Fast Nonnegative Deconvolution for Spike Train Inference From Population Calcium Imaging

Joshua T. Vogelstein,1 Adam M. Packer,2,3 Timothy A. Machado,2,3 Tanya Sippy,2,3 Baktash Babadi,4 Rafael Yuste,2,3 and Liam Paninski4,5

1Department of Neuroscience, Johns Hopkins University, Baltimore, Maryland; 2Howard Hughes Medical Institute, Chevy Chase, Maryland; 3Department of Biological Sciences, 4Center for Theoretical Neuroscience, and 5Department of Statistics, Columbia University, New York, New York

Submitted 9 December 2009; accepted in final form 3 June 2010

Vogelstein JT, Packer AM, Machado TA, Sippy T, Babadi B, Yuste R, Paninski L. Fast nonnegative deconvolution for spike train inference from population calcium imaging. J Neurophysiol 104: 3691–3704, 2010. First published June 16, 2010; doi:10.1152/jn.01073.2009. Fluorescent calcium indicators are becoming increasingly popular as a means for observing the spiking activity of large neuronal populations. Unfortunately, extracting the spike train of each neuron from a raw fluorescence movie is a nontrivial problem. This work presents a fast nonnegative deconvolution filter to infer the approximately most likely spike train of each neuron, given the fluorescence observations. This algorithm outperforms optimal linear deconvolution (Wiener filtering) on both simulated and biological data. The performance gains come from restricting the inferred spike trains to be positive (using an interior-point method), unlike the Wiener filter. The algorithm runs in linear time, and is fast enough that even when simultaneously imaging >100 neurons, inference can be performed on the set of all observed traces faster than real time. Performing optimal spatial filtering on the images further refines the inferred spike train estimates. Importantly, all the parameters required to perform the inference can be estimated using only the fluorescence data, obviating the need to perform joint electrophysiological and imaging calibration experiments.

INTRODUCTION

Simultaneously imaging large populations of neurons using calcium sensors is becoming increasingly popular (Yuste and Katz 1991; Yuste and Konnerth 2005), both in vitro (Ikegaya et al. 2004; Smetters et al. 1999) and in vivo (Göbel and Helmcchen 2007; Luo et al. 2008; Nagayama et al. 2007), and will likely continue to improve as the signal-to-noise ratio (SNR) of genetic sensors continues to improve (Garaschuk et al. 2007; Mank et al. 2008; Wallace et al. 2008). Whereas the data from these experiments are movies of time-varying fluorescence intensities, the desired signal consists of spike trains of the observable neurons. Unfortunately, finding the most likely spike train is a challenging computational task, due to limitations on the SNR and temporal resolution, unknown parameters, and analytical intractability.

A number of groups have therefore proposed algorithms to infer spike trains from calcium fluorescence data using very different approaches. Early approaches simply thresholded \(dF/F\) [typically defined as \((F - F_b)/F_b\), where \(F_b\) is baseline fluorescence; e.g., Mao et al. 2001; Schwartz et al. 1998] to obtain “event onset times.” More recently, Greenberg et al. (2008) developed a dynamic programming algorithm to identify individual spikes. Holekamp et al. (2008) then applied an optimal linear deconvolution (i.e., the Wiener filter) to the fluorescence data. This approach is natural from a signal processing standpoint, but does not realize the knowledge that spikes are always positive. Sasaki et al. (2008) proposed using machine learning techniques to build a nonlinear supervised classifier, requiring many hundreds of examples of joint electrophysiological and imaging data to “train” the algorithm to learn what effect spikes have on fluorescence. Vogelstein and colleagues (2009) proposed a biophysical model-based sequential Monte Carlo (SMC) method to efficiently estimate the probability of a spike in each image frame, given the entire fluorescence time series. Although effective, that approach is not suitable for on-line analyses of populations of neurons because the computations run in about real time per neuron (i.e., analyzing 1 min of data requires about 1 min of computational time on a standard laptop computer).

In the present work, a simple model is proposed relating spiking activity to fluorescence traces. Unfortunately, inferring the most likely spike train, given this model, is computationally intractable. Making some reasonable approximations leads to an algorithm that infers the approximately most likely spike train, given the fluorescence data. This algorithm has a few particularly noteworthy features, relative to other approaches. First, spikes are assumed to be positive. This assumption often improves filtering results when the underlying signal has this property (Cunningham et al. 2008; Huys et al. 2006; Lee and Seung 1999; Lin et al. 2004; Markham and Conchello 1999; O’Grady and Pearlmutter 2006; Paninski et al. 2009; Portugal et al. 1994). Second, the algorithm is fast: it can process a calcium trace from 50,000 images in about 1 s on a standard laptop computer. In fact, filtering the signals for an entire population of \(>100\) neurons runs faster than real time. This speed facilitates using this filter on-line, as observations are being collected. In addition to these two features, the model may be generalized in a number of ways, including incorporating spatial filtering of the raw movie, which can improve effective SNR. The utility of the proposed filter is demonstrated on several biological data sets, suggesting that this algorithm is a powerful and robust tool for on-line spike train inference. The code (which is a simple Matlab script) is available for free download from http://www.optophysiology.org.

METHODS

Data-driven generative model

Figure 1 shows data from a typical in vitro epifluorescence experiment (for data collection details see Experimental methods...
later in this section). The top panel shows the mean frame of this movie, including four neurons, three of which are patched. To build the model, the pixels within a region of interest (ROI) are selected (white circle). Given the ROI, all the pixel intensities of each frame can be averaged, to get a one-dimensional fluorescence time series, as shown in the bottom left panel (black line). By patching onto this neuron, the spike train can also be directly observed (black bars; bottom left). Previous work suggests that this fluorescence signal might be well characterized by convolving the spike train with an exponential and adding noise (Yuste and Konnerth 2005). This model is confirmed by convolving the true spike train with an exponential (gray line; bottom left) and then looking at the distribution of the residuals. The bottom right panel shows a histogram of the residuals (dashed line) and the best-fit Gaussian distribution (solid line).

The preceding observations may be formalized as follows. Assume there is a one-dimensional fluorescence trace $F$ from a neuron [throughout this text $X$ indicates the vector $(X_1, \ldots, X_T)$, where $T$ is the index of the final frame]. At time $t$, the fluorescence measurement $F_t$ is a linear-Gaussian function of the intracellular calcium concentration at that time $[Ca^{2+}]_t$:

$$F_t = \alpha[Ca^{2+}]_t + \beta + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2). \quad (1)$$

The parameter $\alpha$ absorbs all experimental variables influencing the scale of the signal, including the number of sensors within the cell, photons per calcium ion, amplification of the imaging system, and so on. Similarly, the offset $\beta$ absorbs, for example, the baseline calcium concentration of the cell, background fluorescence of the fluorophore, and imaging system offset. The noise at each time $\epsilon_t$ is independently and identically distributed according to a normal distribution with zero mean and $\sigma^2$ variance, as indicated by the notation $\sim \mathcal{N}(0, 1)$. This noise results from calcium fluctuations independent of spiking activity, fluorescence fluctuations independent of calcium, and other sources of imaging noise.

Then, assuming that the intracellular calcium concentration $[Ca^{2+}]_t$ jumps by $A \, \mu M$ after each spike and subsequently decays back down to baseline $C_b \, \mu M$, with time constant $\tau$ s, one can write:

$$[Ca^{2+}]_{t+1} = (1 - \Delta/\tau)[Ca^{2+}]_t + (\Delta/\tau)C_b + An_t. \quad (2)$$

where $\Delta$ is the time step size—which is the frame duration, or $1/($frame rate$)$—and $n_t$ indicates the number of times the neuron spiked in frame $t$. Note that because $[Ca^{2+}]_t$ and $F_t$ are linearly related to one another, the fluorescence scale $\alpha$ and calcium scale $A$ are not identifiable. In other words, either can be set to unity without loss of generality because the other can absorb the scale entirely. Similarly, the fluorescence offset $\beta$ and calcium baseline $C_b$ are not identifiable, so either can be set to zero without loss of generality. Finally, letting $\gamma = (1 - \Delta/\tau)$, Eq. 2 can be rewritten by replacing $[Ca^{2+}]_t$ with its nondimensionalized counterpart $C_t$:

$$C_t = \gamma C_{t-1} + n_t. \quad (3)$$

Note that $C_t$ does not refer to absolute intracellular concentration of calcium, but rather, a relative measure (for a more general model see Vogelstein et al. 2009). The gray line in the bottom left panel of Fig. 1 corresponds to the putative $C$ of the observed neuron.

To complete the “generative model” (i.e., a model from which simulations can be generated), the distribution from which spikes are sampled must be defined. Perhaps the simplest first-order description of spike trains is that at each time, spikes are sampled according to a Poisson distribution with some rate:

$$n_t \sim \text{Poisson} (\lambda \Delta) \quad (4)$$

where $\lambda \Delta$ is the expected firing rate per bin and $\Delta$ is included to ensure that the expected firing rate is independent of the frame rate. Thus Eqs. 1, 3, and 4 complete the generative model.

**Goal**

Given the above model, the goal is to find the maximum a posteriori (MAP) spike train, i.e., the most likely spike train $\hat{n}$, given the fluorescence measurements, $F$:

$$\hat{n} = \arg\max_{n \in \mathbb{N}_0} P(n|F),$$

where $P(n|F)$ is the posterior probability of a spike train $n$, given the fluorescent trace $F$, and $n_t$ is constrained to be an integer $\mathbb{N}_0 = \{0, 1,$...
Inferring the approximately most likely spike train, given a fluorescence trace

The goal here is to develop an algorithm to efficiently approximate \( \hat{n} \), the most likely spike train given the fluorescence trace. Because of the intractability described earlier, one can approximate Eq. 4 by replacing the Poisson distribution with an exponential distribution of the same mean (note that potentially more accurate approximations are possible, as described in the DISCUSSION). Modifying Eq. 10 to incorporate this approximation yields:

\[
\hat{n} = \arg\max \prod_{n_i > 0} \prod_{t=1}^{T} \left[ \frac{1}{2\sigma^2 n_i!} \exp\left\{ -\frac{1}{2} (F_t - \alpha C_t - \beta)^2 \right\} \exp\left\{ -\lambda \Delta \right\} \right] \tag{11a}
\]

\[
\hat{C} = \arg\max \sum_{c_{t-1}} \left[ -\frac{1}{2\sigma^2} (F_t - \alpha C_t - \beta)^2 - (C_t - \gamma C_{t-1})\lambda \Delta \right] \tag{12}
\]

where the second equality follows from taking the log of the right-hand side of Eq. 9, which is twice analytically differentiable, one can approximate solve for \( \hat{n} \) and \( \hat{C} \). Unfortunately, solving Eq. 11b exactly is analytically intractable because it requires a nonlinear search over an infinite number of possible spike trains. The search space could be restricted by imposing an upper bound \( k \) on the number of spikes within a frame. However, in that case, the computational complexity scales exponentially with the number of image frames—i.e., the number of computations required would scale with \( k^T \)—which for pragmatic reasons is intractable.

Inferring the approximately most likely spike train, given a fluorescence trace

The goal here is to develop an algorithm to efficiently approximate \( \hat{n} \), the most likely spike train given the fluorescence trace. Because of the intractability described earlier, one can approximate Eq. 4 by replacing the Poisson distribution with an exponential distribution of the same mean (note that potentially more accurate approximations are possible, as described in the DISCUSSION). Modifying Eq. 10 to incorporate this approximation yields:

\[
\hat{n} = \arg\max \prod_{n_i > 0} \prod_{t=1}^{T} \left[ \frac{1}{2\sigma^2 n_i!} \exp\left\{ -\frac{1}{2} (F_t - \alpha C_t - \beta)^2 \right\} \exp\left\{ -\lambda \Delta \right\} \right] \tag{11a}
\]

\[
\hat{C} = \arg\max \sum_{c_{t-1}} \left[ -\frac{1}{2\sigma^2} (F_t - \alpha C_t - \beta)^2 - (C_t - \gamma C_{t-1})\lambda \Delta \right] \tag{12}
\]

where the second equality follows from taking the log of the right-hand side of Eq. 9, which is twice analytically differentiable, one can approximate solve for \( \hat{n} \) and \( \hat{C} \). Unfortunately, solving Eq. 11b exactly is analytically intractable because it requires a nonlinear search over an infinite number of possible spike trains. The search space could be restricted by imposing an upper bound \( k \) on the number of spikes within a frame. However, in that case, the computational complexity scales exponentially with the number of image frames—i.e., the number of computations required would scale with \( k^T \)—which for pragmatic reasons is intractable.

Inferring the approximately most likely spike train, given a fluorescence trace

The goal here is to develop an algorithm to efficiently approximate \( \hat{n} \), the most likely spike train given the fluorescence trace. Because of the intractability described earlier, one can approximate Eq. 4 by replacing the Poisson distribution with an exponential distribution of the same mean (note that potentially more accurate approximations are possible, as described in the DISCUSSION). Modifying Eq. 10 to incorporate this approximation yields:

\[
\hat{n} = \arg\max \prod_{n_i > 0} \prod_{t=1}^{T} \left[ \frac{1}{2\sigma^2 n_i!} \exp\left\{ -\frac{1}{2} (F_t - \alpha C_t - \beta)^2 \right\} \exp\left\{ -\lambda \Delta \right\} \right] \tag{11a}
\]

\[
\hat{C} = \arg\max \sum_{c_{t-1}} \left[ -\frac{1}{2\sigma^2} (F_t - \alpha C_t - \beta)^2 - (C_t - \gamma C_{t-1})\lambda \Delta \right] \tag{12}
\]
Learning the parameters

In practice, the model parameters \( \theta = \{ \alpha, \beta, \sigma, \gamma, \lambda \} \) tend to be unknown. An algorithm to estimate the most likely parameters \( \hat{\theta} \) could proceed as follows: 1) initialize some estimate of the parameters \( \hat{\theta} \), then 2) recursively compute \( \hat{n} \) using those parameters and update \( \hat{\theta} \) given the new \( \hat{n} \) until some convergence criterion is met. This approach may be thought of as a pseudoexpectation-maximization algorithm (Dempster et al. 1977; Vogelstein et al. 2009). In the following text, details are provided for each step.

INITIALIZING THE PARAMETERS. Because the model introduced earlier is linear, the scale of \( F \) relative to \( \hat{n} \) is arbitrary. Therefore before filtering, \( F \) is linearly mapped between zero and one, i.e., \( F \leftarrow (F - F_{\text{min}})/(F_{\text{max}} - F_{\text{min}}) \), where \( F_{\text{min}} \) and \( F_{\text{max}} \) are the observed minimum and maximum of \( F \), respectively. Given this normalization, \( \alpha \) is set to one. Because spiking is sparse in many experimental settings, \( F \) tends to be around baseline, so \( \beta \) is initialized to be the median of \( F \) and \( \sigma \) is initialized as the median absolute deviation of \( F \), i.e., \( \sigma = \text{median}(|F|)\). Ideally, one could integrate out the hidden variables, to find the most likely parameters:

\[
\hat{n} = \arg\max_{\theta} P(F, C | \theta) dC = \arg\max_{\theta} P(F | C; \theta) P(C | \theta) dC.
\]

However, evaluating those integrals is not currently tractable. Therefore Eq. 17 is approximated by simply maximizing the parameters given the MAP estimate of the hidden variables:

\[
\hat{n} = \arg\max_{\theta} P(F, \hat{C} | \theta) = \arg\max_{\theta} P(F | \hat{C}; \theta) P(\hat{C} | \theta).
\]

where \( \hat{C} \) and \( \hat{n} \) are determined using the above-described inference algorithm. The approximation in Eq. 18 is good whenever most of the mass in the integral in Eq. 18 is around the MAP sequence \( \hat{C} \).

Note that the right-hand side of Eq. 19 decouples \( \lambda \) from the other parameters. The maximum likelihood estimate (MLE) for the observation parameters \( \{\alpha, \beta, \sigma\} \) is therefore given by:

\[
\{\hat{\alpha}, \hat{\beta}, \hat{\sigma}\} = \arg\max_{a, b, \sigma} \sum_{i=1}^{T} \ln P(F_i | \hat{C}_i; \alpha, \beta, \sigma) = \arg\max_{a, b, \sigma} -\frac{1}{2} (2\pi \sigma^2)^{-1} - \frac{1}{2} \left( F_i - \alpha \hat{C}_i - \beta \right)^2.
\]
\[
\hat{\sigma} = \sqrt{\frac{1}{T} \sum (F_t - \hat{C}_t)^2},
\]
which is simply the root-mean-square of the residual error. Finally, the MLE of \(\lambda\) is given by solving:
\[
\hat{\lambda} = \arg\max_{\lambda > 0} \sum \ln(\lambda \Delta) - \hat{\sigma}_i \lambda |\Delta|,
\]
where, again, computing the gradient with respect to \(\lambda\), setting it to zero, and solving for \(\hat{\lambda}\), yields \(\lambda = \tilde{T}(\Delta \sum \hat{\beta}_i)\), which is the inverse of the inferred average firing rate.

Iterations stop whenever 1) the iteration number exceeds some upper bound or 2) the relative change in likelihood does not exceed some lower bound. In practice, parameter estimates tend to converge after several iterations, given the above initializations.

**Spatial filtering**

In the preceding text, we assumed that the raw movie of fluorescence measurements collected by the experimenter had undergone two stages of preprocessing before filtering. First, the movie was segmented, to determine ROIs, yielding a vector \(F_r = (F_{1r}, \ldots, F_{Nr})\), which corresponded to the fluorescence intensity at time \(t\) for each of the \(N_r\) pixels in the ROI (note that we use the \(X\) throughout to indicate row vectors in space vs. \(X\) to indicate column vectors in time). Second, at each time \(t\), that vector was projected into a scalar, yielding \(F_r\), the assumed input to the filter. In this section, the optimal projection is determined by considering a more general model:

\[
F_{ix} = \alpha_i C_i + \beta_i + s_{ex,ix}, \quad e_{ix} \sim N(0,1),
\]

where \(\alpha_i\) corresponds to the number of photons that are contributed due to calcium fluctuations \(C_i\), and \(\beta_i\) corresponds to the static photon emission at pixel \(x\). Further, the noise is assumed to be both spatially and temporally white, with standard deviation (SD) \(\sigma\), in each pixel (this assumption can always be approximately accurate by prewhitening; alternately, one could relax the spatial independence by representing joint noise over all pixels with a covariance matrix \(\Sigma_x\), with arbitrary structure). Performing inference in this more general model proceeds in a nearly identical manner as before. In particular, the maximization, gradient, and Hessian become:

\[
\hat{C}_x = \arg\max_{MC \geq 0} -\frac{1}{2\sigma^2} \|\hat{F} - \hat{C}_x - 1\|_F^2 - (MC)^T \lambda + z \ln |(MC)^T|\]
\[
g = (\hat{F} - \hat{C}_x - 1\|_F^2 - (MC)^T \lambda + z |MC)^T|^{-1}\]
\[
H = -\frac{\hat{C}_x^T}{\sigma^2} - (MC)^T M.
\]

where \(\hat{F}\) is an \(N_r \times T\) element matrix, \(I_x\) is a column vector of ones with length \(T\), \(I\) is an \(N_r \times N_r\) identity matrix, and \(\|x\|_F^2 = \sum x_i^2\), and the exponents and log operator on the vector \(MC\) again indicate element-wise operations. Note that to speed up computation, one can first project the background subtracted \((N_r \times T)\)-dimensional movie onto the spatial filter \(\hat{C}\), yielding a one-dimensional time series \(\hat{F}\), reducing the problem to evaluating a \(T \times 1\) vector norm, as in Eq. 15.

The parameters \(\alpha\) and \(\beta\) tend to be unknown and thus must be estimated from the data. Following the strategy developed in the previous section, we first initialize the parameters. Because each voxel contains some number of fluorophores, which sets both the baseline fluorescence and the fluorescence due to calcium fluctuations, let both the initial spatial filter and initial background be the median image frame, i.e., \(\hat{\alpha}_x = \hat{\beta}_x = \text{median}(F_{ix})\). Given these robust initializations, the maximum likelihood estimator for each \(\alpha_x\) and \(\beta_x\) is given by:

\[
\arg\max_{\alpha,\beta} \left\{ \alpha_x, \beta_x \right\} = \arg\max_{\alpha,\beta} P[F_x|\hat{C}_x]
\]

\[
= \arg\max_{\alpha,\beta} \sum \ln P[F_x|\hat{C}_x]
\]

\[
= \arg\max_{\alpha,\beta} \left\{ -\frac{1}{2} \ln(2\pi \sigma^2) - \frac{1}{2\sigma^2} (F_{ix} - \alpha_i \hat{C}_i - \beta_i)^2 \right\}
\]

\[
= \arg\max_{\alpha,\beta} -\sum (F_{ix} - \alpha_i \hat{C}_i - \beta_i)^2,
\]

where the first equalities follow from Eq. 1 and the last equality follows from dropping irrelevant constants. Because this is a standard linear regression problem, let \(A = [\hat{C}_i, 1] I^T\) be a \(2 \times T\) element matrix and \(Y_i = [\alpha_i, \beta_i]^T\) be a \(2 \times 1\) element column vector. Substituting \(A\) and \(Y_i\) into Eq. 28d yields:

\[
\hat{Y}_i = \arg\max_{Y_i} -\|F_i - A^T Y_i\|_2^2.
\]

which can be solved by computing the derivative of Eq. 29 with respect to \(Y_i\) and setting to zero, or using Matlab notation: \(\hat{Y}_i = A^T (A A^T)^{-1} F_i\). Note that solving \(N_r\) two-dimensional quadratic problems is more efficient than solving a single \((2 \times N_r)\)-dimensional quadratic problem. Also note that this approach does not regularize the parameters at all, by smoothing or sparsening, for instance. In the discussion we propose several avenues for further development, including the elastic net (Zou and Hastie 2005) and simple parametric models of the neuron. As in the scalar form case, we iterate estimating the parameters of this model \(\theta = [\hat{\alpha}_i, \hat{\beta}_i, \hat{\sigma}, \hat{\gamma}, \hat{\lambda}]\) and the spike train \(n\). Because of the free scale term discussed earlier, the absolute magnitude of \(\hat{\alpha}\) is not identifiable. Thus convergence is defined here by the “shape” of the spike train converging, i.e., the norm of the difference between the inferred spike trains from subsequent iterations, both normalized such that \(max(n_i) = 1\). In practice, this procedure converged after several iterations.

**Overlapping spatial filters**

It is not always possible to segment the movie into pixels containing only fluorescence from a single neuron. Therefore the above-cited model can be generalized to incorporate multiple neurons within an ROI. Specifically, letting the superscript \(i\) index the \(N_c\) neurons in this ROI yields:

\[
\hat{F}_i = \sum_{m=1}^{N_c} \hat{F} C_i + \hat{\beta} + \hat{e}_i, \quad \hat{e}_i \sim N(0,\sigma^2 I)
\]

\[
\hat{C}_i = \hat{\gamma} C_{i-1} + n_i, \quad n_i \sim \text{Poisson}(n_i; \lambda, \Delta)
\]

where each neuron is implicitly assumed to be independent and each pixel is conditionally independent and identically distributed with variance \(\sigma^2\), given the underlying calcium signals. To perform inference in this more general model, let \(n = [n_1, \ldots, n_c]\) and \(C_i = [C_{i1}, \ldots, C_{ic}]\) be \(N_c\)-dimensional column vectors. Then, let \(\Gamma = \text{diag}(\hat{\gamma}, \ldots, \hat{\gamma})\) be an \(N_c \times N_c\) diagonal matrix and let \(I\) and \(0\) be an identity and zero matrix of the same size, respectively, yielding:
and proceed as before. Note that Eq. 32 is very similar to Eq. 14, except that $M$ is no longer bidiagonal, but rather, block bidiagonal (and $C_i$ and $n_i$ are vectors instead of scalars), making the Hessian block-tridiagonal. Importantly, the Thomas algorithm, which is a simplified form of Gaussian elimination, finds the solution to linear equations with block-tridiagonal matrices in linear time, so the efficiency gained from using the tridiagonal structure is maintained for this block-tridiagonal structure (Press et al. 1992). Performing inference in this more general model proceeds similarly as before, letting $\bar{a} = [\bar{a}_1, \ldots, \bar{a}_n]$:

$$
\hat{C}_i = \max_{MC \in \mathbb{R}} -\frac{1}{2\sigma^2} \left\| \tilde{F} - C\hat{\alpha} - 1_f \beta_\gamma^2 - (MC)^T \lambda + z \ln \sigma (MC)^T 1, 
\right.$$

$$
g = (\tilde{F} - C\hat{\alpha} - 1_f \beta_\gamma^2) \frac{\alpha^T}{\sigma^2} - M^T \lambda + z M^T (MC)^{-1} 1, \tag{33}
$$

$$
H = -\frac{\hat{\alpha} \alpha^T}{\sigma^2} - z M^T (MC)^{-1} M. \tag{34}
$$

If the parameters are unknown, they must be estimated. Initialize $\hat{\beta}$ as above. Then, define $\alpha = [\alpha_0, \ldots, \alpha_n]^T$ and initialize manually by assigning some pixels to each neuron (of course, more sophisticated algorithms could be used, as described in the discussion). Given this initialization, iterations and stopping criteria proceed as before, with the minor modification of incorporating multiple spatial filters, yielding:

$$
\alpha, \hat{\beta} \text{ s.t. } \min_{\alpha, \hat{\beta}} -\frac{1}{2} \sum_{t=1}^T \left( F_{t,j} - \sum_{i=1}^N \alpha_i \tilde{C}_i - \hat{\beta}_j \right)^2. \tag{36}
$$

Now, generalizing the above single spatial filter case, let $A = [\tilde{C}, 1]_T$ be an $(N + 1) \times T$ element matrix and $Y = [\alpha, \beta]^T$ be an $(N + 1)$-dimensional column vector. Then, one can again use Eq. 29 to solve for $\tilde{\alpha}$ and $\hat{\beta}$, for all $x$.

**Experimental methods**

**SLICE PREPARATION AND IMAGING.** All animal handling and experimentation were done according to the National Institutes of Health and local Institutional Animal Care and Use Committee guidelines. Somatosensory thalamocortical or coronal slices 350–400 μm thick were prepared from C57BL/6 mice at age P14 as described (MacLean et al. 2005). Pyramidal neurons from layer V somatosensory cortex were filled with 50 μM Oregon Green BAPTA 1 hexapotassium salt (OGB-1; Invitrogen, Carlsbad, CA) through the recording pipette or bulk loaded with an acetoxymethyl ester of Fura-2 (Fura-2 AM; Invitrogen). The pipette solution contained 130 mM K-methylsulfate, 2 mM MgCl$_2$, 0.6 mM EGTA, 4 mM ATP-Mg, and 0.3 mM GTP-Tris (pH 7.2, 295 mOsm). After cells were fully loaded with dye, imaging was performed in one of two ways. First, when using Fura-2, images were collected using a modified BX50-WI upright microscope (Olympus, Melville, NY) with a confocal spinning disk (Solamere Technology Group, Salt Lake City, UT) and an Orca charge-coupled device (CCD) camera from Hamamatsu Photonics (Shizuoka, Japan), at 33 Hz. Second, when using Oregon Green, images were collected with epifluorescence with the C9100-12 CCD camera from Hamamatsu Photonics, with arc-lamp illumination with excitation and emission band-pass filters at 480–500 and 510–550 nm, respectively (Chroma, Rockingham, VT). Images were saved and analyzed using custom software written in Matlab (The MathWorks, Natick, MA).

**ELECTROPHYSIOLOGY.** All recordings were made using the Multi-clamp 700B amplifier (Molecular Devices, Sunnyvale, CA), digitized with National Instruments 6259 multichannel cards and recorded using custom software written using the LabVIEW platform (National Instruments, Austin, TX). Square pulses of sufficient amplitude to yield the desired number of action potentials were given as current commands to the amplifier using the LabVIEW and National Instruments system.

**FLUORESCENCE PREPROCESSING.** Traces were extracted using custom Matlab scripts to segment the mean image into ROIs. The Fura-2 fluorescence traces were inverted. Because some slow drift was sometimes present in the traces, each trace was Fourier transformed, and all frequencies <0.5 Hz were set to zero (0.5 Hz was chosen by eye); the resulting fluorescence trace was then normalized to be between zero and one.

**RESULTS**

**Main result**

The main result of this study is that the fast filter can find the approximately most likely spike train $\hat{n}$, very efficiently, and that this approach yields more accurate spike train estimates than optimal linear deconvolution. Figure 2 depicts a simulation showing this result. Clearly, the fast filter’s inferred “spike train” (third panel) more closely resembles the true spike train (second panel) than the optimal linear deconvolution’s inferred spike train (bottom panel; Wiener filter). Note that neither filter results in an integer sequence, but rather, each infers a real number at each time.

![Figure 2](http://jn.physiology.org/)

**FIG. 2.** A simulation showing that the fast filter’s inferred spike train is significantly more accurate than the optimal linear deconvolution (Wiener filter). Note that neither filter constrains the inference to be a sequence of integers; rather, the fast filter relaxes the constraint to allow all nonnegative numbers and the Wiener filter allows for all real numbers. The restriction of the fast filter to exclude negative numbers eliminates the ringing effect seen in the Wiener filter output, resulting in a much cleaner inference. Note that the magnitude of the inferred spikes in the fast filter output is proportional to the inferred calcium jump size. **Top panel:** fluorescence trace. **Second panel:** spike train. **Third panel:** fast filter inference. **Bottom panel:** Wiener filter inference. Note that the gray bars in the bottom panel indicate negative spikes. Gray + symbols indicate true spike times. Simulation details: $T = 400$ time steps, $\Delta = 33.3$ ms, $\alpha = 1, \beta = 0, \sigma = 0.2, \tau = 1 s$, $\lambda = 1 Hz$. Parameters and conventions are consistent across figures, unless indicated otherwise.
versus the Poisson, which is discrete. The height of each spike is therefore proportional to the inferred calcium jump size and can be thought of as a proxy for the confidence with which the algorithm believes a spike occurred. Importantly, by using the Gaussian elimination and interior-point methods, as described in METHODS, the computational complexity of the fast filter is the same as an efficient implementation of the Wiener filter. Note that whereas the Gaussian approximation imposes a shrinkage prior on the inferred spike trains (Wu et al., 2006), the exponential approximation imposes a sparse prior on the inferred spike trains (Seeger, 2008).

Figure 3 quantifies the relative performance of the fast and Wiener filters. The top left panel shows a typical simulated spike train (bottom), a corresponding relatively low SNR fluorescence trace (middle), and a relatively high SNR fluorescence trace (top), as examples. The top right panel compares the mean-squared-error (MSE) of the inferred spike trains using the fast (solid) and Wiener (dashed) filters, as a function of expected firing rate. Clearly, the fast filter has a better (lower) MSE for all expected firing rates. Error bars show SD over 10 repeats. Simulation parameters: \( \sigma = 0.2, T = 1,000 \) time steps. Bottom left: receiver-operator-characteristic (ROC) curve comparing the fast (solid line) and Wiener (dashed-dotted line) filter. Note that for any given threshold, the Wiener filter has a better (higher) ratio of true positive rate to false positive rate. Simulation parameters as in top right panel, except \( \sigma = 0.35 \) and \( T = 10,000 \) time steps. Bottom right: area under the curve (AUC) for fast (solid line) and Wiener (dashed-dotted line) filter as a function of SD (\( \sigma \)). Note that the fast filter has a better (higher) AUC for all \( \sigma \) values until noise gets very high. The 2 simulated fluorescence traces in the top left panel show the bounds for SD here. Error bars show SD over 10 repeats.

The Wiener filter implicitly approximates the Poisson spike rate with a Gaussian spike rate (see APPENDIX B for details). A Poisson spike rate indicates that in each frame, the number of possible spikes is an integer, e.g., 0, 1, 2, .... The Gaussian approximation, however, allows any real number of spikes in each frame, including both partial spikes (e.g., 1.4) and negative spikes (e.g., -0.8). Although a Gaussian well approximates a Poisson distribution when rates are about 10 spikes per frame, this example is very far from that regime, so the Gaussian approximation performs relatively poorly. Further, the Wiener filter exhibits a “ringing” effect. Whenever fluorescence drops rapidly, the most likely underlying spiking signal is a proportional drop. Because the Wiener filter does not impose a nonnegative constraint on the underlying spiking signal, it infers such a drop, even when it causes \( n_t \) to go below zero. After such a drop has been inferred, since no corresponding drop occurred in the true underlying signal here, a complementary jump is often then inferred, to realign the inferred signal with the observations. This oscillatory behavior results in poor inference quality. The nonnegative constraint imposed by the fast filter prevents this because the underlying signal never drops below zero, so the complementary jump never occurs either.

The inferred “spikes,” however, are still not binary events when using the fast filter. This is a by-product of approximating the Poisson distribution on spikes with an exponential (cf. Eq. 11a) because the exponential is a continuous distribution.
fluctuating fluorescence trace. Again, it is clear that the fast filter far outperforms the Wiener filter.

Given the preceding two results, the fast filter was applied to biological data. More specifically, by jointly recording electrophysiological and imaging, the true spike times are known and the accuracy of the two filters can be compared. Figure 5 shows a result typical of the 12 joint electrophysiological and imaging experiments conducted (see METHODS for details). As in the simulated data, the fast filter output is much “cleaner” than the Wiener filter: spikes are more well defined, and not spread out, due to the sparse prior imposed by the exponential approximation. Note that this trace is typical of epifluorescence techniques, which makes resolving individual spikes quite difficult, as evidenced by a few false positives in the fast filter. Regardless, the fast filter output is still more accurate than the Wiener filter, both as determined qualitatively by eye and as quantified (described in the following text). Furthermore, although it is difficult to see in this figure, the first four events are actually pairs of spikes, reflected by the width and height of the corresponding inferred spikes when using the fast filter. This suggests that although the scale of $n$ is arbitrary, the fast filter can correctly ascertain the number of spikes within spike events.

Figure 6 further evaluates this claim. While recording and imaging, the cell was forced to spike once, twice, or thrice for each spiking event. The fast filter infers the correct number of spikes in each event. On the contrary, there is no obvious way to count the number of spikes within each event when using the Wiener filter. We confirm this impression by computing the correlation coefficient, $r^2$, between the sum of each filter’s output and the true number of spikes, for all 12 joint electrophysiological and imaging traces. Indeed, whereas the fast filter’s $r^2$ was 0.47, the Wiener filter’s $r^2$ was $-0.01$ (after thresholding all negative spikes), confirming that the Wiener filter output cannot reliably convey the number of spikes in a fluorescence trace, whereas the fast filter can. Furthermore, varying the magnitude of the threshold for the Wiener filter to discard more “low-amplitude noise” could increase the magnitude of $r^2$ ($=0.24$), still significantly lower than the fast filter’s $r^2$ value. On the other hand, no amount of thresholding the fast filter yielded an improved $r^2$, indicating that thresholding the output of the fast filter is unlikely to improve spike inference quality.

**On-line analysis of spike trains using the fast filter**

A central aim for this work was the development of an algorithm that infers spikes fast enough to use on-line while imaging a large population of neurons (e.g., $>100$). Figure 7 shows a segment of the results of running the fast filter on 136 neurons, recorded simultaneously, as described earlier in Experimental methods. Note that the filtered fluorescence signals show fluctuations in spiking much more clearly than the unfiltered fluorescence trace. These spike trains were inferred in less than imaging time, meaning that one could infer spike trains for the past experiment while conducting the subsequent experiment. More specifically, a movie with 5,000 frames of 100 neurons can be analyzed in about 10 s on a standard desktop computer. Thus if that movie was recorded at 50 Hz, whereas collecting the data would require 100 s, inferring spikes would require only 10 s, a 10-fold improvement over real time.

**Extensions**

Earlier in METHODS, Data-driven generative model describes a simple principled first-order model relating the spike trains to the fluorescence trace. A number of the simplifying assumptions can be straightforwardly relaxed, as described next.

Replacing Gaussian observations with poisson. In the preceding text, observations were assumed to have a Gaussian distribution. The statistics of photon emission and counting, however, suggest that a Poisson distribution would be more natural in some conditions, especially for two-photon data (Sjulson and Miesenböck 2007), yielding:

$$F_{\text{Poisson}}(\alpha C_i + \beta),$$

where $\alpha C_i + \beta \geq 0$. One additional advantage to this model over the Gaussian model is that the variance parameter $\sigma^2$ no
longer exists, which might make learning the parameters simpler. Importantly, the log-posterior is still concave in $C$, as the prior remains unchanged, and the new log-likelihood term is a sum of terms concave in $C$:

$$\ln P(F|C) = \sum_{t=1}^{T} \ln P(F_t|C_t) = \sum_{t=1}^{T} \left\{ F_t \ln (\alpha C_t + \beta) - (\alpha C_t + \beta) - \ln (F_t!)} \right\}. \quad (38)$$

The gradient and Hessian of the log-posterior can therefore be computed analytically by substituting the above likelihood terms for those implied by Eq. 1. In practice, however, modifying the filter for this model extension did not seem to significantly improve inference results in any simulations or data available at this time (not shown).

Allowing for a time-varying prior. In Eq. 4, the rate of spiking is a constant. Often, additional knowledge about the experiment, including external stimuli or other neurons spiking, can provide strong time-varying prior information (Vogelstein et al. 2009). A simple model modification can incorporate that feature:

$$n_i^{t_{id}} \sim \text{Poisson} (\lambda_t \Delta), \quad (39)$$

where $\lambda_t$ is now a function of time. Approximating this time-varying Poisson with a time-varying exponential with the same time-varying mean (similar to Eq. 11a) and letting $\lambda = [\lambda_1, \ldots, \lambda_T]^\top \Delta$, yields an objective function very similar to Eq. 15, so log-concavity is maintained and the same techniques may be applied. However, as before, this model extension did not yield any significantly improved filtering results (not shown).

Saturating fluorescence. Although all the abovementioned models assumed a linear relationship between $F_t$ and $C_t$, the relationship between fluorescence and calcium is often better approximated by the nonlinear Hill equation (Pologruto et al. 2004). Modifying Eq. 1 to reflect this change yields:

$$F_t = \alpha \frac{C_t}{C_t + k_d} + \beta + \varepsilon_t^{t_{id}} \sim \text{N}(0, \sigma^2). \quad (40)$$

Importantly, log-concavity of the posterior is no longer guaranteed in this nonlinear model, meaning that converging to the global maximum is no longer guaranteed. Assuming a good initialization can be found, however, and Eq. 40 is more accurate than Eq. 1, then ascending the gradient for this model is likely to yield improved inference results. In practice, initializing with the inference from the fast filter assuming a linear model (e.g., Eq. 30) often resulted in nearly equally accurate inference, but inference assuming the above nonlinearity was far less robust than the inference assuming the linear model (not shown).

Using the fast filter to initialize the SMC filter. A sequential Monte Carlo (SMC) method to infer spike trains can incorporate this saturating nonlinearity, as well as other model extensions discussed earlier (Vogelstein et al. 2009). However, this SMC filter is not nearly as computationally efficient as the fast filter proposed here. Like the fast filter, the SMC filter estimates the model parameters in a completely unsupervised fashion, i.e., from the fluorescence observations, using an expectation-maximization algorithm (which requires iterating between computing the expected value of the hidden variables—$C$ and $n$—and updating the parameters). In Vogelstein and colleagues (2009), parameters for the SMC filter were initialized based on other data. Although effective, this initialization was often far from the final estimates and thus required a relatively large number of iterations (e.g., 20–25) before converging. Thus it seemed that the fast filter could be used to obtain an improvement to the initial parameter estimates, given an appropriate rescaling to account for the nonlinearity, thereby reducing the required number of iterations to convergence. Indeed, Fig. 8 shows how the SMC filter outperforms the fast filter on biological data and required only three to five iterations to converge on these data, given the initialization from the fast filter (which was typical). Note that the first few events of the spike train are individual spikes, resulting in relatively small fluorescence fluctuations, whereas the next events are actually spike doublets or triplets, causing a much larger fluorescence fluctuation. Only the SMC filter correctly infers the presence of isolated spikes in this trace, a frequently occurring result when the SNR is poor. Thus these two inference algorithms are complementary: the fast filter can be used for rapid, on-line inference, and for initializing the SMC filter, which can then be used to further refine the spike train.

**FIG. 7.** The fast filter infers spike trains from a large population of neurons imaged simultaneously in vitro, faster than real time. Specifically, inferring the spike trains from this 400-s-long movie including 136 neurons required only about 40 s on a standard laptop computer. The inferred spike trains much more clearly convey neural activity than the raw fluorescence traces. Although no intracellular “ground truth” is available from these population data, the noise seems to be reduced, consistent with the other examples with ground truth. *Left:* mean image field, automatically segmented into regions of interest (ROIs), each containing a single neuron using custom software. *Middle:* example fluorescence traces. *Right:* fast filter output corresponding to each associated trace. Note that neuron identity is indicated by color across the 3 panels. Data were collected using a confocal microscope and Fura-2, as described in METHODS.
Spatial filter

In the preceding text, the filters operated on one-dimensional fluorescence traces. The raw data are in fact a time series of images that are first segmented into regions of interest (ROIs) and then (usually) spatially averaged to obtain a one-dimensional time series $F$. In theory, one could improve the effective SNR of the fluorescence trace by scaling each pixel according to its SNR. In particular, pixels not containing any information about calcium fluctuations can be ignored and pixels that are partially anticorrelated with one another could have weights with opposing signs.

Figure 9 demonstrates the potential utility of this approach. The top row shows different depictions of an ROI containing a single neuron. On the far left panel is the true spatial filter for this neuron. This particular spatial filter was chosen based on experience analyzing both in vitro and in vivo movies; often, it seems that the pixels immediately around the soma are anti-correlated with those in the soma (MacLean et al. 2005; Watson et al. 2008). This effect is possibly due to the influx of calcium from the extracellular space immediately around the soma. The standard approach, given such a noisy movie, would be to first segment the movie to find an ROI corresponding to the soma of this cell and then spatially average all the pixels found to be within this ROI. The second panel shows this standard “boxcar spatial filter.” The third panel shows the mean frame. The fourth panel shows the learned filter, using Eq. 29 to estimate the spatial filter and background. Clearly, the learned filter is very similar to the mean filter and the true filter.

The middle panels of Figure 9 show the fluorescence traces obtained by background subtracting and then projecting each frame onto the corresponding spatial filter (black line) and true spike train (gray + symbols). The bottom panels show the inferred spike trains (black bars) using these various spatial filters and, again, the true spike train (gray + symbols). Although the

---

**FIG. 8.** In vitro data with SNR of only about 3 (estimated by dividing the fluorescent jump size by the SD of the baseline fluorescence) for single action potentials depicting the fast filter, effectively initializing the parameters for the sequential Monte Carlo (SMC) filter, significantly reducing the number of expectation-maximization iterations to convergence, using OGB-1. Note that whereas the fast filter clearly infers the spiking events in the end of the trace, those in the beginning of the trace are less clear. On the other hand, the SMC filter more clearly separates nonspiking activity from true spikes. Also note that the ordinate on the third panel corresponds to the inferred probability of a spike having occurred in each frame.

---

**FIG. 9.** A simulation demonstrating that using a better spatial filter can significantly enhance the effective SNR. The true spatial filter was a difference of Gaussians: a positively weighted Gaussian of small width and a negatively weighted Gaussian with larger width (both with the same center). Each column shows the spatial filter (top), one-dimensional fluorescence projection using that spatial filter (middle), and inferred spike train (bottom). From left to right, columns use the true, boxcar, mean, and learned spatial filter obtained using Eq. 29. Note that the learned filter’s inferred spike train has fewer false positives and negatives than the boxcar and mean filters. Simulation parameters: $\alpha = N(0, 2I) - 0.5 N(0, 2.5I)$, where $N(\mu, \Sigma)$ indicates a 2-dimensional Gaussian with mean $\mu$ and covariance matrix $\Sigma$, $\beta = 0$, $\sigma = 0.2$, $\tau = 0.85 s$, $\lambda = 5 Hz$, $\Delta = 5 ms$, $T = 1,200$ time steps.
performance is very similar for all of them, the boxcar filter’s inferred spike train is not as clean.

**Overlapping spatial filters**

The preceding text shows that if the ROI contains only a single neuron, the effective SNR can be enhanced by spatially filtering. However, this analysis assumes that only a single neuron is in the ROI. Often, ROIs are overlapping, or nearly overlapping, making the segmentation problem more difficult. Therefore it is desirable to have an ability to crudely segment, yielding only a few neurons in each ROI, and then spatially filter within each ROI to pick out the spike trains of each neuron. This may be achieved in a principled manner by generalizing the model as described in *Overlapping spatial filters* in METHODS. The true spatial filters of the neurons in the ROI are often unknown and thus must be estimated from the data. This problem may be considered a special case of blind source separation (Bell and Sejnowski 1995; Mukamel et al. 2009). Figure 10 shows that given reasonable assumptions of spiking correlations and SNR, multiple signals can be separated. Note that separation occurs even though the signal is significantly overlapping (*top panels*). To estimate the spatial filters, they are initialized using the boxcar filters (*middle panels*). After a few iterations, the spatial filters converge to very close approximation to the true spatial filters [compare *left* and learned (*right*) spatial filters for the two neurons]. Note that both the true and learned spatial filters yield much improved spike inference relative to the boxcar filter. This suggests that even when spatial filters of multiple neurons are significantly overlapping, each spike train is potentially independently recoverable.

**DISCUSSION**

**Summary**

This work describes an algorithm that finds the approximate maximum a posteriori (MAP) spike train, given a calcium fluorescence movie. The approximation is required because finding the actual MAP estimate is not currently computationally tractable. Replacing the assumed Poisson distribution on spikes with an exponential distribution yields a log-concave optimization problem, which can be solved using standard gradient ascent techniques (such as Newton–Raphson). This exponential distribution has an advantage over a Gaussian distribution by restricting spikes to be positive, which improves inference quality (cf. Fig. 2), is a better approximation to a Poisson distribution with low rate, and imposes a sparse constraint on spiking. Furthermore, all the parameters can be estimated from only the fluorescence observations, obviating the need for joint electrophysiology and imaging (cf. Fig. 4). This approach is robust, in that it works “out of the box” on all the in vitro data analyzed (cf. Figs. 5 and 6). By using the special banded structure of the Hessian matrix of the log-posterior, this approximate MAP spike train can be inferred fast enough on standard computers to use it for on-line analyses of over 100 neurons simultaneously (cf. Fig. 7).

Finally, the fast filter is based on a biophysical model capturing key features of the data and may therefore be straightforwardly generalized in several ways to improve ac-
Innovative Methodology

VOGELSTEIN ET AL.

accuracy. Unfortunately, some of these generalizations do not improve inference accuracy, perhaps because of the exponential approximation. Instead, the fast filter output can be used to initialize the more general SMC filter (Vogelstein et al. 2009), to further improve inference quality (cf. Fig. 8). Another model generalization allows incorporation of spatial filtering of the raw movie into this approach (cf. Fig. 9). Even when multiple neurons are overlapping, spatial filters may be estimated to obtain improved spike inference results (cf. Fig. 10).

Alternate algorithms

This work describes but one specific approach to solving a problem that does not admit an exact solution that is computationally feasible. Several other approaches warrant consideration, including 1) a Bayesian approach, 2) a greedy approach, and 3) different analytical approximations.

First, a Bayesian approach could use Markov Chain Monte Carlo methods to recursively sample spikes to estimate the full joint posterior distribution of the entire spike train, conditioned on the fluorescence data (Andrieu et al. 2001; Joucla et al. 2010; Mishchenko et al. 2010). Although enjoying several desirable statistical properties that are lacking in the current approach (such as consistency), the computational complexity of such an approach renders it inappropriate for the aims of this work.

Second, a common relatively expedient approximation to Bayesian sampling is a so-called greedy approach. Greedy algorithms are iterative, with each iteration adding another spike to the putative spike train. Each spike that is added is the most likely spike (thus the greedy term) or the one that most increases the likelihood of the fluorescence trace. Template matching, projection pursuit regression (Friedman and Stuetzle 1981), and matching pursuit (Mallat and Zhang 1993) are examples of such a greedy approach (the algorithm proposed by Grewe et al. (2010) could also be considered a special case of such a greedy approach).

Third, approximations other than the exponential distribution are possible. For instance, the Gaussian approximation is more appropriate for high firing rates, although in simulations, this more accurate approximation did not improve the Wiener filter output relative to the fast filter output (cf. Fig. 3). Perhaps the best approximation would use the closest log-concave relaxation to the Poisson model (Koenker and Mizera 2010).

More formally, let $P(i)$ represent the Poisson mass at $i$ and let $\ln Q$ be some concave density. Then, one could find the log-density $Q$ such that $Q$ maximizes $\sum_i P(i)Q(i) - \lambda \int \exp(Q(x))\,dx$ over the space of all concave $Q$. The first term corresponds to the log-likelihood, equivalent to the Kullback–Leibler divergence (Cover and Thomas 1991), and the second is a Lagrange multiplier to ensure that the density $\exp(Q(x))$ integrates to unity. This is a convex problem because the space of all concave $Q$ is convex and the objective function is concave in $Q$. In addition, it is easy to show that the optimal $Q$ has to be piecewise linear; this means that one need not search over all possible densities, but rather, simply vary $Q(i)$ at the integers. Note that $\int \exp(Q(x))\,dx$ can be computed explicitly for any piecewise linear $Q$. This optimization problem can be solved using simple interior point methods and, in fact, the Hessian of the inner loop of the interior point method will be banded (because enforcing concavity of $Q$ is a local constraint). This approximation could potentially be more accurate than our exponential approximation. Further, this approximation encourages integer solutions for $n$, and is therefore of interest for future work.

The abovementioned three approaches may be thought of as complementary because each has unique advantages relative to the others. Both the greedy methods and the analytic approximations could potentially be used to initialize a Bayesian approach, possibly limiting the burn-in period, which can be computationally prohibitive in certain contexts. A greedy approach has the advantage of providing actual spike trains (i.e., binary sequences), unlike the analytic approximations. However, the actual spike trains could be quite far from the MAP spike train because greedy approaches, in general, have no guarantee of consistency. The analytic approximations, on the other hand, are guaranteed to converge to solutions close to the MAP spike train, where closeness is determined by the accuracy of the above approximation. Thus developing these distinct approaches and combining them is a potential avenue for further research.

Spatial filtering

Spatial filtering could be improved in a number of ways. For instance, pairing this approach with a crude but automatic segmentation tool to obtain ROIs would create a completely automatic algorithm that converts raw movies of populations of neurons into populations of spike trains. Furthermore, this filter could be coupled with more sophisticated algorithms to initialize the spatial filters when they are overlapping (for instance, principal component analysis (Horn and Johnson 1990) or independent component analysis (Mukamel et al. 2009)). One could also use a more sophisticated model to estimate the spatial filters. One option would be to assume a simple parametric form of the spatial filter for each neuron (e.g., a basis set) and then merely estimate the parameters of that model. Alternately, one could regularize the spatial filters, using an elastic net type approach (Grosenick et al. 2009; Zou and Hastie 2005), to enforce both sparseness and smoothness.

Model generalizations

In this work, we made two simplifying assumptions that can easily be relaxed: 1) instantaneous rise time of the fluorescence transient after a spike and 2) constant background. In practice, often either or both of these assumptions are inaccurate. Specifically, genetic sensors tend to have a much slower rise time than that of organic dyes (Reiff et al. 2005). Further, the background often exhibits slow baseline drift due to movement, temperature fluctuations, laser power, and so forth, not to mention bleaching, which is ubiquitous for long imaging experiments. Both slow rise and baseline drift can be incorporated into our forward model using a straightforward generalization.

Consider the following illustrative example: the fluorescence rise time in a particular data set is quite slow, much slower than that of a single image frame. Thus fluorescence might be well modeled as the difference of two different calcium extrusion mechanisms, with different time constants. To model this scenario, one might proceed as follows: posit the existence of a two-dimensional time-varying signal, each like the calcium
signal assumed in the simpler models described earlier. Therefore each signal has a time constant and each signal is dependent on spiking. Finally, the fluorescence could be a weighted difference of the two signals. To formalize this model and to generalize it, let 1) $X = (X_1, \ldots, X_d)$ be a $d$-dimensional time-varying signal; 2) $\Gamma$ be a $d \times d$ dynamics matrix, where diagonal elements correspond to time constants of individual variables, and off-diagonal elements correspond to dependencies across variables; 3) $A$ be a $d$-dimensional binary column vector encoding whether each variable depends on spiking; and 4) $\alpha$ be a $d$-dimensional column vector of weights, determining the relative impact of each dimension on the total fluorescence signal. Given these conventions, we have the following generalized model:

$$F_t = \alpha^T X_t + \beta + e, \quad e \sim \mathcal{N}(0, \sigma^2) \quad (41)$$

$$X_t = \Gamma X_{t-1} + A n_t, \quad n_t \sim \text{Poisson}(\lambda \Delta) \quad (42)$$

Note that this model simplifies to the model proposed earlier when $d = 1$. Because $X$ is still Markov, all the theory developed above still applies directly for this model. There are, however, additional complexities with regard to identifiability. Specifically, the parameters $\alpha$ and $A$ are closely related. Thus we enforce that $A$ is a known binary vector, simply encoding whether a particular element responds to spiking. The matrix $\Gamma$ will not be uniquely identifiable, for the same reason that $\gamma$ was not identifiable, as described in Learning the parameters in Methods. Thus we would assume $\Gamma$ was known, a priori. Note that other approaches to dealing with baseline drift are also possible, such as letting $\beta$ be a time-varying state: $\beta_t = \beta_{t-1} + \varepsilon_t$, where $\varepsilon_t$ is a normal random variable with variance $\sigma^2_\beta$ that sets the effective drift rate. Both these models are the subjects of further development.

**Concluding thoughts**

In summary, the model and algorithm proposed in this work potentially provide a useful tool to aid in the analysis of calcium-dependent fluorescence imaging and establish the groundwork for significant further development.

**APPENDIX B: WIENER FILTER**

The Poissonian in Eq. 4 can be replaced with a Gaussian instead of an exponential distribution, i.e., $n_t \sim \mathcal{N}(\lambda \Delta, \lambda \Delta)$ that, when plugged into Eq. 7, yields:

$$\tilde{h} = \arg\max_{\alpha_t} \sum_{t=1}^{T} \left[ \frac{1}{2\sigma^2} (F_t - \alpha C_t - \beta)^2 + \frac{1}{2\lambda \Delta} (n_t - \lambda \Delta)^2 \right]$$

(B1)

Note that since fluorescence integrates over $\Delta$, it makes sense that the mean scales with $\Delta$. Further, since the Gaussian here is approximating a Poisson with high rate (Sjulson and Miesenböck 2007), the variance should scale with the mean. Using the same tridiagonal trick as before, Eq. 11b can be solved using Newton–Raphson once (because this expression is quadratic in $n_t$). Writing the above in matrix notation, substituting $C_t - \gamma C_{t-1}$ for $n_t$, and letting $\alpha = 1$ yields:

$$\tilde{C} = \arg\max_{C} \sum_{t=1}^{T} \left[ \frac{1}{2\sigma^2} \|F_t - C - \beta t\|_2^2 + \frac{1}{2\lambda \Delta} \|MC - \lambda \Delta I\|_2^2 \right]$$

(B2)

which is quadratic in $C$. The gradient and Hessian are given by:

$$g = \frac{1}{\sigma^2} (C - F) + \frac{1}{\lambda \Delta} (MC^T) M + \frac{1}{\lambda \Delta} M^T 1$$

(B3)

$$H = \frac{1}{\sigma^2} I + \frac{1}{\lambda \Delta} M^T M$$

(B4)

Note that this solution is the optimal linear solution, under the assumption that spikes follow a Gaussian distribution, and is often referred to as the Wiener filter, regression with a smoothing prior, or ridge regression (Boyd and Vandenberghe 2004). Estimating the parameters for this model follows a pattern similar to that described in Learning the parameters in Methods.

**ACKNOWLEDGMENTS**

We thank V. Bonin for helpful discussions.

**GRANTS**

This work was supported by National Institute on Deafness and Other Communication Disorders Grant DC-00109 to J. T. Vogelstein; National Science Foundation (NSF) Faculty Early Career Development award, an Alfred P. Sloan Research Fellowship, and a McKnight Scholar Award to L.
Innovative Methodology

Paninski; National Eye Institute Grant EY-11787 and the Kavli Institute for Brain Studies award to R. Yuste and the Yuste laboratory; and an NSF Collaborative Research in Computational Neuroscience award IIS-0904353, awarded jointly to L. Paninski and R. Yuste.

DISCLOSURES

No conflicts of interest, financial or otherwise, are declared by the author(s).

REFERENCES


