07 12, 2006

To the Graduate School:

This thesis entitled "Eye Movement Analysis & Prediction with the Kalman Filter" and written by Thomas Grindinger is presented to the Graduate School of Clemson University. I recommend that it be accepted in partial fulfillment of the requirements for the degree of Masters of Science with a major in Computer Science.

Dr. Andrew Duchowski

We have reviewed this thesis
and recommend its acceptance:

Dr. Robert Geist

Dr. Brian Dean

Accepted for the Graduate School:

# EYE MOVEMENT ANALYSIS & PREDICTION WITH THE KALMAN FILTER

---

A Thesis

Presented to

the Graduate School of

Clemson University

---

In Partial Fulfillment

of the Requirements for the Degree

Masters of Science

Computer Science

---

by

Thomas Grindinger

07 12, 2006

Advisor: Dr. Andrew Duchowski

# Abstract

Eye movement analysis is a fundamental component of eye tracking research, especially in real-time environments. Prediction of eye movements is particularly desirable for the amelioration of sensor lag. Previous analysis approaches concentrated on binary classification of eye movements into saccades and non-saccades. Although effective and fairly simple, this approach confuses the classification of smooth pursuit and fixations by lumping them into the non-saccadic category. This thesis derives a model of simulated eye movements suitable for analysis with the Kalman filter. The resulting simulation classifies real-time mouse movement into the three distinct categories of saccades, fixations, and smooth pursuits. The Kalman filter's inherent prediction/correction framework allows for first order movement prediction that can be exploited to reduce sensor lag. The simulation allows inspection of the Kalman filter's predictive accuracy through comparison of the predicted and observed signal.

# Acknowledgements

I would like to thank the graduate Computer Science department at Clemson University for providing me with the many opportunities to learn and do research with their very knowledgeable and helpful staff. I would like to specifically thank Dr. Duchowski, who provided very helpful and meaningful direction on this project. Without his help, this paper could not have come together, much less be as polished and detailed as it is now. I would also like to thank Dr. Geist and Dr. Dean for serving on my committee. These two, specifically, have been among the most influential professors that I have had, contributing greatly to my development as a computer scientist. My graduate student peers are also deserving of acknowledgement, for their support, interest, and friendship. I would especially like to thank my family for their support and encouragement throughout my entire academic career, from my first year in undergraduate as a Mechanical Engineer, up through my graduation as a Computer Scientist and subsequent introduction into the graduate department at Clemson University. I couldn't have made it this far without you.

# Table of Contents

# Table of Figures

# Chapter 1

# Introduction

While gaining acceptance in the Human-Computer Interaction community and elsewhere, eye trackers generally lack the inherent capability of providing meaningful real-time data beyond instantaneous gaze point coordinates augmented by a time stamp, $(x, y, t)$. As such, raw data does not adequately convey the user's cognitive state or intended action (e.g., selection of a user interface item by dwell time [Jacob 1990]). What is required is categorization of the real-time signal into one of three classes of eye movements: fixations, saccades, and smooth pursuits. Saccades are rapid eye movements used in repositioning the fovea to a new location in the visual environment. Pursuit movements are involved when visually tracking a moving target. Depending on the range of target motion, the eyes are capable of matching the target's velocity. Fixations are eye movements which stabilize the retina over a stationary object of interest, and are characterized by miniature eye movements: tremor, drift, and microsaccades. The miniature eye movements that characterize fixations may be considered noise present in the control system (possibly distinct from the smooth pursuit circuit) attempting to hold gaze steady.

Beyond eye movement classification, it is desirable to predict the future gaze position of a noise-reduced (smoothed) signal. Smoothing is needed to overcome the eye tracker's limited accuracy (generally 0.5-1° visual angle), and prediction is needed to overcome the tracker's sampling lag (5-33 milliseconds @ 200-30 Hz frame rate, respectively). The latter is particularly relevant for distributed systems where real-time gaze data needs to be communicated over the network (e.g., depicting shared gaze in a Collaborative Virtual Environment, or CVE [Duchowski et al. 2004]).

This thesis proposes the use of the Kalman filter as a suitable signal analysis framework that can simultaneously classify the eye movement signal as well as predict future gaze position. Additionally, by numerically interpolating gaze position, the Kalman is expected to provide a more fluid interaction between the eye tracker and the user.

In Chapter 2, previous eye movement analysis techniques are reviewed. Although various eye movement signal analysis techniques have been developed, including the Kalman, a complete account of the filter's capability for ternary classification and prediction seems to be missing from the literature. Initialization, a key component of Kalman filtering, is also not well documented, particularly in the specific domain of eye movement analysis. In Chapter 3, the complete signal modeling and initialization framework is provided.

Classification parameters can be found in Chapter 4. Chapter 5 presents results of a mouse-driven real-time simulation of eye movements, where real-time mouse motions are analyzed in place of gaze data. Chapter 6 offers concluding remarks and notes for future work.

# Chapter 2

# Background

Fixations may be the most important form of eye movement for the design of interactive systems since most of the information gathered visually by the user occurs during these periods of relative retinal stability. Due to saccadic suppression, humans are effectively blind during saccades. However, from a signal analysis point of view, saccades are the easiest to detect. Since saccades abruptly relocate gaze position, the saccadic signal profile resembles an edge, and hence can be fairly easily distinguished, at least offline. From a signal modeling point of view, saccades are eye movements of fast, non-constant velocity of short duration. Conversely, fixations can be modeled by constant, zero velocity eye movements. In between these two extremes lie smooth pursuits, which can be modeled as constant non-zero velocity movements.

Simultaneously considering the importance of fixations and the ease of saccade detection, two approaches have generally been used to analyze eye movements [Anliker 1976]. First, *position-variance* approaches keep track of eye movement position over a given temporal window. The criterion for fixation detection is proximity of a cluster of $(x, y, t)$ points to recent mean coordinates, within some tolerance, usually variance. Position-variance schemes therefore attempt to define and detect fixations directly. Second, *velocity-based* approaches tend to perform fixation detection indirectly by detecting saccades. Velocity-based approaches strive to detect eye movement velocity over a given threshold. Then, signal intervals above threshold, considered to be part of a saccade, are then ignored, or filtered out, with the remnants assumed to represent fixations. For example, Duchowski et al. [2002] described the use of both velocity- and acceleration-based filters for saccade detection.

While both position-variance and velocity-based schemes have been successfully employed in the past, neither approach attempts to classify smooth pursuits. Pursuit movements are generally considered to be contained within the fixation signal. It is possible to combine these methods either by checking the two threshold detector outputs (e.g., for agreement) or by deriving the state-probability estimates, as has been proposed by the use of Hidden Markov Models [Salvucci and Goldberg 2000].

Considered as either saccade or fixation "pickers" [Karn 2000], previous eye movement analysis approaches lack two critical components: classification of smooth pursuits as well as saccades and fixations, and prediction of future gaze position. What is needed is a unified framework that provides ternary classification of eye movements into saccades, fixations, and smooth pursuits, while simultaneously maintaining

sufficient information for prediction. Prediction can be linear in the signal velocity sense, e.g., dead reckoning, but should be available at little or no extra cost within the framework.

Anliker [1976] described a saccade and fixation classification algorithm that also provided prediction. This prediction allowed the eye tracking application to estimate the final destination of a saccade when the system detected its peak velocity (the saccade in mid-stream). Although intuitive, this requires additional processing beyond initial saccade and fixation classification, thereby potentially offsetting any reduction in sensor lag. To reduce perceived latency, saccade prediction is not as desirable as smooth pursuit prediction.

The Kalman filter is a statistical framework that enables the prediction of a signal, given its prior history. It consists of two basic pieces, the predictor and the corrector. Once the Kalman filter has been initialized, the prediction component may be used to predict eye movement position at any time in the future. The prediction error is of course expected to degrade the farther in time the prediction is made, i.e., short-term prediction will be more accurate than long-term prediction. To correct predictive error, the Kalman uses its corrector component. This component is evaluated at intervals when the given signal (eye movements) is sampled, i.e., when the next measurement is available. Actual signal measurements (or observations) tend to align the filter's prediction with the observation.

Sauter et al. [1991] were possibly the first to use the Kalman filter for eye movement classification. In their implementation, the filter's innovation served as a detector of saccades. Modeling eye movement position in constant motion, whenever the Kalman innovation exceeded a threshold signaled a saccade. Everything else was considered a smooth pursuit. Unfortunately, Sauter et al. do not describe the initialization and calculations of their Kalman filter implementation. They do not explicitly identify fixations.

To gain an appreciation for the importance of Kalman filter initialization, Kohler's [1997] work is of particular relevance, although it is specific to tracking gross human (arm) motion in the context of computer vision. However, Kohler's report provides a foundation upon which any human movement can be tracked given an appropriate set of covariance matrices and interpolation formulae.

Finally, in a more recent attempt at detecting smooth pursuits, Abd-Almageed et al. [2002] used a Kalman filter to reduce the noise inherent in eye trackers during smooth pursuits. Their process is similar to the one presented here, although Abd-Almageed et al. do not classify eye data into the three basic types.

# Chapter 3

# Kalman Filtering

To ameliorate eye tracking sensor lag, first-order velocity-based prediction can be invoked (dead reckoning). Concomitantly, the signal can be classified by its velocity profile into the three primary states: fixation, saccade, and smooth pursuit. To do so within a unified framework, Kohler's [1997] derivation of the discrete Kalman filter for human interactive motion tracking is adapted to model eye movements.

The Kalman filter provides a recursive statistical framework for estimation and prediction of the state of a process, given a mathematical model of that process. In the case of eye movements, this framework provides the ability to both predict future gaze position, given knowledge of the state of the eye movement signal, as well as to smooth the signal by attenuating noise in the form of measurement and estimation error.

The Kalman estimates the process state by minimizing the mean of the squared estimate error $\mathbf{x}_k - \hat{\mathbf{x}}_k$, where $\mathbf{x}_k$ denotes the (unknown, ideal) process state vector (at iteration $k$, see below), and $\hat{\mathbf{x}}_k$ is the state estimate [Welch and Bishop 2004]. To do so, the goal of the Kalman filter is to minimize the estimate error covariance,

$$\mathbf{P}_k = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T]. \tag{3.1}$$

The *a posteriori* state estimate $\hat{\mathbf{x}}_k$ reflects the mean of the state distribution, i.e., $E[\mathbf{x}_k] = \hat{\mathbf{x}}_k$, and is computed as a linear combination of an *a priori* estimate $\hat{\mathbf{x}}_k^-$ and a weighted difference between an actual measurement $\mathbf{z}_k$ and a measurement prediction $\mathbf{H} \cdot \hat{\mathbf{x}}_k^-$:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k \cdot \hat{\mathbf{x}}_k^-). \tag{3.2}$$

The difference $(\mathbf{z}_k - \mathbf{H}_k \cdot \hat{\mathbf{x}}_k^-)$ is called the measurement *innovation*, or the *residual*, and reflects the discrepancy between predicted measurement $\mathbf{H} \cdot \hat{\mathbf{x}}_k^-$ and actual measurement $\mathbf{z}_k$. A residual of zero means the two are in complete agreement. Minimization of the *a posteriori* error covariance $\mathbf{P}_k$ is accomplished through appropriate computation of $\mathbf{K}_k$, the Kalman gain—a matrix that stipulates a blending factor between the *a priori* state estimate and the residual.

The calculation of the Kalman gain $\mathbf{K}_k$ is accomplished by substituting (3.2) into (3.1), performing the

indicated expectations, taking the derivative of the trace of the result with respect to $\mathbf{K}_k$, setting the result equal to zero, and solving for $\mathbf{K}_k$. One form of the resulting gain matrix that minimizes (3.1) is:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$$

where $\mathbf{R}_k$ is the measurement error covariance and $\mathbf{H}_k$ provides the connection between the state $\mathbf{x}_k$ and measurement $\mathbf{z}_k$—constituents of the above Kalman gain equation are defined in detail below. What is important to note here is the blending behavior of the Kalman. If the process is measured perfectly and the measurement error approaches zero, i.e., $\mathbf{R}_k \to 0$ and

$$\lim_{\mathbf{R}_k \to 0} \mathbf{K}_k = \mathbf{H}_k^{-1},$$

the gain weights the residual, thus favoring the measurement $\mathbf{z}_k$:

$$\begin{aligned} \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{H}_k^{-1}(\mathbf{z}_k - \mathbf{H}_k \cdot \hat{\mathbf{x}}_k^-) \\ &= \hat{\mathbf{x}}_k^- + \mathbf{H}_k^{-1}\mathbf{z}_k - \hat{\mathbf{x}}_k^- \\ &= \mathbf{H}_k^{-1}\mathbf{z}_k. \end{aligned}$$

Conversely, if the state is estimated perfectly, and the *a priori* estimate error covariance $\mathbf{P}_k^-$ approaches zero, i.e., $\mathbf{P}_k^- \to 0$ and

$$\lim_{\mathbf{P}_k^- \to 0} \mathbf{K}_k = 0,$$

the gain ignores the residual, in favor of the estimate $\hat{\mathbf{x}}_k$:

$$\begin{aligned} \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{0}\,(\mathbf{z}_k - \mathbf{H}_k \cdot \hat{\mathbf{x}}_k^-) \\ &= \hat{\mathbf{x}}_k^-. \end{aligned}$$

Note that the minimization of the estimate error hinges on the assumption of statistical normality, i.e.,

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{z}_k) &\sim N\left(E[\mathbf{x}_k], E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T]\right) \\ &= N(\hat{\mathbf{x}}_k, \mathbf{P}_k). \end{aligned}$$

The state estimate is normally distributed provided that the process model's estimate and measurement errors

($\mathbf{w}_k$, $\mathbf{e}_k$, respectively) are modeled by processes that are independent (of each other), white, with normal probability distributions, as defined below.

The discrete Kalman filter is defined by the state-space model of eye movement motion,

$$\mathbf{x}_{k+1} = \mathbf{\Phi}_k \mathbf{x}_k + \mathbf{w}_k,$$

and the discrete-time process observation (measurement),

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{e}_k,$$

where:

- $\mathbf{x}_k$ is the ($n$×1) vector state of the eye movement discrete-time process at time $t_k$ modeling the current gaze position and velocity,
$$\mathbf{x}_k = \begin{pmatrix} \mathbf{p}_k \\ \mathbf{v}_k \end{pmatrix},$$
- $\mathbf{\Phi}_k$ describes the ($n$×$n$) state transition matrix that relates $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$ in the absence of a forcing function,
$$\mathbf{\Phi}_k = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix},$$
- $\mathbf{w}_k$ is the ($n$×1) timewise uncorrelated zero-mean white noise sequence with known covariance structure $\mathbf{Q}_k$, i.e., $p(\mathbf{w}_k) \sim N(0, \mathbf{Q}_k)$,
- $\mathbf{e}_k$ represents the ($m$×1) measurement error, also assumed to be a white sequence with known covariance structure $\mathbf{R}_k$, i.e., $p(\mathbf{e}_k) \sim N(0, \mathbf{R}_k)$, but uncorrelated with the $\mathbf{w}_k$ sequence, and
- $\mathbf{H}_k$ describes the ($m$×$n$) ideal (noiseless) connection between the state vector $\mathbf{x}_k$ and measurement vector $\mathbf{z}_k$ at time $t_k$,
$$\mathbf{H}_k = \begin{pmatrix} \mathbf{I} & \mathbf{0} \end{pmatrix},$$

so that

$$\begin{aligned}
\mathbf{z}_k &= \begin{pmatrix} \mathbf{I} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{p}_k \\ \mathbf{v}_k \end{pmatrix} + \mathbf{e}_k \\
&= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} p_{k,x} \\ p_{k,y} \\ v_{k,x} \\ v_{k,y} \end{pmatrix} + \begin{pmatrix} e_{k,s,x} \\ e_{k,s,y} \end{pmatrix} \\
&= \mathbf{p}_k + \mathbf{e}_k.
\end{aligned}$$

Note that $\mathbf{H}_k$ only extracts the first component (position) from the state vector, suggesting what is in fact being measured, eye movement position.

For efficiency reasons, the above linear systems should be split into two independent systems, if the forces in the system are orthogonal. Kohler [1997] notes that physically, orthogonal forces that induce acceleration and velocities do not influence each other and any translational force, acceleration, and velocity in 2D or 3D space is a superposition of two or three orthogonal forces, accelerations, or velocities, respectively. Thus, the Kalman filter may be split into two independent systems,

$$\begin{pmatrix} p_{k+1,x} \\ v_{k+1,x} \end{pmatrix} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} p_{k,x} \\ v_{k,x} \end{pmatrix} + \begin{pmatrix} w_{k,p,x} \\ w_{k,v,x} \end{pmatrix},$$

and

$$\begin{pmatrix} p_{k+1,y} \\ v_{k+1,y} \end{pmatrix} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} p_{k,y} \\ v_{k,y} \end{pmatrix} + \begin{pmatrix} w_{k,p,y} \\ w_{k,v,y} \end{pmatrix},$$

with the linear measurement system taking the form,

$$\begin{pmatrix} z_{k,x} \end{pmatrix} = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} p_{k,x} \\ v_{k,x} \end{pmatrix} + e_{k,x}$$

and

$$\begin{pmatrix} z_{k,y} \end{pmatrix} = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} p_{k,y} \\ v_{k,y} \end{pmatrix} + e_{k,y}$$

resulting in

$$\begin{aligned} \mathbf{x}_{k+1,x} &= \mathbf{\Phi}_k \mathbf{x}_{k,x} + \mathbf{w}_{k,x} \\ \mathbf{x}_{k+1,y} &= \mathbf{\Phi}_k \mathbf{x}_{k,y} + \mathbf{w}_{k,y}, \end{aligned}$$

with the discrete-time process observation (measurement),

$$\begin{aligned} \mathbf{z}_{k,x} &= \mathbf{H}_k \mathbf{x}_{k,x} + \mathbf{e}_{k,x} \\ \mathbf{z}_{k,y} &= \mathbf{H}_k \mathbf{x}_{k,y} + \mathbf{e}_{k,y}. \end{aligned}$$

The Kalman recursion is defined as the following sequence of computations:

1. Compute gain (blending factor):

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$$

where:

- $\mathbf{P}_k$ is the covariance matrix of the estimation error (the negative superscript denotes projection from the previous step; see below), and
- $\mathbf{R}_k$ is the measurement error covariance matrix,

$$\begin{aligned} \mathbf{R}_k &= E[\mathbf{e}_k \mathbf{e}_k^T] \\ &= E[(X_{e,x}(t_k), X_{e,y}(t_k))^T (X_{e,x}(t_k), X_{e,y}(t_k))] \\ &= \begin{pmatrix} E[(X_{e,x}(t_k) X_{e,x}(t_k))] & E[(X_{e,x}(t_k) X_{e,y}(t_k))] \\ E[(X_{e,x}(t_k) X_{e,y}(t_k))] & E[(X_{e,y}(t_k) X_{e,y}(t_k))] \end{pmatrix} \end{aligned}$$

with $X_{e,x}(t_k)$ and $X_{e,y}(t_k)$ denoting the random variables that describe the measurement error. If the square standard deviation of the $x$ and $y$ measurement error is $E[(X_{e,x}(t_k) X_{e,x}(t_k))] = \sigma_{e,x}^2$ and $E[(X_{e,y}(t_k) X_{e,y}(t_k))] = \sigma_{e,y}^2$, respectively, and the $x$ and $y$ measurements are completely independent, then

$$\mathbf{R}_k = \begin{pmatrix} \sigma_{e,x}^2 & 0 \\ 0 & \sigma_{e,y}^2 \end{pmatrix} \quad \forall k.$$

For example, if the measurement error appears to be an error of plus or minus one pixel with the same probability, the square of the standard deviation is

$$E[(X_{e,x}(t_k)X_{e,x}(t_k))] = \frac{1}{3} \cdot ((-1)^2 + 0^2 + (+1)^2) = \frac{2}{3},$$

and the same holds for $E[(X_{e,y}(t_k)X_{e,y}(t_k))] = \frac{2}{3}$.

In our case, the measurement error is related to the known accuracy of the eye tracker, reported as $\theta = 0.5^\circ$ visual (half-) angle. Assuming a typical viewing distance of $d = 50$ cm $= 19.68''$ from the $17''$ screen with 1280×1024 pixel resolution (96.42 dpi), the expected (biased, half-angle) measurement error distribution is:

$$X_{e,x}(t_k) = d\tan(\theta) = 19.68 \cdot \tan(0.5) = 0.17'' = \pm 16.56 \approx \pm 17 \quad \text{pixels},$$

giving

$$E[(X_{e,x}(t_k)X_{e,x}(t_k))] = \frac{1}{35}\sum_{i=-17}^{17}(i^2) = \frac{2 \cdot (1^2 + 2^2 + \cdots + 17^2)}{35} = \frac{2(1785)}{35} = 102$$

and

$$\mathbf{R}_k = 102 \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \ \forall k.$$

2. Update estimate with measurement $\mathbf{z}_k$ (correction step):

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k \cdot \hat{\mathbf{x}}_k^-)$$

where $\mathbf{z}_k - \mathbf{H}_k \cdot \hat{\mathbf{x}}_k^-$ (the Kalman innovation or measurement residual) is the difference between the real measurement and the measurable components of the process state vector.

3. Update the error covariance matrix:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-.$$

4. Project ahead (prediction step):

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{\Phi}_k\hat{\mathbf{x}}_k.$$

5. Obtain best estimate of the error covariance matrix:

$$\mathbf{P}_{k+1}^{-} = \mathbf{\Phi}_k \mathbf{P}_k \mathbf{\Phi}_k^T + \mathbf{Q}_k,$$

where $\mathbf{Q}_k$ is the covariance matrix of the white acceleration,

$$\mathbf{Q}_k = E[\mathbf{w}_k \mathbf{w}_k^T].$$

For translational motion of constant velocity and random acceleration, Kohler [1997] derives:

$$\mathbf{Q}_k = \frac{a^2 \Delta t}{6} \begin{pmatrix} 2\mathbf{I}(\Delta t)^2 & 3\mathbf{I}\Delta t \\ 3\mathbf{I}\Delta t & 6\mathbf{I} \end{pmatrix} s,$$

where $\mathbf{I}$ is the 2×2 identity matrix, $a$ is the spectral amplitude of the white noise, and $s$ denotes unit seconds. In the ARGUS motion tracking system used by Kohler, spectral amplitude was estimated as

$$a < 11 \frac{m}{s^2} \approx 1225 \frac{pixel}{s^2}.$$

This derivation of $\mathbf{Q}_k$ is applicable to eye movement modeling, but Kohler's spectral amplitude value does not model eye movements well. Empirically, the value of $a = 100$ appears adequate.

Eye movement analysis (characterization into fixations, saccades, and smooth pursuits), as well as prediction, by the Kalman filter depends on the appropriate specification of the stochastic process, described to a large extent by the matrices $\mathbf{P}_k$, $\mathbf{Q}_k$, and $\mathbf{R}_k$. The recursive update of the matrix $\mathbf{P}_k$ at time $t_k$ relies on the *a priori* estimates $\mathbf{P}_k^-$ prior to assimilation of the measurement at time $t_k$ and the initial determination of $\mathbf{P}_0^-$. The estimation error is defined as $\mathbf{x}_k - \hat{\mathbf{x}}_k^-$ where vector $\mathbf{x}_k$ is the (unknown) ideal process state vector. The associated error covariance matrix is then

$$\mathbf{P}_k^- = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T].$$

Since $\mathbf{P}_k^-$ is updated during the Kalman recursion, it is sufficient to determine $\mathbf{P}_0^-$, although generally this is a

difficult task. For the discrete-time Kalman model of human motion above, $\mathbf{P}_0^-$ takes the form

$$\mathbf{P}_0^- = \begin{pmatrix} \sigma_{p,p}^2 & \sigma_{p,v}^2 \\ \sigma_{p,v}^2 & \sigma_{v,v}^2 \end{pmatrix}$$

where $\sigma_{p,p}$ is the standard deviation of the position estimation error and $\sigma_{p,v}^2 = E[(\mathbf{p}_k - \hat{\mathbf{p}}_k^-)(\mathbf{v}_k - \hat{\mathbf{v}}_k^-)]$ since $\mathbf{x}_k = (\mathbf{p}_k, \mathbf{v}_k)^T$. Over time, $E[\mathbf{p}_k - \hat{\mathbf{p}}_k^-] = E[\mathbf{v}_k - \hat{\mathbf{v}}_k^-] = 0$ and position and velocity estimation errors can be assumed to be independent and uncorrelated, leading to $\sigma_{p,v} = 0$. Kohler sets $\mathbf{P}_0^-$ to

$$\mathbf{P}_0^- = \frac{1}{16}\begin{pmatrix} p^2 & 0 \\ 0 & v^2 \end{pmatrix}$$

with $p$ and $v$ set to maximal initial distance and velocity, respectively, and

$$\mathbf{x}_0^- = \begin{pmatrix} \mathbf{p}_0^- \\ \mathbf{v}_0^- \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} p_{0,x}^- \\ p_{0,y}^- \end{pmatrix} \\ \begin{pmatrix} v_{0,x}^- \\ v_{0,y}^- \end{pmatrix} \end{pmatrix} \cong \begin{pmatrix} p_{0,x}^- \\ p_{0,y}^- \\ v_{0,x}^- \\ v_{0,y}^- \end{pmatrix} = \begin{pmatrix} 0.5 \\ 0.5 \\ 0.0 \\ 0.0 \end{pmatrix},$$

if we can assume $\mathbf{p}_0^-$ is the center of gravity of the (two) domains of an object detected to be in motion, i.e., in our case, the initial eye position initialized to the coordinates of the first (and potentially subsequent) calibration point(s), or alternatively to the (normalized) coordinates of the center of the screen, i.e., $\mathbf{p}_0^- = (0.5, 0.5)$. Similarly, since eye movements are observed in normalized coordinates, it is possible to set $\mathbf{P}_0^-$ to

$$\mathbf{P}_0^- = \frac{1}{16}\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

assuming maximal gaze position and no velocity (e.g., fixation).

# Chapter 4

# Eye Movement Classification

The Kalman filtering method described by Sauter et al. [1991] to detect saccades is adopted here for the same purpose. This method consists of performing a Chi-square test on the error between the observed and predicted eye velocities at read times (observations; note that read times generally occur at longer intervals than predictions). A thresholded on the Chi-square test determines when a saccade begins and ends.

An example of a generic Chi-square test is:

$$\chi^2 = \frac{\left(\hat{\mathbf{v}}_1 - \hat{\mathbf{v}}_1^-\right)^2}{\sigma^2} + \frac{\left(\hat{\mathbf{v}}_2 - \hat{\mathbf{v}}_2^-\right)^2}{\sigma^2} + \ldots + \frac{\left(\hat{\mathbf{v}}_k - \hat{\mathbf{v}}_k^-\right)^2}{\sigma^2},$$

where $k$ is the size of the sample window, $\hat{\mathbf{v}}_i$ is the recorded eye velocity, $\hat{\mathbf{v}}_i^-$ is the predicted eye velocity, and $\sigma^2$ is the variance of the sample window, calculated as follows:

$$\sigma^2 = \frac{\sum (\hat{\mathbf{v}} - \mu)^2}{N},$$

where $\hat{\mathbf{v}}$ is the recorded value of each member of the population, $\mu$ is the mean of the population, and $N$ is the size of the population. While this would yield a dynamic variance value that could be used in the Chi-square test, using a constant value yields a more well-behaved filter and allows the filter to run more efficiently. One of the side effects observed with a dynamic variance was that some events were being improperly marked as saccades, such as transitioning from a fixation to a smooth pursuit. The variance that we have determined to generate a well-behaved filter is $\sigma^2 = 1000.0$.

In our implementation, we have chosen to use a sample window size of five ($k = 5$). Choosing a larger window size may lead to the data being corrupted by events too far into the past to have a real impact on the current event. Choosing a smaller window size may cause certain actions to be inappropriately marked as saccades.

For our purposes, testing for $\chi^2 > 5.0$ appeared to be sufficient for detecting saccades in our mouse simulator (see Chapter 5). This value was derived by examining the plot of the Chi-square test and noting that true saccades have a high chance of occurring when the value of the Chi-square test exceeds this threshold. Sauter et al. [1991] described a similar evaluation, in which their threshold was $\chi^2 > 20.5$.

It should be noted that the parameters we have described thus far are dependent on specific hardware and software characteristics. For example, a device with faster or slower read timers may need several of these parameters to be changed.

The above effectively describes the Kalman filter as a “saccade picker”. Fixations may easily be determined by detecting when eye velocities fall below a certain fixed threshold (in our case 0.5 pixels per timestep). This threshold will be dependent on the accuracy of the device and what measurement (degrees or pixels) is being used. The remaining portion of the signal is assumed to be eye movement smooth pursuit.

# Chapter 5

# Results

Our test application uses a model of an eye tracker, with the computer mouse as user input. A graph control is placed at the bottom of the window that records a plot of $\chi^2$ that has been color-coded to specify, at specific moments in input history, whether a saccade, smooth pursuit, or fixation is occurring.

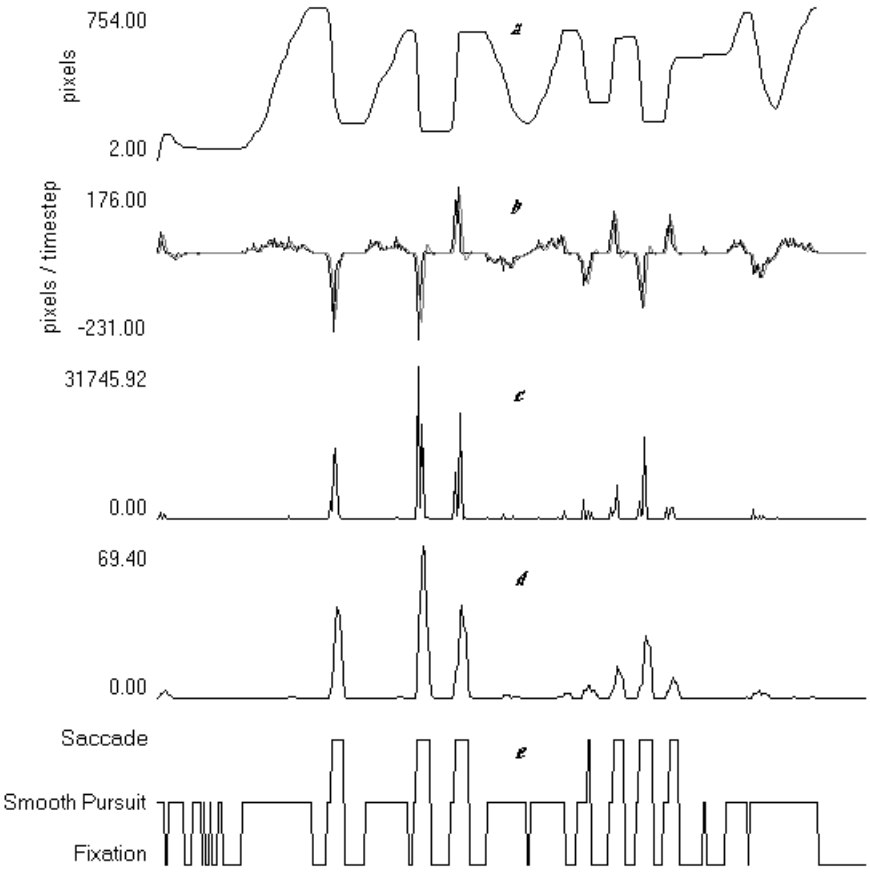


**Figure 5.1:** Saccade detection.

Figure 5.1 shows example results gathered by our simulator. Figure 5.1($a$) shows the raw position data in pixel coordinates, Figure 5.1($b$) displays the measured and predicted velocities, Figure 5.1($c$) shows the squared innovation, Figure 5.1($d$) shows the result of the Chi square test, and Figure 5.1($e$) shows the categorization of the eye movements into fixations, saccades, and smooth pursuits.

Before discussing the accuracy of our method, some implementation details should be explained. Our

implementation consists of a read timer and a display timer. The read timer contains all the operations to read the user's position data and update the Kalman filter. This timer executes after a given amount of time has elapsed. The display timer contains the predicting portion of the Kalman filter and executes as often as possible. This timer also records the user's actual location data, independent of the Kalman filter. This allows us to analyze the accuracy of our method since the read timer effectively acts as an artificial sampling delay (simulating system lag).

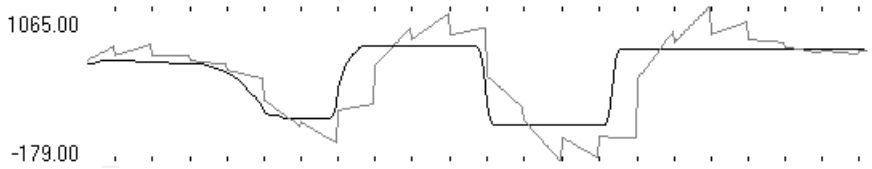


**Figure 5.2:** Predicted vs. recorded positions (500ms).

Figure 5.2 shows an example of the accuracy of our application in the case of a 500 millisecond (half-second) read timer and a 40 millisecond display timer. The solid line represents the recorded position data, and the gray line represents the predicted position data. The notches that are lined up above and below the plotted data denote read updates. Display updates occur rapidly between read updates.

As can be seen, the accuracy of the Kalman prediction is influenced by the action being performed during the last observation. Take, for example, the case of a very high position being read before a very low position is read. In between those readings, it is possible that the user either moved smoothly between them at a steady velocity, or the user was fixating at the first reading, jumped quickly to the position of the second reading, and fixated on that point until the second reading occurred (executed a saccade). The Kalman filter is unable to recognize which action actually occurred, and so could either have a nominal level of accuracy or a very poor level of accuracy.

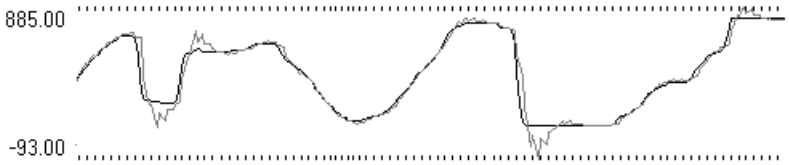


**Figure 5.3:** Predicted vs. recorded positions (100ms).

Figure 5.3 shows another example of the accuracy of our application, but this case is with a 100 millisecond read timer. As can be seen, the accuracy improves over the 500 millisecond read timer example. This

accuracy improves as the read timer is decreased.

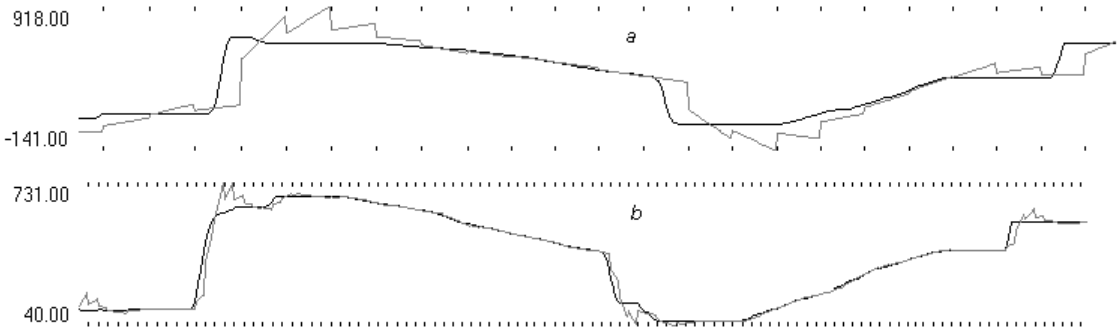


**Figure 5.4:** Similar waveforms with different read timers (500ms vs. 100ms).

In an attempt to quantify the improvement gained from faster sampling, results were compared over two similar waveforms of mouse position data, as shown in Figure 5.4. Figure 5.4(*a*) represents the user's waveform recorded with a 500 millisecond read timer, while Figure 5.4(*b*) represents the user's waveform recorded with a 100 millisecond read timer. Each waveform is a separate instance of the task, but they are very similar. The average error for Figure 5.4(*a*) is 69.88 pixels per timestep, while the average error for Figure 5.4(*b*) is 10.82 pixels per timestep. Anecdotally, inspection of the waveforms reveals that most of the error occurs around the end of the saccades. If the waveforms contained no saccades, the average error for both of them would be much smaller. The average error of the waveform appears to rise linearly with the number of saccades contained in it.

# Chapter 6

# Conclusion and Future Work

We have constructed a Kalman filter to categorize synthetic gaze data into saccades, smooth pursuits, and fixations in real-time while simultaneously using the filter to predict gaze position during smooth pursuit movement. Trinomial eye movement classification allows the eye tracking investigator to examine the context in which specific eye movements occurred. Reduction of sensor lag through eye movement prediction should allow the user to interact more fluidly with real-time gaze-contingent applications. Our preliminary results suggest an error of about 10 pixels per timestep, on average, at a 100 ms delay, which is no worse than the eye tracker's expected accuracy error.

Unfortunately, while the prediction of smooth pursuits appears to be fairly accurate, the prediction of saccades is still unreliable. This is because the constant velocity model on which the Kalman filter is based is violated when saccades are encountered. Of course, violation of the model is what facilitates saccade detection. An interesting future development would be to combine this framework with the saccade prediction algorithm of Abd-Almageed et al. [2002]. Knowing that saccade prediction is inaccurate, however, and with the classification of eye movements built into the filter, prediction in a real application may simply be restricted to function only during a smooth pursuit movement.

The Kalman filtering framework has potential to be used across networks to facilitate gaze-contingent collaboration, in an environment where latency is inherently much more pronounced. Of course, network latency can never be completely eliminated but prediction holds potential for improvement.

# Bibliography


Abd-Almageed, W., Fadali, M. S., and Bebis, G. 2002. A Non-intrusive Kalman Filter-Based Tracker for Pursuit Eye Movement. In *Proceedings of the 2002 American Control Conference*.

Anliker, J. 1976. Eye Movements: On-Line Measurement, Analysis, and Control. In *Eye Movements and Psychological Processes*, R. A. Monty and J. W. Senders, Eds. Lawrence Erlbaum Associates, Hillsdale, NJ, 185–202.

Duchowski, A., Medlin, E., Cournia, N., Gramopadhye, A., Melloy, B., and Nair, S. 2002. 3D Eye Movement Analysis for Visual Inspection Training. In *Eye Tracking Research & Applications (ETRA) Symposium*. ACM, New Orleans, LA, 103–110.

Duchowski, A. T., Cournia, N., Cumming, B., McCallum, D., Gramopadhye, A., Greenstein, J., Sadasivan, S., and Tyrrell, R. A. 2004. Visual Deictic Reference in a Collaborative Virtual Environment. In *Eye Tracking Research & Applications (ETRA)*. ACM, San Antonio, TX.

Jacob, R. J. 1990. What You Look at is What You Get: Eye Movement-Based Interaction Techniques. In *Human Factors in Computing Systems: CHI '90 Conference Proceedings*. ACM Press, 11–18.

Karn, K. S. 2000. "Saccade Pickers" vs. "Fixation Pickers": The Effect of Eye Tracker Choice on Research Findings (Panel Discussion). In *Eye Tracking Research & Applications (ETRA) Symposium*. ACM, New York, NY, 87–88.

Kohler, M. 1997. Using the Kalman Filter to track Human Interactive Motion—Modelling and Initialization of the Kalman Filter for Translational Motion. Tech. Rep. #629, Informatik VII, University of Dortmund.

Salvucci, D. D. and Goldberg, J. H. 2000. Identifying Fixations and Saccades in Eye-Tracking Protocols. In *Eye Tracking Research & Applications (ETRA) Symposium*. ACM, Palm Beach Gardens, FL, 71–78.

Sauter, D., Martin, B. J., Di Renzo, N., and Vomscheid, C. 1991. Analysis of Eye Tracking Movements Using Innovations Generated by a Kalman Filter. *Medical & Biological Engineering & Computing 29*, 63–69.

Welch, G. and Bishop, G. 2004. An Introduction to the Kalman Filter. Tech. Rep. #95-041, University of North Carolina at Chapel Hill.