Final preprint version of paper by Yassin et al. in
ACM Transactions on Embedded Computing Systems (TECS),
Vol 17, Issue 3, June 2018, Article No. 67.
https://doi.org/10.1145/3190854

Algorithm/architecture co-optimisation technique for automatic data reduction of wireless read-out in high-density electrode arrays

YAHYA H. YASSIN, Norwegian University of Science and Technology (NTNU) and KU Leuven
FRANCKY CATTHOOR, IMEC and KU Leuven
FABIAN KLOOSTERMAN, NERF, VIB, and KU Leuven
JYH-JANG SUN, NERF and IMEC
JOÃO COUTO, NERF
PER GUNNAR KJELDSBERG, Norwegian University of Science and Technology (NTNU)
NICK VAN HELLEPUTTE, IMEC

High-density electrode arrays used to read out neural activity will soon surpass the limits of the amount of data that can be transferred within reasonable energy budgets. This is true for wired brain implants when the required bandwidth becomes very high, and even more so for untethered brain implants that require wireless transmission of data. We propose an energy efficient spike data extraction solution for high-density electrode arrays, capable of reducing the data to be transferred by over 85%. We combine temporal and spatial spike data analysis with low implementation complexity, where amplitude thresholds are used to detect spikes and the spatial location of the electrodes is used to extract potentially useful sub-threshold data on neighboring electrodes. We tested our method against a state-of-the-art spike detection algorithm, with prohibitively high implementation complexity, and found that the majority of spikes are extracted reliably. We obtain further improved quality results when ignoring very small spikes below 30% of the voltage thresholds, resulting in 91% accuracy. Our approach uses digital logic and is therefore scalable with increasing number of electrodes.

CCS Concepts: • **Computer systems organization** → **Embedded and cyber-physical systems**; **Embedded systems**;

Additional Key Words and Phrases: Neural probes, data reduction, low energy, digital design, embedded systems

Author's addresses: Yahya H. Yassin, Department of Electronic Systems (IES), Norwegian University of Science and Technology (NTNU), Trondheim, Norway and Department of Electrical Engineering (ESAT), Katholieke Universiteit Leuven (KU Leuven), Leuven, Belgium; Francky Catthoor, IMEC, Kapeldreef 75, 3000 Leuven, Belgium and KU Leuven, Leuven, Belgium; Fabian Kloosterman, NERF, Kapeldreef 75, 3000 Leuven, Belgium and VIB, Leuven, Belgium and Brain and Cognition Research unit, KU Leuven, Leuven, Belgium; João Couto, NERF, Kapeldreef 75, 3000 Leuven, Belgium; Jyh-Jang Sun, NERF and IMEC, Kapeldreef 75, 3000 Leuven, Belgium; Per Gunnar Kjeldsberg, Department of Electronic Systems (IES), Norwegian University of Science and Technology (NTNU), Trondheim, Norway; Nick Van Helleputte, IMEC, Kapeldreef 75, 3000 Leuven, Belgium .

1 INTRODUCTION

The behavior of neural cells and networks are studied through extracellular monitoring of the action potentials in animals' brains in vivo. The activity is normally recorded using very small electrodes on implantable neural probes. The digitized signals are then transmitted to a remote receiver where detailed analysis is performed using advanced signal processing algorithms.

The electrodes detect extracellular voltage deflections that are a mixture of action potentials (spike band: 300 Hz - 6 kHz) emitted by nearby neurons and slower fluctuations that reflect population activity (local field potential band: 0.1 Hz - 300 Hz). In this paper we focus on a data reduction method applied to the spike band, based on the detection of action-potentials (also known as "spike detection") [21].

At least 20 kHz sampling rate is required to capture the spike waveform necessary for spike detection and further analysis. Figure 1 illustrates a typical probe and its electrode arrangement. The figure also shows the bandpass filtered signal (0.250 - 5.938 kHz) from each electrode at 25 kHz sampling rate, which are analyzed together in order to extract relevant spike data for post-processing. Due to the high electrode density, a spike detected in one electrode is also detected in neighboring electrodes. One of the main challenges of using such electrodes is to capture temporal and spatial spike data in noisy environments. Temporal spike data is captured by inspecting a single channel over time, while spatial spike data is captured by inspecting multiple channels with different spatial placement on the probe.

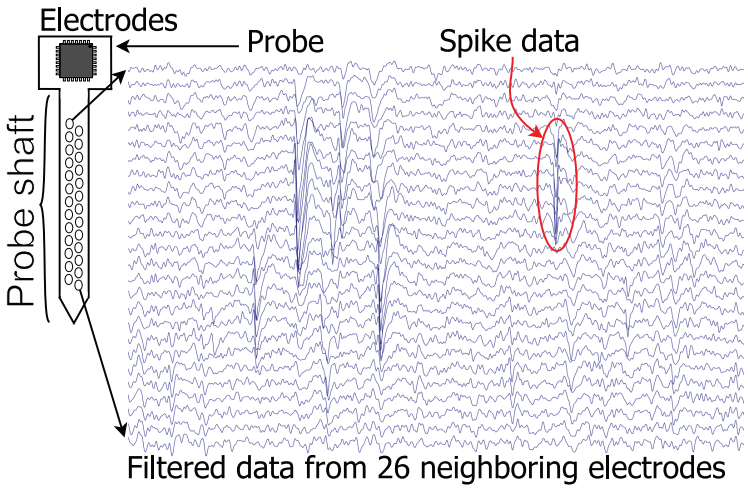


Fig. 1. The filtered recorded signal of a probe with multiple electrodes, clearly showing the spikes and spatial signal dependency.

The number of electrode channels will soon reach the limits of how much data we can transfer within reasonable energy budgets, even while transmitting the data through electrical wiring [22]. The reason is the fast growing number of concurrent sensor nodes (electrodes) on a neural probe. The state-of-the-art neural probe produced with silicon technology contains 1356 electrodes [22]. Raducanu et al. [22] show that the transmission circuitry contributes significantly to the power consumption. Other research projects aim towards thousands of electrodes on a single neural probe [26], [6]. This will exacerbate the transmission problem even more.

Researchers also aim at reducing the number of wires connected to the probe, or eliminate them all together, in order to have more freedom of movement while recording. A wireless transmitter embedded on a battery powered probe will limit the amount of data that can be transmitted in "raw form" even further compared to wired probes. Such wireless probes introduce a long term energy constraint, i.e., the total available energy based on the battery capacity. Even if the electrodes are implemented with optimal energy efficient circuitry, it is still a challenge to meet the energy constraints related to wired and wireless data transmission for future high-density electrodes. The only way to meet the requirements of the long term energy constraint is to perform data reduction to be able to transmit less data. For all data reduction approaches it is therefore important to capture sufficiently detailed information about the spike waveforms, such that these can be assigned to distinct neurons remotely (e.g., through a process called spike sorting [21]).

Mainly three different methods for data reduction exist as indicated by the three root nodes in Figure 2. The first method uses compression techniques to compress the samples of the recorded data, thus transmitting less bits. The second method is to select recording sites along slender probe shafts independently for multiple channels, such as the Electronic Depth Control (EDC) technique [29]. The third method for data reduction is to discard irrelevant data. Compression and EDC techniques are currently used to transmit continuous recordings in "raw form", but may not be enough to meet the energy constraints of future high electrode count probes. Hence, a need is present for data elimination in order to meet the stringent requirements for data transfer.

The signal analysis circuitry should be integrated on the probe closer to the electrodes in order to reduce the data to be transmitted as early as possible. Neural function is affected by temperature so any solution must not increase the brain temperature by more than 1-2 °C [34]. This limits how much short term processing is allowed on an implant. In high-density electrode arrays the heat generation and energy consumption related to the signal filtering, digitization, and especially data transfer, increases with the number of electrodes. In addition, the space available for signal analysis circuitry is limited, which poses a significant challenge. To avoid over-heating of brain tissue, state-of-the-art solutions like the one presented by Raducanu et al. [22] place the signal processing and transmission circuitry on top of the probe outside the brain.

Different approaches have been proposed to meet the challenges presented in this section, such as online spike detection [30] and compression [36] solutions. Common for all approaches is their attempt to reduce the data to be transmitted by extracting the most valuable information. The definition of "valuable" is mainly dictated by what will be done with the data in further analyses. This means that the chosen data reduction technique should match the data analysis goals. These solutions will be discussed in Section 2. In this paper we focus on spike data extraction for remote post-processing, such as spike sorting and clustering. We propose a novel and effective data reduction technique, which keeps the run-time and

storage overhead limited. The technique is mainly comprised of a digital online temporal spike detector implemented on the probe (the part outside the brain), based on voltage amplitude thresholding in combination with a spatial data extraction mechanism. This technique solves the energy problem by performing data selection, to reduce the data to be transmitted significantly. This is, to our knowledge, the first spike detection method that performs spatial and temporal analysis locally on the probe. This algorithm/architecture co-optimization technique identifies and extracts essential spike data in a way that combines accuracy and area/energy efficiency. The spatial mechanism extracts all samples around neighboring electrodes when a threshold crossing is detected. This data extraction allows remote post-processing algorithms to get sufficient information for every detected spike.

Related work and alternative solutions are presented in Section 2. In Section 3, we give a brief overview of the general goal of the data reduction and post-processing steps in order to better understand our proposed technique. Section 4 discusses wireless requirements needed to transfer large amounts of data. The temporal and spatial spike data extraction technique is presented in Section 5, and the estimated hardware (HW) requirements for our technique is discussed in Section 6. Sections 7 and 8 presents our experimental setup and quality metrics, respectively. We present and discuss our results in Section 9. Finally, we present our conclusions in Section 10.

2 RELATED WORK

Different methods for data reduction exist today, and one common aim for all solutions is to reduce the loss of potentially wanted data (false negatives). Some of these methods are implemented with analog circuitry and other solutions use digital logic for the spike detection. Alternative solutions to spike detection focus on data compression [36], or using EDC [29]. These alternative solutions can also be used complimentary to a spike detection solution. All solutions presented in this section are summarized in Figure 2.

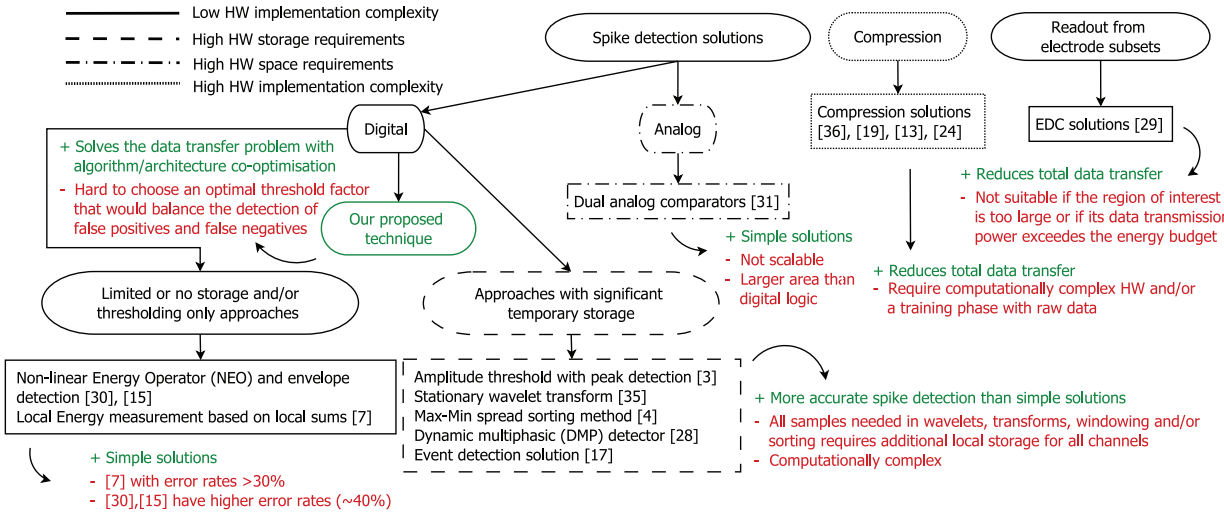


Fig. 2. Summary of current spike detection solutions.

Solutions using analog circuitry, such as detection with dual analog comparators [31], work well when the number of electrodes is small. For high-density electrode arrays, the cost of the analog comparators in terms of area limits the number of electrodes that can be supported, compared to simple digital logic. Solutions using digital circuitry can be divided

into two categories: those with low complexity and minimal local storage requirements, versus solutions that have higher complexity and/or large local storage requirements.

Advanced digital solutions need temporary storage for history, which is used for sorting, filtering and/or wavelet transforms. An example is spike detection using a stationary wavelet transform for the adaptive threshold [35]. Another example, such as a FPGA solution combining amplitude thresholding and peak detection on +/- 1 ms samples [3], requires data buffering. Other spike detection solutions using adaptive thresholds based on a max-min spread sorting method [4], analyze every 2 ms of recorded data with 50% overlap. This solution also requires buffering and sorting. Solutions using a dynamic multiphasic (DMP) detector [28] or solutions with event detection based on median and baseline estimation [17] require storage for data sorting.

Simple digital solutions on the other hand use limited or no temporary storage for spike data history, and/or use simple thresholds. Examples are spike detection with adaptive thresholding, based on Non-linear Energy Operator (NEO) and envelope detection [30], [15]. These solutions are simple to implement and only require storage for the previous and next sample for all sensor nodes. Other solutions with limited or no temporary storage use a local energy measurement based on local sums as thresholds for spike detection [7]. However, these simple digital solutions typically have error rates higher than 30%.

In general, advanced digital solutions require additional temporal storage per sample for all electrodes using a solution based on wavelets, transforms, windowing and/or sorting. They are therefore taking too much area, or produce too much energy overhead. Analog solutions usually do not scale well compared to digital solutions, and they are also more complex to implement. Simple digital solutions have error rates above 30%. To our knowledge, only temporal spike detection is performed locally on the probe in existing solutions, where the spatial spike data analysis or selection is performed remotely.

Alternative approaches focus on data compression, combined with spike detection. These are considered as a separate root node in Figure 2. Zhang et al. [36] proposes a signal dependent compressed sensing (CS) approach used to compress detected spikes before they are transmitted off-chip. Their solution outperforms previous compression works in terms of compression rate and reconstruction quality. However, this solution is dependent on a dictionary matrix (D-matrix) created offline before the CS-circuit is activated for compression. The D-matrix is based on recorded raw signals that are transmitted off-chip. A dictionary matrix requires significant storage per electrode if different channels are compressed independently. Other compression solutions exist based on wavelet transforms ([19], [13]), and threshold crossing and windowing ([24]). Compression solutions based on wavelet transforms are able to achieve good compression rates while maintaining excellent signal reconstruction quality. On the other hand, these solutions require complex hardware (HW). Simpler compression solutions based on threshold crossing and windowing require less complex HW. For very high-density electrode arrays, any solution that requires buffering, such as windowing approaches, would increase the amount of required storage significantly. Reducing the amount of required temporal storage per sample is therefore a necessity in high-density electrode arrays.

Another approach is to record from only a subset of available electrodes with EDC [29]. This technique is already applied in recent probe designs (i.e. "electronic depth control" probes) that use a switching matrix to select electrodes of interest. This is a valid approach, since it is likely that some electrodes are not positioned close to neurons, not in brain regions of interest, or some electrodes may be noisy. This approach is good when the specific

region of interest is small enough. In cases where the region of interest is too large for continuous recording, a simple and more energy efficient solution is needed to meet the energy constraints. Such a solution must be scalable with increasing number of electrodes, and use a minimal amount of local storage per processed sample.

As mentioned, simple temporal spike detection solutions have high error rates. These error rates include both false positive and false negatives. From a data reduction perspective, it is desirable to reduce the false positives, but this usually comes at the cost of an increased number of false negatives. In contrast, if detecting as many spikes as possible is important, then it is desirable to reduce the false negatives, at the expense of higher false positive rates. The trade-off is usually controlled by the value of a threshold (or configurable scaling factor). The threshold selection depends on how the experimenter wants to weight the false positives and negatives. Hence, no single optimal threshold exists. Our focus is to select enough spike data locally for a successful remote post-processing, where the desired result is fine-tuned. Our proposed solution falls between the simple and advanced digital solution category. Hence, we optimize the energy consumption by adding only the required storage needed to preserve enough spike samples for successful remote post-processing.

3 GENERAL GOAL OF DATA REDUCTION APPROACHES

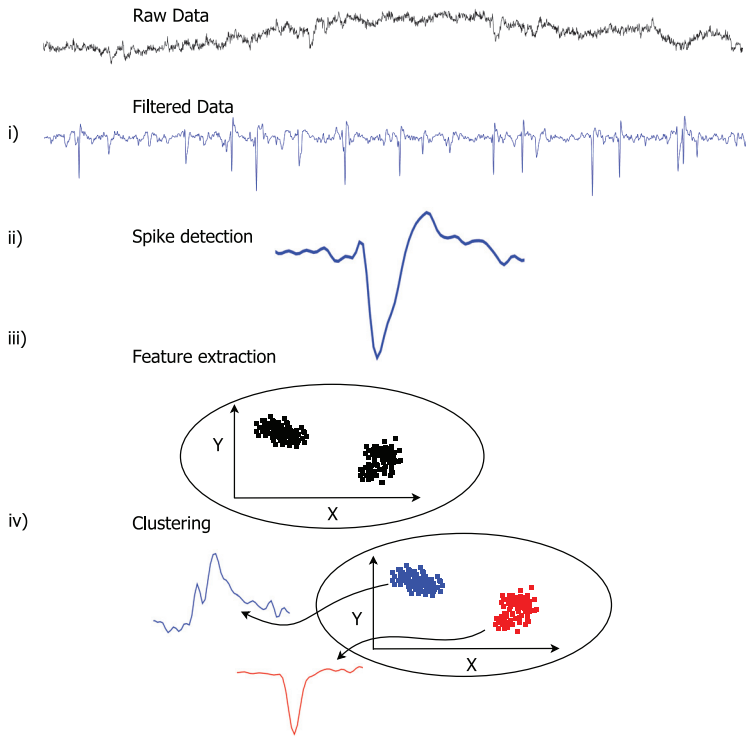


Fig. 3. Typical spike sorting overview as illustrated in [21].

The goal of data reduction approaches is to reduce the transmitted data while preserving all relevant information. What constitutes relevant information is application dependent. Spikes generated by individual neurons are the result of integration of input signals and transmitted to other neurons. For some applications it is sufficient to know the time of a spike regardless of the neuronal source, for example neural decoding approaches based on activities from multiple neurons (multi-unit activity) [27]. However, for most applications additional

information about the wave shape is needed, as this relates to the specific neuron that emitted the spike (different neurons emit spikes with different wave shapes). Wave shape features can be used to improve neural decoding approaches [14]. More often, waveform features are used to identify groups of spikes that originate from the same neuronal source ("spike sorting"; [21]). The spike sorting process relies on differences in intrinsic properties of neurons and their location relative to the electrodes, which are reflected in the spike amplitude and wave shape. After detection and extraction of spike waveforms from high-pass filtered (>300 Hz) raw signals, the sorting procedure is usually performed by deriving informative features from the waveforms (e.g. peak amplitude or components from principal component analysis, PCA[1]) across a set of electrodes, followed by a feature-based classification algorithm, as summarized in Figure 3. Step iii in Figure 3 shows an illustration where the x- and y-axis represents the first and second PCA components, respectively. In the final spike sorting step, similar features are clustered together with a label representing its cluster. Conventional approaches transmit the filtered data from the neural probe to a remote receiver, and process steps ii to iv remotely. The goal of our approach is to detect relevant spike waves (temporally and spatially) with minimal amount of resources on the neural probe. We then reduce the amount of data by only transmitting these waveforms to a remote receiver, where steps iii to iv are processed remotely.

4 WIRED AND WIRELESS DATA TRANSFER LIMITATIONS

Existing wired high-density electrode arrays with close to 1000 electrodes are able to transmit data in its raw form through wires connected to the probe shaft. As motivated in Section 1, the transmission circuitry in wired solutions contributes significantly to the power consumption, which will be even higher for wireless transfer. As an example, a probe with 120 electrodes is estimated to require between 50 and 58 Mbps to send all data in its raw form. This estimate is based on a 25 kHz sampling rate, using 16-bit samples. In our estimate we have assumed the coding overhead to be between 5% and 20% of the required bandwidth [18], i.e., the processing overhead required to prepare and transmit the data to the remote wireless receiver. Today's wired solutions have 2-5 meters cables running from the head stage that sits on the probe to the readout system (like a PC). The distance between the transmitter and the receiver is therefore assumed to be up to 5 m in our power consumption estimation for wireless data transmission.

Wireless low-power networks such as Personal Area Networks (PAN) and Body Area Networks (BAN) cannot handle the throughput requirements for 120 electrodes, because these networks are designed for lower bandwidths. To meet our bandwidth requirements we need to use networks similar to the wifi 802.11n or 802.11ac standards.

Halperin et al. [10] cites 1.28 W for the 802.11n transmitter in single-input and single-output (SISO) mode (which is barely sufficient for our 120 channel example). This power consumption is nearly independent of the exact data rate, unless data is buffered up and transmitted in chunks. Chunking can reduce the power consumption, but requires additional bandwidth overhead which is simply not possible in SISO mode with our bandwidth requirements.

The 802.11n multiple-input and multiple-output 3 (MIMO3) mode allows up to 405 Mbps for 2.1 W at the cost of multiple transmitters on-chip. In the most optimistic scenario (leading to a lower bound), the required power then becomes a fraction as shown in Equation 1, because the power consumption will depend on the data rate when it is buffered in chunks.

$$\text{Power consumption} = \frac{\text{data rate}}{405 \text{ Mbps}} \cdot 2.1 \text{ W} \quad (1)$$

Without any data reduction the estimated 50 to 58 Mbps transfer requires a power consumption of more than 1 W in SISO mode and 0.30 W in MIMO3 mode in addition to the buffering cost. If the amount of data is reduced 8 times, the power consumption for data transfer would become less than 40 mW.

Also more advanced BAN oriented wireless transceivers are being developed but in the end their power efficiency will still not remove the bandwidth-energy bottleneck fully and similar equations as Equation 1 will be valid, with other constants. We believe our general approach will hold also for those.

If the data is streamed with lower speed and bandwidth by means of buffering, we would have latency violations. Nevertheless, many applications have feed-forward processing so that added latency would be fine. However, a more severe limitation in our scope is that the probe does have very strictly limited space. This prevents us from implementing a reasonable sized buffer.

Figure 4 shows our envisioned platform where the electrodes are filtered through filters on the probe, before all samples are time-multiplexed into our custom ASIC. Other researchers have successfully implemented analog filters on the probe [16]. Data to be transmitted are then buffered up and transmitted through the radio circuitry. It should be noted though that the actual layout may change for future probe designs.

The size of the probe is too small to carry anything but a very compact ASIC, which is custom made as illustrated in Figure 4. To transmit the full data set out to an external implant will therefore not work for high density electrode arrays. Data elimination is hence essential as an initial stage for any other processing on external devices.

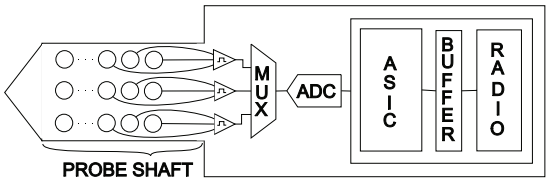


Fig. 4. Block scheme of our envisioned platform.

5 TEMPORAL AND SPATIAL SPIKE DATA EXTRACTION

We focus on spike data extraction, using low power and simple online digital circuitry. Samples around detected spikes are selected locally on the probe, and all other data are discarded before the selected samples are transmitted to a remote receiver for post-processing. This solution is scalable with increasing number of electrodes, and uses a minimal amount of local storage. Pruning away unneeded data results in large reductions in transmit energy, because we transmit significantly less data with minimal increase in processing/memory access energy only.

We have divided the spike data extraction in three consecutive phases as illustrated in Figure 5. All these phases are performed for each sample per electrode. In the initial phase, a temporal spike detection is performed using a crude amplitude thresholding mechanism. The second phase performs a spatial analysis of all detected spikes and marks neighboring samples for extraction. In the final phase, all samples marked for extraction are transmitted. All three phases are described in more detail in Sections 5.1, 5.2 and 5.3.

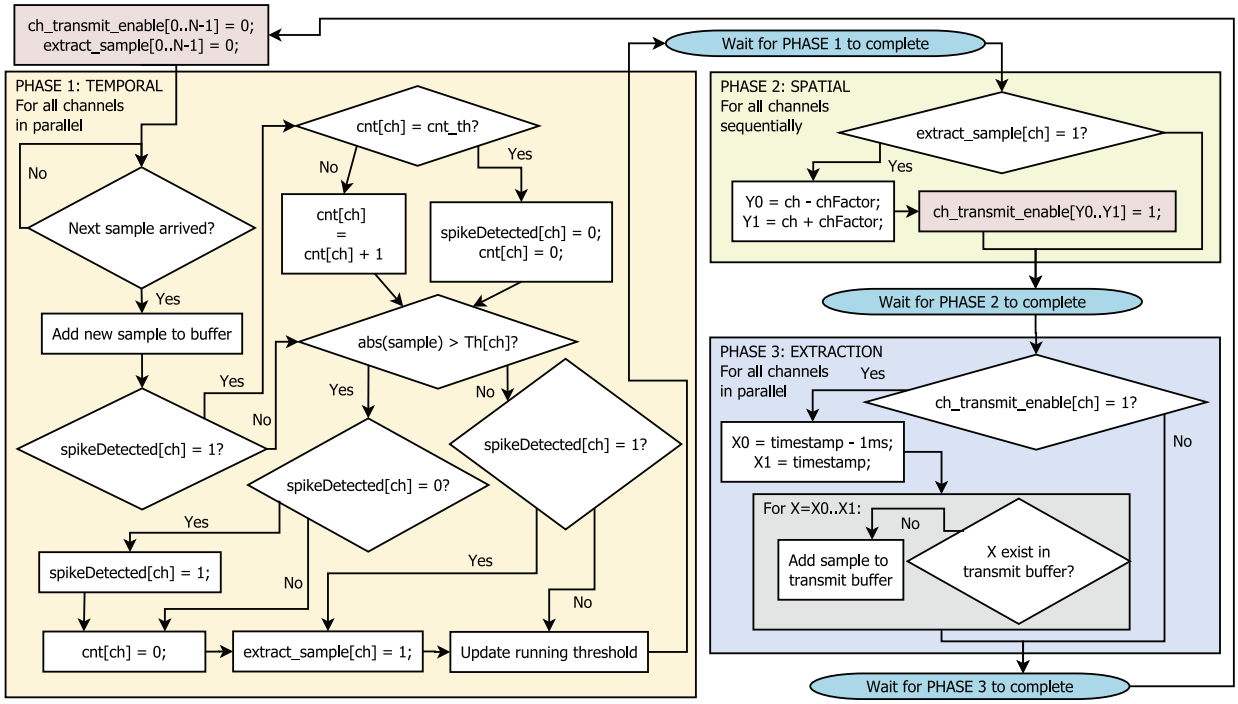


Fig. 5. 3-phase spike data extraction

5.1 Phase 1: Temporal spike detection

As motivated in Section 2, we use a threshold crossing method to detect spikes. Our adaptive threshold calculation is based on a simple amplitude threshold solution used by Jadhav et al. [12], [11], where they use five standard deviations above the mean as threshold.

In our approach, we use the mean (M), together with a factor (ThFactor) times the standard deviation (STD) of samples as a crude threshold for spike detection. Based on the stringent HW and energy requirements for wireless data transmission, the algorithm cannot afford to store more than a few temporary samples when the threshold is calculated locally on the probe. Other amplitude threshold spike detection solutions using a median calculation, such as the one surveyed by Rey et al. [23] and the absolute value method validated by Dragas et al. [7], requires additional storage to calculate the median value. Therefore, we implement a running mean and sample variance calculation algorithm, based on Welford's method for calculating the sum of squares [33], as our crude thresholding algorithm.

The sample variance algorithm is chosen simply because the underlying distribution is not known at run-time. The standard deviation is estimated using the square root of the bias corrected sample variance [32], which is shown in Equation 2. Equation 3 and 4 shows the calculation of the running mean and sum of squares, respectively. In the following equations, x is the absolute value of the sample, k is the sample number, V is the variance, S is the sum of squares, STD is the standard deviation, and M is the mean value.

$$STD_k = \sqrt{V_k}, \; V_k = \frac{S_k}{k-1} \tag{2}$$

$$M_k = M_{k-1} + \frac{1}{k}(x_k - M_{k-1}) \tag{3}$$

$$S_k = S_{k-1} + \frac{k-1}{k}(x_k - M_{k-1})(x_k - M_{k-1}) \tag{4}$$

Our method for calculating the sample variance is based on a one-pass updating algorithm with comparable accuracy to the two-pass method derived form the definition of variance. One-pass updating algorithms are also recommended by Chan et al. [5] when two-pass algorithms cannot be chosen due to storage limitations. Equation 3 can be re-structured as shown in Equation 5, and we use it to replace the $\frac{k-1}{k}(x_k - M_{k-1})$ part of Equation 4 with $(x_k - M_k)$, which is equivalent. We therefore use Equation 6 for S_k to simplify the HW implementation.

$$M_k = M_{k-1} + \frac{1}{k}(x_k - M_{k-1}) = \frac{1}{k}x_k + \frac{k-1}{k}M_{k-1} \tag{5}$$

$$S_k = S_{k-1} + (x_k - M_k)(x_k - M_{k-1}) \tag{6}$$

Our adaptive threshold mechanism calculates the running mean and standard deviation (STD), and uses the result to calculate the running threshold according to Equation 7. Compared to Jadhav et al. [12], [11], we replace the factor five with the configurable ThFactor. In addition, we use the running mean and STD formulas presented in this section.

$$Th_k = M_k + (ThFactor \cdot STD_k) \tag{7}$$

In Equation 7, Th is the running threshold, M is the running mean, STD is the standard deviation, and ThFactor is a configurable factor that is used to tune the sensitivity of the threshold. All samples with an absolute value above the threshold are marked for extraction by setting the variable extract_sample to 1 for this channel (Figure 5). At the same time the variable spikeDetected is set to 1. The calculation of Th_k and marking of samples is performed for all electrodes individually and simultaneously in parallel. When a threshold crossing is detected on the absolute value of a sample, the last millisecond of previously buffered samples are marked for extraction (if they are not marked from before), and a counter (cnt) starts counting for the inclusion of the next millisecond of samples. With a frame rate of 25 kHz, extracting 1 ms of samples (approx. 25 samples) before a threshold crossing with a buffer and after a threshold crossing (using a counter without additional buffering) is assumed to include sufficient samples around the spikes (i.e., peri-spike samples) for post-processing. This extraction of data is also a requirement in related work for post-processing. In Figure 5, the variable cnt_th is used to determine the length of this extraction period. spikeDetected is set to 0 as soon as the 1 ms period has passed.

From Equation 2 and 3, we can observe that new sample values have smaller impact on the running mean and standard deviation when k increases. Depending on the goal of the spike data analysis, the designer or neuroscience expert may want to reset the variance history periodically. We therefore include another tunable parameter called the Variance History Update (VHU), which resets the previous history of M and STD values (i.e., variance history). The Variance History Reset Interval (VHRI) is determined by the VHU parameter and the sampling frequency (FS) according to Equation 8, where FS is equivalent to 1 second of data samples (25k samples for 25 kHz sampling frequency). In other words, when k = VHRI then the variance history is reset. The VHU variable must be stored in a register, and the size of this register may vary depending on how often the designer needs to reset the variance history.

$$\text{VHRI} = \text{VHU} \cdot \text{FS} \tag{8}$$

Using up to 18-bits for VHU is sufficient to have variance history reset periods of more than 47 hours, given a k-register of 32-bits (the k-register wraps around to zero after 2^{32} samples at a rate of 25k samples/second, i.e., after 47.7 hours). If longer recording time is needed, then the VHU- and k-register must be adjusted accordingly. The VHU can be removed completely if no reset of the variance history is needed (the k-register must still be large enough).

5.2 Phase 2: Spatial spike data analysis

Phase 2 is a sequential phase that iterates through all electrodes one by one. If extract_sample is marked on an electrode channel, then it marks this sample for transmission by setting the variable ch_transmit_enable to 1 (Figure 5). A specific number (chFactor) of closely placed electrode channels surrounding the marked channel is also marked for transmission. Phase 1 has to be completed for all channels before Phase 2 can start, because the analysis of each electrode sample is dependent on the situation of all other electrode channels in the same time stamp. In other words, all channels are processed sequentially in Phase 2, where ch_transmit_enable is set for all channels that are close (in distance) to a channel with extract_sample marked (determined by the chFactor tuning parameter).

How the neighboring channels are selected is determined by a geometrical matrix unique for each probe design based on the placement of each electrode. It is up to the probe designer and neuroscience expert to determine which electrodes are neighbors to each other in the geometric matrix. The HW mechanism which selects the electrode neighbors should be as simple as possible because the sequential Phase 2 can become the bottleneck of the application if it takes too much time. One simple solution is to assign a unique channel id for each electrode, such that the neighboring electrodes can be marked by incrementing and decrementing the channel id according to the chFactor. In our experiments, we use electrode channel ids starting from 0 for the electrode at top of the probe and using ascending channel ids on probes placed below. The spatial extraction is then done according to PHASE 2 in Figure 5.

5.3 Phase 3: Spike data extraction

When all the samples to be transmitted are marked, our algorithm enters Phase 3. Phase 3 is a parallel phase where for all electrodes simultaneously it is checked if the corresponding ch_transmit_enable is marked. If it is marked, the previous 1 ms of samples (including the current sample) are added to a transmit buffer, unless they have already been added. After Phase 3 completes, the ch_transmit_enable and extract_sample variables in Figure 5 are reset for all channels.

6 ESTIMATED HW REQUIREMENTS

The HW requirements are estimated assuming there is one time-multiplexed arithmetic logic unit (ALU) available for processing, and that output buffers and other necessary logic are available. Our local storage requirement only considers the storage required to process the data from all electrodes, which is illustrated in Figure 6.

Three control bits are required per electrode to mark samples for extraction, detected spikes and to start a counter to include the future 1 ms samples. One global control bit is needed to transmit the samples marked for extraction. A 5-bit counter (supports up to

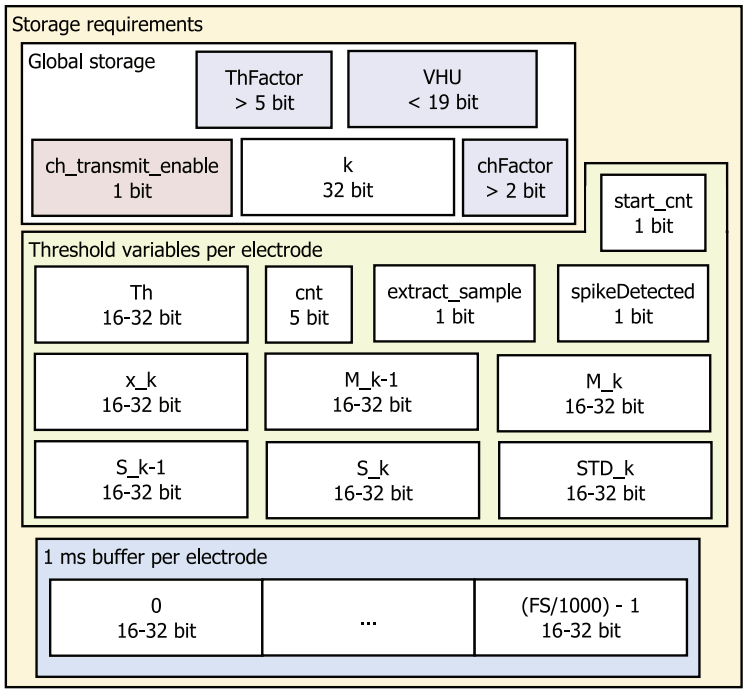


Fig. 6. Storage estimate

32 kHz sampling rate to count 1 ms of data) and a 1 ms buffer (25 samples for 25 kHz sampling rate) of 16- to 32-bit samples is required per electrode to include enough peri-spike samples before a spike is detected. A 16- to 32-bit threshold register is needed with six 16- to 32-bit variables for intermediate mean, STD and absolute value (ABS) variables, which are used to calculate the running means and STD values. All these registers are needed per electrode. 32-bit variables are only needed if high signal accuracy is required, but can be anywhere between 16 and 32 bit. More bits per sample will require more storage. In our estimate we consider either 16- or 32-bit samples. A global 32-bit register is needed for the k-register in order to support up to approximately 47 hours of neural recordings (if FS = 25 kHz) without variance history reset. This register can be smaller or larger depending on how long recordings the probe is designed for. Finally, we need to store the configurable tuning variables ThFactor (at least 6 bits), chFactor (at least 3 bits) and VHU (up to 18 bits) globally. If 16-bit variables are used for each sample, then 520 bits are needed per electrode, and 1032 bits per electrode if 32-bit variables are used. In addition, 60 bits are needed for the global transmit bit, ThFactor, chFactor, VHU and k parameters. The storage for each electrode requires the memory to be accessed concurrently, i.e., one port for each electrode if one bank of SRAM is used, which is not feasible unless one bank of SRAM is used for each electrode. 6T SRAMs requires less area than standard cell-based memories (i.e., memories consisting of standard cells only) [9]. However, Gemmeke et al. [9] show that although near threshold voltage (NTV) SRAMs have an area benefit, the use of cell-based memories have a better power benefit. Andersson et al. [2] makes the same conclusion, i.e., cell-based memories are up to 4x faster than SRAMs with orders of magnitude lower energy dissipation (at the cost of larger area per bit for large memory sizes only). Andersson et al. [2] also mention that standard cell memories scale with less effort than custom SRAMs.

In Table 1, we have estimated how many electrodes are supported for 16- and 32-bit samples per 5 mm^2 digital chip using a 40 nm process from Taiwan Semiconductor Manufacturing Company (TSMC) with and without 3D stacking. For simplicity we assume 6 bits for ThFactor, 3 bits for chFactor, 18 bits for VHU, and 32 bits for k in our estimation. The area requirement assumes one D edge-triggered flip-flop for 1-bit register, and 6 nand gates per flip-flop. 40 nm TSMC supports up to 2000K gates/mm^2.

Table 1. Supported number of electrodes using 40 nm cell-based memories

	16-bit samples	32-bit samples
5 mm^2 (TSMC)	3200	1610
3D stacking (TSMC)	6400	3220

The required area for ALU, output buffers and other logic are not included in the estimation, but are expected to be relatively small. For instance, in each electrode Phase 1 performs approximately 4-5 additions, 4 subtractions, 2 divisions and 3 multiplications per sample in addition to calculating a square root and absolute value. Therefore, a basic time-multiplexed ALU can be configured for a group of electrodes instead of having one ALU per electrode. The 1 ms buffer is intended to be a part of the transmit buffer, where we have assumed that up to 20% coding overhead is required for the transmit system (based on the discussion in Section 4), depending on what kind of wireless technology is used. Phase 2 and 3 perform significantly less calculations because these phases mostly enables control bits and transmits samples.

7 EXPERIMENTAL SETUP

Our 3-phase approach is tested on data obtained with a 128-channel silicon probe in the visual cortex of an awake mouse during the presentation of visual stimuli. Signals were digitized and stored using the Janelia WHISPER recording system (https://www.janelia.org/node/46162). Data were acquired at NERF using a 128-channel passive probe developed at IMEC. More recently, a newer passive probe with 966 electrodes and 384 configurable channels has been developed at IMEC [16]. These recordings are processed with our 3-phase approach in Matlab, where the size of the extracted data is compared relative to transmitting all samples, i.e., the reduction in transmitted data.

Our extracted spike data are also compared to the results from unsorted spike data after a flood fill algorithm [25] performed remotely offline on the data set. I.e., when spikes are detected the algorithm finds the mask for each spike using upper and lower threshold bounds. The method using the flood fill algorithm is a much more advanced method compared to our simpler method. The flood fill method is also not feasible for local low power implementation on the probe because of its complexity. Therefore we use the results from the flood fill algorithm method as our golden reference. It should be noted that even the flood fill algorithm results on our data set contain false positives and false negatives which are not annotated. Publicly available annotated data sets for spike detection only contain annotated data for one channel. Therefore, we are unable to validate our method with such data sets because our method performs spatial analysis across channels. To overcome this literature limitation, we use the flood fill algorithm results as our golden reference, and validate the quality of our algorithm relative to the flood fill algorithm results. We present our quality metrics in Section 8, before introducing the data reduction and quality results in Section 9.

The signal loss of our method (i.e., spike amplitudes below the threshold) will depend on what the focus of the experiment is. E.g., if detecting large spikes is a priority, then it could be considered that spikes below the threshold is not of any interest. This will in other words strongly depend on the neuroscientist analysing the data and the intended post processing steps. This aspect is thus very domain specific.

8 QUALITY METRICS

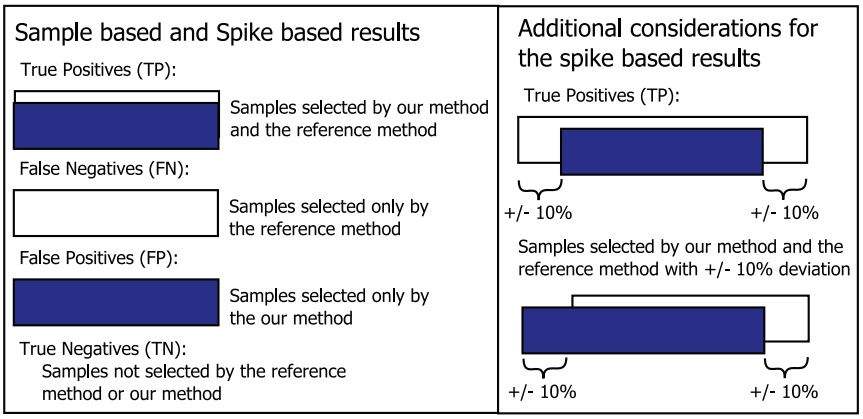


Fig. 7. Definitions for quality measurement.

We measure the quality of our method by collecting the amount of false positives (FP), false negatives (FN), true positives (TP) and true negatives (TN) compared to our reference algorithm. We use rate metrics based on receiver operation characteristics (ROC) analysis [8], [20]. How we define these parameters is illustrated in Figure 7. TPs are samples extracted by our method and the reference algorithm. FPs are samples extracted by our method and not the reference algorithm. TNs are samples that are not extracted by any of the methods. The FNs are the samples extracted by the reference method and not our method.

Lower FNs means higher sensitivity for our method compared to our reference algorithm. Equation 9 shows the sensitivity, i.e., the TP rate (TPR) as a function of TPs and FNs. TPR is a value between 0 and 1, where 1 means that there are no FNs, i.e., our method extracts all the samples that are extracted by the reference algorithm. The higher the TPR value, the better is the quality. Equation 9 also shows the FP rate (FPR). The FN rate (FNR) of our method and the precision is shown in Equation 10. The FPR and FNR metrics indicate the errors compared to our reference algorithm, having values between 0 and 1, where 0 is optimal. The precision of our method, i.e., the positive predictive value (PPV) indicates how much of the detected spikes are TPs (between 0 and 1, where 1 is optimal). A low PPV value therefore indicates that additional samples (FPs) are transmitted.

$$TPR = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \; FPR = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{9}$$

$$FNR = \frac{\mathrm{FN}}{\mathrm{TP} + \mathrm{FN}}, \; PPV = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{10}$$

The accuracy of our method, i.e., how well our method is able to find the TP and TN compared to the reference algorithm, is calculated according to Equation 11.

$$ACC = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}} \tag{11}$$

9 DATA REDUCTION AND QUALITY RESULTS

In Section 9.1, we present our data reduction results. The quality results of our method is presented in Section 9.2. All our results are discussed in Section 9.3.

9.1 Reduction of transmitted samples

The amount of extracted data is compared to the amount of raw data for two different threshold sensitivities. Figure 8 shows the data reduction in percent by using our methodology on 0.1 second (2500 samples) of the recorded data, averaged over different recording lengths up to 10 minutes.

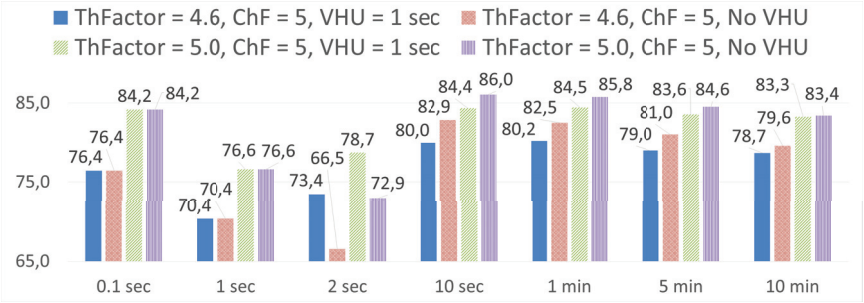


Fig. 8. Data reduction (%) of 0.1 second neural data averaged over various recording lengths up to 10 minutes.

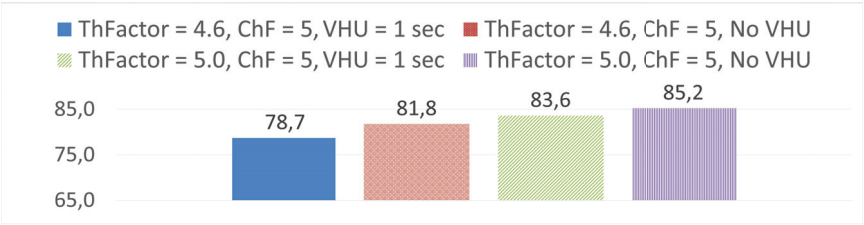


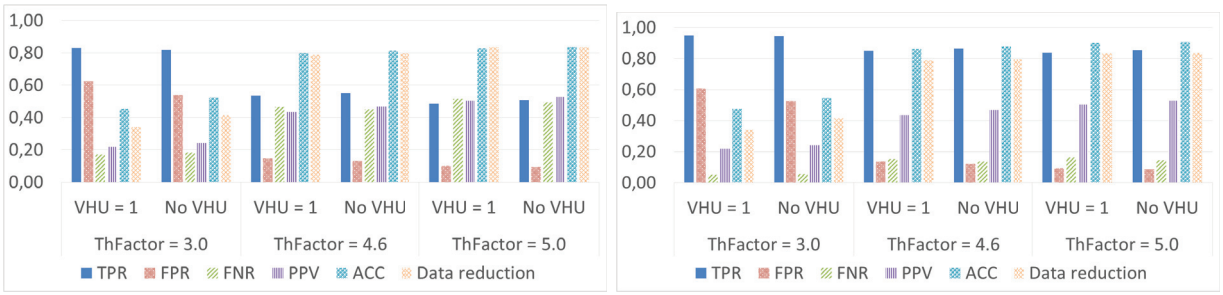
Fig. 9. Total data reduction (%) of a 55 minute recording.

The data reduction varies for short recording lengths, but converges towards approximately 78-80 % and 83-84% for our two ThFactor settings over longer recording lengths. This is confirmed by the results in Figure 9, which shows the total data reduction in percent of our complete 55 minute recording.

The VHU parameter does have an impact in reduced data when the variance history is reset for every recorded second compared to not resetting the variance history. The impact is however only about 3%.

We have estimated the BW requirement based on our results from the ten minute recording averaged over 0.1 second chuncks. If we assume 5-20% coding overhead to transmit the data, the required bandwith is between 10.3 Mbps and 11.8 Mbps (with ThFactor = 4.6, no VHU) or between 8.4 Mbps and 9.6 Mbps (with ThFactor = 5.0, no VHU). If a compression algorithm is used on top of our data reduction method, the bandwidth requirement would most likely decrease even further.

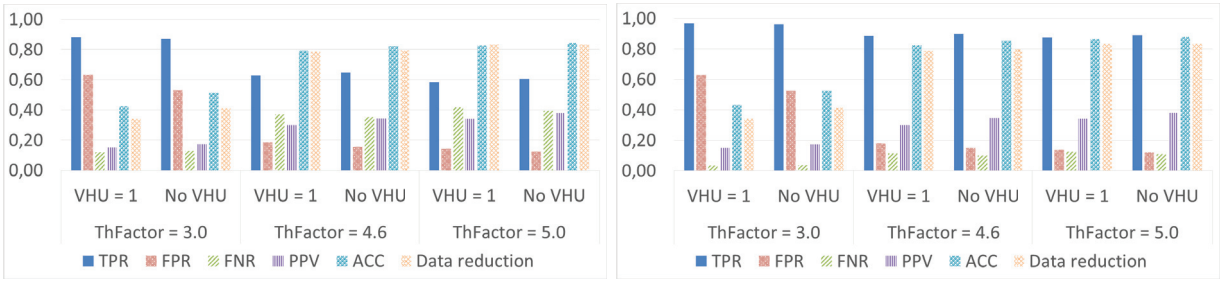
9.2 Quality results



(a) Sample-based results of our method on averaged 0.1 sec segments over a 10 minute recording.

(b) Same results as Figure 10a when ignoring reference spikes with amplitudes lower than 30% of Th.

Fig. 10. Sample-based quality results.



(a) Spike-based results of our method on averaged 0.1 sec segments over a 10 minute recording.

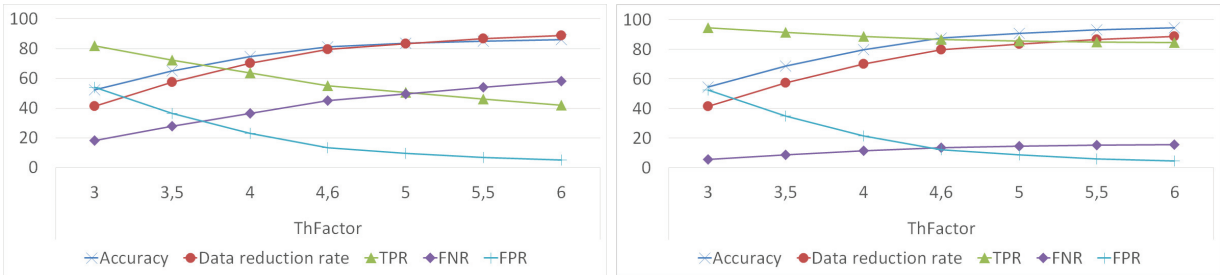
(b) Same results as Figure 11a when ignoring reference spikes with amplitudes lower than 30% of Th.

Fig. 11. Spike-based quality results.

We have measured the quality of our method by calculating the metrics described in Section 8 for averaged 0.1 second segments of the recorded data, over different recording lengths. The quality results of our method converges in a similar way as our data reduction results. We therefore present our quality results for averaged 0.1 second segments over a 10 minute recording in Figures 10 and 11.

These results have been calculated by counting samples (sample-based) and spikes (spike-based), where multiple spike time stamps within a 1 ms (25 samples) window (e.g., such as overlapping spikes) are considered as the same spike in the spike-based results. The results show that our method reduces the amount of data to be transmitted significantly at the expense of discarding spike data with amplitudes significantly lower than the threshold crossings. In other words, our method is able to detect all spikes that crosses the threshold, and the majority of smaller amplitude spikes that are not detected have amplitudes lower than 30% of the threshold. We confirm this observation by calculating the metrics presented in Section 8 where undetected spike samples (FNs) with amplitudes below 30% of our thresholds are ignored. These results are shown in Figure 10b and 11b, while Figure 10a and 11a show the results including all reference spikes. From Figures 10 and 11, we can observe minimal difference with or without the VHU factor. These figures also show that the FPR

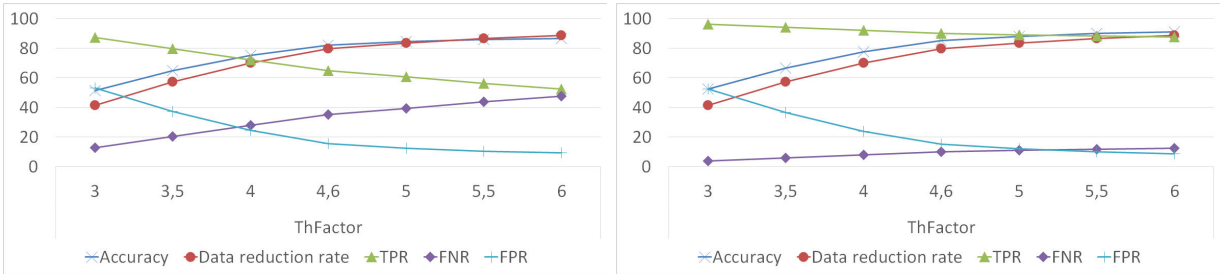
and TPR values are higher with lower ThFactor at the cost of lower accuracy and data reduction. For ThFactor around 4.6 and 5.0, we achieve a significantly improved accuracy and data reduction rate. However, if the ThFactor is too high, then the TPR decreases and the FNR increases as shown in Figures 12 and 13.



(a) Sample-based results of our method on averaged 0.1 sec segments over a 10 minute recording.

(b) Same results as Figure 10a when ignoring reference spikes with amplitudes lower than 30% of Th.

Fig. 12. Sample-based quality results with different ThFactor values.



(a) Spike-based results of our method on averaged 0.1 sec segments over a 10 minute recording.

(b) Same results as Figure 11a when ignoring reference spikes with amplitudes lower than 30% of Th.

Fig. 13. Spike-based quality results with different ThFactor values.

9.3 Discussion

Our methodology detects all the large spikes which the reference algorithm detects. Samples are discarded if they do not cross the threshold, and no nearby samples in time or on neighboring electrodes cross the threshold. The majority of all FNs have an amplitude below 30% of our threshold crossings, as shown in Figure 10b and 11b. The spike data marked by the reference program uses a more complex thresholding algorithm to detect spikes. Our FNs, which are a result of not having high enough amplitudes to cross our statistical threshold, are therefore assumed to be out of range for our coarse grained spike data extraction mechanism.

In Figures 10 and 11, the sensitivity (TPR) is between 0.55 and 0.65 when considering all FNs for ThFactor 4.6 and 5.0. Our ThFactor has to be decreased in order to increase the TPR, but at the expense of less data reduction (only up to 41%). Lowering ThFactor to 3.0 will also increase the FPR, and as a result reduce the precision (PPV) significantly. On the other hand, we obtain good TPR results when ignoring FNs below 30% of Th (TPR between 0.84 and 0.9 for ThFactor 4.6 and 5.0 in the sample-based and spike-based results).

When ignoring FNs below 30% of Th, we also obtain improved results in FPR (0.09-0.18), FNR (0.10-0.16), PPV (0.30-0.53) and ACC (0.82-0.91).

The total reduction in data to be transmitted for our recording, relative to transmitting all the raw data, is significant for ThFactor 4.6 and 5.0 (as shown in Figures 8 and 9). Compared to compression techniques combined with spike detection, these data reduction results are similar [36]. However, our solution is scalable with increasing number of electrodes and does not depend on training phases with all the raw data available. We expect that solutions requiring the raw data to be recordable for all electrodes simultaneously will not be feasible to implement (see specifications and constraints in Section 4 and 6) in future neuronal probes due to the stringent energy constraints. A compression solution can be added as a complimentary solution to our approach where our reduced data-set is compressed before it is transmitted.

If the very low amplitude spikes are critical for the post-processing analysis, then the threshold must be lowered accordingly until all the important spike samples are detected, at the expense of more FPs and less data reduction. Since the majority of our FNs have amplitudes below 30% of the thresholds, we conclude that the spike samples our method does not detect simply come from neurons that cannot be well isolated with a simple thresholding mechanism. A clear tradeoff exists between the desired reduction required vs discarding the FNs which are below the threshold. When energy efficiency is the bottleneck of the system (i.e., in wireless probes) then significantly high data reduction rates are important in order to reduce the energy consumption significantly, as discussed in Section 4 and shown in Equation 1.

The highest bandwidth requirement is 11.8 Mbps in our measured case with ThFactor equal to 4.8. This means that our method would (in average) buffer up less than 30 16-bit words for each iteration of our method. The analog frontend bandwidth used to collect all the raw data is in our case 48 Mbps (with 25 KHz sampling rate). Therefore, we assume that the on-chip communication bandwidth between the ASIC and the radio interface would not be a problem at 11.8 Mbps.

In general, our method reduces the amount of data needed to be transmitted significantly (up to 86%), while maintaining a high sensitivity when ignoring false negatives with very small amplitudes (below 30% of the threshold). Thus we conclude that the data transmit energy will be reduced significantly for high-density electrode probes when using our method. Since the HW required per electrode is independent of other electrodes, the required digital logic is scalable with increasing number of electrodes (at least up to 3200 for 40 nm TSMC).

10 CONCLUSION

We have proposed an energy efficient spike data extraction solution for high-density electrodes capable of reducing the data to be transferred by over 85%. Our proposal extracts spike data temporally and spatially locally on the probe, with minimal digital logic per electrode. Only the reduced data is transmitted to a remote receiver, making our proposal also applicable for wireless high-density electrode arrays. None of the spike detection solutions discussed in related work are able to achieve this significant reduction, while exhibiting an acceptable hardware overhead to enable local processing on the neural probe itself.

Spike samples for all the major spikes are extracted with our proposal, because the major spikes exceed our thresholds with good margin, and our spatial solution includes samples from all sub-threshold spikes in neighboring electrodes. Spikes in electrodes with no threshold crossings in their channels, nor any threshold crossings in nearby channels are considered

irrelevant. These small spikes can be acquired by reducing the spike detection threshold at the cost of low reduction of the amount of data transferred. These usually refer to neurons that cannot be well isolated and are not necessarily relevant depending on the experimental conditions.

Our method is able to obtain an accuracy rate of up to 91% compared to the results of an offline flood fill algorithm when our threshold is five standard deviations above the mean value of the signal amplitude (ThFactor = 5.0). We obtain in general good quality when ignoring spike data with amplitudes below 30% of the temporal thresholds (sensitivity = 0.85, precision = 0.53, false positive rate = 0.09 and false negative rate = 0.15 with ThFactor = 5.0).

The use of digital logic makes our proposal scalable with increasing number of electrodes. As an example, a 40nm TSMC process would support up to 3200 electrodes per 5 mm^2 digital chip and twice as much if 3D-stacking technology is used.

11 ACKNOWLEDGEMENTS

Fabian Kloosterman was supported by Research Project FWO G0D7516N.

REFERENCES

[1] Hervé Abdi and others. 2010. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics* 2, 4 (2010), 433–459. DOI:https://doi.org/10.1002/wics.101

[2] Oskar Andersson and others. 2016. A Wide-Operating Range Standard-Cell Based Memory in 28nm FD-SOI. (2016). http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:hbz:386-kluedo-43308

[3] Emilia Biffi and others. 2010. Development and validation of a spike detection and classification algorithm aimed at implementation on hardware devices. *Computational intelligence and neuroscience* 2010 (2010), 8.

[4] Hsiao-Lung Chan and others. 2008. Detection of neuronal spikes using an adaptive threshold based on the max–min spread sorting method. *Journal of Neuroscience Methods* 172, 1 (2008), 112–121.

[5] Tony F. Chan and others. 1983. Algorithms for Computing the Sample Variance: Analysis and Recommendations. *The American Statistician* 37, 3 (1983), 242–247. http://www.jstor.org/stable/2683386

[6] Guillaume Charvet and others. 2010. $BioMEA^{TM}$: A versatile high-density 3D microelectrode array system using integrated electronics. *Biosensors and Bioelectronics* 25, 8 (2010), 1889 – 1896. DOI:https://doi.org/10.1016/j.bios.2010.01.001

[7] Jelena Dragas and others. 2013. An unsupervised method for on-chip neural spike detection in multi-electrode recording systems. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, Osaka, Japan, 2535–2538.

[8] Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern Recognition Letters* 27, 8 (2006), 861 – 874. DOI:https://doi.org/10.1016/j.patrec.2005.10.010 ROC Analysis in Pattern Recognition.

[9] T. Gemmeke and others. 2014. Resolving the memory bottleneck for single supply near-threshold computing. In *2014 Design, Automation Test in Europe Conference Exhibition (DATE)*. ACM, Dresden, Germany, 1–6. DOI:https://doi.org/10.7873/DATE.2014.215

[10] Daniel Halperin and others. 2010. Demystifying 802.11 n power consumption. In *Proceedings of the 2010 international conference on Power aware computing and systems*. ACM, Vancouver, BC, Canada, 1.

[11] Shantanu P Jadhav and others. 2012. Awake hippocampal sharp-wave ripples support spatial memory. *Science* 336, 6087 (2012), 1454–1458.

[12] Shantanu P Jadhav and others. 2012. Supporting Online Material for Awake Hippocampal Sharp-wave Ripples Support Spatial Memory. (2012). www.sciencemag.org/cgi/content/full/science.1217230/DC1

[13] Awais M Kamboh and others. 2008. Analysis of lifting and B-spline DWT implementations for implantable neuroprosthetics. *Journal of Signal Processing Systems* 52, 3 (2008), 249–261.

[14] Fabian Kloosterman and others. 2014. Bayesian decoding using unsorted spikes in the rat hippocampus. *Journal of neurophysiology* 111, 1 (2014), 217–227.

[15] E. Koutsos and others. 2013. A 1.5 μW NEO-based spike detector with adaptive-threshold for calibration-free multichannel neural interfaces. In *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*. IEEE, Beijing, China, 1922–1925. DOI:https://doi.org/10.1109/ISCAS.2013.6572243

[16] C. M. Lopez and others. 2016. 22.7 A 966-electrode neural probe with 384 configurable channels in 0.13 μm SOI CMOS. In *2016 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, San Francisco, CA, USA, 392–393. DOI:https://doi.org/10.1109/ISSCC.2016.7418072

[17] Jens-Oliver Muthmann and others. 2015. Spike detection for large neural populations using high density multielectrode arrays. *Frontiers in neuroinformatics* 9 (2015), 21.

[18] Pouya Ostovari and others. 2014. *Network Coding Techniques for Wireless and Sensor Networks*. Springer Berlin Heidelberg, Berlin, Heidelberg, 129–162. DOI:https://doi.org/10.1007/978-3-642-40009-4_5

[19] Karim G Oweiss and others. 2007. A scalable wavelet transform VLSI architecture for real-time signal processing in high-density intra-cortical implants. *IEEE Transactions on Circuits and Systems I: Regular Papers* 54, 6 (2007), 1266–1278.

[20] David Martin Powers. 2011. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. (2011).

[21] Rodrigo Quian Quiroga. 2012. Spike sorting. *Current Biology* 22, 2 (2012), R45 – R46. DOI:https://doi.org/10.1016/j.cub.2011.11.005

[22] B. C. Raducanu and others. 2016. Time multiplexed active neural probe with 678 parallel recording sites. In *2016 46th European Solid-State Device Research Conference (ESSDERC)*. IEEE, Lausanne, Switzerland, 385–388. DOI:https://doi.org/10.1109/ESSDERC.2016.7599667

[23] Hernan Gonzalo Rey and others. 2015. Past, present and future of spike sorting techniques. *Brain Research Bulletin* 119, Part B (2015), 106 – 117. DOI:https://doi.org/10.1016/j.brainresbull.2015.04.007 Advances in electrophysiological data analysis.

[24] Alberto Rodriguez-Perez and others. 2012. A low-power programmable neural spike detection channel with embedded calibration and data compression. *IEEE transactions on biomedical circuits and systems* 6, 2 (2012), 87–100.

[25] Cyrille Rossant and others. 2016. *Spike sorting for large, dense electrode arrays*. Technical Report. Nature Publishing Group.

[26] Micha E Spira and others. 2013. Multi-electrode array technologies for neuroscience and cardiology. *Nature nanotechnology* 8, 2 (2013), 83–94.

[27] Eran Stark and others. 2007. Predicting movement from multiunit activity. *Journal of Neuroscience* 27, 31 (2007), 8387–8394.

[28] Nicholas V Swindale and others. 2015. Spike detection methods for polytrodes and high density microelectrode arrays. *Journal of computational neuroscience* 38, 2 (2015), 249–261.

[29] T. Torfs and others. 2011. Two-Dimensional Multi-Channel Neural Probes With Electronic Depth Control. *IEEE Transactions on Biomedical Circuits and Systems* 5, 5 (Oct 2011), 403–412. DOI:https://doi.org/10.1109/TBCAS.2011.2162840

[30] L. Traver and others. 2007. Adaptive-threshold neural spike detection by noise-envelope tracking. *Electronics Letters* 43, 24 (Nov 2007), 1333–1335. DOI:https://doi.org/10.1049/el:20071631

[31] Paul T Watkins and others. 2004. Validation of adaptive threshold spike detector for neural recording. In *Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE*, Vol. 2. IEEE, San Francisco, CA, USA, 4079–4082.

[32] Eric W. Weisstein. 2017. Variance. (2017). From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/Variance.html.

[33] B. P. Welford. 1962. Note on a method for calculating corrected sums of squares and products. (1962).

[34] Patrick D Wolf. 2008. Thermal considerations for the design of an implanted cortical brain–machine interface (BMI). (2008). https://www.ncbi.nlm.nih.gov/books/NBK3932/

[35] Yuning Yang and others. 2010. Adaptive threshold spike detection using stationary wavelet transform for neural recording implants. In *2010 Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, Paphos, Cyprus, 9–12.

[36] J. Zhang and others. 2014. An Efficient and Compact Compressed Sensing Microsystem for Implantable Neural Recordings. *IEEE Transactions on Biomedical Circuits and Systems* 8, 4 (Aug 2014), 485–496. DOI:https://doi.org/10.1109/TBCAS.2013.2284254