

Bayesian microsaccade detection

Andra Mihali*

Center for Neural Science, New York University,
New York, NY, USA



Bas van Opheusden*

Center for Neural Science, New York University,
New York, NY, USA



Wei Ji Ma

Center for Neural Science, New York University,
New York, NY, USA
Department of Psychology, New York University,
New York, NY, USA



Microsaccades are high-velocity fixational eye movements, with special roles in perception and cognition. The default microsaccade detection method is to determine when the smoothed eye velocity exceeds a threshold. We have developed a new method, Bayesian microsaccade detection (BMD), which performs inference based on a simple statistical model of eye positions. In this model, a hidden state variable changes between drift and microsaccade states at random times. The eye position is a biased random walk with different velocity distributions for each state. BMD generates samples from the posterior probability distribution over the eye state time series given the eye position time series. Applied to simulated data, BMD recovers the “true” microsaccades with fewer errors than alternative algorithms, especially at high noise. Applied to EyeLink eye tracker data, BMD detects almost all the microsaccades detected by the default method, but also apparent microsaccades embedded in high noise—although these can also be interpreted as false positives. Next we apply the algorithms to data collected with a Dual Purkinje Image eye tracker, whose higher precision justifies defining the inferred microsaccades as ground truth. When we add artificial measurement noise, the inferences of all algorithms degrade; however, at noise levels comparable to EyeLink data, BMD recovers the “true” microsaccades with 54% fewer errors than the default algorithm. Though unsuitable for online detection, BMD has other advantages: It returns probabilities rather than binary judgments, and it can be straightforwardly adapted as the generative model is refined. We make our algorithm available as a software package.

Introduction

Even when people attempt to fixate, their eyes are always in motion. Fixational eye movements are often categorized as drift, tremor, and microsaccades (Ciuffreda & Tannen, 1995). During most of the fixation time, the eye is in a drift state, which is low-amplitude, low-velocity motion, sometimes modeled as a random walk (Cornsweet, 1956; Ditchburn & Ginsborg, 1953; Ratliff & Riggs, 1950). A few times a second, the eye makes a microsaccade, which is a high-velocity movement along a relatively straight trajectory (Cornsweet, 1956; Engbert & Kliegl, 2003; Engbert, Mergethaler, Sinn, & Pikovsky, 2011; Rolfs, 2009).

Microsaccades have been implicated in several perceptual and cognitive functions, including aiding performance in high-acuity visual tasks (Ko, Poletti, & Rucci, 2010; Poletti, Listorti, & Rucci, 2013; Rucci, Iovin, Poletti, & Santini, 2007) and shifts of covert spatial attention in both humans and monkeys (Engbert & Kliegl, 2003; Hafed & Clark, 2002; Hafed, Lovejoy, & Krauzlis, 2011; Lara & Wallis, 2012; Laubrock, Engbert, & Kliegl, 2005; Laubrock, Engbert, Rolfs, & Kliegl, 2007; Rolfs, 2009; Rolfs, Engbert, & Kliegl, 2004; Yuval-Greenberg, Merriam, & Heeger, 2014), though there has been some disagreement regarding this last role (Collewijn & Kowler, 2008; Horowitz, Fine, Fencsik, Yurgenson, & Wolfe, 2007). Arguments about the functional roles of microsaccades rely on accurate definition and detection of microsaccades (Poletti & Rucci, 2016). Microsaccade detection is complicated by motor noise in the eye and measurement noise in the eye tracker. The latter is particularly important in view of the widespread use of

Citation: Mihali, A., van Opheusden, B., & Ma, W. J. (2017). Bayesian microsaccade detection. *Journal of Vision*, 17(1):13, 1–23, doi:10.1167/17.1.13.



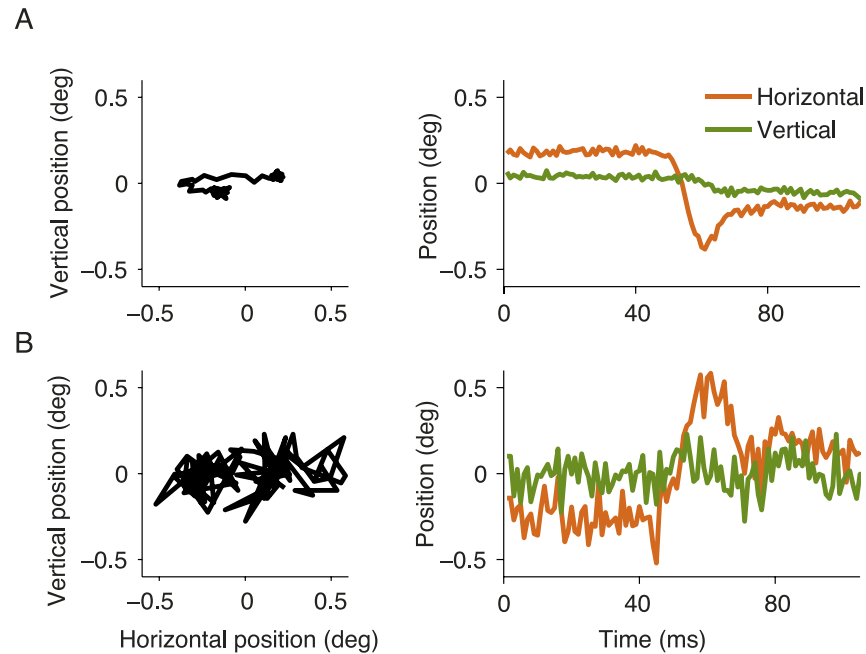


Figure 1. Microsaccades under different noise levels. Example single-trial eye position data from two subjects, measured with the EyeLink eye tracker with the “Heuristic filter” option turned off. (A) Measured eye position in the plane (left) and horizontal and vertical position as a function of time (right) for an easily detectable microsaccade. (B) Another trace, which contains an apparent microsaccade buried in measurement noise. EK with the threshold multiplier set at six identifies a microsaccade in (A) but not in (B).

video-based infrared eye trackers, which are less invasive than magnetic scleral search coils, but noisier (Hermens, 2015; Träisk, Bolzani, & Ygge, 2005). For example, the popular EyeLink II video-based infrared eye tracker reports a precision of 0.01 deg of visual angle; however, in practice this precision can be worse (Holmqvist et al., 2011). The low sensitivity, precision, and resolution of video-based eye trackers can cause difficulties in resolving microsaccades (Nyström, Hansen, Andersson, & Hooge, 2016; Poletti & Rucci, 2016).

How can microsaccades be reliably detected in the presence of other fixational eye movements and measurement noise? The most commonly used microsaccade detection method, especially in human studies, is a velocity-threshold algorithm proposed by Engbert and Kliegl (Engbert, 2006; Engbert & Kliegl, 2003). This method, which we refer to as EK, detects a microsaccade when the magnitude of the eye velocity exceeds a given threshold for a sufficiently long duration. Because a fixed threshold would ignore differences in noise across trials and individuals, the threshold was adaptively chosen to be a multiple of the standard deviation of the velocity distribution (Engbert & Kliegl, 2003). However, the value of the multiplier is arbitrary and affects the algorithm’s performance, as expected from signal detection theory: If the multiplier is too high, the algorithm misses microsaccades, while too low a multiplier causes false alarms. For example, EK with the threshold multiplier set to its standard value of 6 (Engbert & Kliegl, 2003) labels the eye

position data in Figure 1A as a microsaccade, but not the data in Figure 1B. However, lowering the threshold multiplier to three causes EK to label both examples as microsaccades. This ambiguity in the identification of microsaccades can cause ambiguity in conclusions about their functional roles.

More recent algorithms have tried to eliminate the need for an arbitrary velocity threshold by taking into account more details of the statistics of fixational eye movements. Bettenbuehl et al. (2010) assumed that microsaccades are discontinuities embedded in drift and used wavelet analysis to detect them. Otero-Millan, Castro, Macknik, and Martinez-Conde (2014) have proposed an unsupervised clustering algorithm based on three features: peak velocity, initial acceleration peak, and final acceleration peak.

Here we propose to detect microsaccades with a Bayesian algorithm. Bayesian algorithms have been used previously for saccade detection. Salvucci and colleagues (Salvucci & Anderson, 1998; Salvucci & Goldberg, 2000) used a hidden Markov model to separate fixations from saccades. However, their algorithm requires the user to specify a set of fixation targets, which are regions of interest based on a cognitive process model of the task. By contrast, our algorithm is entirely task independent. More recently, Daye and Optican (2014) used particle filters to estimate the posterior over a hidden position variable in a generic and simple model for eye velocity. Whenever this model fails to capture the data, their algorithm

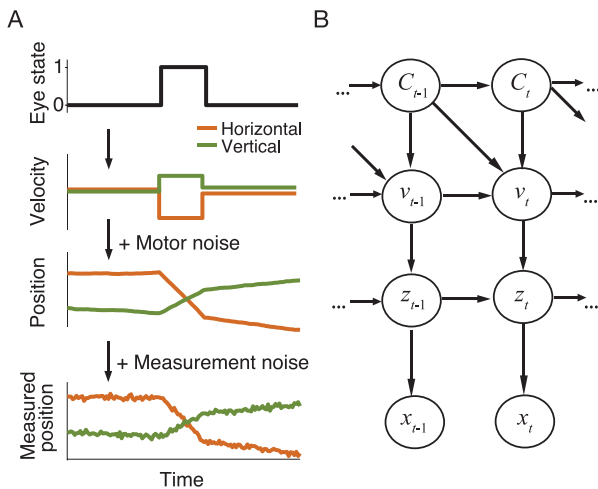


Figure 2. Generative model of fixational eye movements. (A) Example time courses of the variables in our model. (B) Graphical representation of our generative model for eye position data, which is a hidden semi-Markov model. The eye state, a latent binary variable C_t , is either a low-velocity drift/tremor state (0) or a high-velocity microsaccade state (1). The latent eye state informs the velocity v_t , which together with the preceding eye position z_{t-1} and motor noise yield the current eye position z_t . This eye position is contaminated with measurement noise, yielding the measured eye position x_t .

concludes that a microsaccade or saccade has occurred. Instead, we build an explicit model of both microsaccades and drift, and compute the full posterior over the eye state. Santini, Fuhl, Kubler and Kasneci (2016) have proposed using a Bayesian classifier to separate fixations, saccades, and smooth pursuits based on two features: eye speed and a feature which distinguishes smooth from abrupt motion. This algorithm was applied to much lower resolution eye tracking data (30 Hz) than are typically used in psychology laboratories, and while principled, it still relies on preprocessing using a heuristic filter. This method seems to work for separating saccades from drift, but we focus on the harder problem of separating microsaccades from drift.

Methods

We develop a Bayesian algorithm for microsaccade detection. First, we make explicit assumptions about the statistical process by which the eye movement data are generated from an underlying sequence of hidden states alternating between drift and microsaccades. Optimal Bayesian inference then entails inverting this generative model to infer the probability of the hidden eye state sequence given the measured eye position data. The fact that our algorithm returns a probability

distinguishes it from earlier algorithms, which return only a binary judgment.

The input of our inference algorithm is a time series of measured eye positions. We conceptualize this time series as being generated from an unknown internal state, which at each time step is either drift/tremor (0) or microsaccade (1). We distinguish the two by asserting that they correspond to different velocity distributions; this statistical definition stands in contrast to the traditional method, which uses a threshold. The probability distributions that describe the process by which the measure eye position time series arises from the internal state are together called the generative model.

Assuming this generative model, we derive an inference algorithm that estimates the time series of hidden eye states given a particular measured eye position time series. The algorithm considers many candidate time series (e.g., 0001111100... 00111111111000) and calculates how consistent each candidate is with the data; this is called the likelihood of that candidate time series. Combining the likelihoods with prior information about frequencies and durations of microsaccades yields the posterior distribution over time series. Because the space of candidate time series is very large ($2^{60,000}$ for 1 min of data sampled at 1 kHz), we use a suitable search algorithm from the class of Markov-chain Monte Carlo (MCMC) sampling algorithms.

Software

A computer package implementing our algorithm is available at: <https://github.com/basvanopheusden/BMD>.

Generative model

We formulate our model to generate eye position data in 1-ms time bins, but this can be easily extended to different sampling rates. We use boldface to denote a time series of a variable. The eye state time series \mathbf{C} has length T . We assume that at time t , the eye is in a hidden state C_t , which is 0 for a drift/tremor state and 1 for a microsaccade state (Figure 2). As long as the eye remains in the same state, its two-dimensional velocity v_t remains constant; when the eye switches state, its velocity changes. Of note, velocity here does not represent the derivative of the measured eye position, but rather an unobservable underlying variable.

At every time step, the eye's two-dimensional position z_t gets augmented with the velocity and Gaussian motor noise with covariance matrix Σ_z . This eye position is augmented with Gaussian measurement

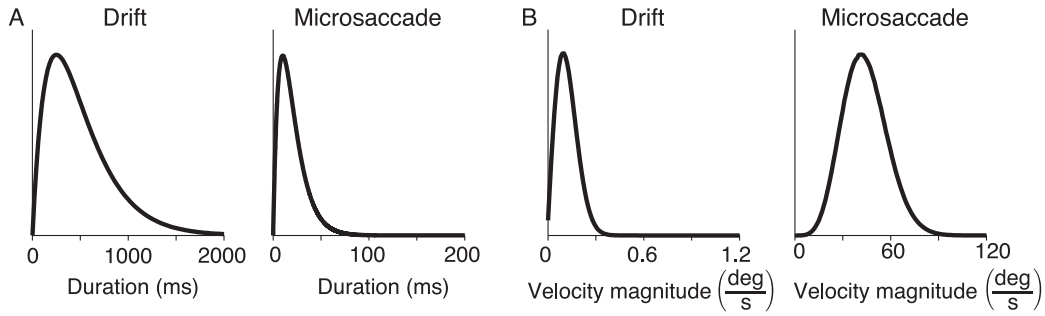


Figure 3. Prior distributions used in the algorithm. (A) Prior distributions over the durations of drift and microsaccade states. These priors are fixed in the inference process. (B) Priors for eye velocity for drift and microsaccade states. The inference process estimates the parameters of these priors from data; here we show the priors estimated for one example subject (EyeLink S1; Table 2). Note that these distributions are not normalized.

noise with covariance matrix Σ_x (independent across time points), yielding the measured eye position x_t .

We define a change point of \mathbf{C} as a time t where $C_t \neq C_{t-1}$, and denote the i th change point as τ_i . The duration for which the eye stays in a given state is then $\Delta\tau_i = \tau_{i+1} - \tau_i$. We assume that \mathbf{C} is a semi-Markov process, which means that these durations are independent. In a hidden Markov model (Bishop, 2006), the probability of C_t depends only on the previous state, C_{t-1} ; however, in a hidden semi-Markov model (also called an explicit-duration hidden Markov model; Yu, 2010), the durations over which the state remains unchanged are independent. Then the prior probability of \mathbf{C} is

$$p(\mathbf{C}) = \prod_i p(\Delta\tau_i | C_{\tau_i}), \quad (1)$$

where $p(\Delta\tau_i | C_{\tau_i})$ is the state-specific probability of the duration. Specifically, we use a gamma distribution with shape parameter 2 and scale parameter k :

$$p(\Delta\tau | C) \propto \Delta\tau e^{-k\Delta\tau}, \quad (2)$$

where $k = k_0$ if $C = 0$ (drift/tremor) and $k = k_1$ if $C = 1$ (microsaccade). We choose this distribution because it makes very short and very long durations unlikely, consistent with previously reported distributions of durations for drift and microsaccades (Engbert, 2006). Assumptions about the frequency and duration of microsaccades are reflected in the choices of parameters k_0 and k_1 . We chose $k_0 = 4 \text{ s}^{-1}$ and $k_1 = 100 \text{ s}^{-1}$, corresponding to median durations of 260 ms for drift and 10 ms for microsaccades (Figure 3A, B), which are realistic (Engbert, 2006). We will later examine the robustness of our results to variations in k_0 and k_1 (Figure A6).

At each change point τ_i , we draw the velocity v_{τ_i} from a state-specific probability distribution $p(v_{\tau_i} | C_{\tau_i})$; this velocity remains constant until the eye switches state at τ_{i+1} . The distribution of the velocity time series \mathbf{v} given an eye state time series \mathbf{C} is

$$p(\mathbf{v} | \mathbf{C}) = \prod_i \left[p(v_{\tau_i} | C_{\tau_i}) \prod_{t=\tau_i+1}^{\tau_{i+1}-1} \delta(v_t - v_{t-1}) \right]. \quad (3)$$

To define the state-specific velocity distribution, we write v in polar coordinates, $v = (r \cos \theta, r \sin \theta)^T$, and assume that in both states, the direction of the velocity θ is uniformly distributed and its magnitude r follows a generalized gamma distribution

$$p(r | C) = \frac{2}{\left(\Gamma\left(\frac{d+1}{2}\right) (\sigma\sqrt{2})^{d+1} \right)} r^d e^{-\frac{r^2}{2\sigma^2}}, \quad (4)$$

where $d = d_0$ and $\sigma = \sigma_0$ if $C = 0$, and $d = d_1$ and $\sigma = \sigma_1$ if $C = 1$. Note that our definition of the generalized gamma distribution differs from that of Stacy (1962) by a reparametrization $d \rightarrow d + 1$, $\sigma \rightarrow \sigma\sqrt{2}$. We fix d_0 to 1, which is equivalent to assuming that the distribution of the two-dimensional velocity in the drift/tremor state is a circularly symmetric Gaussian with standard deviation σ_0 . The other parameters d_1 and σ_1 control the shape and scale, respectively, of the distribution of microsaccade velocities. Figure 3B shows examples of these velocity distributions.

The eye position time series \mathbf{z} is piecewise linear with velocity \mathbf{v} , plus motor noise, which follows a Gaussian distribution with covariance matrix Σ_z :

$$p(\mathbf{z} | \mathbf{v}) = \prod_{t=1}^T p(z_t | z_{t-1}, v_t) = \prod_{t=1}^T \mathcal{N}(z_t; z_{t-1} + v_t, \Sigma_z). \quad (5)$$

The observed eye position time series \mathbf{x} is equal to \mathbf{z} plus Gaussian measurement noise that is independent across time and has covariance matrix Σ_x :

$$p(\mathbf{x} | \mathbf{z}) = \prod_{t=1}^T p(x_t | z_t) = \prod_{t=1}^T \mathcal{N}(x_t; z_t, \Sigma_x). \quad (6)$$

Motor and measurement noise are in principle distinguishable, because changes in the eye position due

Step	Operation
0	Initialize \mathbf{C} , $\hat{\sigma}_1$, \hat{d}_1 , $\hat{\sigma}_0$
1	Estimate the motor and measurement noise: $\hat{\sigma}_z$, $\hat{\sigma}_x$
2	Estimate $\hat{\mathbf{z}}$ from observations \mathbf{x} : Kalman smoother
3	Sample from the posterior over \mathbf{C} : MCMC
4	Estimate the velocity distribution parameters: maximum likelihood estimation (MLE) Return to Step 1

Table 1. BMD algorithm.

to motor noise are added over time, whereas measurement noise is independently added at each time point. We assume that both covariance matrices are isotropic: $\Sigma_z = \sigma_z^2 \mathbf{I}$ and $\Sigma_x = \sigma_x^2 \mathbf{I}$. Before we analyze data, we rescale the vertical dimension of the measured eye positions so that the isotropy assumption is approximately satisfied (see Preprocessing, later).

Inference of the eye state time series \mathbf{C}

Our goal is to infer the eye state time series \mathbf{C} given a time series of measured eye positions \mathbf{x} , using the generative model. To perform optimal inference, we need to compute the posterior distribution over \mathbf{C} . By Bayes's rule, this posterior is proportional to the product of the prior $p(\mathbf{C})$ and the likelihood $p(\mathbf{x}|\mathbf{C})$:

$$p(\mathbf{C}|\mathbf{x}) \propto p(\mathbf{C})p(\mathbf{x}|\mathbf{C}). \quad (7)$$

The prior can be directly evaluated using Equations 1 and 2, but computing the likelihood requires marginalization over nuisance parameters, the velocity time series \mathbf{v} and the eye position time series \mathbf{z} , using the dependencies given by the generative model:

$$p(\mathbf{x}|\mathbf{C}) = \iint p(\mathbf{x}|\mathbf{z})p(\mathbf{z}|\mathbf{C}, \mathbf{v})p(\mathbf{v}|\mathbf{C}) d\mathbf{v} d\mathbf{z}. \quad (8)$$

Plugging in the functional form of these distributions, and performing some algebra (see Computation of the likelihood in the Appendix), yields the likelihood of \mathbf{C} :

$$p(\mathbf{x}|\mathbf{C}) \propto \int \prod_i \left[e^{-\frac{(z_i - z_{i-1})^T (z_i - z_{i-1})}{2\sigma_z^2} - \frac{(x_i - z_i)^T (x_i - z_i)}{2\sigma_x^2}} \right] \times \prod_i \left[\int p(v_{\tau_i} | C_{\tau_i}) e^{-\frac{(\tau_{i+1} - \tau_i)^T v_{\tau_i} v_{\tau_i}}{2\sigma_z^2} + \frac{(z_{\tau_{i+1}} - z_{\tau_i})^T v_{\tau_i}}{\sigma_z^2}} dv_{\tau_i} \right] d\mathbf{z}. \quad (9)$$

Approximate inference

The goal of our algorithm is to draw samples from the posterior $p(\mathbf{C}|\mathbf{x})$. First we need to evaluate the likelihood in Equation 9. This is difficult, because we

need to integrate both over the velocities at all change points, $\{v_{\tau_i}\}$, and over the eye position time series \mathbf{z} . The velocity integral is numerically tractable, but the eye position one is not. Moreover, the likelihood also depends on the unknown parameters σ_0 , σ_1 , d_1 , σ_z , and σ_x . A fully Bayesian algorithm would require priors over these parameters to jointly infer the parameters together with the eye state time series \mathbf{C} . This too is intractable.

Instead, we use a multistep approximate inference algorithm, which we name Bayesian microsaccade detection (BMD), outlined in Table 1. A key idea in this algorithm is to replace the marginalization over \mathbf{z} by a single estimate, reminiscent of expectation maximization. Our algorithm then alternates between estimating \mathbf{C} , \mathbf{z} , and the parameters for six iterations, which in practice suffices for the estimates to converge. To run BMD on an eye position time series of 1 min (60,000 time steps) takes approximately 40 s on a MacBook Pro with an Intel Core i7 with a 2.9 GHz processor and 8 GB of RAM.

Although BMD returns a probability over eye state at every time point, for most of the following analyses we will threshold these probabilities at 0.5 in order to obtain binary judgments.

We now describe the details of the steps of the BMD algorithm.

Preprocessing

We split the eye position data into blocks of ~ 1 min, which we process independently. Before we perform inference, we preprocess the data to match the isotropy assumption of the measurement and motor noise in our generative model. To do so, we observe that within our model, eye velocity is piecewise constant, and therefore its derivative is zero except at change points. This means that the acceleration of the measured eye position depends only on the motor and measurement noise, except at change points. For this reason, we use the median absolute deviation of the acceleration to estimate the noise level. We calculate this quantity separately in the horizontal and vertical dimensions, and rescale the vertical-position time series by the ratio of the outcomes. After rescaling, the noise distribution is approximately isotropic.

The algorithm utilizes measured eye position at boundary unobserved time points x_0 and x_{T+1} . For these, we choose $x_0 = x_1 - \varepsilon$ and $x_{T+1} = x_T + \varepsilon$, where $\varepsilon = (10^{-4}, 10^{-4})$ deg. We need to include the offset ε to avoid numerical instabilities in our implementation. Finally, we subtract the resulting value of x_0 from every point in the time series, so that $x_0 = 0$; this has no effect on the detected microsaccades.

Step 0: Initialize \mathbf{C} , $\hat{\sigma}_1$, \hat{d}_1 , $\hat{\sigma}_0$

We fix d_0 to 1. We initialize $\hat{\sigma}_0$, $\hat{\sigma}_1$, and \hat{d}_1 to random values drawn from reasonable ranges ($\hat{\sigma}_0$: [0.1, 5] deg/s, $\hat{\sigma}_1$: [5, 100] deg/s, and \hat{d}_1 : [1.1, 5]). We initialize \mathbf{C} by setting C_t to 1 for time points t where $\|x_t - x_{t-1}\|$ is in the highest percentile, and to 0 otherwise.

Step 1: Estimate the motor and measurement noise

Our first goal is to estimate σ_x and σ_z given a measured eye position time series \mathbf{x} and an estimated eye state time series $\hat{\mathbf{C}}$. As stated before, we can disentangle motor and measurement noise because, in our generative model, motor noise accumulates over time while measurement noise does not. Specifically, the autocovariance function of \mathbf{x} conditioned on \mathbf{v} at time lag s is

$$\text{cov}(x_t, x_{t-s}) = 2\sigma_z^2 s + 4\sigma_x^2. \quad (10)$$

To use this relationship, we first estimate \mathbf{v} by fitting \mathbf{x} as a piecewise linear function with discontinuities at the change points of \mathbf{C} . Then we calculate the empirical autocovariance function of the residual,

$$c_{\text{emp}}(s) = \frac{1}{T} \sum_{t=s+1}^T \tilde{x}_t^T \tilde{x}_{t-s}, \quad (11)$$

and fit this as a linear function of s ; this gives a slope and a y-intercept. Our estimates of the motor noise and measurement noise are $\hat{\sigma}_z = \sqrt{\text{slope}/2}$ and $\hat{\sigma}_x = \sqrt{y_{\text{intercept}}/4}$.

Step 2: Estimate \mathbf{z} from observations \mathbf{x} with Kalman smoother

We cannot compute the likelihood of the eye state time series, $p(\mathbf{x}|\mathbf{C})$ in Equation 9, because the integral over \mathbf{z} is both analytically and numerically intractable. However, the integral over v_{τ_i} depends only on $z_{\tau_{i+1}} - z_{\tau_i}$. The expected value of this difference is equal to $(\tau_{i+1} - \tau_i)\bar{v}$, where \bar{v} is the average velocity between the change points; its standard deviation is of the order of σ_z . Therefore, if either \bar{v} or $\tau_{i+1} - \tau_i$ is sufficiently large (we expect the former to hold for microsaccades and the latter for drift), we can neglect the uncertainty in $z_{\tau_{i+1}} - z_{\tau_i}$ and approximate it by a point estimate.

We obtain the point estimate of \mathbf{z} given \mathbf{x} by maximizing the first integral in Equation 9. This maximization turns out to be equivalent to applying a Kalman smoother to \mathbf{x} (Kalman, 1960; Welch & Bishop, 2006). In general, a Kalman smoother estimates the system state in a time interval from noisy observations during the same interval. The

optimal estimate turns out to be a linear filter. We implement the Kalman smoother with the Rauch–Tung–Striebel (RTS) algorithm, which applies a Kalman filter to \mathbf{x} followed by another Kalman filter to the output of the first filter, backward in time (Rauch, Tung, & Striebel, 1965; Terejanu, 2008). The Kalman filter estimates the system state at each time only from earlier observations. In our case, the RTS algorithm reduces to

$$\hat{y}_t = \hat{y}_{t-1} + K_{\text{forward}}(x_t - \hat{y}_{t-1}), \quad (12)$$

with $K_{\text{forward}} = (1 + \sqrt{1 + 4R^2}) / (1 + 2R^2 + \sqrt{1 + 4R^2})$, where $R = \frac{\sigma_x}{\sigma_z}$, and

$$\hat{z}_t = \hat{z}_{t+1} + K_{\text{backward}}(y_t - \hat{z}_{t+1}), \quad (13)$$

with $K_{\text{backward}} = (\sqrt{1 + 4R^2} - 1) / (\sqrt{1 + 4R^2} + 1)$. For more details, see Kalman smoother in the Appendix.

Given our generative model, the Kalman smoother is the optimal filter to denoise the measured eye position. The EyeLink eye tracker software also has a denoising option, called ‘‘Heuristic filter,’’ which is based on an algorithm by Stampe (1993). This filter is suboptimal given our generative model and therefore, assuming that our generative model is realistic, will perform worse in separating signal from noise than the Kalman smoother.

Step 3: Sample from the posterior over the eye state time series \mathbf{C}

We draw samples from the posterior $p(\mathbf{C}|\hat{\mathbf{z}}, \sigma_0, \sigma_1, d_1, \sigma_z)$ using MCMC sampling with Metropolis–Hastings acceptance probabilities. Using the prior over velocities, Equation 5, and the property of the delta function, we can compute the posterior as

$$p(\mathbf{C}|\hat{\mathbf{z}}) = \prod_i \int p(v_{\tau_i} | C_{\tau_i}) e^{-\frac{(\tau_{i+1} - \tau_i)v_{\tau_i}^T \hat{z}_{\tau_i}}{2\sigma_z^2} + \frac{(z_{\tau_{i+1}} - z_{\tau_i})^T v_{\tau_i}}{\sigma_z^2}} dv_{\tau_i}. \quad (14)$$

Each term in this product is an independent integral over v_{τ_i} , which depends only on $z_{\tau_{i+1}} - z_{\tau_i}$, $\tau_{i+1} - \tau_i$, and implicitly the eye state C_{τ_i} through the parameters d and σ in the prior $p(v_{\tau_i})$. We can therefore write

$$p(\hat{\mathbf{z}}|\mathbf{C}) = \prod_i I(z_{\tau_{i+1}} - z_{\tau_i}, \tau_{i+1} - \tau_i, d_{C_{\tau_i}}, \sigma_{C_{\tau_i}}), \quad (15)$$

with

$$I(\Delta z, \Delta \tau, d, \sigma) = \frac{1}{2\pi} \iint p(r|d, \sigma) \times e^{-\frac{\Delta z_r^2}{2\sigma_z^2} + \frac{1}{\sigma_z^2} \Delta z \cdot \begin{pmatrix} r \cos \theta \\ r \sin \theta \end{pmatrix}} dr d\theta. \quad (16)$$

This integral can be evaluated to

$$I(\Delta z, \Delta \tau, d, \sigma) = \frac{2^{\frac{1-d}{2}}}{\sigma^{d+1} \Gamma(\frac{d+1}{2})} \left(\frac{\sigma_z^2}{\|\Delta z\|} \right)^{d+1} \times A \left(d, \frac{\sigma_z^4}{2\|\Delta z\|^2} \left(\frac{1}{\sigma^2} + \frac{\Delta \tau}{\sigma_z^2} \right) \right), \quad (17)$$

where

$$A(d, \alpha) = \int_0^\infty s^d e^{-\alpha s^2} I_0(s) ds, \quad (18)$$

with I_0 the modified Bessel function of the first kind. (For details, see the Appendix under From Equation 16 to Equation 17.) We solve this integral analytically in the limits $a \rightarrow 0$ and $a \rightarrow \infty$, which correspond to $\|\Delta z\| \rightarrow \infty$ and $\|\Delta z\| \rightarrow 0$, respectively. For intermediate α , we solve the integral numerically.

The details of the MCMC algorithm we use to sample from the posterior $p(\mathbf{C}|\hat{\mathbf{z}})$ are presented in the Appendix under MCMC sampling. The MCMC algorithm returns a set of 40 samples $\hat{\mathbf{C}}^j$. On the last iteration, we convert these samples into a probability time series by averaging them. For some applications, we subsequently transform from probabilities to binary values by thresholding at 0.5. This operation minimizes the absolute-value cost function

$$\text{Cost}(\hat{\mathbf{C}}, \mathbf{C}) = \sum_{t=1}^T |\hat{C}_t - C_t|. \quad (19)$$

Step 4: Estimate the velocity distribution parameters

We infer the global velocity distribution parameters σ_0 , σ_1 , and d_1 by maximizing $p(\hat{\mathbf{z}}|\hat{\mathbf{C}}^j, \sigma_0, \sigma_1, d_1, \sigma_z)$ with a grid search for each sample $\hat{\mathbf{C}}^j$ and then taking the median across samples. The grid ranges are [0.1, 100] deg/s for σ_0 and σ_1 and [1.1, 5] for d_1 .

Alternative algorithms

EK velocity threshold

The EK algorithm starts by averaging the measured eye position time series across a triangular sliding window and differentiating it to obtain a velocity time series. The algorithm detects microsaccades whenever the velocity exceeds a threshold η_x for the horizontal dimension and η_y for the vertical dimension for a sufficiently long duration. The thresholds are adaptively set as a multiple of the standard deviation of the eye movement velocity, using a median-based estimate of the standard deviation:

$$\eta_{x,y} = \lambda \left(\text{median}(v_{x,y}^2) - \text{median}(v_{x,y})^2 \right). \quad (20)$$

The size of the sliding window, the multiplier λ , and the minimum duration are free parameters set by the user. Of these, λ tends to have the largest effect on the detected microsaccades.

In their original article, Engbert and Kliegl (2003) used a triangular sliding-window size of 6 for 500-Hz data, a duration threshold of 12 ms, and a relatively conservative velocity threshold multiplier of $\lambda = 6$. This value is used in most subsequent studies. Other studies have used a more liberal threshold (Engbert & Mergenthaler, 2006). We consider two particular cases of $\lambda = 3$ and $\lambda = 6$, which we will refer to as EK3 and EK6, respectively.

Unsupervised clustering

More recently, Otero-Milan et al. (2014) have proposed a threshold-free microsaccade detection method, which we will refer to as OM. It uses an unsupervised clustering algorithm, k -means, to group putative events obtained from the EK algorithm into clusters of microsaccades or drift. The algorithm separates drift and microsaccade events using three features: peak velocity, initial acceleration peak, and final acceleration peak. Here we use the implementation provided by Otero-Milan et al. as obtained from their website (<http://smc.neuralcorrelate.com/sw/microsaccade-detection/>).

EyeLink experimental methods

This study was approved by the New York University Institutional Review Board, in accordance with the Declaration of Helsinki. Five subjects (two women and three men) of median age 26 years (range: 20–36 years) performed the task after providing informed consent.

Apparatus

We displayed stimuli on a 21-in. Sony GDMF520 CRT monitor (resolution: 1280 × 960 pixels, refresh rate: 100 Hz). Subjects used a headrest located approximately 57 cm from the screen. The screen background was gray (57 cd/m²). An Apple iMac computer running MATLAB 7.1 (MathWorks, Natick, MA) with the Psychtoolbox (Brainard, 1997; Kleiner et al., 2007; Pelli, 1997) and EyeLink (Cornelissen, Peters, & Palmer, 2002) extensions controlled stimulus presentation and response collection. We recorded eye movements using a remote infrared video-oculographic system (EyeLink 1000; SR Research, Ltd., Mississauga, Ontario, Canada) with a 1-kHz sampling rate, precision of 0.01 deg, and average accuracy of 0.25°–0.5 deg,

according to the manual (but see Holmqvist et al., 2011; Poletti & Rucci, 2016). We acquired eye position data with the EyeLink software. We set the “Heuristic filter” option to off to obtain the raw data.

Procedure

Subjects performed a delayed-estimation-of-orientation delayed-estimation task, as introduced by Wilken and Ma (2004). A trial sequence started with the appearance of a central white fixation cross subtending a visual angle of 0.3 deg, which lasted for 500 ms or until the subject successfully fixated. We defined fixation to be successful when the eye position remained within a 2 deg circle centered at the fixation cross. Next, two stimuli appeared 6 deg to the right and left of the central fixation cross. The stimuli were circularly windowed gratings with radius 0.35 deg, spatial frequency 2.5 cycles/deg, and uniformly drawn orientation. The stimuli stayed on the screen for 11 frames (about 110 ms), followed by a delay period of 1000 ms. If the subject broke fixation at any point during the stimulus or delay period, the trial was aborted and a new trial sequence started. We eliminated these trials from our data set. After the delay period, the subject was probed about one of the locations and responded by using the mouse to estimate the orientation. More precisely, when the subject moved the mouse, a windowed grating appeared inside that circle. The subject had to rotate it using the mouse to match the orientation of the grating that had been in that location, and then press the space bar to submit a response. The experiment consisted of eight blocks, each consisting of 60 completed (nonaborted) trials with 30-s breaks in between blocks.

Dual Purkinje Image experimental methods

The Dual Purkinje Image (DPI) eye tracker data were made available by Martina Poletti and Michele Rucci. Their study was approved by the institutional review board of Boston University. The method and data were described in detail elsewhere (Cherici, Kuang, Poletti, & Rucci, 2012); we summarize them here.

Apparatus

Stimuli were presented on a custom-developed system for flexible gaze-contingent display control on a fast-phosphor CRT monitor (Iiyama HM204DT) with a vertical refresh rate of 150 Hz. The movements of the right eye were measured with a Generation 6 DPI eye tracker (Fourward Technologies, Buena Vista, VA) at a 1-kHz sampling rate. While most video-based eye trackers detect only the first corneal reflection (Purkinje

reflection), DPI eye trackers detect both the first and fourth Purkinje reflections, allowing discrimination between eye movements of rotation and translation. The DPI eye tracker has a high precision, of 0.006° (Cherici et al., 2012; Crane & Steele, 1985).

Procedure

Subjects observed the screen with the right eye while wearing an eye patch on their left eye. A dental-imprint bite bar and a headrest prevented head movements. Subjects were asked to maintain sustained fixation while looking at a marker displayed on the screen. Two subjects performed the task.

Results

Comparison of algorithms on simulated data

We created 36 data sets with eye position time series of length $T = 60,000$ ms according to the generative model. We created every combination of six chosen values of motor noise and six values of measurement noise. We fixed the velocity distribution parameters at $\sigma_0 = 0.3^\circ/\text{s}$, $d_1 = 4.4$, and $\sigma_1 = 30^\circ/\text{s}$, to approximate realistic microsaccade kinematics (Engbert, 2006). We inferred the eye state time series with the BMD algorithm and the standard EK algorithm, which uses a velocity threshold multiplier of 6 (referred to as EK6). After thresholding the BMD inferences, we evaluated their performance in terms of the hit rate (defined as the proportion of 1s correctly identified in the **C** time series) and the false-alarm rate (the proportion of 1s wrongly identified in the **C** time series; Figure 4). While the velocity distribution parameters were not perfectly recovered (Figure A3), the BMD hit rates were very high (Figure 4A). The hit rate of the BMD algorithm decreases with increased motor noise, as in standard signal detection theory, but it is remarkably robust to increased measurement noise. By contrast, the hit rate of EK6 is lower and more affected by the noise level. In EK6, the false-alarm rate decreases with increasing noise because the threshold adapts to the noise level. Across the board, BMD has false-alarm rates comparable to EK6's but much higher hit rates, especially at high noise.

For a more comprehensive evaluation, we also compare BMD against OM and an EK variant with a velocity threshold multiplier $\lambda = 3$ (EK3; Figure 5). As performance metrics, we use the error rate in identifying the eye state at every time point, the number of microsaccades per unit time, and the hit and false-alarm rates. BMD has a lower error rate than all alternative algorithms in 30 out of 36 noise levels. As in Figure 4, the improvement of BMD over alternative

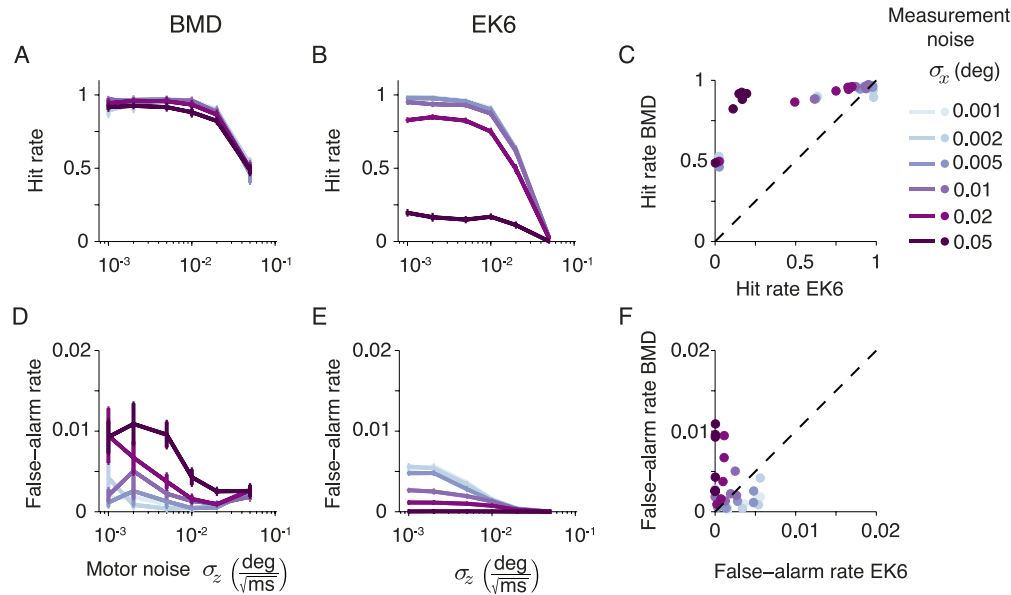


Figure 4. Performance of the BMD and EK6 algorithms on simulated data. (A) Hit rates of the BMD algorithm as a function of the motor noise σ_z for several values of measurement noise σ_x . Points and error bars represent means and standard errors across eight simulated data sets. (B) Hit rates of the EK6 algorithm. (C) Scatterplot comparing hit rates of both algorithms. Each point corresponds to a different pair (σ_z, σ_x) . (D–F) The same as (A–C) for false-alarm rates.

algorithms is larger for higher noise. BMD has a hit rate close to 1 in all but the highest level of motor noise, whereas the false-alarm rate is comparable to those of other algorithms. The BMD algorithm is more robust than all other algorithms: Its hit rate and microsaccade rate vary only weakly with increasing measurement noise.

As expected from signal detection theory, there is a trade-off between false alarms and misses in the EK algorithm. EK6 is too conservative, leading to more misses than BMD; however, EK3 is too permissive and has more false alarms. To test whether the EK algorithm with any threshold can match BMD's performance, we compute a receiver operating characteristic (ROC; Figure 6). At low noise, both BMD and EK perform close to perfectly. Overall, BMD outperforms or matches EK at all other noise levels. However, in cases where BMD performance matches that of EK, BMD intersects the EK ROC curves for different thresholds at different noise levels. This makes choosing a single best threshold problematic.

Applications to real data

The results on simulated data suggest that BMD recovers microsaccades more faithfully than alternative algorithms, especially at high noise. This confirms that the approximations in our inference algorithm do not significantly impair its performance. However, we created data according to our generative model, so we expected the BMD algorithm to be superior. Next, we

apply our algorithm to real eye-tracking data measured with two different eye trackers: EyeLink and DPI.

EyeLink data

In Figure 7, we show six example measured eye position sequences and the inferred change points by BMD and EK6. When the signal-to-noise ratio is high (Figure 7A through C), BMD generally infers the same microsaccades as EK6. Additionally, BMD returns a probabilistic judgment of the beginning and end time of the microsaccade. In some cases, BMD detects a small microsaccade immediately after a larger one, in the opposite direction (Figure 7B, C), corresponding to the overshoot. For low signal-to-noise data (Figure 7D through F), the BMD algorithm tends to detect potential microsaccades that EK6 misses, but they could be false positives. BMD assigns low confidence to its judgments in ambiguous cases like Figure 7D and F.

The microsaccades detected by BMD have similar kinematics as previously reported (Engbert, 2006; Figure A4). The inferred velocity and duration distributions of BMD and EK6 are similar, except for the duration cutoff in EK6. Most importantly, the microsaccades detected by BMD follow the main sequence: Their amplitude is monotonically related to their peak velocity (Zuber, Stark, & Cook, 1965). As in Engbert and Kliegl (2003), we consider the approximate recovery of the main-sequence relationship to be evidence for the validity of our detection algorithm. Our algorithm estimates the mean velocity for drift as 0.1253 deg/s for all but one subject, and 22.64 ± 8.4 deg/s (mean and

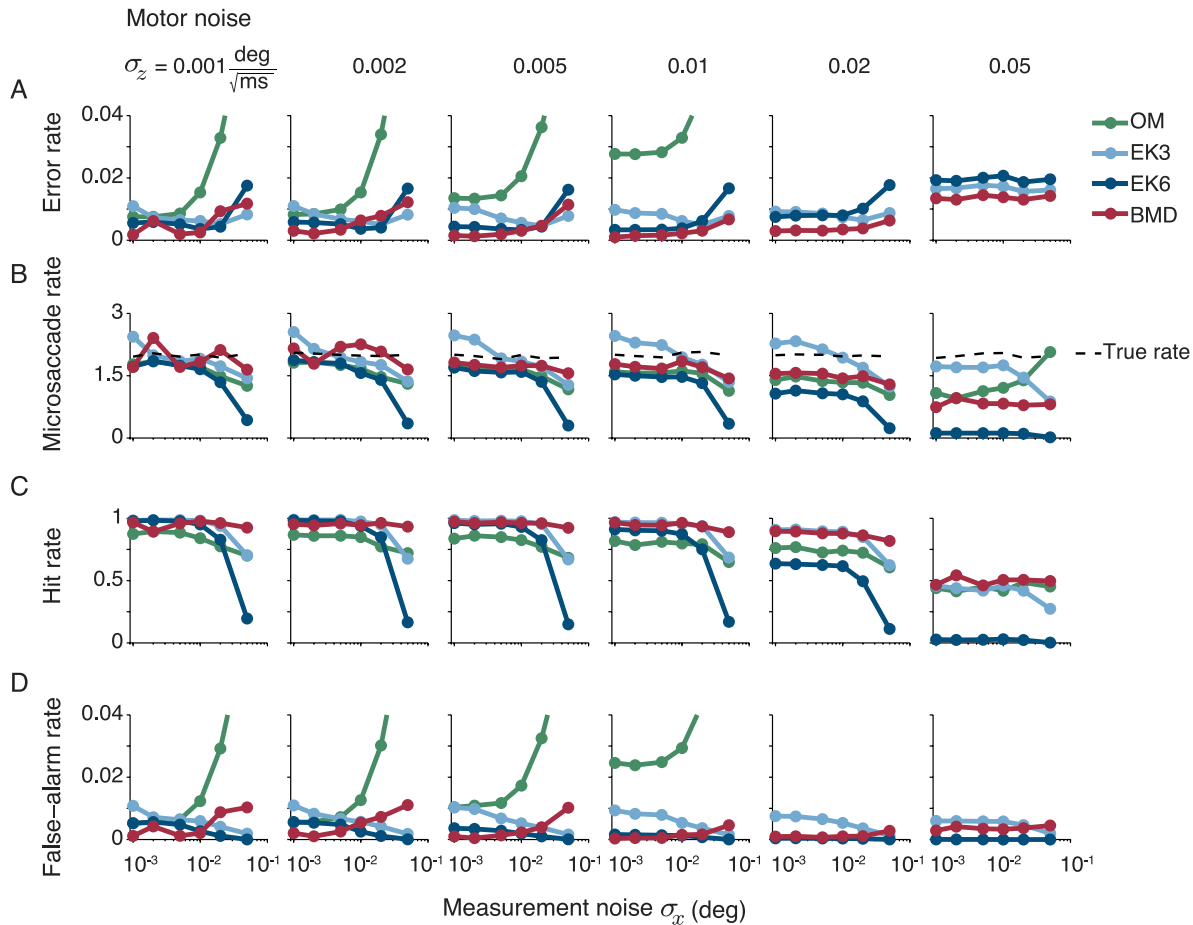


Figure 5. Performance of several algorithms on simulated data. Colors represent four different algorithms: OM, two versions of EK, and BMD. We evaluate performance with four different metrics: (A) error rate, (B) microsaccade rate, (C) hit rate, and (D) false-alarm rate. The motor noise σ_z increases across columns, and the measurement noise σ_x increases within each subplot. BMD has the lowest error rates at high noise levels and is the most robust against increases in both σ_z and σ_x . BMD hit rates and microsaccade rates are the most robust against increases in either σ_z or σ_x without a major increase in false-alarm rates.

standard error across subjects) for microsaccades. These values are in line with literature reports: mean drift velocity of $0.85^\circ/s$ (Poletti, Aytikin, & Rucci, 2015) and below $0.5^\circ/s$ (Engbert, 2006; Rolfs, 2009), and mean microsaccade velocity of ~ 30 deg/s.

Overall, BMD detects more microsaccades than EK6 for all five subjects (Figure A5). This difference can be dramatic: For two subjects (S3 and S4), EK6 infers no microsaccades at all, whereas BMD infers microsaccade rates up to 2.1 per second. This further suggests that EK6 is too conservative and misses microsaccades when the measurement noise is high. The other algorithms (OM and EK3) are less conservative, but their inferred microsaccade rates vary widely, reinforcing the need for a more principled microsaccade detection algorithm.

Finally, we ask how dependent the microsaccade rate inferred by BMD is on the choice of parameters in the priors over the frequency and duration of microsaccades. We vary both k_0 and k_1 by an order of

magnitude and show that the inferred microsaccade rate is approximately constant (Figure A6), making the BMD algorithm robust to the choice of the prior in a plausible range.

These results suggest that BMD outperforms EK6 with real data. Specifically, BMD detects many plausible microsaccades that EK6 misses, especially when their amplitude is small and the noise is high. However, an alternative interpretation is that BMD detects false positives. We cannot distinguish these possibilities because, in contrast to the simulated data, we do not know the ground truth. In general, we know that all four algorithms give different inferences, but without ground truth we have no way of establishing which one is better.

DPI data

To address this problem, we use another data set, provided by Poletti and Rucci (Cherici et al., 2012).

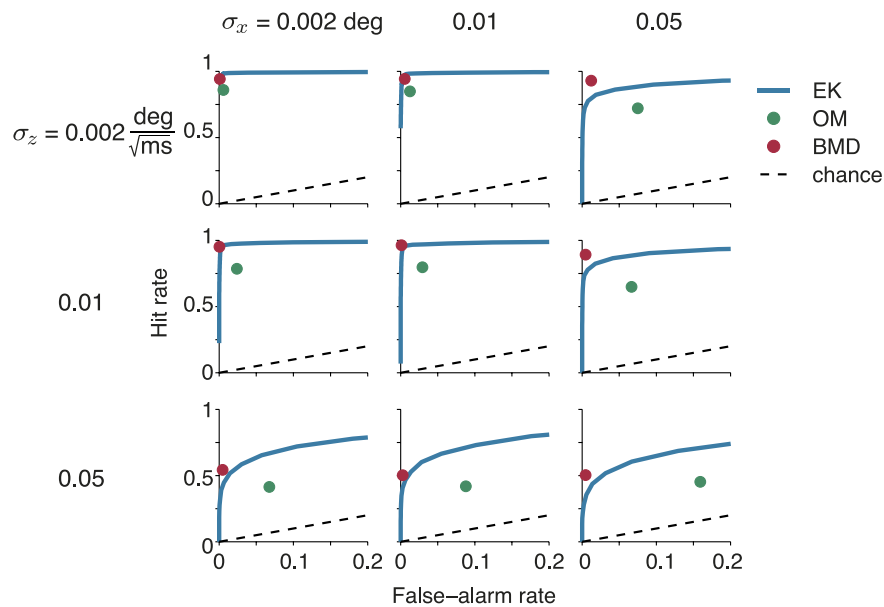


Figure 6. Performance of the algorithms on simulated data visualized relative to the EK ROC curve. The red and green dots represent the combination of hit rate and false-alarm rate for BMD and OM, respectively. The EK ROC curves were created with different values of the threshold multiplier λ . EK3 and EK6 correspond to points on the curve. For all noise levels tested, including the ones presented here, BMD either outperforms both OM and EK or matches EK.

These eye movements were measured with the more precise DPI eye tracker (Cherici et al., 2012; Crane & Steele, 1985). Indeed, BMD infers that the geometric mean of the measurement noise level in DPI data is almost an order of magnitude lower than in EyeLink data (Table 2). In simulated data with the same noise level as BMD infers for DPI, all algorithms perform close to perfectly. In view of this high performance, we

can treat the microsaccades inferred from the raw DPI data (averaged across algorithms) as ground truth. Our strategy is to artificially add increasing amounts of measurement noise to the raw data and see how much the inference of each algorithm degrades as a result. This allows us to compare the robustness of the algorithms with an objective metric.

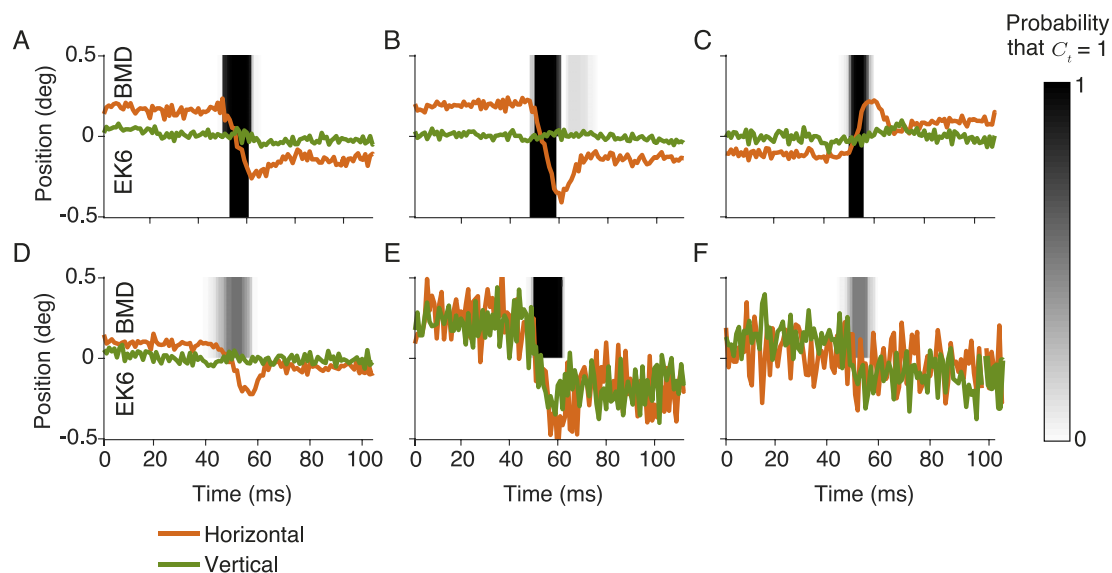


Figure 7. Inferences of microsaccades by BMD and EK6 on example eye position sequences measured with the EyeLink eye tracker. The black and white shading represents the probability that the eye is in a microsaccade state, with black indicating certainty. Every subplot shows the BMD inference in the top half and the EK6 inference in the bottom half. (A–C) Often, BMD and EK6 infer nearly identical microsaccade sequences. (D–F) BMD infers potential microsaccades that EK6 misses, especially when they are small or noisy.

Subject	$\hat{\sigma}_z$ ($\frac{\text{deg}}{\sqrt{\text{ms}}}$)	$\hat{\sigma}_x$ deg	\hat{d}_1	$\hat{\sigma}_0$ (deg/s)	$\hat{\sigma}_1$ (deg/s)
EyeLink S1	0.01397	0.0249	1.1	0.1	27.5
EyeLink S2	0.00723	0.0165	1.1	0.1	13.1
EyeLink S3	0.01317	0.10379	4.961	0.1	7.58
EyeLink S4	0.01265	0.0182	1.1	0.1	0.1
EyeLink S5	0.00637	0.02327	1.1	15.8	33.1
DPI S1	0.014	0.025	1.1	0.1	27.5
DPI S2	0.014	0.025	1.1	0.1	27.5

Table 2. Parameter inference.

We compare the error rates as well as the micro-saccade rates, hit rates, and false-alarm rates between BMD, OM, EK3, and EK6 (Figure 8). BMD outperforms EK3, EK6, and OM at all except the lowest noise levels. In particular, at measurement noise levels comparable to the ones inferred in EyeLink data (0.02 deg), the error rate for EK6 is 3.22% (averaged across subjects), while BMD achieves 1.48%—a 54% improvement. Note that all algorithms have low error rates, primarily because microsaccades are rare. As in simulated data, we compare BMD to EK with different thresholds by plotting an ROC curve; BMD outperforms EK regardless of its threshold (Figure 9).

Variants of BMD

A common risk in Monte Carlo methods is that the samples aggregate near potential local maxima of the posterior and miss the global maximum. One method to mitigate this problem, albeit at increased compu-

tational cost, is parallel tempering (Earl & Deem, 2005; Newman & Barkema, 1999). BMD with parallel tempering does not significantly outperform BMD either for simulated data (Figures 10 and 11) or for real DPI data with added noise (Figure 12), suggesting that the posterior probability landscape did not contain many local maxima. To investigate which components of our method are necessary for its performance, we compare BMD against three reduced variants. We obtain the first variant by reducing the number of iterations in the approximate inference method from six to two. The second variant has only one iteration, which is equivalent to applying a Kalman smoother to obtain $\hat{\mathbf{z}}$ from \mathbf{x} , then sampling from $p(\mathbf{C}|\hat{\mathbf{z}})$.

Finally, a third version, $\text{BMD}_{\text{reduced}} + \text{threshold}$, starts with Steps 0–2 of the BMD algorithm. However, instead of sampling from the posterior $p(\mathbf{C}|\hat{\mathbf{z}})$ in Step 3, it estimates \mathbf{C} by applying a Kalman smoother (after the Kalman smoother of Step 2) to $\hat{\mathbf{z}}$ to obtain a smoothed eye position time series, differentiating that to obtain eye velocities, and thresholding the velocity

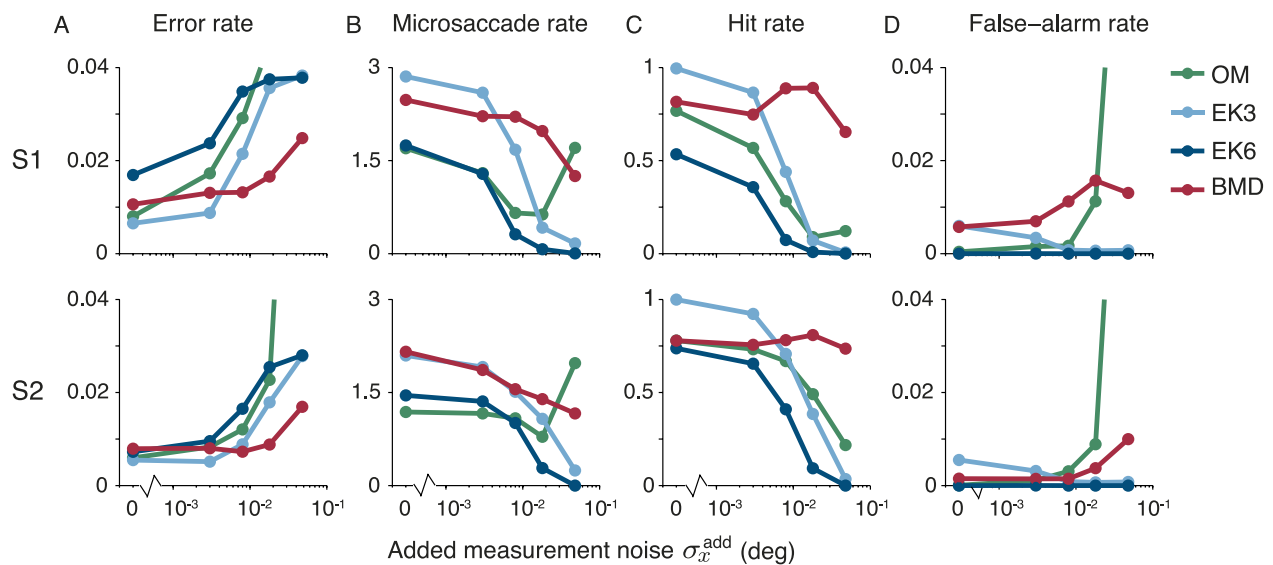


Figure 8. Performance of the algorithms on DPI data. We took DPI data from two subjects (rows), collected by Cherici et al. (2012), and artificially added measurement noise to the eye position traces. Colors represent algorithms. BMD shows the highest robustness to adding measurement noise; specifically, error rates are lowest and hit rates tend to stay the same.

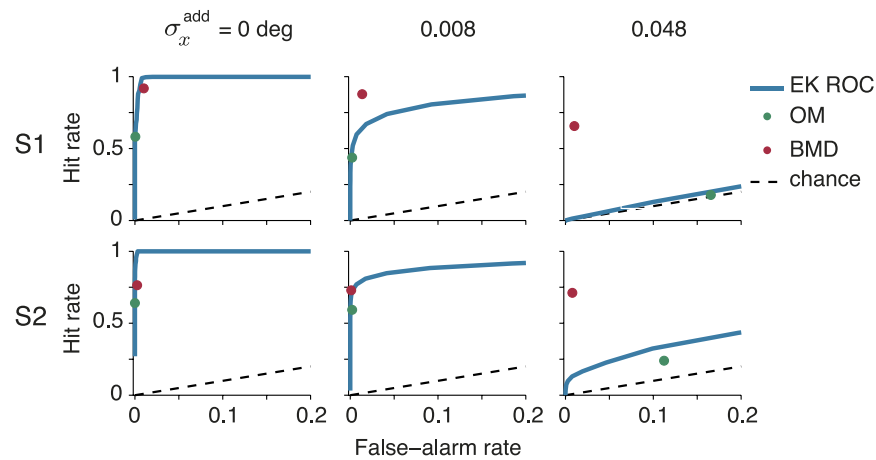


Figure 9. Performance of the algorithms on DPI data with added noise relative to the EK ROC curve. We show the same two subjects (S1 and S2) as in Figure 8. The level of the added measurement noise varies across columns. The EK ROC curves were created with different values of the threshold multiplier λ . EK3 and EK6 correspond to points on the curve. As more measurement noise is added, BMD outperforms EK and OM by larger amounts.

time series (Figure 13). We fix the window size of the Kalman smoother to 5.32 ms and use a threshold which scales linearly with the inferred motor noise level: $\text{threshold} = a\hat{\sigma}_z + b$, with $a = 32\sqrt{s}^{-1}$ and $b = 1$ deg/s. We chose these values to approximately match the output of BMD and $\text{BMD}_{\text{reduced}} + \text{threshold}$ in real and simulated data. This method performs about as well as the full inference algorithm. However, it is unprincipled, does not return a probabilistic estimate, and cannot be directly extended to more sophisticated generative models.

Discussion

We developed a Bayesian algorithm for detecting microsaccades among drift/tremor; it returns probabilistic rather than binary judgments. Given our assumptions about the statistical process generating a measured eye position time series, this algorithm is optimal. BMD has lower error rates than the algorithms proposed by Engbert and Kliegl (2003) and Otero-Millan et al. (2014), especially at high noise. This is a particularly useful feature given the relatively high measurement noise of current infrared eye trackers. However, a hybrid between BMD and velocity-threshold algorithms, $\text{BMD}_{\text{reduced}} + \text{threshold}$, can sometimes approach BMD's performance.

In our model, microsaccades are defined through prior probability distributions over velocity and duration that are different from those for drift/tremor (Figure 2). This definition contrasts with the more common one that uses an arbitrary velocity threshold. The BMD algorithm (and the actual code) allows researchers to easily build in their own prior beliefs and

state clearly which of their findings depend on those beliefs.

We designed the BMD algorithm for off-line analysis of eye tracker data. An online detection method—for example for closed-loop experiments that require real-time detection of microsaccades, such as in Chen and Hafed (2013) and Yuval-Greenberg et al. (2014)—would require a modified inference algorithm. If it is crucial to detect microsaccades online, we recommend using $\text{BMD}_{\text{reduced}} + \text{threshold}$, with a Kalman filter (only the forward filter) instead of the Kalman smoother.

We designed and tested BMD for detecting microsaccades in fixational eye movement data obtained under head-fixed conditions, where the fixation point does not move. Would the algorithm readily apply to other kinds of eye movement data? First, head-free recordings are sometimes used in order to better mimic naturalistic conditions (Benedetto, Pedrotti, & Bridgeman, 2011; Martinez-Conde, Macknik, Troncoso, & Dyar, 2006; Poletti et al., 2015). In theory, our algorithm is suitable for inferring microsaccades in head-free recordings. However, studies have reported higher velocities for drift in head-free fixation (Poletti et al., 2015; Skavenski, Hansen, Steinman, & Winterson, 1979)—for example, on average 1.5°/s for head-free versus 0.85 deg/s for head-fixed in Poletti et al. (2015). Therefore, we expect the velocity distributions presented in Figure 3B to be less separable, which in turn would impair microsaccade detection. Second, our algorithm is not immediately applicable to smooth pursuit, in which the eye continuously tracks the motion of an object. Santini, Fuhl, Kubler, and Kasneci (2016) used a Bayesian classification algorithm to separate drift, saccades, and smooth pursuit based on features derived from the eye position data, but this

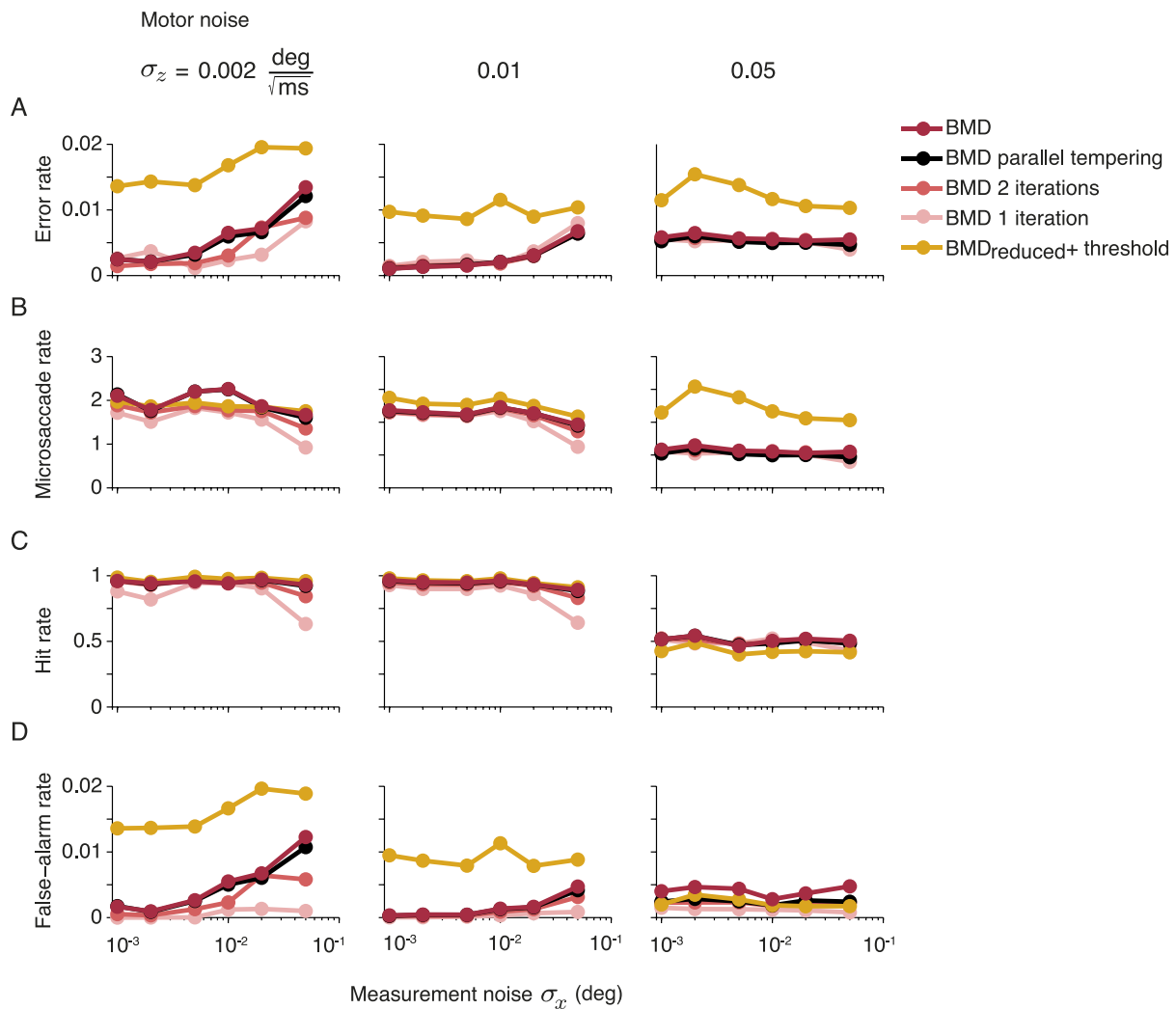


Figure 10. Performance of BMD variants on simulated data. The variants we examine are BMD with parallel tempering, BMD with fewer iterations (two and one), and a reduced variant of BMD with a threshold ($BMD_{reduced} + \text{threshold}$). In the latter model, the threshold is dependent on motor noise through the equation $\text{threshold} = 32\sqrt{5}^{-1}\hat{\sigma}_z + 1 \text{ deg/s}$, chosen because it gave the lowest error rates on DPI data. The motor noise σ_z increases across columns, and the measurement noise σ_x increases within each subplot. BMD with parallel tempering is only a slight improvement over BMD, while BMD performs slightly better than BMD with two and one iterations. $BMD_{reduced} + \text{threshold}$ only performs comparably with BMD under high motor and measurement noise.

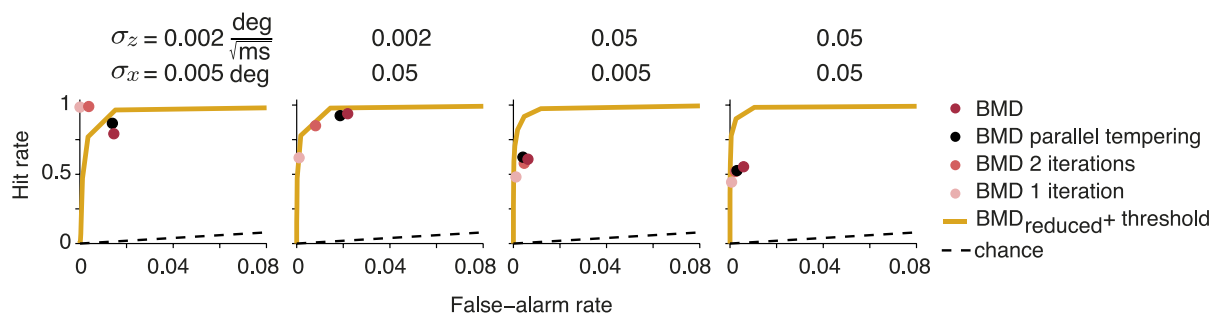


Figure 11. Performance of BMD variants on simulated data visualized relative to the ROC curves for $BMD_{reduced} + \text{threshold}$. Here we show hit rates with false-alarm rates points for BMD variants and ROC curves for $BMD_{reduced} + \text{threshold}$ for several values of the threshold multiplier. In contrast to Figure 10, where we choose one threshold, here we see that the BMD-variant points are on the ROC curves at low noise (first two subplots). However, as the motor and measurement noise increase (last two subplots), the full ROC curve can reach higher performance than the variant BMD algorithms.

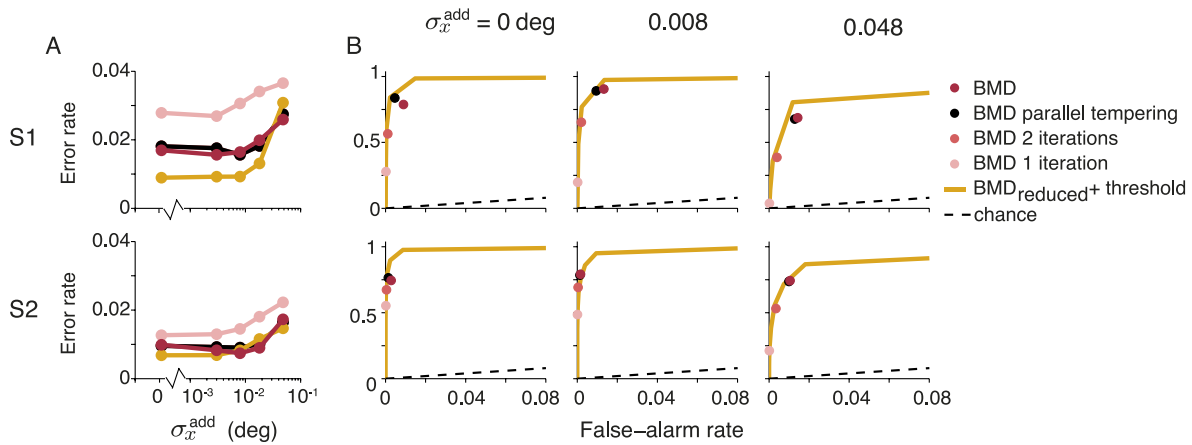


Figure 12. Performance of BMD variants on DPI data to which we add measurement noise. We show the same two subjects (S1 and S2) as in Figure 8. We measure performance on the same metrics as before. For brevity, we show in (A) the error rates with fixed threshold. In (B), we show the hit rates with false-alarm rates for the variant BMD algorithms relative to the ROC curve for BMD_{reduced} + threshold. Adding parallel tempering to BMD makes little difference. Using fewer iterations negatively affects the hit rate. BMD_{reduced} + threshold gives slightly lower error rates and seems to match the performance of BMD.

algorithm does not have a generative model of the entire time series. In our approach, we could amend our generative model to include a third state, smooth pursuit, with different duration and velocity distributions, but this would require a more complex inference algorithm.

Caveat

BMD outperforms the alternative algorithms for simulated and real data, on average. However, it sometimes makes idiosyncratic mistakes. In simulated data with low noise, for some visually salient microsaccades, BMD incorrectly identifies a microsaccade as mostly drift, with very short microsaccades immediately before and after (see Figure A7). This mistake happened 17 times in 36×8 simulations. This particular mistake coincides with instances when the algorithm overestimates σ_0 and underestimates d_1 , which makes the drift and microsaccade velocity distributions less separable. We could solve the incorrect microsaccade inference with some post hoc processing; however, this would introduce arbitrariness. Instead, we accept this as a failure mode of our algorithm: rare, exclusively at low noise, and easily detectable.

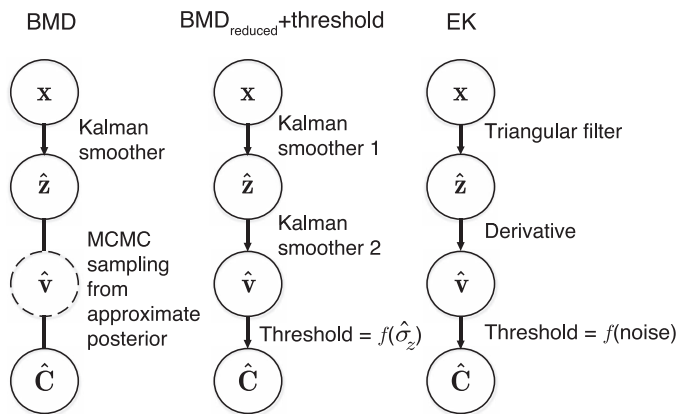


Figure 13. Schematic comparison of microsaccade detection algorithms. All algorithms first perform a filtering operation to eliminate the noise from the measured eye position time series x . EK removes noise with a heuristically chosen filter; in contrast, BMD and BMD_{reduced} + threshold use a Kalman smoother, which optimally eliminates measurement noise in our generative model. EK estimates the eye state time series by taking the derivate of the eye position to yield the eye-velocity time series, then thresholding those velocities. BMD, on the other hand, marginalizes over velocity and samples from the posterior distribution over eye states. BMD_{reduced} + threshold uses a second Kalman smoother to eliminate some of the motor noise and ultimately uses a velocity threshold which depends on the motor noise.

Conceptual extensions of the algorithm

The inferred microsaccades depend on assumptions in our generative model, which are simplistic and incorrect. We can flexibly adjust these assumptions in the generative model and modify the BMD algorithm accordingly.

Correlated state durations

Our generative model assumes that the durations over which the eye remains in either state are independent. We can relax this assumption by changing the duration prior; this does not affect the likelihood.

Binocular data

Our algorithm is designed to operate on monocularly recorded eye position time series, but it can be extended to binocular data. This can be accomplished simply by changing all position and velocity vectors from 2-D to 4-D and adjusting the noise covariance matrices.

Tremor and saccades

We can add tremor (low-amplitude, high-frequency oscillations; Ratliff & Riggs, 1950) or saccades as additional states in our generative model, given statistical descriptions of these processes. However, it has been argued that microsaccades and saccades are produced by the same process (Hafed & Krauzlis, 2012; Otero-Millan, Troncoso, Macknik, Serrano-Pedraza, & Martinez-Conde, 2008; Zuber et al., 1965).

Microsaccade dynamics

Our generative model assumes that the eye velocity is constant throughout each microsaccade or drift state, resulting in linear microsaccade trajectories. However, real microsaccades, such as the ones in Figures 1 and 7, have a smooth velocity profile, for which specific shapes have been proposed (Abadi & Gowen, 2004). We could incorporate a template for the characteristic temporal profile of microsaccades into our generative model, which would require only minor changes to the inference algorithm.

Correlated measurement noise

We assumed that the measurement noise is uncorrelated across time, which allowed us to estimate the eye position using a Kalman smoother (Step 2). We can incorporate noise correlations into our generative model if we replace the Kalman smoother in the inference algorithm with a Gaussian process estimator.

Conclusion

We conclude that Bayesian methods could significantly improve microsaccade detection. The BMD algorithm both is more principled and produces the lowest errors on both simulated and real data. In particular, it is substantially more robust to measurement noise (which is especially useful given the relatively high measurement noise of current infrared eye trackers). The BMD algorithm can be extended to build in more knowledge about the processes underlying microsaccades.

Keywords: fixational eye movements, microsaccades, detection, Bayesian model

Acknowledgments

WJM was supported by Grant R01EY020958 from the National Eye Institute and Grant W911NF1410476 from the Army Research Office. We thank Martina Poletti and Michele Rucci for providing the DPI data set and for valuable discussions. We thank Marisa Carrasco and her lab members, especially Bonnie Lawrence, Ian Donovan, and Nick Murray-Smith, for advice and access to their eye tracker. In addition, we thank Eero Simoncelli, Roozbeh Kiani, Luigi Acerbi, Emin Orhan, Aspen Yoo, and Will Adler for useful conversations. We thank Paul Bays for sharing with us a data set that allowed us to test an early version of our algorithm.

*AM and BvO contributed equally to this article.

Commercial relationships: none.

Corresponding author: Andra Mihali.

Email: alm652@nyu.edu.

Address: Center for Neural Science, New York University, New York, NY, USA.

References

- Abadi, R. V., & Gowen, E. (2004). Characteristics of saccadic intrusions. *Vision Research*, *44*(23), 2675–2690, doi:10.1016/j.visres.2004.05.009.
- Abramowitz, M., & Stegun, I. (1965). *Handbook of mathematical functions*. Mineola, NY: Dover.
- Benedetto, S., Pedrotti, M., & Bridgeman, B. (2011). Microsaccades and exploratory saccades in a naturalistic environment. *Journal of Eye Movement Research*, *4*(2), doi:10.16910/jemr.4.2.2.
- Bettenbuehl, M., Paladini, C., Mergenthaler, K., Kliegl, R., Engbert, R., & Holschneider, M. (2010). Microsaccade characterization using the continuous wavelet transform and principal component analysis. *Journal of Eye Movement Research*, *3*(5), doi:10.16910/jemr.3.5.1.
- Bishop, C. M. (2006). *Pattern recognition and machine learning*. Secaucus, NJ: Springer-Verlag.
- Brainard, D. H. (1997). The Psychophysics Toolbox. *Spatial Vision*, *10*(4), 433–436.
- Chen, C.-Y., & Hafed, Z. M. (2013). Postmicrosaccadic enhancement of slow eye movements. *The Journal*

- of *Neuroscience*, 33(12), 5375–5386, doi:10.1523/jneurosci.3703-12.2013.
- Cherici, C., Kuang, X., Poletti, M., & Rucci, M. (2012). Precision of sustained fixation in trained and untrained observers. *Journal of Vision*, 12(6):31, 1–16, doi:10.1167/12.6.31. [PubMed] [Article]
- Ciuffreda, K. J., & Tannen, B. (1995). *Eye movement basics for the clinician*. St. Louis, MO: Mosby.
- Collewijn, H., & Kowler, E. (2008). The significance of microsaccades for vision and oculomotor control. *Journal of Vision*, 8(14):20, 1–21, doi:10.1167/8.14.20. [PubMed] [Article]
- Cornelissen, F. W., Peters, E. M., & Palmer, J. (2002). The Eyelink Toolbox: Eye tracking with MATLAB and the Psychophysics Toolbox. *Behavior Research Methods, Instruments, & Computers*, 34(4), 613–617, doi:10.3758/bf03195489.
- Cornsweet, T. N. (1956). Determination of the stimuli for involuntary drifts and saccadic eye movements. *Journal of the Optical Society of America*, 46(11), 987–988.
- Crane, H. D., & Steele, C. M. (1985). Generation-V Dual Purkinje Image eyetracker. *Applied Optics*, 24(4), 527–537, doi:10.1364/ao.24.000527.
- Daye, P. M., & Optican, L. M. (2014). Saccade detection using a particle filter. *Journal of Neuroscience Methods*, 235, 157–168, doi:10.1016/j.jneumeth.2014.06.020.
- Ditchburn, R. W., & Ginsborg, B. L. (1953). Involuntary eye movements during fixation. *The Journal of Physiology*, 119(1), 1–17.
- Earl, D. J., & Deem, M. W. (2005). Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7(23), 3910–3916.
- Engbert, R. (2006). Microsaccades: A microcosm for research on oculomotor control, attention, and visual perception. *Progress in Brain Research*, 154, 177–192.
- Engbert, R., & Kliegl, R. (2003). Microsaccades uncover the orientation of covert attention. *Vision Research*, 43(9), 1035–1045.
- Engbert, R., & Mergenthaler, K. (2006). Microsaccades are triggered by low retinal image slip. *Proceedings of the National Academy of Sciences, USA*, 103(18), 7192–7197, doi:10.1073/pnas.0509557103.
- Engbert, R., Mergenthaler, K., Sinn, P., & Pikovskiy, A. (2011). An integrated model of fixational eye movements and microsaccades. *Proceedings of the National Academy of Sciences, USA*, 108(39), 16149–16150, doi:10.1073/pnas.1102730108.
- Hafed, Z. M., & Clark, J. J. (2002). Microsaccades as an overt measure of covert attention shifts. *Vision Research*, 42(22), 2533–2545, doi:10.1016/s0042-6989(02)00263-8.
- Hafed, Z. M., & Krauzlis, R. J. (2012). Similarity of superior colliculus involvement in microsaccade and saccade generation. *Journal of Neurophysiology*, 107(7), 1904–1916, doi:10.1152/jn.01125.2011.
- Hafed, Z. M., Lovejoy, L. P., & Krauzlis, R. J. (2011). Modulation of microsaccades in monkey during a covert visual attention task. *The Journal of Neuroscience*, 31(43), 15219–15230, doi:10.1523/jneurosci.3106-11.2011.
- Hermens, F. (2015). Dummy eye measurements of microsaccades: Testing the influence of system noise and head movements on microsaccade detection in a popular video-based eye tracker. *Journal of Eye Movement Research*, 8(1), <http://dx.doi.org/10.16910/jemr.8.1.1>.
- Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Jarodzka, H., & van de Weijer, J. (2011). *Eye tracking: A comprehensive guide to methods and measures*. Oxford, UK: Oxford University Press.
- Horowitz, T. S., Fine, E. M., Fencsik, D. E., Yurgenson, S., & Wolfe, J. M. (2007). Fixational eye movements are not an index of covert attention. *Psychological Science*, 18(4), 356–363.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1), 35–45.
- Kleiner, M., Brainard, D., Pelli, D., Ingling, A., Murray, R., & Broussard, C. (2007). What's new in Psychtoolbox-3. *Perception*, 36, ECVF Abstract Supplement.
- Ko, H., Poletti, M., & Rucci, M. (2010). Microsaccades precisely relocate gaze in a high visual acuity task. *Nature Neuroscience*, 13(12), 1549–1553, doi:10.1038/nn.2663.
- Lara, A. H., & Wallis, J. D. (2012). Capacity and precision in an animal model of visual short-term memory. *Journal of Vision*, 12(3):13, 1–12, doi:10.1167/12.3.13. [PubMed] [Article]
- Laubrock, J., Engbert, R., & Kliegl, R. (2005). Microsaccade dynamics during covert attention. *Vision Research*, 45(6), 721–730, doi:10.1016/j.visres.2004.09.029.
- Laubrock, J., Engbert, R., Rolfs, M., & Kliegl, R. (2007). Microsaccades are an index of covert attention: Commentary on Horowitz, Fine, Fencsik, Yurgenson, and Wolfe (2007). *Psychological Science*, 18(4), 364–366, doi:10.1111/j.1467-9280.2007.01904.x.

- Martinez-Conde, S., Macknik, S. L., Troncoso, X. G., & Dyar, T. A. (2006). Microsaccades counteract visual fading during fixation. *Neuron*, *49*(2), 297–305, doi:10.1016/j.neuron.2005.11.033.
- Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., & Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, *21*, 1087–1092.
- Newman, M. E. J., & Barkema, G. T. (1999). *Monte Carlo methods in statistical physics*. Oxford, UK: Clarendon Press.
- Nyström, M., Hansen, D. W., Andersson, R., & Hooge, I. (2016). Why have microsaccades become larger? Investigating eye deformations and detection algorithms. *Vision Research*, *118*, 17–24, doi:10.1016/j.visres.2014.11.007.
- Otero-Millan, J., Castro, J., Macknik, S. L., & Martinez-Conde, S. (2014). Unsupervised clustering method to detect microsaccades. *Journal of Vision*, *14*(2):18, 1–17, doi:10.1167/14.2.18. [PubMed] [Article]
- Otero-Millan, J., Troncoso, X. G., Macknik, S. L., Serrano-Pedraza, I., & Martinez-Conde, S. (2008). Saccades and microsaccades during visual fixation, exploration, and search: Foundations for a common saccadic generator. *Journal of Vision*, *8*(14):21, 1–18, doi:10.1167/8.14.21. [PubMed] [Article]
- Pelli, D. G. (1997). The VideoToolbox software for visual psychophysics: Transforming numbers into movies. *Spatial Vision*, *10*(4), 437–442, doi:10.1163/156856897x00366.
- Poletti, M., Aytikin, M., & Rucci, M. (2015). Head-eye coordination at a microscopic scale. *Current Biology*, *25*(24), 3253–3259, doi:10.1016/j.cub.2015.11.004.
- Poletti, M., Listorti, C., & Rucci, M. (2013). Microscopic eye movements compensate for nonhomogeneous vision within the fovea. *Current Biology*, *23*(17), 1691–1695, doi:10.1016/j.cub.2013.07.007.
- Poletti, M., & Rucci, M. (2016). A compact field guide to the study of microsaccades: Challenges and functions. *Vision Research*, *118*, 83–97, doi:10.1016/j.visres.2015.01.018.
- Ratliff, F., & Riggs, L. A. (1950). Involuntary motions of the eye during monocular fixation. *Journal of Experimental Psychology*, *40*(6), 687–701.
- Rauch, H. E., Tung, F., & Striebel, C. T. (1965). Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, *3*(8), 1445–1450.
- Rolfs, M. (2009). Microsaccades: Small steps on a long way. *Vision Research*, *49*(20), 2415–2441, doi:10.1016/j.visres.2009.08.010.
- Rolfs, M., Engbert, R., & Kliegl, R. (2004). Microsaccade orientation supports attentional enhancement opposite a peripheral cue: Commentary on Tse, Sheinberg, and Logothetis (2003). *Psychological Science*, *15*(10), 705–707, doi:10.1111/j.0956-7976.2004.00744.x.
- Rucci, M., Iovin, R., Poletti, M., & Santini, F. (2007). Miniature eye movements enhance fine spatial detail. *Nature*, *447*(7146), 852–855, doi:10.1038/nature05866.
- Salvucci, D. D., & Anderson, J. R. (1998). Tracing eye movement protocols with cognitive process models. In J. Editor (Ed.), *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society* (pp. 923–928). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Salvucci, D. D., & Goldberg, J. H. (2000). Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the symposium on eye-tracking research & applications* (pp. 71–78). New York: ACM.
- Santini, T., Fuhl, W., Kubler, T., & Kasneci, E. (2016). Bayesian identification of fixations, saccades, and smooth pursuits. In J. Editor (Ed.), *Proceedings of the ninth biennial ACM Symposium on Eye Tracking Research & Applications* (pp. NN–NN), doi:10.1145/285749/2857512. Location: Publisher.
- Skavenski, A., Hansen, R., Steinman, R., & Winterson, B. (1979). Quality of retinal image stabilization during small natural and artificial body rotations in man. *Vision Research*, *19*(6), 675–683, doi:10.1016/0042-6989(79)90243-8.
- Stacy, E. W. (1962). A generalization of the gamma distribution. *The Annals of Mathematical Statistics*, *33*(3), 1187–1192.
- Stampe, D. M. (1993). Heuristic filtering and reliable calibration methods for video-based pupil-tracking systems. *Behavior Research Methods, Instruments, & Computers*, *25*(2), 137–142, doi:10.3758/bf03204486.
- Terejanu, G. (2008). *Crib sheet: Linear Kalman smoothing* [Online tutorial]. Retrieved from <https://cse.sc.edu/~terejanu/files/tutorialKS.pdf>
- Träisk, F., Bolzani, R., & Ygge, J. (2005). A comparison between the magnetic scleral search coil and infrared reflection methods for saccadic eye movement analysis. *Graefe's Archive for Clinical and Experimental Ophthalmology*, *243*(8), 791–797, doi:10.1007/s00417-005-1148-3.
- Welch, G., & Bishop, G. (2006). *An introduction to the Kalman filter* [Online tutorial]. Retrieved from http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf

- Wilken, P., & Ma, W. J. (2004). A detection theory account of change detection. *Journal of Vision*, 4(12):11, 1120–1135, doi:10.1167/4.12.11. [PubMed] [Article]
- Yu, S.-Z. (2010). Hidden semi-Markov models. *Artificial Intelligence*, 174(2), 215–243, doi:10.1016/j.artint.2009.11.011.
- Yuval-Greenberg, S., Merriam, E. P., & Heeger, D. J. (2014). Spontaneous microsaccades reflect shifts in covert attention. *The Journal of Neuroscience*, 34(41), 13693–13700, doi:10.1523/jneurosci.0582-14.2014.
- Zuber, B. L., Stark, L., & Cook, G. (1965, Dec 10). Microsaccades and the velocity-amplitude relationship for saccadic eye movements. *Science*, 150(3702), 1459–1460, doi:10.1126/science.150.3702.1459.

Appendix

Mathematical details of the BMD algorithm

Computation of the likelihood

We plug in the distributions from Equations 3, 5, and 6 into the likelihood Equation 8:

$$\begin{aligned}
 p(\mathbf{x}|\mathbf{C}) &= \iint p(\mathbf{x}|\mathbf{z})p(\mathbf{z}|\mathbf{C}, \mathbf{v})p(\mathbf{v}|\mathbf{C})d\mathbf{v}d\mathbf{z} \\
 &= \iint \prod_t N(x_t; z_t, \Sigma_x)N(z_t; z_{t-1} + v_t, \Sigma_z) \\
 &\quad \times \prod_i \left[p(v_{\tau_i}|C_{\tau_i}) \prod_{t=\tau_i+1}^{\tau_{i+1}-1} \delta(v_t - v_{t-1}) \right] d\mathbf{v}d\mathbf{z} \\
 &\propto \iint \prod_t e^{-\frac{(x_t - z_t)^T(x_t - z_t)}{2\sigma_x^2}} e^{-\frac{(\Delta z_t - v_t)^T(\Delta z_t - v_t)}{2\sigma_z^2}} \\
 &\quad \times \prod_i \left[p(v_{\tau_i}|C_{\tau_i}) \prod_{t=\tau_i+1}^{\tau_{i+1}-1} \delta(v_t - v_{t-1}) \right] d\mathbf{v}d\mathbf{z},
 \end{aligned}$$

where $\Delta z_t = z_t - z_{t-1}$. We then expand the product $(\Delta z_t - v_t)^T(\Delta z_t - v_t)$ and gather terms that depend on v_t into the second integral:

$$\begin{aligned}
 p(\mathbf{x}|\mathbf{C}) &\propto \int \int \prod_t e^{-\frac{(x_t - z_t)^T(x_t - z_t)}{2\sigma_x^2}} e^{-\frac{\Delta z_t^T \Delta z_t}{2\sigma_z^2}} \times \prod_i \\
 &\quad \left[p(v_{\tau_i}|C_{\tau_i}) \prod_{t=\tau_i+1}^{\tau_{i+1}-1} e^{-\frac{-v_t^T v_t + 2\Delta z_t^T v_t}{2\sigma_z^2}} \delta(v_t - v_{t-1}) \right] d\mathbf{v}d\mathbf{z}. \quad (21)
 \end{aligned}$$

The delta function will collapse the integral over the time series \mathbf{v} to the integral over the change points v_{τ_i} , yielding Equation 9.

Kalman smoother

The goal of Step 2 in the approximate inference algorithm is to maximize the integrand of the first term of Equation 9:

$$e^{-\frac{(z_t - z_{t-1})^T(z_t - z_{t-1})}{2\sigma_z^2} - \frac{(x_t - z_t)^T(x_t - z_t)}{2\sigma_x^2}}. \quad (22)$$

This integrand can be interpreted as the likelihood of a stochastic process with update equations

$$\begin{aligned}
 z_t &= z_{t-1} + \zeta_t \\
 x_t &= z_t + \xi_t,
 \end{aligned}$$

where ζ and ξ represent independent Gaussian noise with standard deviation σ_z and σ_x , respectively. These equations represent a special case of the Kalman update equations (Kalman, 1960; Welch & Bishop, 2006); therefore, the maximum-likelihood estimate of \mathbf{z} given \mathbf{x} is a special case of a Kalman smoother. In a Kalman filter, the goal would be to predict a future state z based on the observations x so far. However, since we have access to the entire time series, the correct inference of z is given by a Kalman smoother. This can be implemented using the RTS algorithm (Rauch et al., 1965; Terejanu, 2008). In our case, this takes the form of a Kalman filter forward in time, followed by another Kalman filter backward in time. The forward Kalman filter is:

$$\begin{aligned}
 K_t &= \frac{P_{t-1} + \sigma_z^2}{P_{t-1} + \sigma_z^2 + \sigma_x^2} \\
 \hat{y}_t &= \hat{y}_{t-1} + K_t(x_t - \hat{y}_{t-1}) \\
 P_t &= (1 - K_t)(P_{t-1} + \sigma_z^2).
 \end{aligned} \quad (23)$$

In these equations, P_t is the variance of the posterior over \hat{y}_t , and K_t is the Kalman gain. If P_t is large, we expect large changes in the states, so we need to be able to update the estimates with new incoming measurements x_t . Higher weighting of the incoming measurements is achieved with increased Kalman gains. However, if the measurement noise σ_x is high, the observation x_t is less reliable and the Kalman gain will decrease accordingly, weighting the observation less when estimating the state \hat{y}_t .

The variance of the posterior P_t does not depend on the observations \mathbf{x} , only on σ_z and σ_x , and on P_{t-1} through a recurrence relation that follows from Equation 23:

$$P_t = \frac{(P_{t-1} + \sigma_z^2)\sigma_x^2}{P_{t-1} + \sigma_z^2 + \sigma_x^2}. \quad (24)$$

This recurrence relation defines P_t at each time point given a choice for P_0 , the variance of the prior over the first time point. The choice of P_0 affects the variance P_t at early times, but not for $t \gg \frac{\sigma_x^2}{\sigma_z^2}$, because the recurrence relation converges. At convergence,

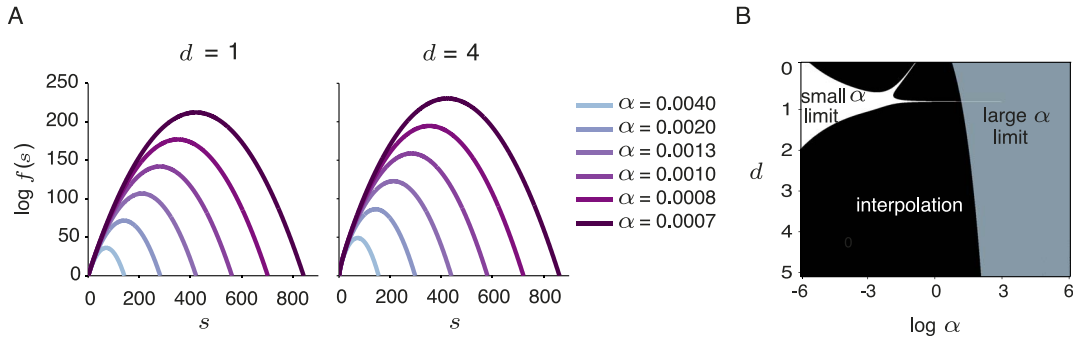


Figure A1. Details of solving the integral $A(\alpha, d) = \int f(s)ds$, with $f(s) = s^d e^{-\alpha s^2} I_0(s)$. (A) $\log f(s)$ for several combinations of s , d , and α . For larger values of α , $f(s)$ is concentrated at lower values of s . For such values, we use the Taylor series expansion of $I_0(s)$. However, for smaller values of α , the larger values of s contribute substantially to the integral and therefore we use the large s approximation of $I_0(s)$. These analytical approximations are much faster than interpolation, though come at the cost of approximation errors. (B) We limit the usage of approximations to ensure that the total approximation error of the integral $A(\alpha, d)$ is less than 0.003. In white and gray, we show the parameter regions that satisfy this criterion.

$\lim_{t \rightarrow \infty} P_{t-1} = \lim_{t \rightarrow \infty} P_t = P$. Plugging this into the forward-update equations yields the quadratic equation $P^2 + P\sigma_z^2 - \sigma_z^2\sigma_x^2 = 0$ with the valid solution

$$P = \frac{-\sigma_z^2 + \sqrt{\sigma_z^4 + 4\sigma_z^2\sigma_x^2}}{2}. \quad (25)$$

We choose $P_0 = P$, which implies $P_t = P$ for each time point. Therefore, the Kalman gain K_t is also constant across time. Plugging this Kalman gain into Equation 12 allows us to express the state estimation equation for \hat{y}_t in terms of the previous estimate \hat{y}_{t-1} , current observation x_t , and process and noise standard deviations:

$$\hat{y}_t = \hat{y}_{t-1} + \frac{\sigma_z^2 + \sqrt{\sigma_z^4 + 4\sigma_z^2\sigma_x^2}}{2\sigma_x^2 + \sigma_z^2 + \sqrt{\sigma_z^4 + 4\sigma_z^2\sigma_x^2}}(x_t - \hat{y}_{t-1}). \quad (26)$$

We denote $R = \sigma_x/\sigma_z$ and then get

$$\hat{y}_t = \hat{y}_{t-1} + \frac{1 + \sqrt{1 + 4R^2}}{1 + 2R^2 + \sqrt{1 + 4R^2}}(x_t - \hat{y}_{t-1}). \quad (27)$$

Next we apply a second Kalman filter, backwards in time, to the output of the first filter $\hat{\mathbf{y}}$ to yield the estimated eye position $\hat{\mathbf{z}}$ (Rauch et al., 1965; Terejanu, 2008). We initialize the eye position at the end of the time series \hat{z}_T to be equal to \hat{y}_T and again set the prior variance of \hat{z}_T equal to the asymptotic limit P . This backwards filter has a different Kalman gain than the first filter; the RTS update equations in our case yield $K = \frac{P}{P + \sigma_z^2}$. We can rewrite it, and thus the update equation for eye position, Equation 13, becomes

$$\hat{z}_t = \hat{y}_t + \frac{\sqrt{1 + 4R^2} - 1}{\sqrt{1 + 4R^2} + 1}(\hat{z}_{t+1} - \hat{y}_t). \quad (28)$$

From Equation 16 to Equation 17

Plugging Equation 4 into Equation 16, we get

$$\begin{aligned} I(\Delta z, \Delta \tau, d, \sigma) &= \frac{2^{\frac{1-d}{2}}}{2\pi\Gamma(\frac{d+1}{2})\sigma^{d+1}} \\ &\times \int \int r^d e^{-\frac{r^2}{2\sigma^2} e^{-\frac{\Delta \tau}{2\sigma_z^2} r^2 + \frac{1}{\sigma_z^2} \Delta z \cdot (r \cos \theta)}} dr d\theta \\ &= \frac{2^{\frac{1-d}{2}}}{\Gamma(\frac{d+1}{2})\sigma^{d+1}} \int_0^\infty r^d e^{-r^2 \left(\frac{1}{2\sigma^2} + \frac{\Delta \tau}{2\sigma_z^2}\right)} \\ &\times \left[\frac{1}{2\pi} \int_0^{2\pi} e^{\frac{r}{\sigma_z^2} \|\Delta z\| \cos(\theta - \frac{\pi}{2})} d\theta \right] dr \\ &= \frac{2^{\frac{1-d}{2}}}{\Gamma(\frac{d+1}{2})\sigma^{d+1}} \int_0^\infty r^d e^{-r^2 \left(\frac{1}{2\sigma^2} + \frac{\Delta \tau}{2\sigma_z^2}\right)} I_0\left(\frac{r}{\sigma_z^2} \|\Delta z\|\right) dr, \quad (29) \end{aligned}$$

where I_0 is the modified Bessel function of the first kind of order zero. Next we change variables from r to $s = \frac{r}{\sigma_z^2} \|\Delta z\|$ and write the final form of the integral as

$$I(\Delta z, \Delta \tau, d, \sigma) = \frac{2^{\frac{1-d}{2}}}{\Gamma(\frac{d+1}{2})\sigma^{d+1}} \left(\frac{\sigma_z^2}{\|\Delta z\|}\right)^{d+1}$$

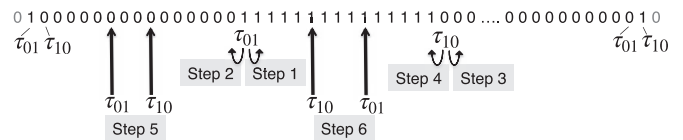


Figure A2. MCMC steps. Visualization of the six types of steps we use to navigate the space of the eye state time series **C**. We ensure that we take samples from the posterior probability distribution $p(\mathbf{C}|\mathbf{x})$.

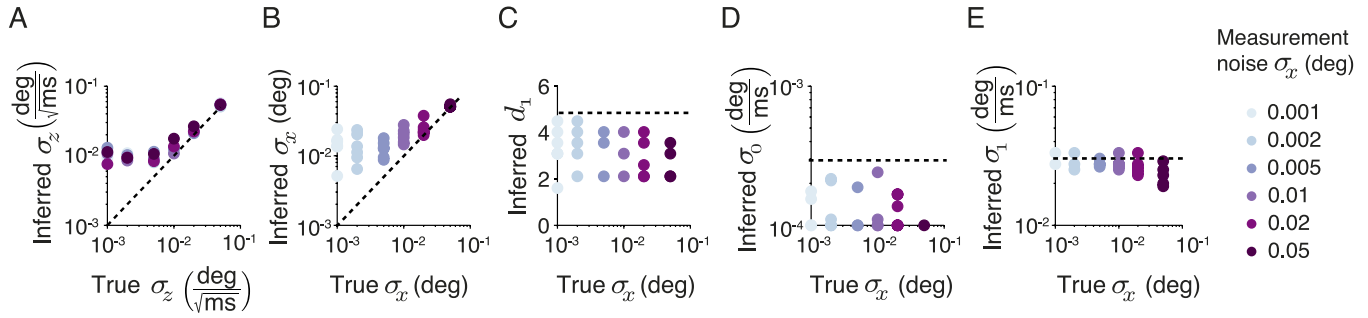


Figure A3. Parameter recovery in simulated data. In all simulated data sets, we fixed the velocity distribution parameters at $d_1 = 4.4$, $\sigma_0 = 0.0003^\circ/\text{ms}$, and $\sigma_1 = 0.03^\circ/\text{ms}$. For every combination of six motor-noise values and six measurement-noise values (colors), we created eight data sets. Here we show the median across the eight data sets of the inferred parameter values as a function of the true value of the same parameter—(A) motor noise, (B) measurement noise—or as a function of the true measurement noise σ_x , in the case of the velocity distribution parameters (C) d_1 , (D) σ_0 , and (E) σ_1 . The dashed black lines correspond to perfect parameter recovery. While these parameters are not always faithfully recovered, the inferred eye state time series **C** is recovered to a great degree of accuracy (Figure 4).

$$\times \int_0^\infty s^d e^{-s^2 \left(\frac{\sigma_z^4}{2\|\Delta z\|^2} \left(\frac{1}{\sigma^2} + \frac{\Delta \tau}{\sigma^2} \right) \right)} I_0(s) ds. \quad (30)$$

This expression shows that we can calculate $I(\Delta z, \Delta \tau, d, \sigma)$ by evaluating the integral

$$A(d, \alpha) = \int_0^\infty s^d e^{-\alpha s^2} I_0(s) ds,$$

and plugging in $\alpha = \frac{\sigma_z^4}{2\|\Delta z\|^2} \left(\frac{1}{\sigma^2} + \frac{\Delta \tau}{\sigma^2} \right)$.

Unfortunately, this integral appears to have no general analytic solution. However, in the limit of small or large α , we can replace the Bessel function with

asymptotic approximations and solve the resulting integrals. Specifically, we define upper and lower bounds $\alpha_\infty(d)$ and $\alpha_0(d)$. For $\alpha < \alpha_0(d)$, we use the large- s approximation to the Bessel function (Abramowitz & Stegun, 1965), $I_0(s) \approx \frac{e^s}{\sqrt{2\pi s}}$, so that

$$\begin{aligned} \log A(d, \alpha) &\approx \log \int_0^\infty s^d e^{-\alpha s^2} \frac{e^s}{\sqrt{2\pi s}} ds \\ &= \frac{1}{4\alpha} - d \log \alpha - d \log 2. \end{aligned} \quad (31)$$

When $\alpha > \alpha_\infty(d)$, we approximate $I_0(s)$ by its Taylor series around $s = 0$ (Abramowitz & Stegun, 1965):

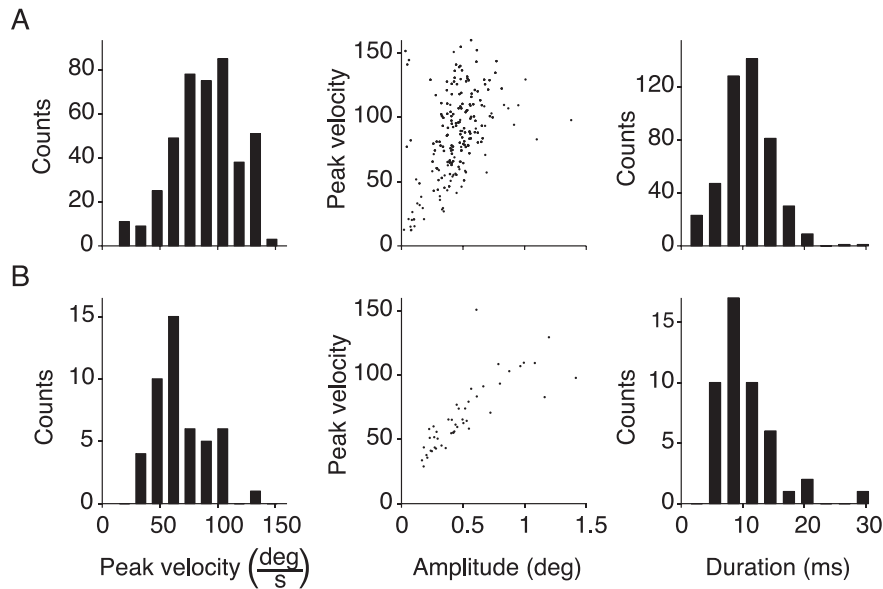


Figure A4. Microsaccade kinematics in EyeLink data. (A) BMD: (left) peak velocity distributions, (middle) main-sequence linear relationship between peak velocity and amplitude, and (right) duration distributions. (B) EK6. Mostly we notice similarities between the kinematics of the sequences detected with the two different algorithms. We spot the velocity threshold for the peak velocity distribution for the microsaccades detected by EK6.

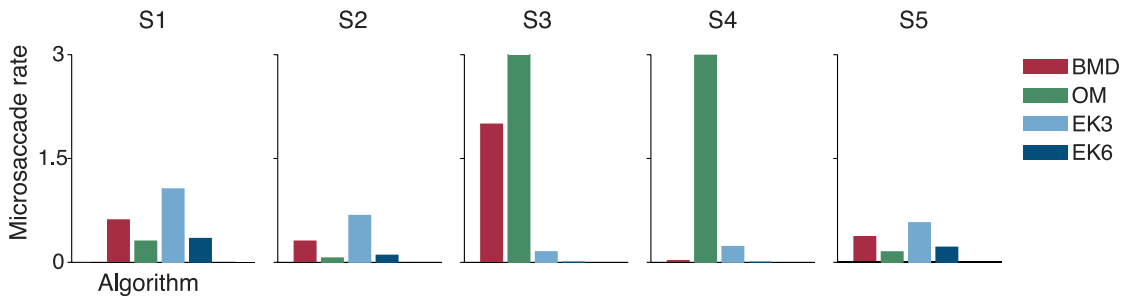


Figure A5. Inferred microsaccade rates in EyeLink data vary across algorithms. Colors for the four algorithms are the same as in previous figures. S1–S5 represent the five subjects.

$I_0(s) \approx \sum_{i=0}^{\infty} \frac{1}{\Gamma(i+1)^2} \left(\frac{s}{2}\right)^{2i}$, so that $A(d, \alpha) \approx \sum_{i=0}^{\infty} \frac{2^{-2i}}{\Gamma(i+1)^2} \int_0^{\infty} s^{d+2i} e^{-\alpha s^2} ds$. Keeping the first two terms and evaluating the integrals, we obtain

$$\log A(d, \alpha) \approx \log 2 + \frac{d+1}{2} \log \alpha + \log \left(\Gamma\left(\frac{d+1}{2}\right) + \frac{\Gamma\left(1 + \frac{d+1}{2}\right)}{4\alpha} \right). \quad (32)$$

We also build a lookup table with a million pairs of α and d , and the corresponding value of $\log A(\alpha, d)$, which we compute numerically using MATLAB's integral command. For $\alpha_0(s) < \alpha < \alpha_{\infty}(s)$, we evaluate $\log A(\alpha, d)$ by linearly interpolating between entries in the table. Interpolation is a slow operation, so we replace $I_0(s)$ with asymptotic approximations in the limit of small and large α . This causes some error, which grows as α deviates from 0 or ∞ . We choose $\alpha_{\infty}(d)$ and $\alpha_0(d)$ such that the total error in $\log A(\alpha, d)$ is less than 0.003 (Figure A1).

MCMC sampling

The goal of Step 3 in the BMD algorithm is to sample possible eye state time series \mathbf{C} from $p(\mathbf{C} | \hat{\mathbf{z}}, \hat{\sigma}_x, \hat{\sigma}_z, \hat{\sigma}_0, \hat{\sigma}_1, \hat{d}_1)$. We use an MCMC method (Newman & Barkema, 1999), which performs a biased random walk in the space of all such time series. On each step, we generate a new time series \mathbf{C}^{new} by randomly mutating the current \mathbf{C} in one of six possible steps (Figure A2). To concisely express these steps, we reparametrize each time series \mathbf{C} in terms of its change points τ , and separately keep track of time points where the eye state changes from drift to microsaccade (τ_{01}) and from microsaccade back to drift (τ_{10}). The six steps in our MCMC sampling scheme are as follows:

1. $\tau_{01} \rightarrow \tau_{01} + 1$
2. $\tau_{01} \rightarrow \tau_{01} - 1$
3. $\tau_{10} \rightarrow \tau_{10} + 1$
4. $\tau_{10} \rightarrow \tau_{10} - 1$
5. Create a new pair $\tau_{01} - \tau_{10}$
6. Create a new pair $\tau_{10} - \tau_{01}$

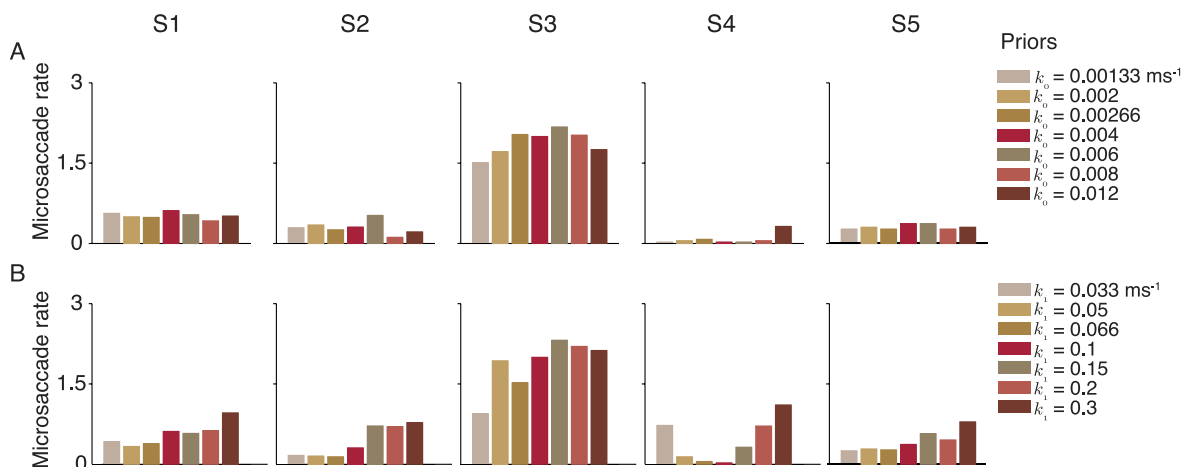


Figure A6. Inferred microsaccade rate in EyeLink data is robust to prior parameters. (A) As we vary k_0 , the parameter that controls the drift-duration prior, the inferred microsaccade rate varies only slightly. The lowest value, $k_0 = 0.012 \text{ ms}^{-1}$, corresponds to a drift-duration distribution with median 80 ms, and the highest value, $k_0 = 0.00133 \text{ ms}^{-1}$, to 760 ms. (B) The inferred microsaccade rate does not depend too much on k_1 (with the exception of subject S4). The highest and lowest values of k_1 correspond to median microsaccade durations of 3.3 and 30.3 ms, respectively. The somewhat larger dependence of the microsaccade rate on k_1 makes intuitive sense, as increasing k_1 allows for very short high-velocity sequences to be labeled as microsaccades.

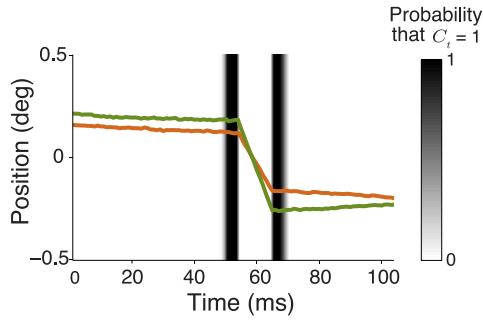


Figure A7. Typical failure mode of BMD in low-noise simulations. Instead of detecting the microsaccade labeled by EK6, BMD detects a microsaccade right before and another microsaccade right after. This error occurs because the Kalman smoother (Step 2) converts the discontinuities at the beginning and end of the change points into more gradual slopes, and the subsequent eye state estimation algorithm (Step 3) infers that these slopes are low-velocity microsaccades. A truly optimal inference algorithm, which marginalizes over the eye position, will not make this error.

These steps dictate the selection probability $g(\mathbf{C} \rightarrow \mathbf{C}^{\text{new}})$, which in general does not necessarily equal $g(\mathbf{C}^{\text{new}} \rightarrow \mathbf{C})$. The MCMC algorithm accepts any of these steps with an acceptance probability $A(\mathbf{C} \rightarrow \mathbf{C}^{\text{new}})$. To sample from the correct posterior distribution, the Markov chain in a Monte Carlo algorithm has to satisfy detailed balance, which ensures that the system makes transitions in and out of every state with compatible probabilities:

$$\begin{aligned} \frac{P(\mathbf{C} \rightarrow \mathbf{C}^{\text{new}})}{P(\mathbf{C}^{\text{new}} \rightarrow \mathbf{C})} &= \frac{g(\mathbf{C} \rightarrow \mathbf{C}^{\text{new}})A(\mathbf{C} \rightarrow \mathbf{C}^{\text{new}})}{g(\mathbf{C}^{\text{new}} \rightarrow \mathbf{C})A(\mathbf{C}^{\text{new}} \rightarrow \mathbf{C})} \\ &= \frac{p(\mathbf{C}^{\text{new}}|\hat{\mathbf{z}}, \hat{\sigma}_x, \hat{\sigma}_z, \hat{\sigma}_0, \hat{\sigma}_1, \hat{d}_1)}{p(\mathbf{C}|\hat{\mathbf{z}}, \hat{\sigma}_x, \hat{\sigma}_z, \hat{\sigma}_0, \hat{\sigma}_1, \hat{d}_1)}. \end{aligned} \quad (33)$$

We guarantee detailed balance using a modified Metropolis–Hastings acceptance probability (Metropolis, Rosenbluth, Rosenbluth, Teller, & Teller, 1953; Newman & Barkema, 1999):

$$\begin{aligned} A(\mathbf{C} \rightarrow \mathbf{C}^{\text{new}}) &= \min \left(1, \frac{g(\mathbf{C}^{\text{new}} \rightarrow \mathbf{C})p(\mathbf{C}^{\text{new}}|\hat{\mathbf{z}}, \hat{\sigma}_x, \hat{\sigma}_z, \hat{\sigma}_0, \hat{\sigma}_1, \hat{d}_1)}{g(\mathbf{C} \rightarrow \mathbf{C}^{\text{new}})p(\mathbf{C}|\hat{\mathbf{z}}, \hat{\sigma}_x, \hat{\sigma}_z, \hat{\sigma}_0, \hat{\sigma}_1, \hat{d}_1)} \right). \end{aligned} \quad (34)$$

Coarsely, this rule accepts all steps which increase the posterior probability of the new time series, and accepts some steps which decrease its posterior. However, the acceptance probability also contains a term $\frac{g(\mathbf{C}^{\text{new}} \rightarrow \mathbf{C})}{g(\mathbf{C} \rightarrow \mathbf{C}^{\text{new}})}$ which compensates for any mismatch in selection probabilities between transitions and their reverse. This compensation term allows the Metropo-

lis–Hastings to be flexible and ensures detailed balance for any choice of steps.

Parallel tempering

We have a high-dimensional problem with a complicated probability landscape that can be hard for Metropolis algorithms to navigate without getting stuck in local maxima. To avoid this, we performed parallel tempering (Earl & Deem, 2005; Newman & Barkema, 1999), also called replica-exchange MCMC sampling, which entails performing the Metropolis–Hastings algorithm concurrently at several inverse temperatures β , which modify the acceptance probability to

$$\begin{aligned} A(\mathbf{C} \rightarrow \mathbf{C}^{\text{new}}) &= \min \left(1, \frac{g(\mathbf{C}^{\text{new}} \rightarrow \mathbf{C})p(\mathbf{C}^{\text{new}})}{g(\mathbf{C} \rightarrow \mathbf{C}^{\text{new}})p(\mathbf{C})} \right. \\ &\quad \left. \left(\frac{p(\hat{\mathbf{z}}|\mathbf{C}^{\text{new}}, \hat{\sigma}_x, \hat{\sigma}_z, \hat{\sigma}_0, \hat{\sigma}_1, \hat{d}_1)}{p(\hat{\mathbf{z}}|\mathbf{C}, \hat{\sigma}_x, \hat{\sigma}_z, \hat{\sigma}_0, \hat{\sigma}_1, \hat{d}_1)} \right)^\beta \right), \end{aligned} \quad (35)$$

where we have split up the posterior into a prior and a likelihood. The lower the temperature (increased β), the less likely the Markov chain will accept steps which reduce the likelihood. Therefore, low-temperature chains are strongly attracted by likelihood maxima (local or global), whereas high-temperature chains explore the space more widely. In the infinite-temperature limit, the Markov chain samples from the prior $p(\mathbf{C})$. The strength of parallel tempering consists in allowing neighboring chains to exchange information by attempting to swap their configurations and accepting swaps with a probability

$$\begin{aligned} A(\{\mathbf{C}_1, \mathbf{C}_2\} \rightarrow \{\mathbf{C}_2, \mathbf{C}_1\}) &= \min \left(1, \left(\frac{p(\hat{\mathbf{z}}|\mathbf{C}_1, \hat{\sigma}_x, \hat{\sigma}_z, \hat{\sigma}_0, \hat{\sigma}_1, \hat{d}_1)}{p(\hat{\mathbf{z}}|\mathbf{C}_2, \hat{\sigma}_x, \hat{\sigma}_z, \hat{\sigma}_0, \hat{\sigma}_1, \hat{d}_1)} \right)^{\beta_2 - \beta_1} \right). \end{aligned} \quad (36)$$

This acceptance probability ensures that we always swap if a hotter chain has stumbled on a state with a higher posterior, thus providing the algorithm with a very high chance to not get stuck in a local maxima, while ensuring that the chain has $\beta = 1$ sample from the correct posterior probability distribution $p(\mathbf{C}|\mathbf{z}, \sigma_z, \sigma_1, d_1, \sigma_0, d_0)$. We choose the set of temperatures in our simulation to span the full range between $\beta = 0$ and $\beta = 1$, with significant overlap in the distribution of posterior values between successive chains, so that swaps are accepted with a nonzero probability.