Chapter 13
Eye Movement Analysis

The goal of eye movement measurement and analysis is to gain insight into the viewer's attentive behavior. As can be seen in Fig. 8.5 at the end of Chap. 8, raw eye movement data, or perhaps data processed to a certain extent such as Gaze Intersection Point (GIP) data in virtual reality, may appear to be informative, however, without further analysis, raw data are for the most part meaningless. Although intuitively (and from the knowledge of the task), it is possible to guess where the subject happened to be paying attention in the environment (over the internal calibration points, as she or he was instructed), it is not possible to make any further quantitative inferences about the eye movement data without further analysis. A method is needed to identify fixations; those eye movements that best indicate the locations of the viewer's (overt) visual attention.

As suggested in Chap. 4, eye movement signals can be approximated by linear filters. Fixations and pursuits can be modeled by a relatively simple neuronal feedback system. In the case of fixations, the neuronal control system is responsible for minimizing fixation error. For pursuit movements, the error is similarly measured as distance off the target, but in this case the target is nonstationary. Fixations and pursuits may be detected by a simple linear model based on linear summation.

The linear approach to eye movement modeling is an operational simplification of the underlying nonlinear natural processes (Carpenter 1977). The linear model assumes that position and velocity are processed by the same neuronal mechanism. The visual system processes these quantities in different ways. The position of a target is signaled by the activation of specific retinal receptors. The velocity of the target, on the other hand, is registered by the firing rate (amplitude) of the firing receptors. Furthermore, nonlinearities are expected in most types of eye movements. Accelerational and decelerational considerations alone suggest the inadequacy of the linear assumption. Nevertheless, from a signal processing standpoint, linear filter analysis is sufficient for the localization of distinct features in eye movement signals. Although this approach is a poor estimate of the underlying system, it nonetheless establishes a useful approximation of the signal in the sense of pattern recognition.

A.T. Duchowski, *Eye Tracking Methodology*, DOI 10.1007/978-3-319-57883-5_13

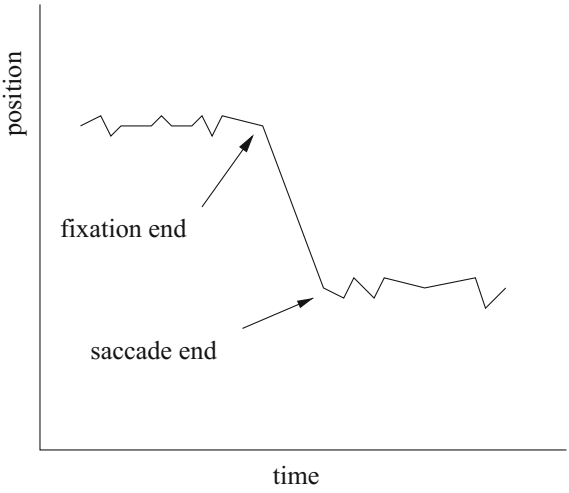


Fig. 13.1 Hypothetical eye movement signal

The goal of eye movement signal analysis is to characterize the signal in terms of salient eye movements, i.e., saccades and fixations (and possibly smooth pursuits). Typically, the analysis task is to locate regions where the signal average changes abruptly indicating the end of a fixation and the onset of a saccade and then again assumes a stationary characteristic indicating the beginning of a new fixation. A hypothetical plot of an eye movement time course is shown in Fig. 13.1. The graph shows the sought points in the signal where a saccade begins and ends. Essentially, saccades can be thought of as signal edges in time.

Two main automatic types of approaches have been used to analyze eye movements: one based on summation (averaging), the other on differentiation.[1] In the first, the temporal signal is averaged over time. If little or no variance is found, the signal is deemed a candidate for fixation. Furthermore, the signal is classified as a fixation, provided the duration of the stationary signal exceeds a predetermined threshold. This is the "dwell-time" method of fixation determination. In the second, assuming the eye movement signal is recorded at a uniform sampling rate, successive samples are subtracted to estimate eye movement velocity. The latter type of analysis is gaining favor, and appears more suitable for real-time detection of saccades. Fixations are either implicitly detected as the portion of the signal between saccades, or the portion of the signal where the velocity falls below a threshold.

Thresholds for both summation and differentiation methods are typically obtained from empirical measurements. The seminal work of Yarbus (1967) is often still referenced as the source of these measurements.

[1] A third type of analysis, requiring manual intervention, relies on slowly displaying the time course of the signal (the scanpath), either in 1D or 2D, one sample at a time, and judging which sample points lie outside the mean. This "direct inspection" method is rather tedious, but surprisingly effective.

13.1 Signal Denoising

Before (or during) signal analysis, care must be taken to eliminate excessive noise in the eye movement signal. Inevitably, noise will be registered due to the inherent instability of the eye, and worse, due to blinks. The latter, considered to be a rather significant nuisance, generates a strong signal perturbation, which (luckily) may often be eliminated, depending on the performance characteristics of the available eye movement recoding device. It is often the case that either the device itself has capabilities for filtering out blinks, or that it simply returns a value of $(0, 0)$ when the eye tracker "loses sight" of the salient features needed to record eye movements.

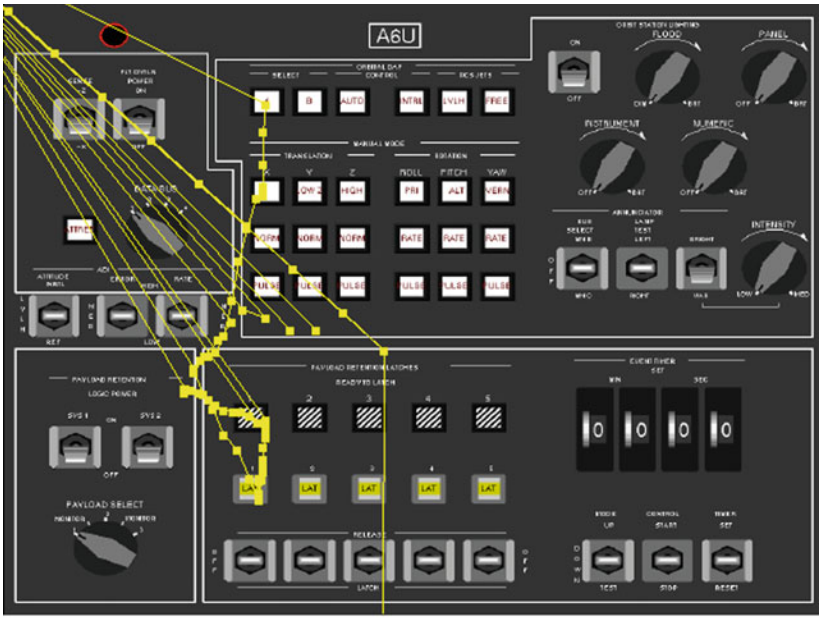
In practice, eye movement data falling outside a given rectangular range can be considered noise and eliminated. Using a rectangular region to denoise the (2D) signal also addresses another current limitation of eye tracking devices: their accuracy typically degrades in extreme peripheral regions. For this reason (as well as elimination of blinks), it may be sensible to simply ignore eye movement data falling outside the "effective operating range" of the device. This range will often be specified by the vendor in terms of visual angle. An example of signal denoising is shown in Fig. 13.2, where an interior rectangular region 10 pixels within the image borders defines the operating range. Samples falling outside this constrained interior image boundary were removed from the record (the original images measured 600×450 pixels).

13.2 Dwell-Time Fixation Detection

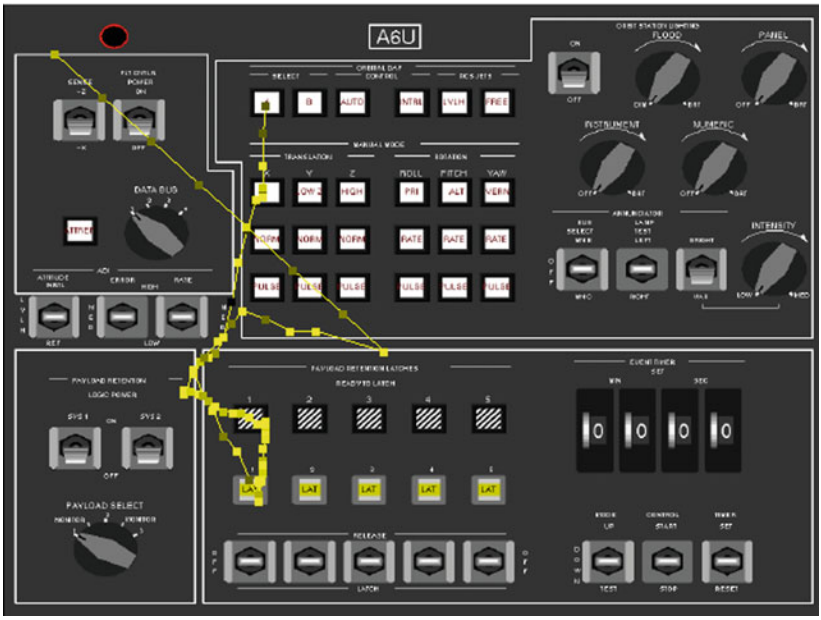
The dwell-time fixation detection algorithm depends on two characterization criteria:

1. Identification of a stationary signal (the fixation)
2. Size of time window specifying an acceptable range (and hence temporal threshold) for fixation duration

An example of such an automatic saccade/fixation classification algorithm, suggested by Anliker (1976), determines whether M of N points lie within a certain distance D of the mean (μ) of the signal. This strategy is illustrated in Fig. 13.3a where two N-sized windows are shown. In the second segment (positioned over a hypothetical saccade), the variance of the signal would exceed the threshold D indicating a rapid positional change, i.e., a saccade. The values of M, N, and D are determined empirically. Note that the value of N defines an a priori sliding window of sample times where the means and variances are computed. Anliker denotes this algorithm as the *position-variance method* because it is based on the fact that a fixation is characterized by relative immobility (low position variance) whereas a saccade is distinguished by rapid change of position (high position variance).



(a) Raw eye movement data.



(b) Eye movement data after eliminating samples falling outside an interior region ten pixels inside the image borders.

Fig. 13.2 Eye movement signal denoising. Courtesy of Wesley Hix, Becky Morley, and Jeffrey Valdez. Reproduced with permission, Clemson University

13.3 Velocity-Based Saccade Detection

An alternative to the position-variance method is the *velocity detection method* (Anliker 1976). In this velocity-based approach, the velocity of the signal is calculated within a sample window and compared to a velocity threshold. If the sampled velocity is smaller than the given threshold, then the sample window is deemed to belong to a fixation signal, otherwise it is a saccade. The velocity threshold is specified empirically. Figure 13.3b shows the hypothetical eye movement time course with the sample window centered over the saccade with its velocity profile above.

Noting Yarbus' observation that saccadic velocity is nearly symmetrical (resembling a bell curve), a velocity-based prediction scheme can be implemented to approximate the arrival time and location of the next fixation. The next fixation location

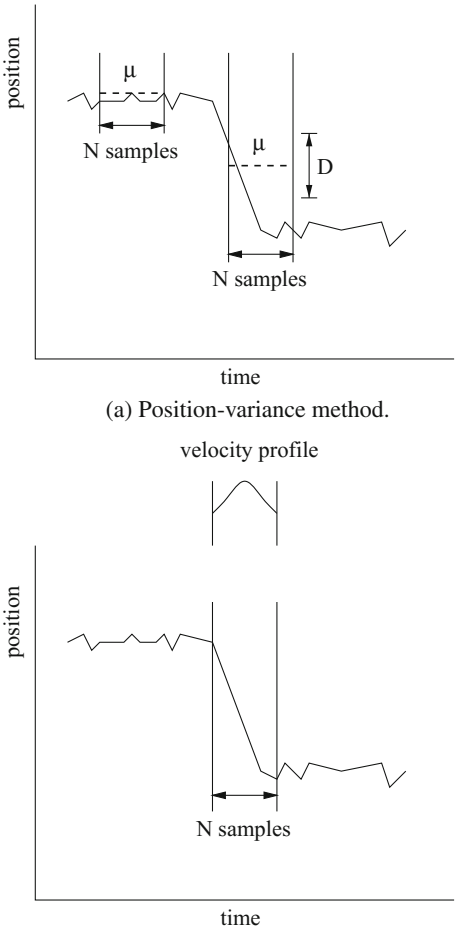


Fig. 13.3 Saccade/fixation detection

(a) Position-variance method.

(b) Velocity-detection method.

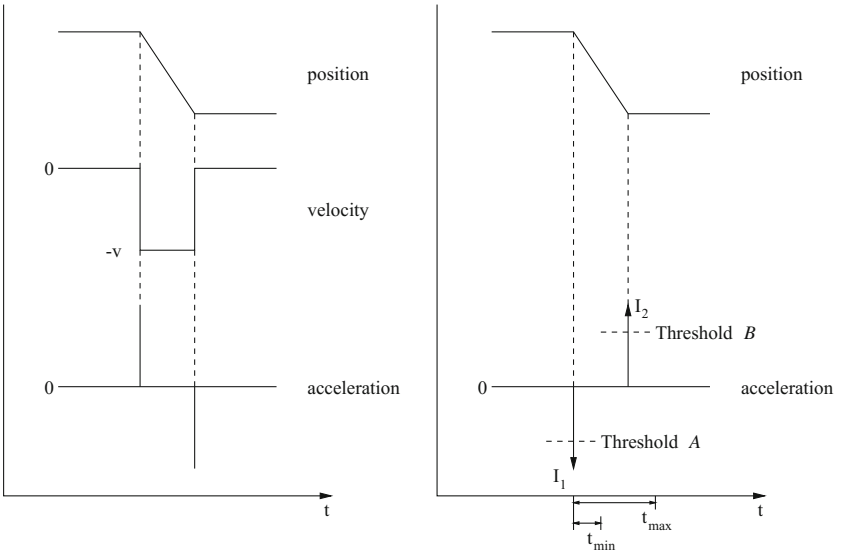
can be approximated as soon as the peak velocity is detected. Measuring elapsed time and distance traveled, and taking into account the direction of the saccade, the prediction scheme essentially mirrors the left half of the velocity profile (up to its peak) to calculate the saccade's end point.

The position-variance and velocity-based algorithms give similar results, and both methods can be combined to bolster the analysis by checking for agreement. The velocity-based algorithm offers a slight advantage in that often short-term differential filters can be used to detect saccade onset, decreasing the effective sample window size. This speeds up calculation and is therefore more suitable for real-time applications. The image in Fig. 13.2b was processed by examining the velocity,

$$v = \frac{\sqrt{(x_{t+1} - x_t)^2 + (y_{t+1} - y_t)^2}}{dt},$$

between successive sample pairs and normalizing against the maximum velocity found in the entire (short) time sequence. The normalized value, subtracted from unity, was then used to shade the sample points in the diagram. Slow moving points, or points approaching the speed of fixations are shown brighter than their faster counterparts.

Tole and Young (1981) suggest the use of short FIR (Finite Impulse Response) filters for saccade and fixation detection matching idealized saccade signal profiles. An idealized (discrete) saccade time course is shown in Fig. 13.4a. A sketch of the



(a) Velocity and acceleration profiles.

(b) Finite Impulse Response (FIR) acceleration filter algorithm.

Fig. 13.4 Idealized saccade detection

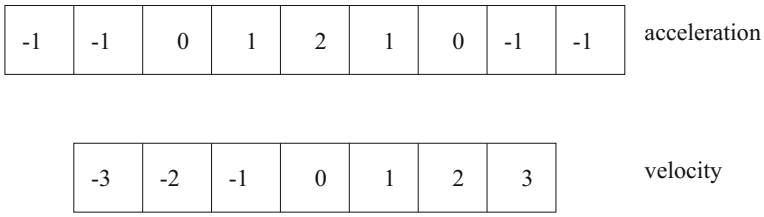


Fig. 13.5 Finite Impulse Response (FIR) filters for saccade detection

algorithm proposed by Tole and Young is presented in Fig. 13.4b, with corresponding FIR filters given in Fig. 13.5. The algorithm relies on four conditions to detect a saccade:

$$| I_1 | > A \tag{13.1}$$
$$| I_2 | > B \tag{13.2}$$
$$Sgn(I_2) \neq Sgn(I_1) \tag{13.3}$$
$$T_{\min} < I_2 - I_1 < T_{\max}. \tag{13.4}$$

If the measured acceleration exceeds a threshold (condition (13.1)), the acceleration buffer is scanned forward to check for a second peak with opposite sign to that of I_1 (condition (13.3)) and greater in magnitude than threshold B (condition (13.2)). Amplitude thresholds are derived from theoretical values, e.g., corresponding to expected peak saccade velocities of 600°/s (visual angle). Condition (13.4) stipulates minimum and maximum durations for the forward search, also based on theoretical limits, e.g., saccade durations in the range 120–300 ms. If all conditions are met, a saccade is detected.

Enhancements to the above algorithm, offered Tole and Young, include adaptive threshold estimation, adjusted to compensate for recently observed signal noise. This is achieved by reformulating the thresholds A and B as functions of an RMS estimate of noise in the signal:

$$\text{Threshold } A = 4000°/s^2 + \text{Accel/RMS} + \text{Accel/DC noise}$$
$$\text{Threshold } B = 4000°/s^2 + \text{Accel/RMS}.$$

The term

$$\text{Accel/DC noise} = \frac{2 \mid \Delta p \mid}{\Delta t^2}$$

estimates noise in acceleration, where Δt is the sampling interval and Δp is the peak-to-peak estimate of noise in position. This term can be estimated by measuring the peak-to-peak amplitude of fluctuations on the input signal when the average eye velocity over the last 2 s is less than 4°/s. This estimate can be calculated once,

for example, during calibration when the signal-to-noise ratio could be expected to remain constant over a test session, e.g., when the subject is known to be fixating a test target. This term may also be updated dynamically whenever velocity is low. The remaining estimate of noise in the output filter,

$$\text{Accel/RMS} = \sqrt{\frac{1}{T}\sum_{i=1}^{T}(\text{Accel}^2(i) - \overline{\text{Accel}}^{\,2})},$$

assumes mean acceleration is zero for time windows greater than 4 s; i.e., $\overline{\text{Accel}} \to 0$ as $T > 4$ s. The noise estimation term can be further simplified to avoid the square root,

$$\text{Accel/RMS} \leq \frac{1}{T}\sum_{i=0}^{T} \mid \text{Accel}(i) \mid, \quad T > 4 \text{ s},$$

because $\sqrt{\sum \text{Accel}^2} \leq \sum \mid \text{Accel} \mid$. This adaptive saccade detection method is reported to respond well to a temporarily induced increase in noise level, e.g., when the subject is asked to grit his or her teeth.

13.4 Eye Movement Analysis in Three Dimensions

Analysis of eye movements in 3D, as recorded in virtual reality, for example, follows the above algorithms, with slight modifications. In general, there are two important considerations: elimination of noise and identification of fixations. Noise may be present in the signal due to eye blinks, or other causes for the eye tracker's loss of proper imaging of the eye. These types of gross noise artifacts can generally be eliminated by knowing the device characteristics, for example, the eye tracker outputs (0, 0) for eye blinks or other temporary missed readings. Currently there are two general methods for identification of fixations: the position-variance strategy or the velocity-detection method. A third alternative may be a hybrid approach that compares the result of both algorithms for agreement.

The traditional two-dimensional eye movement analysis approach starts by measuring the visual angle of the object under inspection between a pair (or more) of raw eye movement data points in the time series (i.e., the POR data denoted (x_i, y_i)). Given the distance between successive POR data points, $r = \parallel (x_i, y_i), (x_j, y_j) \parallel$, the visual angle θ is calculated by the equation: $\theta = 2\tan^{-1}(r/2D)$, where D is the (perpendicular) distance from the eyes to the viewing plane. Note that r and D, expressed in like units (e.g., pixels or inches), are dependent on the resolution of the screen on which the POR data was recorded. A conversion factor is usually required to convert one measure to the other (e.g., screen resolution in dots per inch (dpi) converting D to pixels). The visual angle θ and the difference in timestamps

Δt between the POR data points allows velocity-based analysis, because $\theta / \Delta t$ gives eye movement velocity in degrees visual angle per second.

Note that the arctangent approach assumes that D is measured along the line of sight, which is assumed to be perpendicular to the viewing plane. Traditional 2D eye movement analysis methods can therefore be applied directly to raw POR data in the eye tracker reference frame. As a result, identified fixations could then be mapped to world coordinates to locate fixated ROIs within the virtual environment. A different approach may be followed by mapping raw POR data to world coordinates first, followed by eye movement analysis in three-space. The main difference of this approach is that calculated gaze points in three-space provide a composite three-dimensional representation of both the user's left and right eye movements. Applying the traditional 2D approach prior to mapping to (virtual) world coordinates suggests a componentwise analysis of left and right eye movements (in the eye tracker's reference frame) possibly ignoring depth. In three dimensions, depth information is implicitly taken into account prior to analysis. However, the assumption of a perpendicular visual target plane does not hold because the head is free to translate and rotate within six degrees of freedom.

Operating directly on Gaze Intersection Point (GIP) data in (virtual) world coordinates (see Chap. 7), a fixation detection algorithm based on estimation of velocity proceeds as follows. Given raw gaze intersection points in three dimensions, the velocity-based thresholding calculation is in principle identical to the traditional 2D approach, with the following important distinctions.

1. The head position $\mathbf{h}$ must be recorded to facilitate the calculation of the visual angle.
2. Given two successive GIP data points in three-space $\mathbf{p}_i = (x_i, y_i, z_i)$ and $\mathbf{p}_{i+1} = (x_{i+1}, y_{i+1}, z_{i+1})$, and the head position at each instance $\mathbf{h}_i$ and $\mathbf{h}_{i+1}$, the estimate of instantaneous visual angle at each sample position θ_i is calculated from the dot product of the two gaze vectors defined by the difference of the gaze intersection points and averaged head position:

$$\theta_i = \cos^{-1} \frac{\mathbf{v}_i \cdot \mathbf{v}_{i+1}}{\| \mathbf{v}_i \| \| \mathbf{v}_{i+1} \|}, \; i \in [0, n), \tag{13.5}$$

where n is the sample size and $\mathbf{v}_i = \mathbf{p}_i - \overline{\mathbf{h}}$ and $\overline{\mathbf{h}}$ is the averaged head position over the sample time period. Head position is averaged because the eyes can accelerate to reach a target fixation point much more quickly than the head (Watson et al. 1997).

With visual angle θ_i and timestamp difference between $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$, the same velocity-based thresholding is used as in the traditional 2D case. No conversion between screen resolution and distance to target is necessary because all calculations are performed in world coordinates.

The algorithm generalizes to the use of wider filters (by changing the subscript $i + 1$ to $i + k$ for $k > 1$) for improved smoothing. Using Eq. (13.5) to calculate θ_i, only two successive data points are used to calculate eye movement velocity. This is

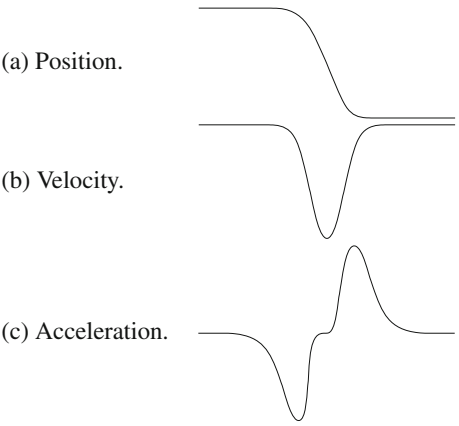


Fig. 13.6 Characteristic saccade signal and filter responses

analogous to the calculation of velocity using a two-tap FIR filter with coefficients $\{1, 1\}$.

To address excessive noise in the eye movement signal the two-tap FIR filter can be replaced by a five-tap FIR filter, as shown in Fig. 13.7a. Due to its longer sampling window, the 5-tap filter is more effective at signal smoothing (antialiasing). Following Tole and Young's 1981 work, an acceleration filter may also be used on 3D GIP data, with slight modification. The acceleration filter is shown in Fig. 13.7b, and is convolved with eye movement velocity data as obtained via either the two-tap or five-tap velocity filter. The filter responses resemble the real velocity and acceleration curves for a saccade characterized in Fig. 13.6.

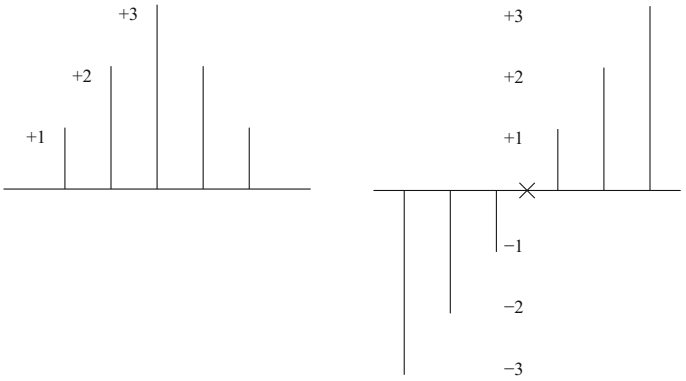
The 3D fixation eye movement analysis algorithm calculates the velocity and acceleration at each instantaneous estimate of visual angle θ_i. Note that θ_i is effectively a measure of instantaneous eye movement magnitude (i.e., amplitude), and therefore implicitly represents eye movement velocity. That is, the signal resembles the positively oriented velocity peaks shown in Fig. 13.6b. Withholding division by the time difference between successive samples (Δt) facilitates the measurement of velocity with arbitrarily long filters.

Velocity is obtained via convolution with pattern-matching FIR filters of variable length. When convolved, these filters respond to sampled data with profiles matching those of the filter. These filters, denoted $\mathbf{h}_k$, are essentially unnormalized low-pass filters which tend to smooth and amplify the underlying signal. Division by the duration of the sampling window yields velocity; i.e.,

$$\dot{\theta}_i = \frac{1}{\Delta t} \sum_{j=0}^{k} \theta_{i+j} \, \mathbf{h}_j, \; i \in [0, n-k),$$

expressed in °/s, where k is the filter length, $\Delta t = k - i$.

Acceleration is obtained via a subsequent convolution of velocity $\dot{\theta}_i$, with the acceleration filter $\mathbf{g}_j$ shown in Fig. 13.7b. That is,



(a) 5-tap velocity filter. (b) 7-tap acceleration filter.

Fig. 13.7 FIR filters

$$\ddot{\theta}_i = \frac{1}{\Delta t} \sum_{j=0}^{k} \dot{\theta}_{i+j} \mathbf{g}_j, \; i \in [0, n-k),$$

where k is the filter length, $\Delta t = k - i$. The acceleration filter is essentially an unnormalized high-pass differential filter. The resulting value $\ddot{\theta}_i$ expressed in $^\circ/s^2$, is checked against threshold A. If the absolute value of $\ddot{\theta}_i$ is greater than A, then the corresponding gaze intersection point $\mathbf{p}_i$ is treated as the beginning of a saccade. Scanning ahead in the convolved acceleration data, each subsequent point is tested in a similar fashion against threshold B to detect the end of the saccade. Two additional conditions are evaluated to locate a saccade, as given by Tole and Young. The four conditions are listed and illustrated in Fig. 13.8.

Note that these velocity and acceleration filters differ from those used by Tole and Young. This is because Tole and Young applied their filters (the reverse of these,

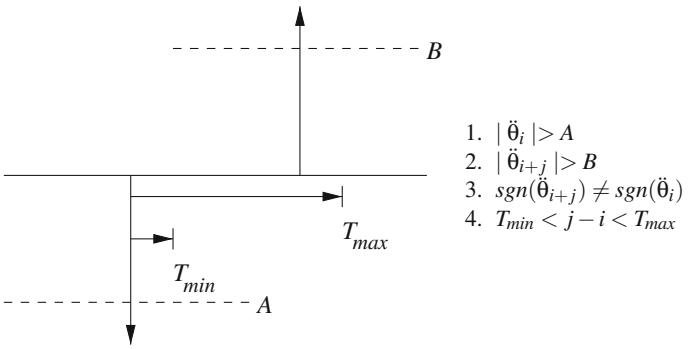


Fig. 13.8 Acceleration thresholding

essentially) to the positional eye movement signal (**p**), whereas the filters given here are applied to the signal amplitude (θ). Pseudocode of the technique is presented in Listing 13.1.

```
// Input: p(n), gaze intersection points,
// h(k), g(k), velocity and accel. filters, resp.
// Output: classification of each p_i as fixation or saccade

// convert GIP data to angular GIP data (visual angle)
for (i = 0 to n - 1)
  θ_i = cos^-1(v_i · v_i+1 / ‖ v_i ‖‖ v_i+1 ‖);

// init accumulation arrays (convolution results)
for (i = 0 to n - k - 1)
  θ̇_i = θ̈_i = 0;

// convolve (vel)
for (i = 0 to n - k - 1)
  for (j = 0 to k)
    θ̇_i = θ̇_i + θ_i+j h_j;

// convolve (accel)
for (i = 0 to n - k - 1)
  for (j = 0 to k)
    θ̈_i = θ̈_i + θ̇_i+j g_j;

// find all saccadic intervals
for (i = 0 to n - k - 1) {
  if (| θ̈_i | ≥ A) {
    // (condition 1)
    for (j = i+T_min to j < (n - k) - i && (j - i) ≤ T_max) {
      // (condition 4 implicit in loop)
      if (| θ̈_i+j | ≥ B && sgn(θ̈_i+j) ≠ sgn(θ̈_i)) {
        // (conditions 2 & 3)
        for (l = i to l < j)
          p_l = saccade
      } else {
        p_i = fixation
      }
    }
  }
}
```

Listing 13.1 Acceleration-based saccade detection

13.4.1 Parameter Estimation

Thresholds are needed for saccade velocity, acceleration, and duration, because the fixation detection algorithm relies on the detection of saccades. Although algorithm

parameters may eventually be determined empirically, algorithm fine tuning is guided by a review of the literature, briefly summarized here for context.

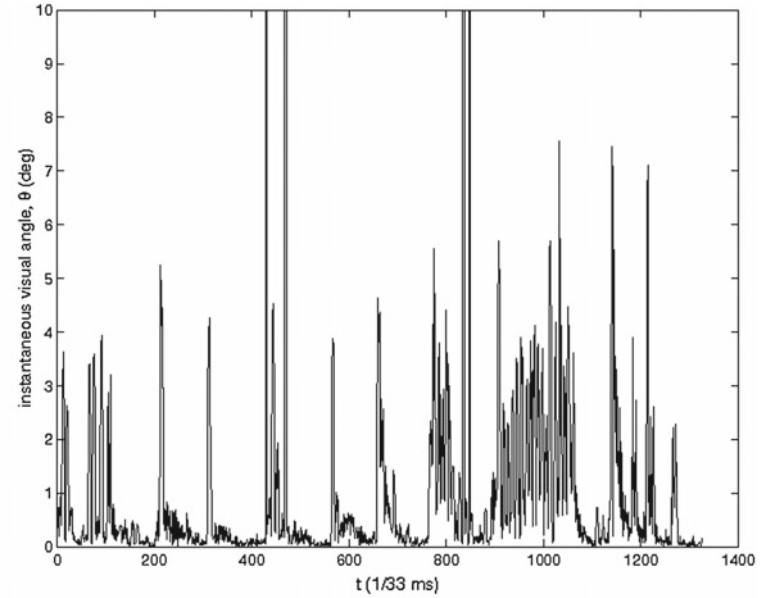
The duration of saccades is related in a nonlinear manner to their amplitude over a thousandfold range (3′–50°) (Bahill et al. 1975). Saccades of less than 15°or 20°in magnitude are physiologically the most important because most naturally occurring saccades fall in this region. When looking at pictures, normal scanpaths are characterized by a number of saccades similar in amplitude to those exhibited during reading. The saccade "main sequence" describes the relationships between saccade duration, peak velocity, and magnitude (amplitude). Because saccades are generally stereotyped, the relationship between saccade amplitude and duration can be modeled by the linear equation $\Delta t = 2.2\,\theta + 21$ (Knox 2001). Peak velocity reaches a soft saturation limit up to about 15° or 20°, but can range up to about 50°, reaching velocity saturation at about 1000°/s (Clark and Stark 1975). In practice, the main sequence relationship between amplitude and velocity can be modeled by the asymptotic equation $\dot{\theta} = \lambda(1 - e^{-\theta/15})$, with velocity upper limit (asymptote λ) set to 750°/s (Hain 1999).

According to accepted saccade amplitude estimates, measured instantaneous eye movement amplitude (θ) is expected to range up to about 20°. Example data captured in VR ranges up to 136°, with mean 1.5° (median 0.27°) and 9.7° s.d., which appears to be within normal limits, except for a few outliers (possibly due to head motion; see Fig. 13.9a).

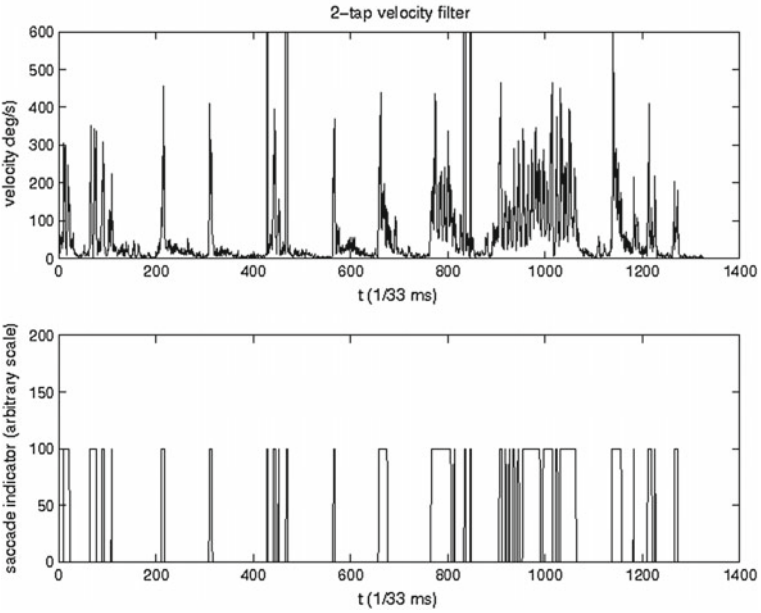
For saccade detection via velocity filtering, a threshold of 130°/s may be chosen for both two-tap and five-tap filters. Using the asymptotic model of the main sequence relationship between saccade amplitude and velocity (limited by 750°/s), this threshold should effectively detect saccades of amplitude roughly greater than 3°. Observed velocity averages are reported in Table 13.1 (see also Fig. 13.9b, c).

Saccade detection via acceleration filtering requires setting a larger number of parameters. Values of 10 and 300 ms for T_{min} and T_{max}, respectively, appear to cover a fairly wide range of saccade acceleration impulse pairs. The choice of the remaining threshold for saccade acceleration is made difficult because no applicable models of saccadic acceleration (e.g., a main sequence) could readily be found. In fact, unlike commonly listed limits of amplitude, duration, and velocity, there seems to be some disagreement regarding upper limits of acceleration. Peak acceleration has been reported to average at about 30,000°/s^2 in saccades of 10° with a saturation limit of 35,000°/s^2 for $\theta < 15°$, whereas other findings are given of 20° saccades with average peak acceleration of 26,000°/s^2 (Becker 1989). Observed acceleration averages are reported in Tables 13.1 and 13.2 (see also Fig. 13.9d). Following Tole and Young's acceleration filtering algorithm (incidentally, these authors report acceleration limits approaching 80,000°/s^2), the authors' recommended thresholds for saccade acceleration are suitable initial estimates (see below).

In Tole and Young's paper the authors point out variable noise characteristics dependent on the subject's actions (e.g., different noise profile while gritting teeth). To adapt to such signal changes the authors recommmend an adaptive thresholding technique that dynamically adjusts the threshold, based on the current estimate of noise level. Indeed, a very large peak-to-peak acceleration signal variance may be

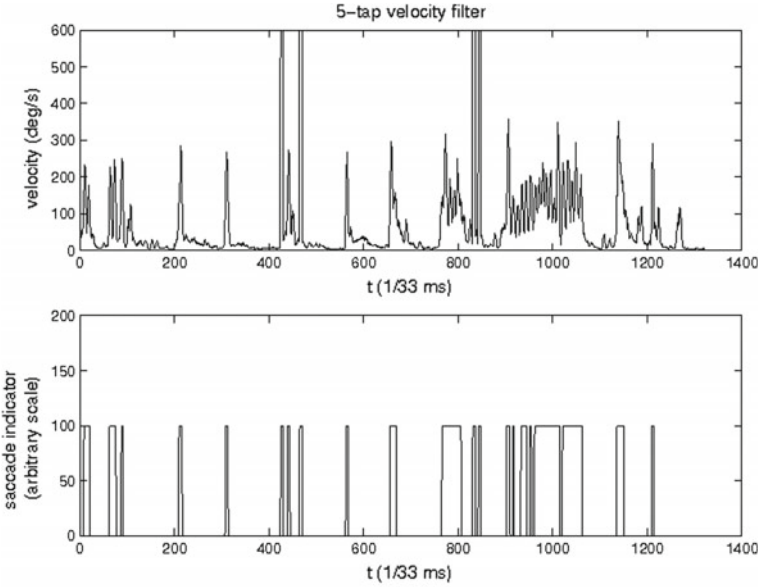


(a) Eye movement amplitude θ (deg. visual angle)

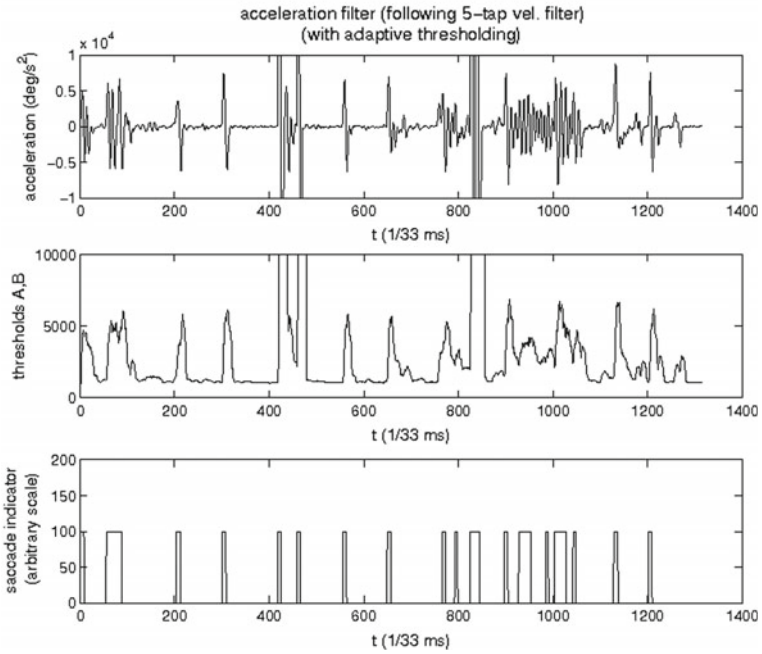


(b) Velocity $\dot{\theta}$ (deg/s) via 2-tap filter; saccade indicator

Fig. 13.9 Eye movement signal and filter responses



(c) Velocity $\dot{\theta}$ (deg/s) via 5-tap filter; saccade indicator



(d) Accel. $\ddot{\theta}$ (deg/s^2); threshold; saccade indicator

Fig. 13.9 (continued)

Table 13.1 Velocity algorithm comparisons

	2-Tap	5-Tap
Fixation groups	30	21
Mean fixation duration (ms)	1079	1450
Time spent in fixations (%)	73	69
Min $\dot{\theta}$ (°/s)	0	0.917
Max $\dot{\theta}$ (°/s)	12,385	5,592
Avg. $\dot{\theta}$ (°/s)	106	106
S.d. $\dot{\theta}$ (°/s)	635	451

Table 13.2 Acceleration algorithm comparisons

	2-Tap		5-Tap	
Thresholding	Adaptive	Constant	Adaptive	Constant
Fixation groups	20	17	17	14
Mean fixation duration (ms)	1633	1583	1937	1983
Time spent in fixations (%)	74	61	74	63
Min $\ddot{\theta}$ (°/s^2)	−257,653		−182,037	
Max $\ddot{\theta}$ (°/s^2)	248,265		167,144	
Avg. $\ddot{\theta}$ (°/s^2)	4,453		3,966	
S.d. $\ddot{\theta}$ (°/s^2)	22,475		17,470	

observed (see Fig. 13.9d). As the authors suggest, an adaptive thresholding technique may aid in automatically setting acceleration thresholds A and B:

$$A = B = 1000 + \sqrt{\frac{1}{k}\sum_{i=0}^{k}(\ddot{\theta}_{i+k})^2}\ °/s^2,$$

where k is the number of samples in time T proportional to the length of the acceleration filter; that is,

$$T = \frac{\text{filter length}}{\text{sampling rate}} = \frac{9}{30\ \text{Hz}} = 300\ \text{ms}.$$

This is a slightly different implementation of adaptive thresholding than Tole and Young's. The threshold value is slightly lower and its adaptive adjustment relies on explicit calculation of the acceleration Root Mean Squared (RMS). Also, the sampling window for this purpose is much shorter than the authors' recommended window of $T > 4$ s. Finally, the adaptive technique given above employs a "lookahead"

scan of the acceleration data, suitable for off-line analysis. Changing the $i + k$ subscript to $i - k$ provides a "lookbehind" scan that can be employed in real-time systems.

13.4.2 Fixation Grouping

The above algorithm classifies each GIP as either part of a fixation or saccade (see the saccade indicator plots in Fig. 13.9). Once each GIP has been classified, each string of consecutive fixation GIPs is condensed to a single fixation point by finding the centroid of the group. However, due to the nature of the new algorithm, we observed that at times isolated noisy GIPs were also included in fixation groups. To prevent the inclusion of such outlying points a simple check can be implemented to verify that each fixation group's duration is greater than or equal to the minimum theoretical fixation duration (i.e., 150 ms; Irwin 1992). Augmenting the acceleration-based saccade detection described above, this check can be considered as a position-variance component (emphasizing temporal coherence over spatial distribution) of a hybridized spatiotemporal approach to eye movement signal analysis.

13.4.3 Eye Movement Data Mirroring

Although the 3D eye movement analysis algorithm is mathematically robust at handling signal noise, the system is still susceptible to noise generated by the eye tracker. In particular, eye tracking equipment may randomly drop POR data. In some cases (e.g., during a blink), null POR values may be recorded for both left and right eyes. However, in some instances, only one eye's POR may be null while the other is not. This may occur due to calibration errors. To address this problem a heuristic technique may be used to mirror the nonnull POR eye movement data (Duchowski et al. 2002). The table in Fig. 13.10 shows an example of this technique. The left eye POR at time $t + 1$ is recorded as an invalid null point. To estimate a nonnull left eye coordinate at $t + 1$, the difference between successive right eye POR values is calculated and used to update the left eye POR values at $t + 1$, as shown in the equation in Fig. 13.10, giving $(x_{l_{t+1}}, y_{l_{t+1}}) = (-0.5 + dx, 0 + dy) = (-0.4, 0)$. Note that this solution assumes static vergence eye movements. It is assumed that the eyes remain at a fixed interocular distance during movement. That is, this heuristic strategy will clearly not account for vergence eye movements occurring within the short corrective time period.

Time	Left Eye	Right Eye
t	$(-0.5, 0)$	$(0.3, 0)$
$t+1$	$(0, 0)$	$(0.4, 0)$

$$dx = x_{r_{t+1}} - x_{r_t} = 0.4 - 0.3 = 0.1$$
$$dy = y_{r_{t+1}} - y_{r_t} = 0.0 - 0.0 = 0.0$$

Fig. 13.10 Heuristic mirroring example and calculation

13.5 Summary and Further Reading

Eye movement analysis for 2D applications is fairly straightforward. The 3D eye movement analysis technique is resolution-independent and is carried out in three-space, which is particularly suitable for VR applications.

There are various collections of technical papers on eye movements, usually assembled from proceedings of focused symposia or conferences. A series of such books was produced by John Senders et al. in the 1970s and 1980s (see, e.g., Monty and Senders 1976; Fisher et al. 1981). Good papers surveying eye movement analysis protocols have also recently appeared. Salvucci and Goldberg (2000) provide a good survey of current techniques (see also Salvucci and Anderson 2001).