Algorithms for Neuroprocessor Spike Sorting

6.1 Introduction

The first stage of processing the neural signals coming from a recording front-end may be the detection of time points of firing events. This task is typically referred to as spike-detection. Since an electrode might sense activity of several adjacent neurons, one may try to classify detected firing events to their originating neurons. This process is called spike-sorting. What makes spike-sorting possible is that action potentials generated by different neurons will have fairly different characteristic shapes. Sorting algorithms try to classify incoming spikes into shape-groups and treat every group as if originated by the same neuron.

Spike detection and sorting algorithms have been the subject of extensive work over many years [96]. Many different methods were proposed, including various clustering methods such as feature analysis [96] or PCA [97], template matching [98], wavelet-transform based methods [99, 100, 101] and artificial neural networks [102].

The purpose of the work presented in this chapter was to develop algorithms for integrated spike detection and sorting. The feasibility of on-chip sorting was shown in [78], which examined the power requirements of hardware implementation of some common sorting algorithms. We show below that spike-sorting power consumption can be reduced significantly by trading some classification accuracy in state-of-the-art clustering algorithms in return for considerable savings in power. Detailed description of the study, including algorithm validation, can be found in [103, 104] and in [105].

6.1.1 Clustering Methods

Clustering methods apply transformations on spike signals mapping them onto a shape-space to form scatter-plots. Under a successful mapping spikes of similar shapes fall close to each other, while spikes of different shapes lie
apart. The resulting scatter plot has a set of clusters, areas with high density of mapped spikes, each representing spikes of a different characteristic shape. Sorting is performed by assigning spikes to clusters, for example, by choosing the closest cluster. Figure 6.1 shows an example of a two-dimensional scatter-plot obtained by measuring two features of spikes. The process is successful if

![Figure 6.1. Example scatter plot of two measured features](image)

the clusters in the shape-space can be distinguished, and the distances between cluster centers (typical shapes) are larger than cluster spread (background noise influence).

Shape-space can be defined by selecting a set of spike shape features, such as the height of the first peak, the height of the second peak or spike “width”, i.e. time interval between points at 50% maximal height. Measured features define the axes of this space. This is the method of feature analysis. To make it work reliably, one must select the most discriminative set of features. This technique was common in the past [96], mostly due to the very modest computation requirements: typically the selected features are measured on the signal and require none or very little computation. Today feature analysis is replaced by a more precise and significantly more computationally complex Principal Component Analysis [97].

Cluster boundaries can be defined on the scatter plot once it has been generated. Spikes that fall inside the boundaries of a particular cluster are assigned to that cluster. Those that fall outside of any cluster boundaries are disregarded. Often, decision boundaries are defined by manually drawing polygons on the shape-space [106]. Automatic approaches exist as well, such as K-means or Bayesian clustering [107, 108].