helpful suggestions of revising the paper.

7. Appendix

7.1. *An Intuitive Explanation of NEO*

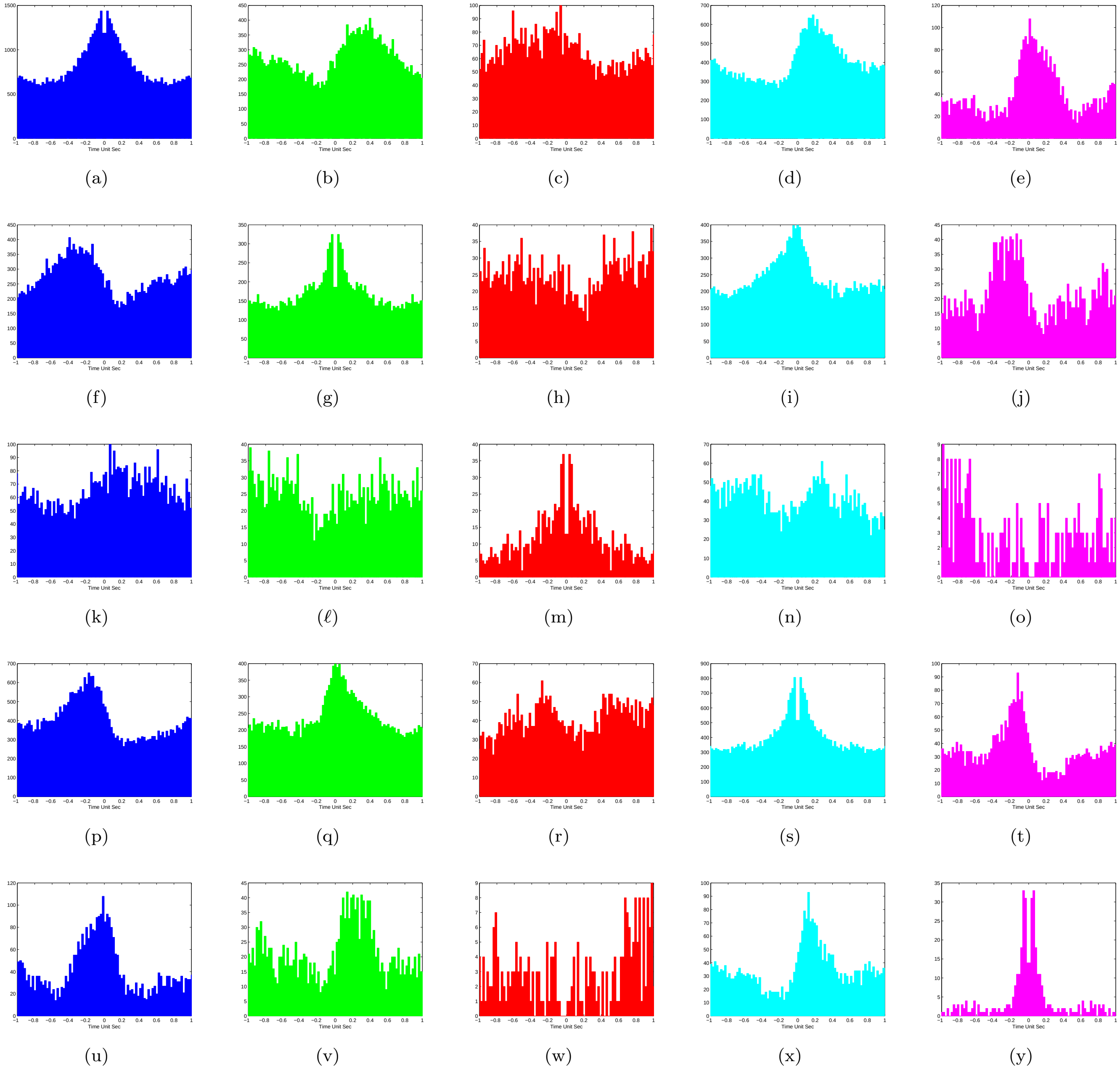
In this section, a simplified explanation of NEO as a neural spike detection algorithm is presented.

For a general detection algorithm, its goal is to decide if neural spike activities present within a time slot of interest. Assuming $V(i)$ and $A(i)$ as neural spikes and background signal, the output of NEO according to Eq. 1 is

$$\psi[\cdot] = [2V(i)A(i) - V(i-1)A(i+1) - V(i+1)A(i-1)] \quad (27)$$

$$+[V(i)^2 - V(i-1)V(i+1)] + [A(i)^2 - A(i-1)A(i+1)]$$

Since a high pass filter is typically placed at a few hundred Hz, $V(i)$ and $A(i)$ can be roughly treated as zero mean, independent waveforms within the time slot. As a result, the averaged correlation terms of $V(i)$ and $A(i)$ approach zero



(a) (b) (c) (d) (e)

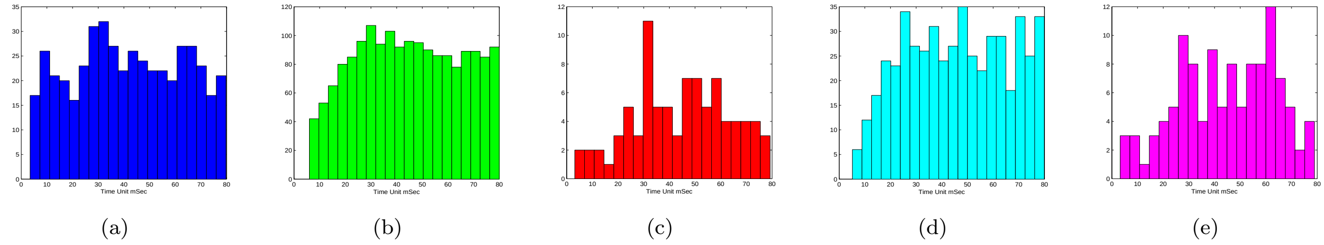
(f) (g) (h) (i) (j)

(k) (ℓ) (m) (n) (o)

(p) (q) (r) (s) (t)

(u) (v) (w) (x) (y)

Fig. 10. the auto- and cross correlograms of classified neuron spikes shown in Figure 8 and Figure 9. The time scale is set to -1 to 1 sec to examine the excitatory and inhibitory firing relations.



(a) (b) (c) (d) (e)

Fig. 11. the autocorrelograms of classified neuron spikes shown in Figure 8 and Figure 9. The time scale is set to 0-80 msec to examine the refractory period.

$$\overline{2V(i)A(i)} - \overline{V(i-1)A(i+1)} - \overline{V(i+1)A(i-1)} \approx 0. (28)$$

Given that the averaged NEO score $\overline{\psi[V(i)+A(i)]}$ is used to decide if neural spikes present, those correlation terms in Eq. 27 should not influence a decision.

For a general sequence x(i), the auto-correlation function $R_x(\tau) = \overline{x(n)x(i+\tau)}$ is by definition the reverse Fourier transform of the power spectrum

$$R_x(\tau) = \int P(f)e^{j2\pi f\tau}df. \tag{29}$$

According to Eq. 27 and 29, the averaged NEO output is

$$\overline{\psi(x(n))} = R_x(0) - R_x(\tau) = \int P(f)(1 - cos2\pi f\tau)df \tag{30}$$

When frequency of interest is much lower than the sampling frequency, $1 - cos2\pi f\tau$ is approximately $2\pi^2 f^2 \tau^2$, which attenuates low frequency components. Since background noise has dominant components at lower frequencies while spikes are relatively high frequency events, $[V(i)^2 - V(i-1)V(i+1)]$ become dominant when spikes are present.

7.2. *Noise Shaping Filter*

In Section 2, we have mentioned applying a frequency-shaping filter before feature extraction.

In general, the power spectrum of the input referred noise at the first stage amplifier exhibits a decaying profile [30,47] and approximates as

$$N(f) = N_{neu} + N_{e.e} + N_{1/f} + N_{therm}$$

$$\approx N_{f_{c1}}(\frac{f_{c1}}{f})^{\alpha} + N_{therm}, \tag{31}$$

where N_{neu} is the neuronal noise, $N_{e.e}$ is the electrode-electrolyte interface noise, $N_{1/f}$ is the flicker noise, N_{therm} is the thermal noise contributed by tissue impedance and transistors, f_{c1} is the high pass corner frequency of the digital filter, and $N_{f_{c1}}$ is the low frequency noise at frequency f_{c1}. Except thermal noise, the remaining noise is featured at low frequency and assumed to have profile following $f^{-\alpha}$. Noise profiles vary among both of objects and recording systems, however, low frequency noise is typically dominant.

Among various frequency shaping filters, taking derivative is a simple one, which almost linearly emphasizes signal spectrum according to frequency. For a discrete time spike sequence, taking the derivative after the analog-to-digital converter (ADC) has the frequency response

$$H(f) = 2e^{j\pi f/2}\sin(\pi f/f_s), \tag{32}$$

where f_s is the sampling frequency of the ADC.

The effect of a frequency shaping filter on noise can be quantitatively evaluated by the expression

$$k = \frac{1}{N_0|H(f_{spike})|^2}\int_{f_{c1}}^{f_{c2}} N(f)|H(f)|^2 df, \tag{33}$$

where f_{ci} are the corner frequencies (3dB attenuation frequency points) of the digital filter before feature extraction, f_{spike} is the center frequency of the spike signal, $N(f)$ is the estimated power spectrum of the noise, and N_0 is the integrated noise over passing band. If k is less than 1, the SNR further increases, which improves waveform differentiation.

After derivative, the noise spectrum density changes to

$$N(f)|H(f)|^2 = 2[N_{f_{c1}}(f_{c1}/f)^{\alpha} + N_{therm}]\sin^2(\pi f/f_s).(34)$$

For integer α, a closed loop expression of the integrated noise after derivative (N_1) can be obtained from Eq. 34 . With a further assumption that the sampling frequency is sufficiently higher than the signal spectrum, the expression of N_1 could be generalized to non-integer α as

$$N_1 \approx \frac{2N_{f_{c1}}f_{c1}\alpha\pi^2}{(3-\alpha)f_s^2}[f_{c2}^{3-\alpha} - f_{c1}^{3-\alpha}] + \frac{2N_{therm}\pi^2}{3f_s^2}[f_{c2}^3 - f_{c1}^3].(35)$$

Combine Eq. 33 and 35, the parameter k that is used to quantify the modification to SNR due to the frequency shaping filter is

$$k = \frac{\frac{N_{f_{c1}}f_{c1}^{\alpha}}{3-\alpha}(f_{c2}^{3-\alpha} - f_{c1}^{3-\alpha}) + \frac{N_{therm}}{3}(f_{c2}^3 - f_{c1}^3)}{N_{f_{c1}}\frac{f_{c1}^{\alpha}}{1-\alpha}(f_{c2^{1-\alpha}} - f_{c1}^{1-\alpha}) + N_{therm}(f_{c2} - f_{c1})}\frac{1}{2f_{spike}^2}.(36)$$

The quantitative impact of frequency shaping filter on noise is affected by the recording system and biological environment. Here, we use $\alpha = 2$ which is a typical value to illustrate the analysis

$$k = \frac{N_{f_{c1}}f_{c1}^2(f_{c2} - f_{c1}) + \frac{N_{therm}}{3}(f_{c2}^3 - f_{c1}^3)}{N_{f_{c1}}f_{c1}^2(\frac{1}{f_{c1}} - \frac{1}{f_{c2}}) + N_{therm}(f_{c2} - f_{c1})}\frac{f_{c2}^2}{2f_{spike}^2} \tag{37}$$

$$\approx \frac{\frac{f_x^2}{f_{c2}} + \frac{f_{c2}}{3}}{\frac{f_x^2}{f_{c1}} + f_{c2}}\frac{f_{c2}^2}{2f_{spike}^2}$$

with

$$f_x = f_{c1}(\frac{N_{f_{c1}}}{N_{therm}})^{1/2}, \tag{38}$$

where f_x is the frequency at which the noise spectrum approximately settles to the thermal noise floor. During the measurement, f_x varies according to the recording system and biological environment (varies from 6 to 12 KHz in measurements).

In the case that the digital filter's low pass corner frequency f_{c2} is designed smaller or comparable to f_x, Eq. 37 can be simplified as

$$k \approx \frac{f_{c1}f_{c2}}{2f_{spike}^2} \approx \frac{2f_{c1}f_{c2}}{(f_{c1}+f_{c2})^2} \leq \frac{1}{2}, \tag{39}$$

where the approximation holds well if the center frequency of the spike signal is close to the middle point of the filter's passing band.

As a summary, the spectrum of the recorded noise exhibits a decaying profile with respect to the frequency within the signal band. Therefore, an appropriate frequency shaping filter could be used to further improve the SNR.

References

[1] M. Lewicki, A review of methods for spike sorting: the detection and classification of neural action potentials, Network Comput. Neural. Syst. 9 (1998) 53–78.

[2] U. Rutishauser, E. Schuman, A. Mamelak, Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, in vivo, J. Neurosci. Methods 154 (1-2) (2006) 204–224.

[3] F. Wood, M. Black, C. Vargas-Irwin, M. Fellows, J. Donoghue, On the variability of manual spike sorting, IEEE Trans. Biomed. Eng. 51 (6) (2004) 912–918.

[4] J. Chapin, Using multi-neuron population recordings for neural prosthetics, Nat. Neurosci. 7 (2004) 452–455.

[5] M. Serruya, N. Hatsopoulos, L. Paninski, M. Fellows, J. Donoghue, Instant neural control of a movement signal, Nature 416 (2002) 141–142.

[6] L. Hochberg, M. Serruya, G. Friehs, J. Mukand, M. Saleh, A. Caplan, A. Branner, D. Chen, R. Penn, J. Donoghue, Neuronal ensemble control of prosthetic devices by a human with tetraplegia, Nature 442 (2006) 164–171.

[7] R. Bellman, Adaptive Control Processes, Princeton University Press, Princeton, NJ, 1961.

[8] W. Liu, K. Vichienchom, M. Clements, S. DeMarco, C. Hughes, E. McGucken, M. Humayun, E. De Juan, J. Weiland, R. Greenberg, A neuro-stimulus chip with telemetry unit for retinal prosthetic device, IEEE Journal of Solid-State Circuits 35 (10) (2000) 1487–1497.

[9] Z. Yang, W. Liu, E. Basham, Inductor modeling in wireless links for implantable electronics, IEEE Transactions on Magnetics 43 (2007) 3851–3860.

[10] L. Theogarajan, J. Wyatt, Minimally invasive retinal prosthesis, in: IEEE International Solid-State Circuits Conference (ISSCC), 2006, pp. 54–55.

[11] Z. Zumsteg, C. Kemere, S. O'Driscoll, G. Santhanam, R. Ahmed, K. Shenoy, T. Meng, Power feasibility of implantable digital spike sorting circuits for neural prosthetic systems, IEEE Trans. Neural Syst. Rehabil. Eng. 13 (3) (2005) 272–279.

[12] P. Thakur, H. Lu, S. S. Hsiao, K. Johnson, Automated optimal detection and classification of neural action potentials in extra-cellular recordings, J. Neurosci. Methods 162 (1) (2007) 364–376.

[13] H. Jung, J. Choi, T. Kim, Solving alignment problems in neural spike sorting using frequency domain pca, Neurocomputing 69 (7-9) (2006) 975–978.

[14] T. Blanche, N. Swindale, Nyquist interpolation improves neuron yield in multiunit recordings, J. Neurosci. Methods 155 (1) (2006) 207–216.

[15] M. Sahani, Latent variable modelss for neural data analysis, Ph.D. dissertation, California Institute of Technology.

[16] M. Lewicki, Bayesian modeling and classification of neural signals, Neural Comput. 6 (1994) 1005–1030.

[17] R. Vollgraf, K. Obermayer, Improved optimal linear filters for the discrimination of multichannel waveform templates for spike-sorting applications, IEEE Signal Proc. Let. 13 (3) (2006) 121–124.

[18] P. Zhang, J. Wu, Y. Zhou, P. Liang, J. Yuan, Spike sorting based on automatic template reconstruction with a partial solution to the overlapping problem, J. Neurosci. Methods 7 (5) (2004) 446–451.

[19] R. Quian Quiroga, Z. Nadasdy, Y. Ben-Shaul, Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering, Neural Comput. 16 (8) (2004) 1661–1687.

[20] A. Pavlov, V. Makarov, I. Makarova, F. Panetsos, Sorting of neural spikes: When wavelet based methods outperform principal component analysis, Natural Computing 6 (3) (2007) 269–281.

[21] T. Borghi, R. Gusmeroli, A. Spinelli, G. Baranauskas, A simple method for efficient spike detection in multiunit recordings, J. Neurosci. Methods 163 (1) (2007) 176–180.

[22] J. Choi, H. Jung, T. Kim, A new action potential detector using the mteo and its effects on spike sorting systems at low signal-to-noise ratios, IEEE Tran. Biomed. Eng. 53 (4) (2006) 738–746.

[23] K. Kim, J. Kim, Neural spike sorting under nearly 0-db signal-to-noise ratio using nonlinear energy operator and artificial neural-network classifier, IEEE Trans. Biomed. Eng. 47 (10) (2000) 1406–1411.

[24] J. Hartigan, Clustering algorithms, John Wiley & Sons, New York, NY, 1975.

[25] A. Jain, M. Murty, P. Flynn, Data clustering: A review, ACM Computing Surveys 31 (3) (1999) 264–323.

[26] K. Harris, D. Henze, J. Csicsvari, H. Hirase, G. Buzsaki, Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements, J Neurophysiol. 84 (2000) 401–414.

[27] N. Schmitzer-Torbert, J. Jackson, D. Henze, K. Harris, A. Redish, Quantitative measures of cluster quality for use in extracellular recordings, Neuroscience 131 (1) (2005) 1–11.

[28] J. HARTIGAN, M. WONG, Algorithm as136: A k-means clustering algorithm, Applied Statistics 28 (1979) 100–108.

[29] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. I: 281–297.

[30] Z. Yang, T. Chen, W. Liu, A neuron signature based spike feature extraction algorithm for on-chip implementation, Proc. 30th Ann. Int. Conf. IEEE EMBS (2008) 4237–4240.

[31] T. Chen, Z. Yang, W. Liu, L. Chen, Neusort2.0: a multiple-channel neural signal processor with systolic array buffer and channel-interleaving processing schedule, Proc. 30th Ann. Int. Conf. IEEE EMBS (2008) 6652–6656.

[32] M. Chae, W. Liu, Z. Yang, T. Chen, J. Kim, M. Sivaprakasam, M. Yuce, A 128 channel 6mw wireless neural recording ic with on-the-fly spike sorting and uwb transmitter, IEEE ISSCC 2008 Dig. Tech. Papers 7 (6) (2008) 241–261.

[33] T. Chen, K. Chen, Z. Yang, K. Cockerham, W. Liu, A biomedical multiprocessor soc for closed-loop neuroprosthetic applications, IEEE ISSCC 2009 Dig. Tech. Papers.

[34] M. Chae, Z. Yang, T. Chen, J. Kim, M. Yuce, L. Hoang, W. Liu, A 128 channel 6mw wireless neural recording ic with spike feature extraction and uwb transmitter, in revision to IEEE Trans. Neural Syst. Rehabil. Eng.

[35] P. Maragos, J. Kaiser, T. F. Quatieri, On amplitude and frequency demodulation using energy operators, IEEE. Trans. Signal Proc.

[36] Z. Yang, Q. Zhao, W. Liu, Spike feature extraction using informative samples, Advances in Neural Information Processing Systems, NIPS.

[37] H. Teager, S. Teager, Evidence of nonlinear sound production mechanisms in the vocal tract, in Speech Production and Speech Modeling, edited by W. J. Hardcastle and A. Marchal (Kluwer Academic, Boston, MA) 241–261.

[38] Z. Yang, Q. Zhao, W. Liu, Improving spike separation using waveform derivatives, Jouranl of Neural Eng. *6 046006 (12 pp)* 4 (2009) 1–1.

[39] K. Fukunaga, L. D. Hostetler, The estimation of the gradient of a density function, with application in pattern recognition, IEEE Trans. Inf. Theory 21 (1975) 180–187.

[40] Y. Cheng, Mean shift, mode seeking, and clustering, IEEE Trans. Pattern Analysis and Machine Intelligence 17 (8) (1995) 790–799.

[41] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE Trans. Pattern Analysis and Machine Intelligence 24 (5) (2002) 603–619.

[42] M. Carreira-Perpiñán, Fast nonparametric clustering with gaussian blurring mean-shift, in: International Conference on Machine Learning, 2006, pp. 153–160.

[43] D. Comaniciu, V. Ramesh, P. Meer, The variable bandwidth mean shift and data-driven scale selection (2001) I: 438–445.

[44] P. Hall, T. Hui, J. Marron, Improved variable window kernel estimate of probability densities, The Annals of Statistics 23 (1) (1995) 1–10.

[45] D. Comaniciu, An algorithm for data-driven bandwidth selection, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (2) (2003) 281–288.

[46] L. Hoang, Z. Yang, W. Liu, Vlsi architecture of neo spike detection with noise shaping filter and feature extraction using informative samples, to appear in Proc. 31th Ann. Int. Conf. IEEE EMBS (2009) 1–1.

[47] V. Gilja, M. Linderman, G. Santhanam, A. Afshar, S. Ryu, T. Meng, K. Shenoy, Multiday electrophysiological recordings from freely behaving primates, Proc. 28th Ann. Int. Conf. IEEE EMBS (2006) 4387–4391.