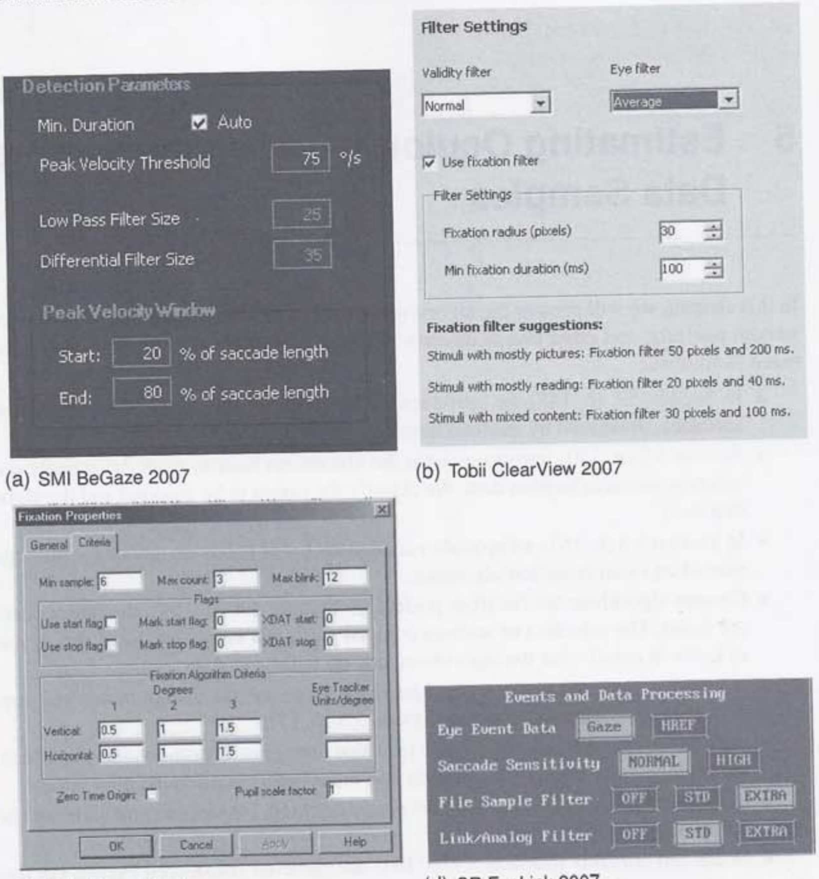
5 Estimating Oculomotor Events from Raw Data Samples

In this chapter, we will present the algorithms responsible for calculating *fixations*, *saccades*, *smooth pursuits*, and other events directly from the raw data samples. This chapter is organized as follows.

- In Section 5.1 (p. 148) we introduce the illusively simple calculation of fixations and saccades, illustrated by manufacturers' analysis software.
- Section 5.2 (p. 150) introduces what the algorithms have to work on; namely position, velocity, and acceleration data. We classify the events to be detected and the algorithms that do it.
- In Section 5.3 (p. 153) we provide a condensed list of hands-on advice for the beginning user of an event detection algorithm.
- Current algorithms are far from perfect. In Section 5.4 (p. 154), the major challenges are listed. The selection of settings is given particular emphasis. Read this if you want to know in detail what the algorithms may do with your data.
- If you are interested in the algorithms themselves and the design issues and computational reasons behind them, read Section 5.5 (p. 171).
- Section 5.6 (p. 175) focuses on data recorded onto gaze-overlaid video, for which manual segmentation of fixation duration and other events is often the only option.
- Blink events (Section 5.7, p. 176) are easily detected, but smooth pursuit is not (Section 5.8, p. 175).
- Noise and artefacts (Section 5.9, p. 181) are not even considered events, but there are good reasons for algorithms to detect such periods, and for us as researchers to decide how to treat them.
- There are detection algorithms also for some of the lesser known events, for instance microsaccades and square-wave jerks (Section 5.10, p. 182).
- The chapter is summarized in Section 5.11 (p. 185) by listing the events that can be detected and the values of which we carry with us for further analysis.

Very often, the first step in data analysis is the calculation of *events* such as fixations and saccades, with all their parameters. Indeed, the fixation and saccade values exported by the algorithms of this chapter are of great importance. They are heavily used in research in themselves, as well as in a multitude of combinations with other ways to measure and visualize eye-tracking data. Sometimes, it is even thought to be impossible to analyse eye-tracking data without this calculation. This is wrong, however. In many cases, fixation and saccade analysis is *not* a prerequisite to data analysis. For instance, heat map visualizations and the dwell time measure (p. 386) and scanpath length (p. 319) can all be calculated on raw unprocessed data just as well. Only raw data, but not fixations, can be taken as input when the analysis is tightly connected to running sample time, as with proportion over time curves (p. 197).



Detection Parameters

Min. Duration ☑ Auto

Peak Velocity Threshold 75 °/s

Low Pass Filter Size 25

Differential Filter Size 35

Peak Velocity Window

Start: 20 % of saccade length

End: 80 % of saccade length

(a) SMI BeGaze 2007

Filter Settings

Validity filter: Normal

Eye filter: Average

☑ Use fixation filter

Filter Settings

Fixation radius (pixels) 30

Min fixation duration (ms) 100

Fixation filter suggestions:

Stimuli with mostly pictures: Fixation filter 50 pixels and 200 ms.

Stimuli with mostly reading: Fixation filter 20 pixels and 40 ms.

Stimuli with mixed content: Fixation filter 30 pixels and 100 ms.

(b) Tobii ClearView 2007

Fixation Properties

General | Criteria

Min sample: 6 Max count: 3 Max blink: 12

Flags

Use start flag □ Mark start flag: 0 XDAT start: 0

Use stop flag □ Mark stop flag: 0 XDAT stop: 0

Fixation Algorithm Criteria

	Degrees 1	2	3	Eye Tracker Units/degree
Vertical:	0.5	1	1.5	
Horizontal:	0.5	1	1.5	

Zero Time Origin: □ Pupil scale factor: 1

(c) ASL Eyenal 2004

Events and Data Processing

Eye Event Data: Gaze | HREF

Saccade Sensitivity: NORMAL | HIGH

File Sample Filter: OFF | STD | EXTRA

Link/Analog Filter: OFF | STD | EXTRA

(d) SR EyeLink 2007

Fig. 5.1 Settings dialogues for fixation analysis in three analysis packages and one recording software from commercial eye-tracking manufacturers.

5.1 The setting dialogues and the output

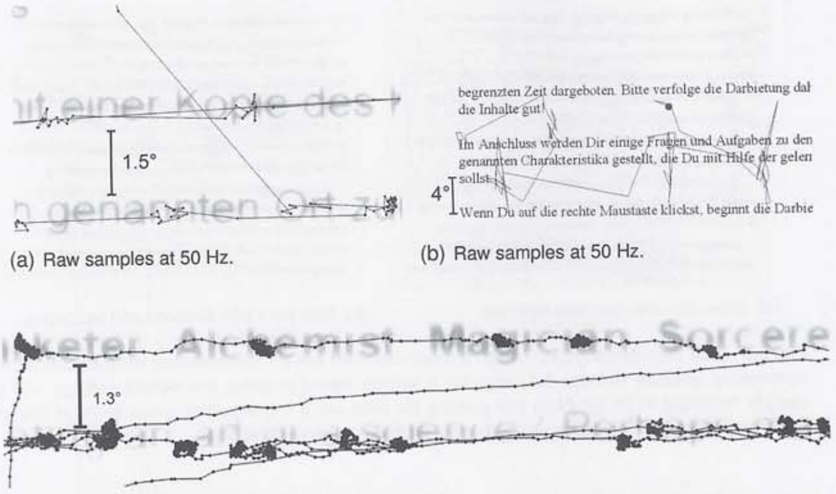
In theory, the event detection algorithm takes raw, possibly filtered data samples and tries to detect events within them. The most reported of such events are fixations and saccades. It sounds simple and something that could be done automatically, and that we should not really have to bother thinking about, and indeed software engineers have made it illusively simple to use fixation algorithms. All you have to do is to accept the pre-set values in a dialogue like those in Figure 5.1, and click OK.

In reality, however, these setting dialogues provoke many questions: What does minimal time or minimum fixation duration actually mean? What is a fixation radius, and how does it relate to monitor resolution, measured in pixels? And what is peak velocity threshold? When is a normal saccade-detection sensitivity better than a high sensitivity; should not high sensitivity always be better? And what do all the ASL Eyenal parameters mean? How sensible are the suggestions given in the Tobii dialogue? Why should there be different settings for different kinds of stimuli: are fixations more stable in reading than in picture viewing when

recording software

ples and tries to and saccades. It should not really illusively simple in a dialogue like

hat does minimal s, and how does ocity threshold? should not high an? How sensible erent settings for re viewing when



(a) Raw samples at 50 Hz.

(b) Raw samples at 50 Hz.

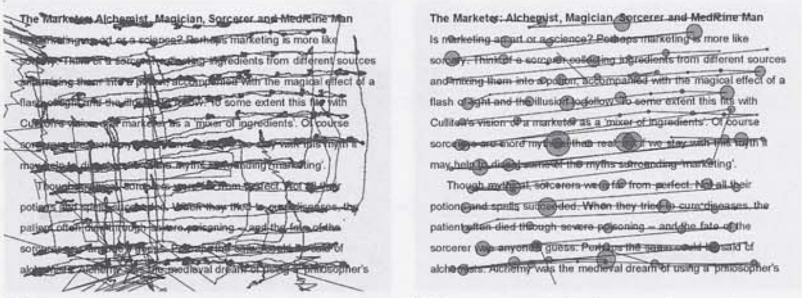
(c) Raw samples at 1250 Hz.

Fig. 5.2 Enlarged views of raw samples plotted against the stimulus background. Note how close samples are to one another in the 1250 Hz recording compared to the 50 Hz data, in particular during long saccades. Also note the low precision in (b) recording compared to (a) and (c).

such recordings are taken on the same eye-tracker? Is there a danger that my data is too noisy to do a fixation analysis, or does filtering automatically fix this? Does it matter what settings I choose? The purpose of this chapter is to give a better understanding of how event detection works, and provide insights into how to approach a settings dialogue such as the ones in Figure 5.1.

Figure 5.2 shows the *input*, the raw data samples, from three different eye-trackers. The raw data constitute the data that you get from your eye-tracker after recording. When plotting raw data against the background of the stimulus, each data sample is a little dot. During a saccade, when the eye is moving quickly, the distance between dots is large. During fixations, the dots aggregate to form one large blob from many dots. How closely the raw sample dots are positioned is directly related to the sampling frequency of your eye-tracker. How smooth the raw data appear is a direct consequence of the precision of the eye-tracker. Both these system properties are crucial to how the fixation and saccade algorithms are designed, and largely decide what a given algorithm can deliver.

In Figure 5.2(c), a full stimulus display is shown with an overlaid raw data plot. Fixation blobs and thinner saccadic lines are clearly seen. Vertical lines are blinks in progress. At the bottom and to the left, there is some high-velocity noise, probably caused by a dual corneal reflection; either a split corneal reflection, or one real and one falsely detected. Overall, this is the type of data you should expect from your eye-tracker. Figure 5.3(b) shows fixations and saccades calculated from the same raw data, using manufacturer software and default settings. Fixations are now seen as circles with a diameter indicating the duration, and abstracted straight lines for saccades. During the fixation analysis, blinks and artefacts were filtered off. In total, the fixation scanpath looks much cleaner than the raw data plot. Nevertheless, there is something deeply wrong with the scanpath of fixations and saccades. A lot of fixations that we can clearly see in the raw data plot are gone. There are two fixations on the word "Magician" in the first line, for instance, that have simply disappeared after event detection. Each line has lost one or more fixations. If you were to export this fixation and saccade data,



(a) Scanpath with raw data samples. (b) Scanpath with fixations and saccades.

Fig. 5.3 Data recorded at 1250 Hz with a tower-mounted eye-tracker. Fixations were calculated with the commercial software BeGaze 2.1, applying a velocity-based analysis and default settings with a peak velocity threshold of 75°/s. Note that parsing the data using this algorithm omits some of the grouped samples of 'blobs' which from manual inspection seem to be valid fixations.

and calculate fixation durations or saccadic amplitudes from it, you would get erroneous data: too few fixations, too large saccadic amplitudes, and the wrong average for fixation duration. You would use corrupted data, and the results you present in your report or journal paper may not be valid.

5.2 Principles and algorithms for event detection

Before we start addressing event detection in more detail, let us stop for a moment and reflect over what an event is. For example, should we always look for blobs in the raw gaze plots when we want to find a fixation, or can the fixation event be defined by other criteria?

The only raw material all algorithms have to work with is the stream of data samples recorded by the eye-tracker. In this stream of data samples, there are sometimes portions that exhibit a prototypical behaviour signifying that an *oculomotor event* has been recorded. For example, the saccade event is loosely defined as a period when the eye 'moves fast', and the fixation event where the eye 'is rather still'. The goal of *event detection* is to, according to a set of rules, robustly extract such events from the stream of data samples. Most often, this is done automatically by applying a detection algorithm to the gaze data, but it can also be done manually using subjective judgements.

Event terms such as 'fixation' are used both for the events algorithmically or manually detected in the data stream, and the oculomotor events of the eye that were recorded. In reality, perfect matches between the fixations detected by an algorithm and moments of stillness of the eye are very rare. To make matters worse, the term fixation is sometimes also used for the period during which the fixated entity is cognitively processed by the participant. The oculomotor, the algorithmically detected, and the cognitive 'fixations' largely overlap, but are not the same. When reading, for instance, it is considered proven that a word can be processed parafoveally prior to being fixated (Rayner, 1998). It is in fact easy to decouple the fixation position from the position where attention is located and processing takes place, if the task and stimuli are simple (Posner, 1980).

Furthermore, there are 'eye-in-head' fixations when the eye is still in its socket, irrespective of whether the head moves or not, and 'eye-on-stimulus' fixations when the eye is fixated on a target but possibly moving inside the head to compensate for head and body motion. Only when the head is immobile relative to the stimulus are they identical.

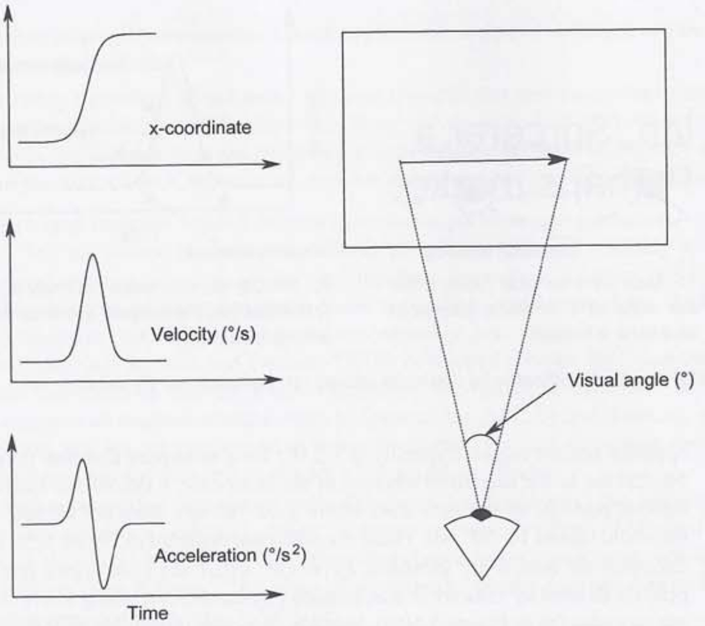


Fig. 5.4 Idealized gaze position, velocity, and acceleration profile over time showing one saccade in-between the end of a fixation (to the left of the saccade) and the beginning of a new fixation (to the right of the saccade).

The term 'fixation' in this chapter refers to an event in the data file that has been detected by an algorithm or subjectively by a person. 'Fixation' in this chapter does *not* refer to the cognitive event during which the fixated entity is processed by the participant (for a discussion about the relationship between fixation and cognitive processing, see page 377).

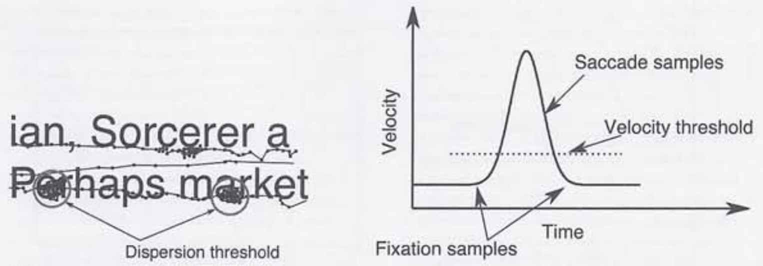
Event detection algorithms make use of three data streams from the recording and subsequent calculations: Gaze position (x, y), gaze velocity (in $^\circ/\mathrm{s}$) and gaze acceleration (in $^\circ/\mathrm{s}^2$). Besides pupil size which is sometimes used to detect blinks, that is all there is. Figure 5.4 illustrates such data from an idealized saccade[15] represented by the vector to the right in the figure. Velocity is calculated using the distance between two data samples (first derivative of gaze position), while acceleration is estimated from three consecutive samples (second derivative of gaze position).[16] Wyatt (1998) proposes the use of jerk (the third derivative of gaze position) to identify saccades, but it is noisy and fairly impractical in software implementations. As we saw in Chapter 2, filtering can significantly influence the data, in particular velocity and acceleration profiles. In the remainder of this chapter, we assume that filtering has already been done, but acknowledge that the results of event detection are tightly coupled to both the precision of the eye-tracker and the filters applied to the recorded gaze positions, as well as used in velocity and acceleration calculations.

There are some general principles many algorithms use to detect specific events:

1. Fixations are predominantly detected by a maximum allowed *dispersion* or *velocity* criterion. In the former case, temporally adjacent samples must be located within a

[15] Saccade taking the shortest path between two fixation positions.

[16] See page 48 for details about these calculations.



(a) Data samples must reside within the circle for a minimum amount of time to be a fixation.

(b) Velocity samples below the threshold belong to fixations and the samples above the threshold belong to saccades.

Fig. 5.5 Fixation identification by maximum allowed (a) *dispersion* and (b) *velocity*.

spatially limited region (typically 0.5–2.0°) for a minimum duration (anywhere from 50–250 ms in the literature) whereas in the latter case, fixations are identified as contiguous portions of the gaze data where gaze velocity does not exceed a predefined threshold (about 10–50°/s) . These scenarios are depicted in Figure 5.5(a) and 5.5(b).

2. Saccades are commonly identified as periods where the eyes 'move fast', and are in practice defined by velocity or acceleration thresholds; everything above the thresholds are saccades (as in Figure 5.5(b)). Saccade detection thresholds vary significantly, but are usually in the range 30–100°/s (velocity) and 4000–8000°/s^2 (acceleration).
3. Smooth pursuit identification does not exist in any current commercial implementation, and it is currently an open research problem to develop a robust and generic algorithm for such a purpose. The few algorithms that do exist mostly use information about the velocity (typically less than 30–40°/s) and direction of smooth pursuit eye movement.
4. Blinks are often identified as $(x = 0, y = 0)$ coordinates or when the pupil diameter is zero, indicative of a closed eyelid. Note, however, that a careful investigation of blink parameters requires us to measure eyelid movement, and this information is only crudely (if at all) related to the coordinates from your eye-tracker.
5. An artefact is a rather ill-defined 'event', but can for example occur when data samples report high velocity movement that physically cannot derive from real movement of the eye. Typically, such parts of the eye-movement data are identified and removed during initial analysis (e.g. the filtering stage), and sometimes even online during recording. More generally, artefacts can be considered as consecutive data samples that do not conform to any known eye-movement event. If the percentage of such 'unknown' data samples is high, this may be an indication that the data is of poor quality and should not be used in further analysis. It can also indicate that the algorithm is not appropriate to use on your recorded data.
6. While the above events, in particular fixation and saccade events, will be of main focus in this chapter, other events indeed exist and algorithms have been developed to detect them. For *nystagmus*, for instance, Juhola (1988) presents an adaptive digital recursive filter capable of detecting all maxima and minima in a sequence of alterations. The *square-wave jerk* (p. 183) is another event that occurs frequently in healthy participants' eye movements. Then there are fixational eye-movement drifts, microsaccades, and tremor, for which a few algorithms exist (the one by Engbert & Kliegl, 2003 for microsaccade detection, for instance). Nyström and Holmqvist (2010) proposed an al-

gorithm to quantify movements known as *glissades*, a type of wobbling eye movement at the end of saccades.

The existing algorithms do not detect all event types; in fact they rarely detect more than one. The common *identification by dispersion threshold algorithm*[17] (I-DT) detects fixations only and does not separate between remaining events. Other algorithms detect only saccadic portions of the data. Overall, the existing methods can be divided into three broad groups:

Dispersion- (and duration-) based *fixation detection algorithms* using positional information, and the related clustering algorithms using Principle 1. By making alterations to the dispersion criterion, several varieties have developed: Salvucci and Goldberg (2000) test five different dispersion-based algorithms, and Urruty, Lew, Ihadaddene, and Simovici (2007) have developed a completely new algorithm based on projection clustering. Santella and DeCarlo (2004) developed a mean shift clustering algorithm that could be used for fixation detection. This group of algorithms is common in commercial implementations, such as Gazetracker, ASL Eyenal, faceLab, and SMI BeGaze, and are not uncommon in research papers. Typically, dispersion-based algorithms are used for data collected with a low-speed eye-tracker.

Velocity and acceleration algorithms using Principles 1 or 2, (mostly) use velocity and/or acceleration data to calculate events. Software packages by Tobii, SMI, and SR Research (EyeLink) include detection algorithms based on such principles, although their details are quite different. This class of algorithms typically requires data collected at higher sampling rates (say > 200 Hz).

Manual detection of events where a number of experienced eye-tracking researchers subjectively parse data samples into events. This is a method to find fixations, but it is not an algorithm. Many researchers trust only manual detection, in particular when data are collected with head-mounted eye-trackers without head tracking.

5.3 Hands-on advice for event detection

If you need to analyse your data with a fixation or saccade detection algorithm, and care about the validity of the output, what should you do? Some general recommendations are:

- Perhaps most importantly, plot your fixations next to your raw data, as in Figure 5.3, and examine what the algorithm does at different settings.
- Examine the distributions of events in your measure at different settings (look at the histograms), before you decide which setting to use.
- Make parallel analyses with several settings, and see how this affects your results; see Green (2006) who does this.
- Recommendations for algorithmic settings depend on factors such as the eye-tracker used for data collection, individual traits such as fixation stability, and the particular circumstances during calibration and recording (see pp. 154–161 for a detailed discussion).
- If you want to compare your results to previous literature, use similar algorithms and settings. Unfortunately, however, not all researchers report the settings they use.
- Fixations with unreasonably low durations often result from the current algorithms. Remove them if they significantly influence your results, or use a better algorithm.

[17] Refers in this book to the implementation described in Salvucci and Goldberg (2000).

- Beware of smooth pursuit or movement that looks like smooth pursuit in your data file. This is likely to be part of your data if you use animated stimuli or a head-mounted eye-tracker. Current algorithms have been designed to analyse only data recorded from static stimuli.
- Dispersion based algorithms are not suited to analyse data collected with higher sampling frequencies (> 200 Hz). If you have access to velocity-based algorithms, they are more likely to produce a good output.
- Using velocity and acceleration data, make sure that you understand how the filters used to generate the data affect detection. For example, lowpass filtering smooths out velocity peaks and thus effects where the peak threshold intersects with the velocity curves.
- Some researchers use measures that require data samples as input, but only from fixations. The fixation detection algorithms can then be used to 'clean' the data from all other events prior to using the measure.
- Beware that some implementations divide events (such as a fixation) that cross a trial border into two parts. This may lead to artificially low first fixation durations for trials; one option is to exclude such partial events from the analysis.
- In you article, always report the algorithm and the detection parameters you have used.
- Clearly define the events you use in your article. For example, do 'fixations' refer to implicitly detected inter-saccadic intervals or explicitly detected oculomotor periods of stillness?

5.4 Challenging issues in event detection

There are a range of issues that influence the results of event detection. As we saw in Figure 5.1, some of them are possible to control in the settings dialogue boxes in commercial software, while others may require post-processing or new algorithmic solutions. The validity of your results depends on how you deal with these issues.

5.4.1 Choosing parameter settings

Given a set of raw data samples, parameter settings are used to identify a specific event or to separate different types of events from each other. Therefore, they largely define the properties of a detected event. While the settings mostly serve to distinguish between event types, they are also commonly used by researchers to exclude data that are unreasonable with respect to what is known about the physiological limitations of eye movements, or with regard to the experimental design. With the important choice of parameter settings in mind, what are the proper values to choose for the settings in event detection algorithms, and how are they motivated?

Recommendations, arguments, and practices

Some manufacturers have provided their customers with recommendations for fixation analysis settings, but how well founded is such advice? In the Tobii Clearview settings dialogue (p. 148), the recommended lower fixation duration threshold of 40 ms for reading studies compared to 200 ms in picture viewing probably reflects the observation that fixations are typically shorter during reading than during picture viewing (p. 377). But what if the researcher has participants making 165 ms fixations during picture viewing? Should she then just lose those fixations from later statistical calculations, as Figure 5.3 exemplifies? And what about the 50 pixel radius suggestion for picture viewing, compared to 20 pixels for reading? Are fixations more stable during reading than while viewing images? If the participant makes two fixations close to one another during reading, the fixation analysis would give two fixations. But if the same person makes two fixations at the same close distance during picture viewing, should the fixation analysis produce just one long fixation?

The ASL Eyenal Manual (2001) offers the following motivation for their thresholds (defaults are 1° and 100 ms): "Specifically, there is research documenting the minimum latency of saccades in response to visual stimuli (thus suggesting a minimum fixation duration) and data defining the maximum amplitude of involuntary eye movements during the fixation (thus establishing maximum fixation boundaries)". Involuntary eye movements such as drift and microsaccades indeed make up part of the movements inside a fixation, but the imprecision in the specific eye-tracker and the specific measurement may be much larger than 1°.

Also, dispersion can be calculated in a number of different ways. Blignaut and Beelders (2009) present the following varieties of dispersion:

1. The maximum horizontal and vertical distance covered by the gaze positions in a fixation, $((\max(x) - \min(x)) + (\max(y) - \min(y)))/2 \leq$ threshold (Salvucci & Goldberg, 2000).
2. The distance between points in the fixation that are the furthest apart (Salvucci & Goldberg, 2000).
3. The distance between any two successive points, which is an estimate of the eye velocity (Shic, Scassellati, & Chawarska, 2008).
4. The distance between points and the centre of the fixation, i.e. the radius (Camilli, Terenzi, & Nocera, 2008).
5. The average or the standard deviation of the distances of all points from the centre of a fixation (Anliker, 1976; Applied Science Laboratories, 2001).

ASL Eyenal uses the standard deviation as dispersion measure and sets it to 1° as default, but for many of the other implementations, it is unclear what dispersion is. Surprisingly many softwares ask for dispersion thresholds in pixels instead of visual degrees. SMI's BeGaze 2.1 software uses 100 pixels as default. To make sense of this value, the experimenter first needs to convert them to degrees of visual angle, taking into account the viewing distance as well as the size and resolution of the screen (p. 24), and also understand which calculation of dispersion is used in the particular implementation they have at hand.

The dispersion setting is closely connected to the imprecision of the recorded data, and some implementations attempt to compensate for such noise. For instance, the ASL Eyenal dispersion algorithm for 50 Hz data requires three data samples to be outside of the dispersion radius for the fixation to end, not just one. Allowing single data samples to deviate is an insurance against low precision in the data; what we saw in Figure 5.2(b). In a way, this can be seen as a temporal increase of the dispersion threshold to allow one or two deviating fixation samples to pass unnoticed. In contrast, the I-DT ends the fixation as soon as the dispersion criteria are violated, which makes it more sensitive to noise, possibly requiring a higher dispersion setting.

Researchers have not paid too much attention to the dispersion setting, but Rötting (2001) reviews studies with dispersion settings ranging from 0.5°–2° of visual angle. Blignaut and Beelders (2009) and Blignaut (2009) argue that the optimal dispersion setting is 1° (for the radius dispersion measure), but relies heavily on the dispersion measure used.

The minimal fixation duration setting has been a long-standing discussion among researchers, however. Inhoff and Radach (1998) write that they themselves mostly use a cutoff point for fixation duration of 50 ms, but that many of their reading research colleagues use cutoff points ranging from 70–100 ms (but do not state which algorithms they use). Rötting

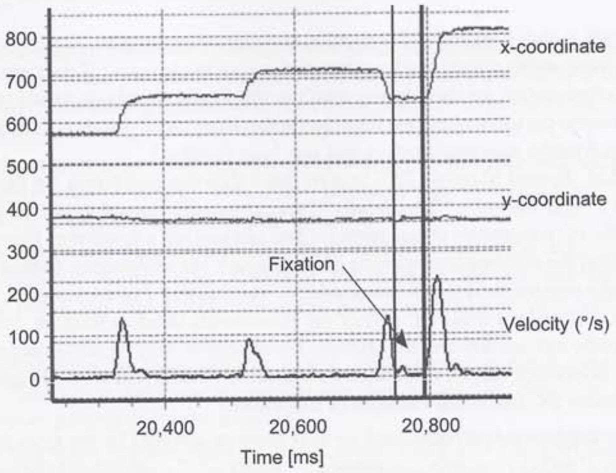


Fig. 5.6 A <45 ms fixation; following a regressive saccade during reading (Data from our example reading set). Excluding the glissade (the little bump in the velocity curve), the fixation duration becomes even shorter; about 30 ms. Data recorded at 1250 Hz with a tower-mounted system.

(2001) summarizes a number of studies, mainly in human factors, that use dispersion algorithms and duration settings ranging from 60–120 ms. At the other end of the scale, Granka, Hembrooke, Gay, and Feusner (2008) use a 200 ms duration setting. Manor and Gordon (2003) notice that 200 ms has become the de facto standard in clinical studies, originally derived from a 1962 study of eye movements in reading. Engmann *et al.* (2009) used no cutoff at all, but found that only 3.9% of the fixations in their study had a duration of less than 100 ms. Is this divergence in settings a problem for researchers who want to compare their results against someone else's?

Perhaps we could decide duration setting on the basis of what is known about how short fixations can be. However, there seems to be no concensus on how common even shorter fixations really are. They do exist, that is clear, as shown by the fixation in Figure 5.6, which measures exactly 45 ms when including the smaller velocity peak after the main saccade. If we exclude the glissade duration, the true fixation is around 30 ms in duration.

As we will see on page 377, information intake may be closed for the entire duration of such short fixations, but is this a good reason to exclude them? Rötting (2001) seems to argue that we should, while others use cutoffs without specifying the motivation. But the short fixations are still real oculomotor events, even if intake is closed. The fixation blob in the raw data files represents an oculomotor event, and it is our task to measure it as best we can, and distinguish it from other short periods of stillness that can be found in the data. The question of intake is very important, but should not be built into the event-detecting algorithms.

The EyeLink velocity algorithm allows for 'low', 'medium', or 'high' saccade sensitivity, although the settings dialogue in Figure 5.1(d) shows only 'normal and high'. Medium sensitivity corresponds to a velocity threshold of 30°/s and an acceleration threshold of $8000°/s^2$, while the high sensitivity uses 22°/s and $4000°/s^2$. The algorithm assumes that a given sample of raw data is part of a saccade in progress, if at least one of the velocity and the acceleration values is above the respective threshold. This is sensible when detecting saccades online. It is safer to use two criteria than only one, as there is only one chance to get it right. For the same reason, the settings for the EyeLink algorithm are chosen *before* recording the data. The EyeLink manual of 2007 recommends the more sensitive setting for oculomotor research, and the medium setting for cognitive and reading research (compare the recommendations in Figure 5.1(b)), arguing that "The larger threshold also reduces the number of microsaccades detected, decreasing the number of short fixations (less than 100 ms in duration) in the data" and noting that "Some short fixations (2% to 3% of total fixations) can be expected, and most researchers simply discard these". Not everyone discards the short fixations, however. Velichkovsky, Dornhöfer, Pannasch, and Unema (2000) not only take them seriously, but name them 'express fixations', after finding that they make up 7% of the total number of fixations given by their EyeLink system in a car simulator task. However, do poor data, noise, microsaccades, smooth pursuit, and a too low velocity threshold—below the precision level of the system—lie behind these frequent 'express fixations', rather than actual oculomotor behaviour?

There is a large spanwidth of velocity threshold settings among researchers. Duchowski (2007, pp. 149–152) makes a theoretical argument about the settings for velocity algorithms, suggesting a lower threshold of 130°/s, which "should effectively detect saccades of amplitudes roughly larger than 3°". Most other researchers use lower velocity threshold settings. For instance, Smeets and Hooge (2003) used a velocity threshold of 75°/s when studying rather large saccades, and Inchingolo and Spanio (1985) compare the settings 10°/s and 50°/s. Beintema, Van Loon, and Van Den Berg (2005) chose the very low setting of 20°/s, but added a minimal saccade amplitude criterion of 1°, and a minimal duration between saccades of 30 ms to distinguish saccades from noise.

While the dispersion setting may be difficult to motivate, the choice of thresholds for velocity algorithms could be made in relation to the purpose of your study: what size of saccades do you want to detect, how much noise is there in the recorded fixations, and where is the line between the velocities of the fastest saccades you want and the slowest movement due to artefacts you have? The precise settings inside these spans could be selected from visual inspection of some typical samples in your data, using a plot of velocity and position, such as Figure 5.7. The following paragraph summarizes issues related to the saccade velocity threshold:

- **Saccade velocity threshold** The major setting. How small are the saccades you need to detect? Detection of small saccades requires a lower threshold. How much noise is there in the fixations? A lot of noise requires a higher threshold. Settings in the literature typically range from 20–130°/s, as discussed above.

In fact, the problem with undetected fixations seen in Figure 5.3 was that the default saccade velocity threshold (75°/s) was set too high. This means that short saccades and their two surrounding fixations are grouped as one single fixation. We can see four such short saccades in Figure 5.7, three of which move with a velocity below 50°/s. An appropriate setting for this data is rather 30–40°/s.

Although the saccade velocity threshold is the most commonly used, there are three additional thresholds for velocity and acceleration based algorithms:

- **Saccade on- and offset velocity** Deciding when a saccade starts and stops, and is always equal or lower than the saccade velocity threshold. For high-quality recordings, a setting of 10–15°/s is often used but there is no consensus on how such thresholds should be set.
- **Maximum velocity threshold** A little-used artefact-removal threshold. The fast-moving artefactual movements from split and false corneal reflections, mascara, and droopy eyelids are above the interval 750–1000°/s, which can be considered as a physical limitation on how fast the eye can move. Not many algorithms use an upper velocity

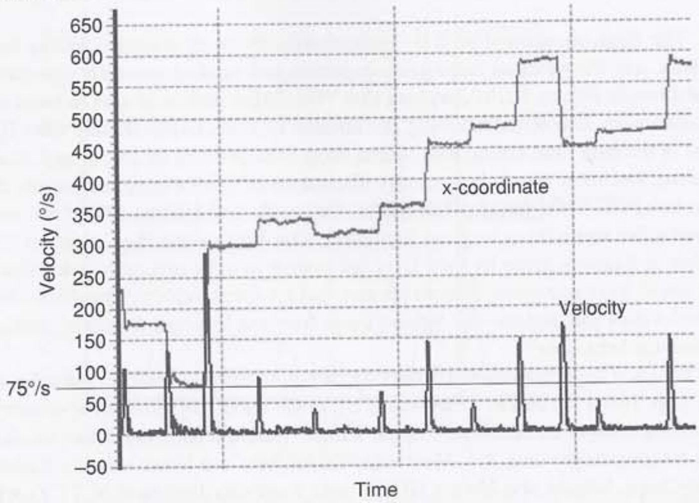


Fig. 5.7 Gaze velocity (black) and gaze x-coordinate (grey) for reading data. The vertical scale is in °/s and pixels, respectively. The horizontal scale is samples (time). 75°/s is marked by a line, which is clearly too high for many of the saccades. 1250 Hz data from a tower-mounted system.

threshold, but Duchowski (2007) makes a theoretical argument suggesting 750°/s as a suitable threshold.

- **Saccade acceleration threshold** Used in the EyeLink algorithm to allow for quick detection of saccades online. An acceleration threshold can also be useful for distinguishing saccades from periods of smooth pursuit; quick pursuit velocity can be larger than slow (small) saccade velocity, but saccades always have larger accelerations (Behrens & Weiss, 1992).

The EyeLink software allows for *post-recording filtering* of the fixation and saccades resulting from the online saccade algorithm. The filter takes minimal fixation duration and minimal saccadic amplitude as settings. Defaults thresholds are 50 ms and 1°. Subsequent fixations that are shorter and closer than the threshold settings stipulate are merged into one fixation. Remedying noisy recordings post-hoc seems to be the major function of this tool. It introduces two new settings, however, making it a total of four settings for the algorithms deciding what fixations and saccades should remain for data analysis. Overall, the many heuristic elements of the EyeLink online saccade algorithm appear to be difficult to overview for the average user, which is perhaps why the settings dialogue primarily provides the summary settings of 'medium' or 'high' saccade sensitivity (Figure 5.1(d)).

Effects of settings

It has long been known that fixation and saccade output is very sensitive to the choice of algorithm settings (Karsh & Breitenbach, 1983, e.g.). Using 60 Hz data and a dispersion-based algorithm, Shic, Scassellati, and Chawarska (2008) show that the effect of parameter changes on mean fixation duration is a linear function of parameters, with a considerable slope. As our reading data (p. 5) in Figure 5.8 show, the effect is that all basic fixation measures are heavily altered when using the common dispersion-based I-DT algorithm. Both the dispersion and the duration settings may give rise to artificially significant differences that may change the result of a study completely. For instance, the average fixation duration at setting

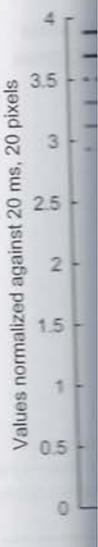


Fig. 5.8 How fixatio
mercial implementat
that from 50 Hz data

60 ms and 60 pi
60 pixels (two-si
thing happens if
duration at 100 m

Fixation dura
ation thresholds
60 pixel setting t
around the 200 r
400–600 ms. Sac
which in turn af
stance). Even th
duration settings

These effects
rithms using vel
affected by para
ple, the average
$p < 0.01$). Shic
the saccade velo
the determining
similar, studies
parable. It is im
that many other
dependent meas

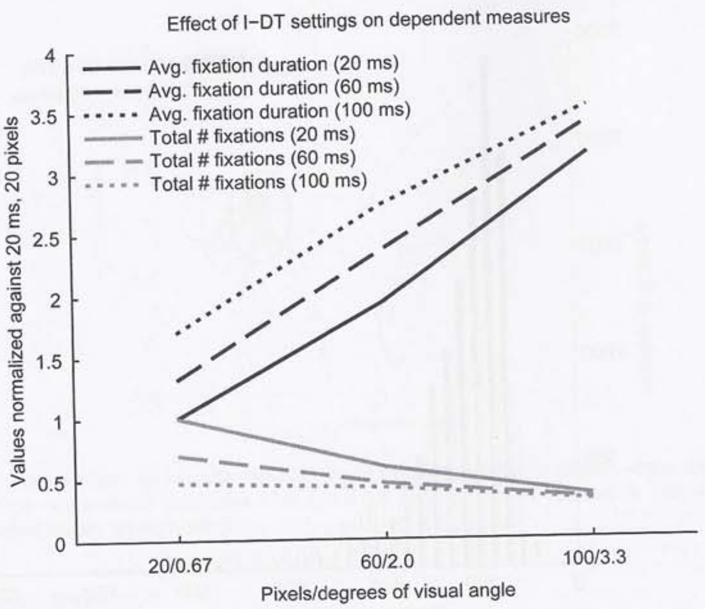


Fig. 5.8 How fixation measures differ with different dispersion diameters and duration settings in a commercial implementation of the I-DT algorithm (1250 Hz reading data from page 5). The slope is similar to that from 50 Hz data in Shic, Scassellati, and Chawarska (2008).

60 ms and 60 pixels differs significantly from the average fixation duration at 100 ms and 60 pixels (two-sided t-test with 36490 fixations each, $t(36489) = 3.07$, $p < 0.01$). The same thing happens if you change the dispersion from 60 pixels to 100 pixels while keeping the duration at 100 ms ($t(26950) = 3.22$, $p < 0.01$).

Fixation durations will not only differ in their averages—a change in dispersion and duration thresholds also alters the distribution, as shown in Figure 5.9. A change from a 100 ms 60 pixel setting to a 100 ms 100 pixel setting dramatically decreases the number of 'fixations' around the 200 ms duration, and increases the number of 'fixations' with durations around 400–600 ms. Such a change in distribution affects averages but also the variance of the data, which in turn affects all your variance-based significance tests (t-tests and ANOVAs, for instance). Even this small examination of the I-DT algorithm clearly shows that dispersion and duration settings should be chosen with the utmost care.

These effects are not unique for dispersion-based algorithms, but are also present in algorithms using velocity data. Figure 5.10 shows how basic saccade and fixation measures are affected by parameter changes in the SMI velocity algorithm; at a 90°/s setting, for example, the average fixation is 2.5 times as long as it is at the 30°/s setting ($t(14340) = 2.85$, $p < 0.01$). Shic, Scassellati, and Chawarska (2008) found similar variation when changing the saccade velocity setting from 18°/s to 81°/s. It is clear that the choice of setting can be *the* determining factor to the success or failure of an eye-tracking study. Although otherwise similar, studies using different settings of the peak saccadic velocity are not directly comparable. It is important to notice that the basic measures in Figure 5.10 are the foundation that many other dependent measures in eye-tracking research are built upon. Virtually all dependent measures will alter their values when this setting is changed.

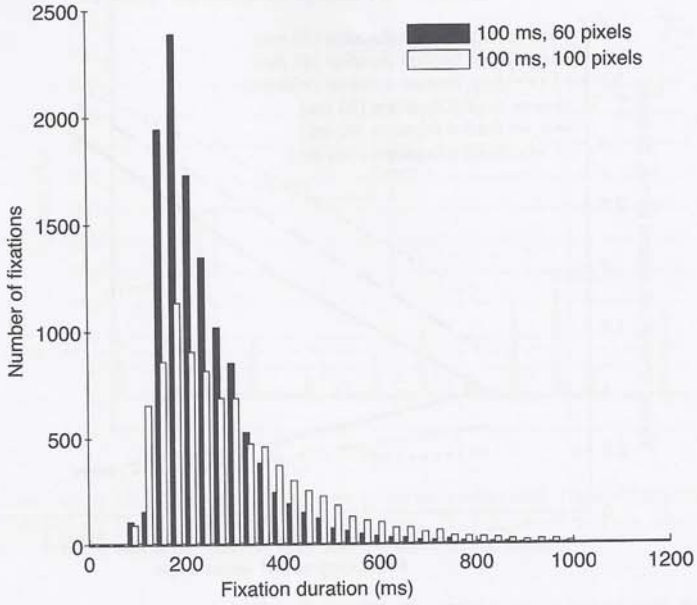


Fig. 5.9 Distribution of fixation durations for two dispersion settings of the I-DT algorithm (data source described on page 5).

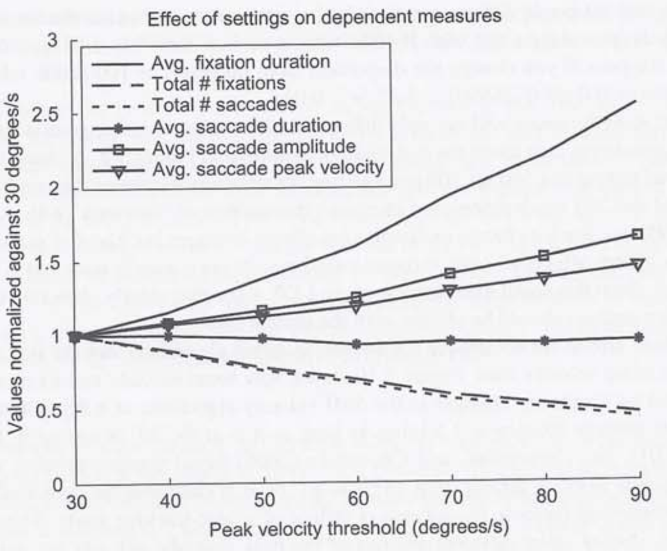


Fig. 5.10 How important dependent variables change with the setting of saccade velocity threshold in the SMI velocity algorithm (a commercial implementation of Smeets & Hooge, 2003). Reading data recorded at 1250 Hz and described on page 5.

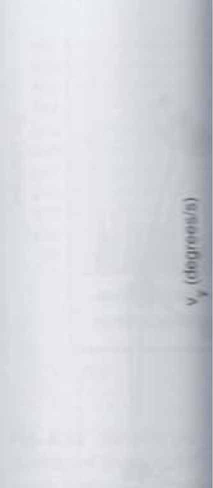


Fig. 5.11 Control ellips
locity is large—are sacc
tower-mounted system

Data driven thresh

It is well known th
tasks, trials, and even
with it throughout th
velop algorithms tha
(1981) suggested loc
pensate for varying
and Schröder-Preiksc
momentary accelera
data acquired at 100
Hollands (1996), wh
each side of the cur
the difference betwe
Niemenlehto (2009)
constant false alarm

Assuming the no
whole trial, and then
dominant principle f
and Kliegl (2003),
and then set the thr
outside the control e
illustrates. Since the
principle can be use
choosing saccade de
Holmqvist, 2010; V

5.4.2 Noise, a

Noise can derive fr
unwanted variation

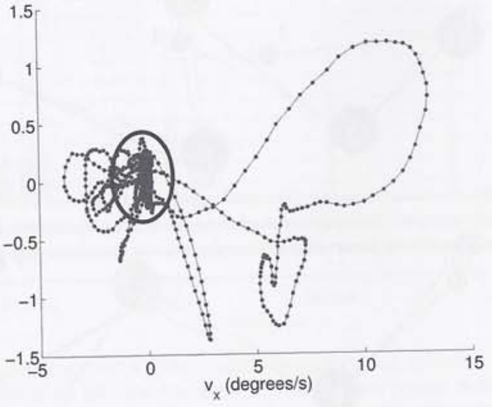


Fig. 5.11 Control ellipse for saccade detection where samples outside the ellipse—where the eye velocity is large—are saccade candidates. One and a half seconds of data collected at 1250 Hz with a tower-mounted system during reading.

Data driven threshold

It is well known that the noise levels in eye-tracking data can change across individuals, tasks, trials, and even within trials, so why should we choose a setting subjectively and stick with it throughout the entire analysis? This particular question has led researchers to develop algorithms that let the data itself assist in how to set the thresholds. Tole and Young (1981) suggested locally adapting the acceleration threshold used to detect saccades to compensate for varying noise levels they observed in the data. Similarly, Behrens, MacKeben, and Schröder-Preikschat (2010) proposed a saccade detection algorithm where an adaptive, momentary acceleration threshold was calculated based on the preceding 200 samples (for data acquired at 1000 Hz). A related algorithm is described by Marple-Horvat, Gilbey, and Hollands (1996), who use a "double-window" technique where two temporal windows on each side of the current velocity sample are subtracted, and a saccade is detected only if the difference between the average value within each window exceeds a certain threshold. Niemenlehto (2009) based the resilience against varying noise for saccade detection on a constant false alarm technique.

Assuming the noise is constant over a trial, one can estimate the noise level over the whole trial, and then use this estimate to set the thresholds. For fixational eye movements, the dominant principle for microsaccade detection is based on the algorithm proposed by Engbert and Kliegl (2003), who first estimate the velocity noise in x and y-dimensions separately, and then set the thresholds as multiples of the estimated variance in the noise; all samples outside the control ellipse formed by such thresholds are saccade candidates, as Figure 5.11 illustrates. Since the dynamics of microsaccades are similar to normal saccades, the same principle can be used to find appropriate saccade detection thresholds. Similar strategies for choosing saccade detection thresholds have been employed in other recent work (Nyström & Holmqvist, 2010; Van Der Lans, Wedel, & Pieters, 2010).

5.4.2 Noise, artefacts, and data quality

Noise can derive from the oculomotor system, the eye-tracker, or the environment, and adds unwanted variation to the acquired data. Artefacts can be seen as a special type of noise,

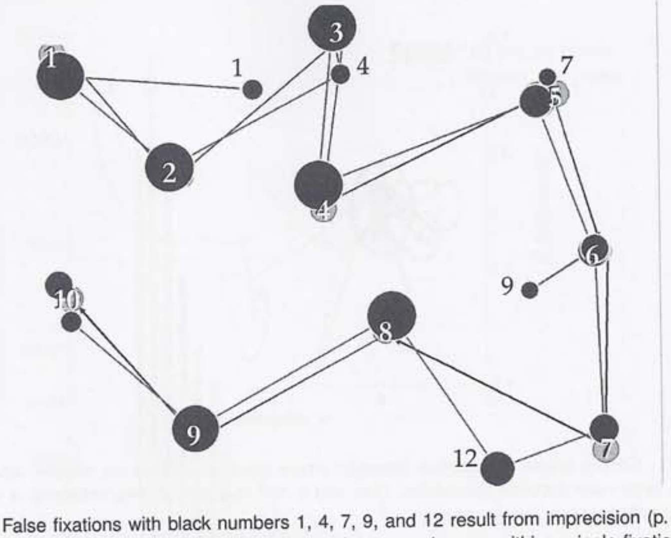


Fig. 5.12 False fixations with black numbers 1, 4, 7, 9, and 12 result from imprecision (p. 33) in data. This means that raw data differ so much from sample-to-sample even within a single fixation that some of the samples end up outside of the dispersion radius, and will be segmented into minute fixations of their own. Recorded at 50 Hz on the remote system on a blue-eyed participant with contact lenses, and analysed using a dispersion-based algorithm with manufacturer standard settings. The task was to look at the centre of each white number in increasing order. Dark filled circles represent detected 'fixations'.

but are typically larger and easier to distinguish from known eye-movement characteristics. Data quality is a more imprecise term, but is related to accuracy, precision, percentage of data loss, perhaps in addition to a subjective rating from the person responsible for the recording. Having access to all these quality indicators gives you an idea of whether the recorded data are useful for further analysis, or should be discarded.

It is generally easier to detect events in recordings with high data quality. Figure 5.8 showed results from data with high quality in the sense that the calibration was judged as good and no problems were reported by the operator during the recording. Unfortunately, not all recorded data have the same high quality, and the algorithms need to deal with the imperfections too. In fact, data quality is an important factor to consider when using the algorithms, which can erroneously interpret various recording imperfections as actual eye-movement events.

In dispersion-based algorithms, high noise levels can make a sample that rightfully belongs to a fixation move outside of the dispersion radius, end the fixation, and trigger a new one. Figure 5.12 shows how a number of such false 'fixations' are created from stray samples in the vicinity of the real fixations. Some varieties of dispersion algorithms attempt to address this problem by temporarily allowing a few samples to exceed the maximum dispersion threshold without ending the fixation (such as ASL Eyenal). High velocity artefactual eye movements will also be assumed to be saccades with intermediate fixations, but the 'fixations' will now be deleted because they are too short.

The velocity algorithms are also best suited for high quality data. High velocity artefacts and imprecision are major obstacles; if the imprecision inside a fixation has a velocity above the velocity threshold used, it gives rise to false 'saccades', effectively ending the fixation. The velocity threshold can be superseded many times, giving a whole array of unrealistically short 'fixations'. Figure 5.13 shows how this happens during the first and fifth fixations.

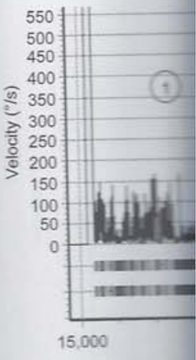


Fig. 5.13 Variable pr
participant looks at th
fourth fixation). Recor
with a threshold of 75

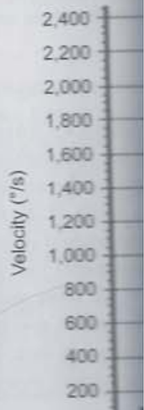


Fig. 5.14 Saccadic
(eye image in Figure
Recorded at 1250 H

With both dis
some extent be a
or a higher veloci
the consistency o
the velocity thres
high that many re
surrounding fixat
of the two real fix

Figure 5.14 s
over 1000°/s and

recision (p. 33) in data.
ingle fixation that some
into minute fixations of
with contact lenses, and
s. The task was to look
nt detected 'fixations'.

ement characteristics.
on, percentage of data
ible for the recording.
her the recorded data

a quality. Figure 5.8
ration was judged as
rding. Unfortunately,
need to deal with the
sider when using the
ections as actual eye-

ple that rightfully be-
ion, and trigger a new
eated from stray sam-
algorithms attempt to
the maximum disper-
gh velocity artefactual
fixations, but the 'fix-

High velocity artefacts
n has a velocity above
ly ending the fixation.
array of unrealistically
and fifth fixations.

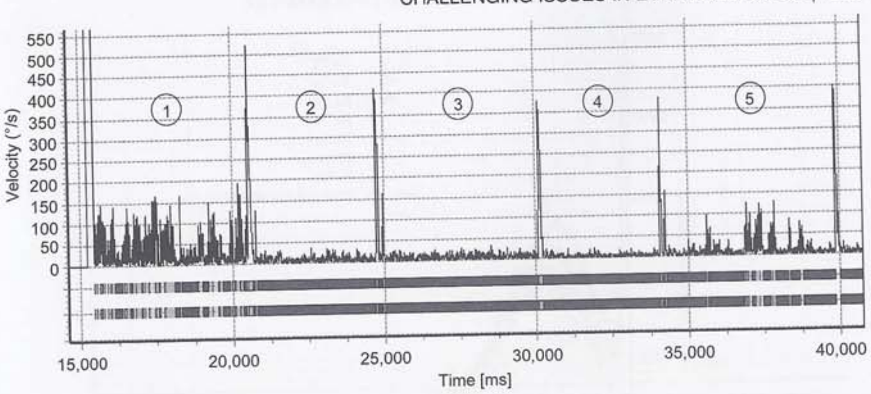


Fig. 5.13 Variable precision. Data acquired for oculomotor fixations 1–5 are noisy (imprecise) when the participant looks at the top of the stimulus (first and fifth fixations) and precise at the bottom (second to fourth fixation). Recorded with a remote system at 250 Hz and analysed with a velocity-based algorithm with a threshold of 75°/s.

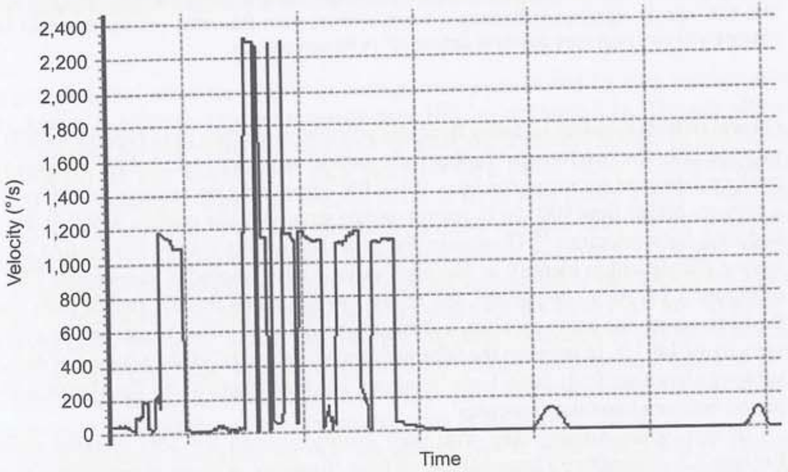


Fig. 5.14 Saccadic velocity plot showing the effect of having multiple competing corneal reflections (eye image in Figure 4.13). High-speed artefacts to the left, and slower reading saccades on the right. Recorded at 1250 Hz a tower-mounted system, participant with contact lenses.

With both dispersion- and velocity-based algorithms, the effect of imprecision can to some extent be alleviated by raising the threshold setting. With a larger dispersion radius, or a higher velocity threshold, sample-to-sample motion can be quicker without endangering the consistency of fixations. This remedy comes at a price, however. For instance, raising the velocity threshold of Figure 5.13 above the peaks of this imprecision will make it so high that many real saccades will not be identified. When a saccade is not identified, the two surrounding fixations are reported as one single 'fixation', with a duration that equals the sum of the two real fixations and the intermediate saccade.

Figure 5.14 shows high-speed *optic artefacts*: false eye movements with velocities well over 1000°/s and virtually infinite acceleration. Such velocities appear for instance when the

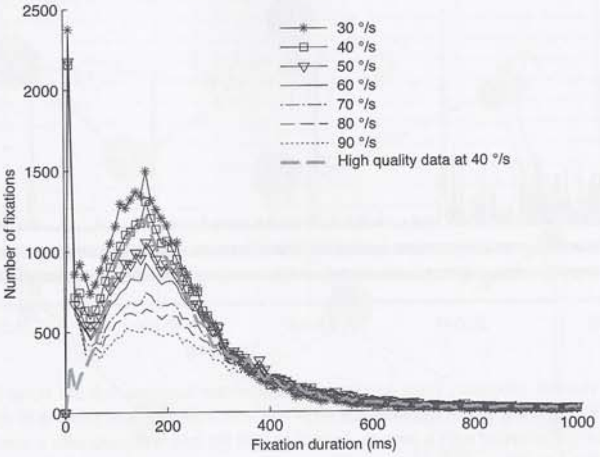


Fig. 5.15 13 readers with very poor recording quality (mascara, contact lenses, drooping eyelids, etc). The SMI velocity algorithm in BeGaze 2.1 with different peak (saccade) velocity threshold settings. 10 readers with very high data quality at setting 40°/s for comparison.

corneal reflection moves instantly from one position to another, as in Figures 4.12(d) and 4.13 on page 124. The two 'bumps' further right are real saccades, included for comparison. In the reading data used for examples here, 0.4–0.9% (depending on settings) of the saccades had a velocity higher than 800°/s. In corresponding data with low quality, 4–9% of the saccades had such high velocities.[18] Obviously, the poor data quality caused this tenfold increase, by having the algorithm identify as saccades various false 'saccadic' movements like the ones in Figure 5.14. As a comparison, the return sweeps, when readers switch from one line to the next, across the entire monitor, a distance about 25° of visual angle, had an average peak velocity of 440°/s. In between the false high speed 'saccades' of the artefactual data, the SMI velocity algorithm finds false 1 ms 'fixations', as it attempts to fill the almost non-existent period between two false 'saccades'.

In fact, when running data with poor quality through the SMI velocity algorithm of BeGaze 2.1, it identifies a huge number of false 'fixations' with durations shorter than 40 ms, as clearly shown in the histogram of Figure 5.15. Similar 'blips' of very short 'fixations' in the related I-VT algorithm were reported by Salvucci and Goldberg (2000). In both cases, the lack in many velocity-based algorithms of a temporal criterion for fixation duration could be one part of the problem. The other part is the lack of an upper velocity threshold that could eliminate high-speed artefacts with intermediate 1-sample false 'fixations'. For very high quality data, the number of unreasonably short 'fixations' is much smaller (dotted comparison line), but they still exist. For durations above ~80 ms, the distribution is very similar. This suggests that with an improved algorithm, a good portion of the poor quality data could in fact be used, at least for some types of analyses.

5.4.3 Glissades

Interestingly enough, the very high quality data in Figure 5.15 *also* exhibit a small proportion of 1 ms 'fixations'. In the high quality data, the unreasonably short fixations are not found

[18] Analysis was made using the SMI velocity algorithm of BeGaze 2.1 with a threshold of 40°/s

enses, drooping eyelids, etc).
elocity threshold settings. 10

n Figures 4.12(d) and 4.13 ded for comparison. In the tings) of the saccades had ality, 4–9% of the saccades ed this tenfold increase, by ' movements like the ones rs switch from one line to angle, had an average peak he artefactual data, the SMI fill the almost non-existent

SMI velocity algorithm of urations shorter than 40 ms, of very short 'fixations' in berg (2000). In both cases, a for fixation duration could oper velocity threshold that e false 'fixations'. For very much smaller (dotted com- e distribution is very similar. f the poor quality data could

so exhibit a small proportion hort fixations are not found

threshold of 40°/s

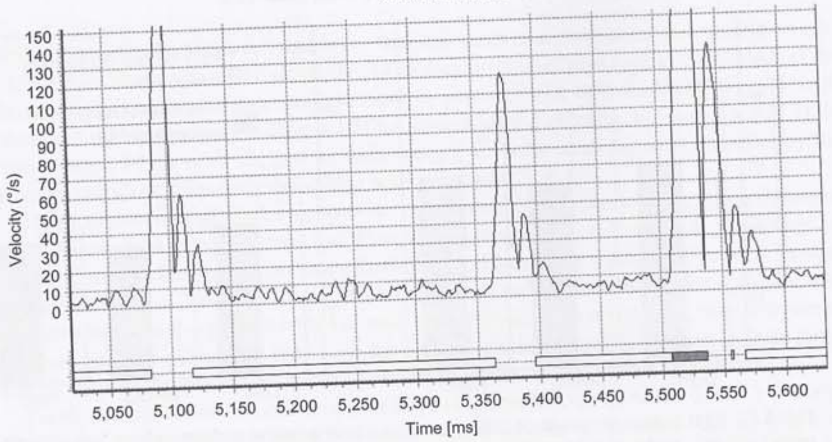


Fig. 5.16 Saccades with multiple velocity peaks and false 1 ms fixations between them. Recorded at 1250 Hz with a tower-mounted system. Fixations (white lines), saccades (grey lines), and undetected events (gaps in between) according to SMI BeGaze 2.1 are indicated at the bottom of the graph.

amongst noise, but when the algorithm faces a main saccade ending with smaller velocity peaks, knows as *glissades* (see definition on page 183), as in Figure 5.16. The saccade of this participant does not stop at the intended fixation goal, but continues beyond it, and then back again, but too far, and thus wobbles back and forth for a while, before it comes to a standstill and the fixation can start. The velocity peaks in these very strong glissadic movements are well beyond the normal velocity threshold (here 40°/s), and therefore the SMI velocity algorithm finds, not a fixation right after the saccade ending, but essentially a new 'saccade'. Therefore, two of the saccades in Figure 5.16 are not recognized as saccades at all. The third saccade *is* recognized, but only its first velocity peak. We see false 1–10 ms 'fixations' between the peaks inside the second and the third saccades. Glissades of this extreme size are not uncommon in data that we have recorded from reading, mathematical problem solving, and many other tasks. Between 20–40% of all saccades end with a glissade, but almost no saccades start with this type of movement (Nyström & Holmqvist, 2010).

Some algorithms treat glissades like just another type of noise. Stampe (1993) describes glissades as noise that "includes ringing or overshoot artefacts following saccades, which can confuse the saccade detector into extending the saccades into the next fixation". Similarly, Duchowski (2007) describes filters that are optimized for idealized saccades (without glissades), thus smoothening out the glissades before the fixation algorithm gets the velocity data. The EyeLink parser seems to assign glissades to fixations, as shown in Figure 5.18. However, from this figure it remains unclear whether the EyeLink algorithm at times generates very short fixations as a result of poor glissade treatment, or whether it is robust enough to avoid that. The manual tends to point in the former direction: "Post-processing or data cleanup may be needed to prepare data during analysis. For example, short fixations may need to be discarded or merged with adjacent fixations, or artefacts around blinks may have to be eliminated" (SR Research, 2007). However, whether the short fixations are due to glissades remains to be investigated.

First fixation duration values are *extremely* sensitive to these short 'fixations' in data, because when a participant makes a saccade into an area of interest, the saccade very often ends with a glissade, and the SMI velocity algorithm often outputs a false 'fixation' before the glissade, and also before the real fixation. Therefore, during analysis, forgetting to remove short

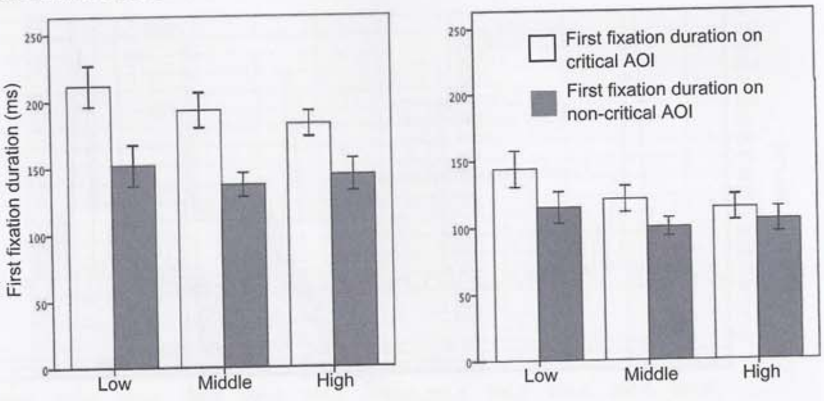


Fig. 5.17 First fixation durations on critical areas in a mathematical problem solving task (described on page 5). 'Low', 'High', and 'Middle' are three groups of participants with varying levels of mathematical competences. Fixations were detected using the SMI velocity algorithm and a 40°/s threshold. The same data is presented both left and right, but to the right the data include the small 'fixations'. Why such a huge difference on first fixations? After a saccade into an area, the very short 'fixation' that the SMI velocity algorithm finds between the saccade and the following glissade is taken as the first fixation, since it precedes the real fixation. Note that the very short 'fixations' not only make the averages much lower, they also introduce noise that conceals significant effects.

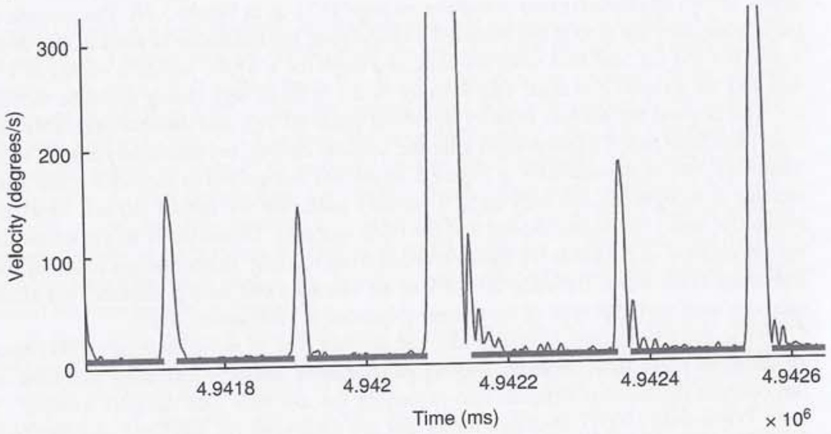


Fig. 5.18 Event detection with the EyeLink parser for reading data ('Normal' sensitivity). Data were collected with the head-mounted system at 500 Hz. Thick lines at the bottom of the graph indicate where 'fixations' have been detected. Note how glissadic movements are systematically assigned to the following fixation and how parts of the saccades are also attributed to fixations.

'fixations' before calculating first fixation duration values means using lots of unreasonably short 'fixations', and getting averages that are lower, as in Figure 5.17.

Glissades have until recently been treated unsystematically and differently across algorithms and even within the same algorithm, sometimes being attributed to saccades, other times to fixations. Some researchers express the need to exclude them completely from further analysis. Gilchrist and Harvey (2006) require the velocity to remain below 30°/s for at least five samples (20 ms) to count as a fixation, which "excludes the interval of ocular in-

stability just after the
of fixation location, al
glissade is assigned to
considered the eye ve
period started 20 ms
dynamic overshoot fr

The prevalence o
larger, shorter, and he
eradicated or transfor
man, 1995; Frens & V
any video-based high
they may be more di
than 20 ms (one samp

Positioned in betw
the glissade to. The
(McConkie & Losch
quence relationships
nature. As glissade-d
more studies using t

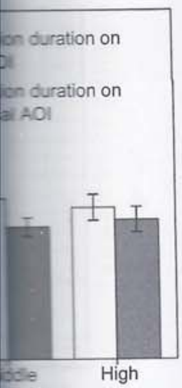
5.4.4 Samplin

There is a tendency
sampling frequency,
pling frequencies (s
velocity-based algo

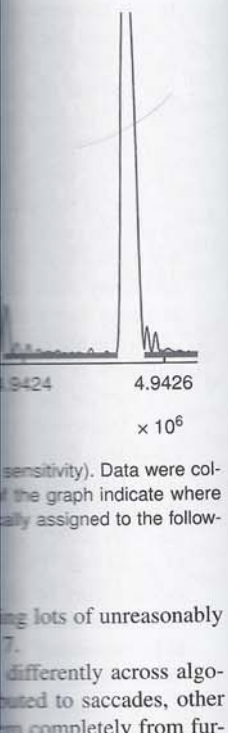
Dispersion-bas
defined by the disp
anytime. The two
black lines, start a
is clearly incorrect
points and dispers
Figure 5.19 that sta
point of the new
away from its lan
data samples are c
the saccade. This
think the dispersi
deep inside the n
with its new cent
accommodate the

Dispersion-ba
vidual saccade a
all sampling freq
since the samplin

Velocity-base
of sampling freq
possible to get cl
The reason it is
in particular acc



olving task (described on ng levels of mathematical 0°/s threshold. The same nall 'fixations'. Why such ort 'fixation' that the SMI as the first fixation, since he averages much lower,



sensitivity). Data were col- f the graph indicate where ally assigned to the follow-

ng lots of unreasonably 7.

differently across algo- uted to saccades, other m completely from fur- main below 30°/s for at he interval of ocular in-

stability just after the saccade", and argue that this leads to a more accurate calculation also of fixation location, although fixation durations may be shorter than typically reported, as the glissade is assigned to the saccade. Investigating post-saccadic drift, Collewijn *et al.* (1988) considered the eye velocity during a 100 ms period after a saccade, but arguing that "This period started 20 ms after the end of each saccade in order to avoid contamination by the dynamic overshoot frequently associated with a saccade".

The prevalence of glissades appears to vary across eye-trackers, being more common, larger, shorter, and heavily curved in DPI-systems compared to video-based, while they are eradicated or transformed to post-saccadic drift in coil-based eye-trackers (Deubel & Bridgeman, 1995; Frens & Van Der Geest, 2002). Glissades can be observed in data collected with any video-based high-speed eye-tracker with good precision. In low-speed, remote systems, they may be more difficult to see, since the average glissade duration is only slightly larger than 20 ms (one sample in a 50 Hz eye-tracker) (Nyström & Holmqvist, 2010).

Positioned in between saccades and the following fixation, the question is which to assign the glissade to. The fact that perceptual visual intake appears to be closed during glissades (McConkie & Loschky, 2002), as well as the fact that glissades follow the same main sequence relationships as saccades (p. 318) tells us that they are predominantly saccadic in nature. As glissade-detecting algorithms become more available, we can surely expect to see more studies using this event.

5.4.4 Sampling frequency

There is a tendency for dispersion-based algorithms to be used for data collected at a low sampling frequency, such as 50 Hz, and velocity algorithms for data collected at higher sampling frequencies (say > 200 Hz), but there are also exceptions. The Tobii Fixation filter is a velocity-based algorithm for fixation detection in data as slow as 30 and 50 Hz, for instance.

Dispersion-based algorithms end a 'fixation' as soon as the raw samples cross the border defined by the dispersion radius. In high-speed data, this border can be crossed just about anytime. The two 'saccades' identified by the I-DT algorithm in Figure 5.19, indicated by black lines, start a bit into a real saccade, and two thirds into a real fixation, respectively. This is clearly incorrect, and reflects the fact that the dispersion algorithm is only aware of centre points and dispersion, but only indirectly velocity and acceleration. Take the 'fixation' in Figure 5.19 that starts in the middle of the first real saccade. I-DT starts calculating the centre point of the new 'fixation' here, even though the eye is in full motion and some distance away from its landing point in the real fixation. Once the eye has reached the real fixation, data samples are close to the dispersion radius border, as most of the distance was spanned by the saccade. This means that even very small movements inside the real fixation make I-DT think the dispersion border has been crossed. In Figure 5.19, this happens at time 8360 ms, deep inside the real fixation. After a minimal 'saccade', the I-DT starts a new 'fixation', with its new centre point and dispersion, which happen to be chosen generously enough to accommodate the next real saccade inside the 'fixation'.

Dispersion-based algorithms have a large problem with their imprecise estimate of individual saccade and fixation durations. The same miscalculation of fixation onsets occurs at all sampling frequencies, but for the lowest sampling frequencies, this is not as big a problem since the sampling frequency by itself is the major limiting factor.

Velocity-based saccade detection algorithms are better suited for use with a wide spectrum of sampling frequencies. With suitable filtering when velocity data is calculated, it is quite possible to get clear velocity peaks even for 50 Hz data, as shown for instance in Figure 5.20. The reason it is uncommon to use velocity algorithms for low-speed data is that velocity, and in particular acceleration, can be calculated only crudely when the sampling frequency is low.

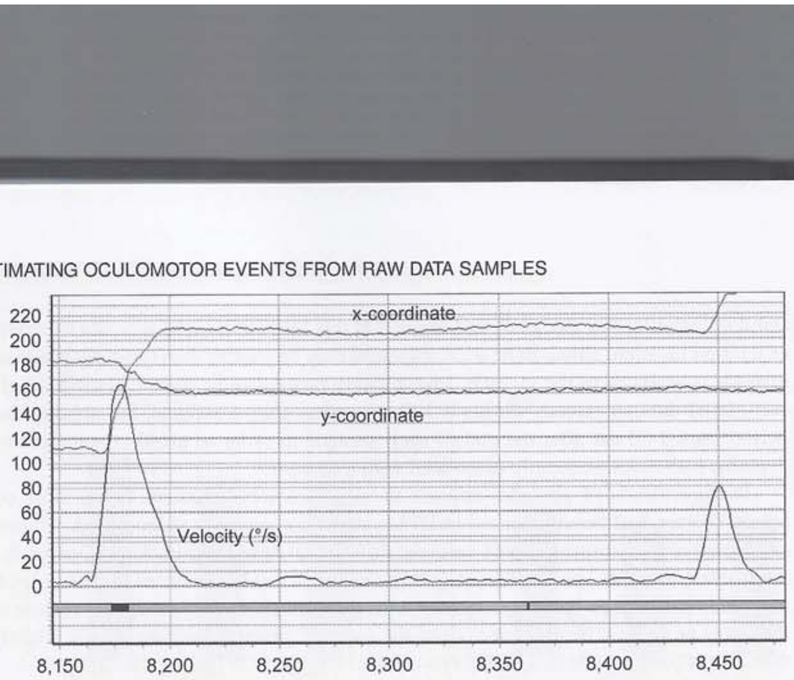


Fig. 5.19 Note how the dispersion algorithm reduces the duration of saccades, and even inserts false 'saccades' in the midst of a fixation in this reading data recorded at 1250 Hz with a tower-mounted system. Grey lines depict the x- and y-coordinates in the coordinate system of the scene video. The dark line is eye velocity. The bottom bar indicates 'fixations' (light) and 'saccades' (darker) according to the I-DT algorithm with 100 ms and 80 pixels settings.

If our task is to detect saccades correctly rather than to measure them with high precision, all we need is a fair estimate of peak velocity, and this we can get even at 50 Hz.

In conclusion, while dispersion-based algorithms do not produce valid event data for higher sampling frequencies, the velocity algorithms have good potential for the entire spectrum of sampling frequencies. The use of acceleration is only suitable for data acquired with high-speed systems, however.

5.4.5 Smooth pursuit

An increasing number of studies use stimuli and experimental set-ups that induce the participant to make smooth pursuit movements, for instance by using animated or video stimuli, or taking the participant out with a head-mounted eye-tracker to make simultaneous head and eye movements. Many of these studies use data recorded at low sampling frequencies, and dispersion algorithms are used to calculate fixations.

Figure 5.20 shows data from a participant walking past a shelf in the supermarket (experiment 3 on page 5). He walks, turns his head, and moves his eyes simultaneously. When applying the I-DT algorithm to the data in this figure, three fairly correct saccades are indeed found, but also four or five false ones. The impact on variables such as fixation duration or saccade rate is therefore disasterous. Such event data cannot be used. Interestingly, the velocity peaks of this 50 Hz data seem to better estimate where the saccades are located, posing the question of whether a velocity-based algorithm would have been a better choice (which finds some support in Munn, Stefano, & Pelz, 2008).

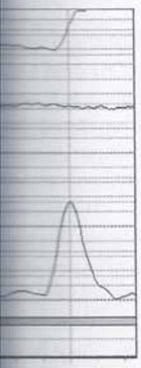
Velocity algorithms typically assign smooth pursuit data into the same category as fixations. Itti (2006), using video data, applies a velocity algorithm to remove all saccades, arguing that the remaining mixture of fixation and smooth pursuit data can be seen as a 'visual intake' category. Depending on the purpose of the study, such a mixed category could be sound or not. Itti wanted to compare all visual intake to that predicted by his algorithm, but made no duration statistics on the data. In most commercial software packages, this is indeed also the best case scenario of how smooth pursuit is handled; in the Tobii Fixation filter im-

Fig. 5.20 Eye mo
a shelf in a superm
video. The grey lin
according to the I-D

Fig. 5.21 Event
were collected a
The lines at the b
smooth pursuit a

plementation,
pursuit data m
these 'fixation
and SMI. Fig
person follow

The worst
movement ca
thresholds onl



and even inserts false
ower-mounted system.
eo. The dark line is eye
g to the I-DT algorithm

h high precision, all
) Hz.
alid event data for
for the entire spec-
r data acquired with

t induce the partici-
or video stimuli, or
ultaneous head and
ng frequencies, and

he supermarket (ex-
multaneously. When
saccades are indeed
fixation duration or
restingly, the veloc-
are located, posing
better choice (which

me category as fix-
emove all saccades,
can be seen as a 'vi-
ed category could be
y his algorithm, but
kages, this is indeed
n Fixation filter im-

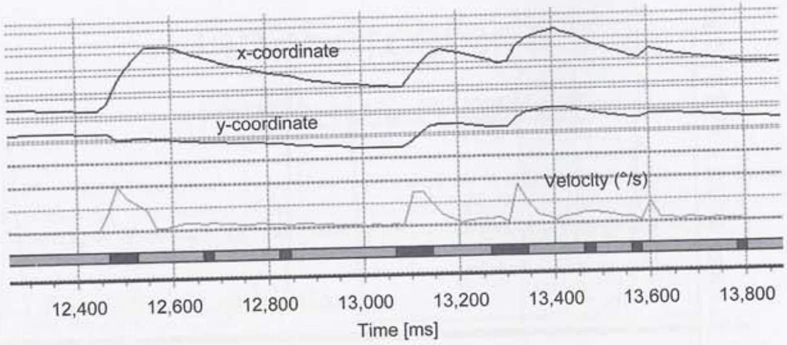


Fig. 5.20 Eye movement data from a head-mounted eye-tracker at 50 Hz on a participant walking past a shelf in a supermarket. Dark lines are the x- and y-coordinates in the coordinate system of the scene video. The grey line is eye velocity. Bottom bar indicates fixations (light grey) and saccades (dark grey) according to the I-DT algorithm with 80 ms and 80 pixels settings.

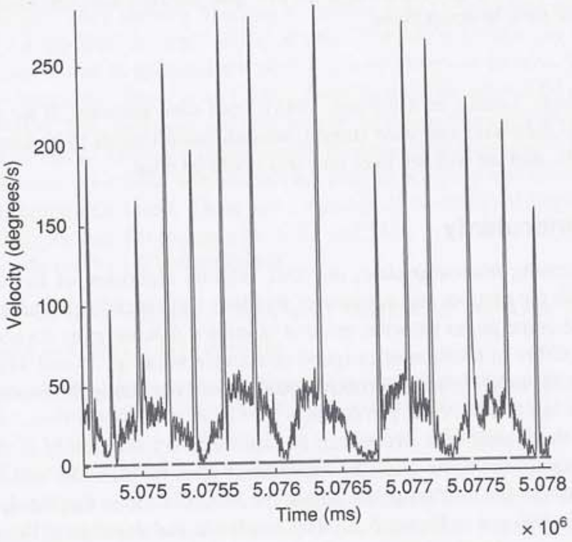


Fig. 5.21 Event detection with the EyeLink parser for smooth pursuit data ('Normal' sensitivity). Data were collected at 500 Hz with the head-mounted system from a person viewing a pendulum movement. The lines at the bottom of the graph indicate where 'fixations' have been detected. Notice how fixation and smooth pursuit are merged into the same category.

plementation, the 'visual intake' category is still labelled 'fixations', and users with smooth pursuit data may be misled into making various statistics on the duration and prevalence of these 'fixations'. This is also the current status for parsers of high-speed data from EyeLink and SMI. Figure 5.21 illustrates how the EyeLink parser treats smooth pursuit data from a person following a pendulum movement on a computer screen.

The worst case scenario, Figure 5.20 is an example of this, is that smooth pursuit eye movement causes an algorithm to output events that are clearly not present. Using velocity thresholds only, it may be hard to separate fast smooth pursuit, which can reach velocities of

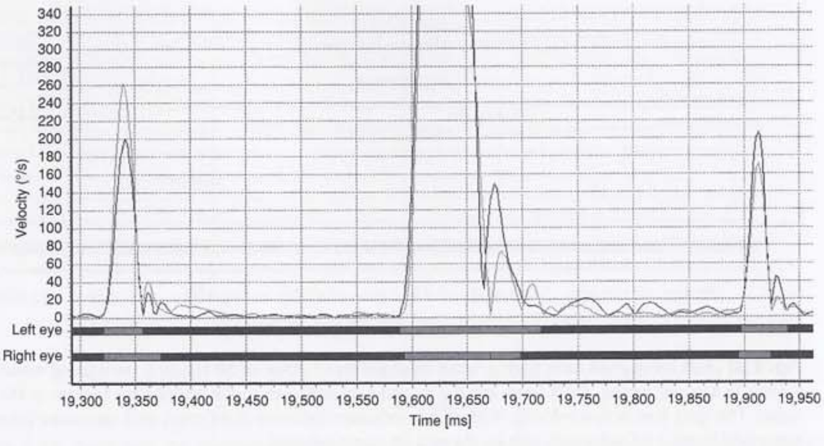


Fig. 5.22 Reading near the borders of a flat monitor: Velocities of saccade far to the right, return sweep, and saccade far to the left. Black line is left eye; grey line right eye. Recorded with a tower-mounted system at 500 Hz in binocular mode.

100°/s (Meyer, Lasker, & Robinson, 1985), from slow saccades. If we want correct fixation and saccade data with animated stimuli, smooth pursuit needs to be identified as an event in its own right, and we will see later how this could be done.

5.4.6 Binocularity

When processing *binocular data*, the SMI velocity algorithm of BeGaze 2.1 finds differences in both the number and duration of fixations and saccades, as indicated by Figure 5.22. This should come as no surprise, since it is known that the eyes do not move in complete synchrony, either in position or in speed and acceleration (p. 24 and 449). Nevertheless, the hard thresholds used by the algorithms could make even subtle differences in eye movement between the left and the right eye count.

Part of the reason why events can be detected very differently is that the two eyes do not make exactly the same glissadic movements after the saccade, and sometimes only one of the eye velocities reaches down below the saccadic offset threshold, examplified by the right (grey curve) eye in Figure 5.22. The amplitude and duration of the saccade then differs between the eyes. Another reason is due to the two eyes having different distances to the right and left part of the monitor. Figure 5.22 shows first a saccade in the far right of the monitor (at around 19,350 ms), then a return sweep (at about 19,650 ms), and finally a reading saccade at the far left of the monitor (at around 19,900 ms). At the right-hand side of the monitor, the right-eye saccades have a larger amplitude and the fixation durations are shorter than for the left eye. Conversely, on the left side of the monitor, the fixations of the left eye will be shorter, and its saccadic amplitudes longer than in the right side of the monitor (compare Figure 2.4 on page 24). This is not really a problem of the algorithm, but rather questions whether we should continue to record monocularly, and accept only the saccades and fixations of the one eye that we happen to select.

Since the majority of eye-tracking research is monocular, velocity algorithms have mostly been applied to monocular data. The I-DT algorithm appears not be used in any real binocular research, probably as it is too imprecise in itself. Binocular event detection algorithms using the covariance between the eyes have been developed (Van Der Lans *et al.*, 2010).

Fig. 5.23 The I-D
minimum duration
dispersed blob on

5.5 Algor

5.5.1 Dispe

Dispersion-base
are implemente
used for low-sp
50 Hz data. In s
to a common c
to one another f
or acceleration
algorithms are
and Schryver (1
to Salvucci and
they tested six
ease of implem
dispersion-base
The pseudo

```
I-DT. Input
1. While t
    (a) Ini
    (b) Whi
    (c) Not
    (d) Rem
```

Dispersion
resent the sam
dow (duration
the temporal t
would then me
is that sequen
5.23. The I-DT
culate the spat
& Chawarska,
different ways

5.5.2 Velo

Fixation de

As for disper
combine it w

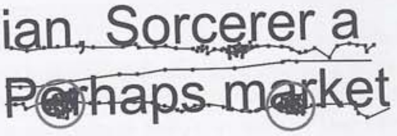


Fig. 5.23 The I-DT detection criteria: gaze must reside within a limited spatial region for a specified minimum duration. Two fixations clearly fulfil the spatial dispersion criterion, but what about the more dispersed blob on Sorcerer? Is that one fixation or two? In the end, your settings will decide.

5.5 Algorithmic definitions

5.5.1 Dispersion-based algorithms

Dispersion-based algorithms are the most common type of event-detection algorithms, and are implemented in many commercial analysis software packages. They have mostly been used for low-speed data, and have long been considered the prime choice when analysing 50 Hz data. In short, dispersion algorithms detect only fixations and collect all other events to a common category. They identify fixations by finding data samples that are close enough to one another for a specified minimal period of time. They do not make any use of velocity or acceleration information to calculate the precise on- or offsets of fixation. Related cluster algorithms are presented by Urruty *et al.* (2007), Santella and DeCarlo (2004), and Goldberg and Schryver (1995b). The most used and also best of the dispersion algorithms is, according to Salvucci and Goldberg (2000), the *identification by dispersion threshold* (I-DT) algorithm; they tested six fixation algorithms with respect not only to accuracy and robustness, but also ease of implementation and speed. There are a number of commercial implementations of dispersion-based algorithms, for example by ASL, and SMI.

The pseudo-code for the I-DT algorithm is:

```
I-DT. Input: (raw data samples, dispersion threshold, duration threshold)
  1. While there are still data samples
      (a) Initialize window over first samples to cover duration threshold
      (b) While dispersion <= threshold, add samples to window
      (c) Note a fixation at the centroid of window samples
      (d) Remove window points from samples
```

Dispersion is defined as $d = [\max(x) - \min(x)] + [\max(y) - \min(y)]$, where (x, y) represent the samples inside the window. The dispersion algorithms combine a temporal window (duration threshold) with a spatial requirement (the dispersion threshold). For instance, the temporal threshold may be 100 ms, and the dispersion threshold 1° of visual angle. This would then mean that only when the data samples stay within a 1° diameter for at least 100 ms is that sequence of data samples considered a fixation. This principle is illustrated in Figure 5.23. The I-DT algorithm has a number of cousins who all use a temporal threshold, but calculate the spatial dispersion criterion somewhat differently (Blignaut, 2009; Shic, Scassellati, & Chawarska, 2008; Salvucci & Goldberg, 2000). Moreover, the algorithmic variations have different ways to deal with noise, as we have pointed out above.

5.5.2 Velocity and acceleration algorithms

Fixation detection algorithms

As for dispersion algorithms, *fixation velocity algorithms* use a duration criterion, but instead combine it with a stillness criterion based on eye velocity. The eye velocity is seldom at the

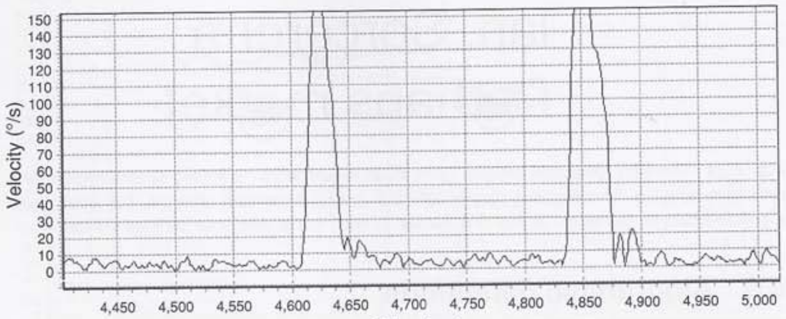


Fig. 5.24 Velocity chart for three fixations and two saccades, recorded at 1250 Hz with a tower-mounted system. Velocity calculated with BeGaze 2.1.

absolute zero level, because of micro-movements in the eye and eye-tracker-related noise. Therefore users of this algorithm must decide an upper velocity threshold for fixations. Figure 5.24 shows a velocity over time chart for three fixations and two intermediate saccades. The velocity during fixations in this reading data has its peaks at 6–10°/s. The shortest saccades typically have velocity peaks of about 30–40°/s. Rötting (2001) summarizes settings for the velocity threshold used for this type of fixation analysis in five quoted studies: < 16°/s, < 20°/s, < 6.58°/s, < 50°/s, < 37.7°/s. The two last settings most likely reflect a considerable noise in the eye-tracking systems used in the quoted studies, and definitely run the risk of categorizing some short saccades as parts of fixations. Thresholds could also vary since the velocity samples have undergone different types of lowpass filtering prior to detection; little filtering requires higher thresholds.

This type of algorithm also requires an additional minimal duration threshold for fixations, which can be set to anything between 60 and 120 ms, according to Röttings review, also see pages 155–156 for an extended discussion. The algorithm thus finds fixations as periods longer than a minimal duration, during which the eye velocity is below a maximum velocity threshold.

Even though the 'fixation radius' and 'Min fixation duration' settings in Figure 5.1(b) on page 148 invite the user to believe that it is a dispersion-based algorithm, the Tobii ClearView fixation algorithm is in principle similar to the I-VT algorithm by Salvucci and Goldberg (2000); the 'fixation radius' setting refers to the maximum distance between two consecutive samples in pixels. Fixations comprise consecutive samples whose distances are shorter than the 'fixation radius' over a period longer than the minimum fixation duration. Note, however, that according to Blignaut and Beelders (2009) classification of dispersion metrics on page 155, the Tobii Clearview algorithm could well fit the under the umbrella of dispersion-based algorithms.

Saccade detection algorithms

A velocity-based saccade detection algorithm focuses on identifying the saccadic velocity peaks. Motion above a velocity threshold, for instance 75°/s, is assumed to be a saccade. In order to differentiate real saccades from artefacts, which can also be fast movements, there are usually additional constraints on saccades, such as a clear speed peak near the middle of the saccade (Smeets & Hooge, 2003), or that the peak saccade velocity cannot be higher than a certain threshold (Nyström & Holmqvist, 2010). What is not identified as a saccade is typically assumed to be a fixation. Surprisingly, very few, if any, algorithms have used the fact

that saccades follow
(i.e. eye movements

Velocity-based alg
are also available in
found in the commerc
lustrate strengths and
is a more elaborate
(1981) and Salvucci
by Inchingolo and Sp
from 1965 and 1967

The particular imp
ples above is based o
paper spells out the
be specific about their
SMI employee a little
used in research paper
their BeGaze software

The SMI velocity
data stream twice. Th
time to find saccadic

```
SMI velocity algor
saccade peak locat

  1. For all sampl
      (a) Calculat
      (b) Detect
      (c) Calculat

  2. For all veloc
      (a) Collect
          is so sl
          threshol
      (b) Collect
          velocity
      (c) Detect
          velocity
          part of
      (d) Detect
          between
      (e) Fixatio
```

In other words,
accepts these as sac
walks down the slop
old tells the algorit
as a saccade, but ra
the saccade on- and
than the average vel
samples. For the tw
and the offset veloc
noise of 6–10°/s. A

that saccades follow what is known as the *main sequence* to exclude 'non-typical' saccades (i.e. eye movements have a typical path from which deviating saccades are obvious).

Velocity-based algorithms have been implemented by many researchers themselves, but are also available in some commercial analysis software. One such implementation can be found in the commercial software BeGaze 2.1 by SMI. It has been used in this book to illustrate strengths and weaknesses of current centre algorithms. The SMI velocity algorithm is a more elaborate version of the I-VT algorithm of Bahill, Brockenbrough, and Troost (1981) and Salvucci and Goldberg (2000), and also very similar to the algorithms tested by Inchingolo and Spanio (1985), and possibly stemming back to the algorithms by Boyce from 1965 and 1967 as referred to in Ditchburn (1973).

The particular implementation of the SMI velocity algorithm we have used in many examples above is based on the algorithm described and used by Smeets and Hooge (2003). This paper spells out the algorithm in detail, because the authors thought that researchers should be specific about their algorithms. In fact, few other papers provide such detail, so when one SMI employee a little later made a literature search for saccade detection algorithms being used in research papers, that particular version came to be the algorithm SMI implemented in their BeGaze software.

The SMI velocity algorithm is a two-pass algorithm, i.e. it looks through the complete data stream twice. The first time to calculate velocities and *detect saccades*, and the second time to find saccadic onsets and offsets. The following is the pseudo-code.

```
SMI velocity algorithm. Input: (raw data samples, velocity threshold,
saccade peak location threshold)
  1. For all samples:
       (a) Calculate angular velocities
       (b) Detect peaks in eye velocity
       (c) Calculate fixation velocity threshold
  2. For all velocity peaks:
       (a) Collect all data samples to the left of the peak, but only until the velocity
           is so slow that the sample must be part of a fixation (the fixation velocity
           threshold).
       (b) Collect all data samples to the right of the peak, but only until the
           velocity is so slow that the sample must be part of a fixation.
       (c) Detect a saccade from the collected data samples only if the
           velocity peak of the saccade is located within the central
           part of the saccade. Otherwise it is discarded.
       (d) Detect blinks as periods where only zero data ($(x,y) = (0,0)$) are found
           between two saccades.
       (e) Fixations are everything that are not saccades or blinks.
```

In other words, this algorithm finds the velocity peaks that rise above the threshold, and accepts these as saccade candidates. Originating at the saccadic peak, the algorithm then walks down the slopes on both sides (see Figure 5.24). The calculated fixation velocity threshold tells the algorithm when the speed is so low (the eye is so still) as to stop counting this as a saccade, but rather as a fixation. The SMI implementation of this algorithm takes it that the saccade on- and offsets are when the saccade velocity is three standard deviations higher than the average velocity of fixations, as calculated from the beginning of the stream of data samples. For the two saccades in Figure 5.24, the onset velocities are 12.56°/s and 10.95°/s, and the offset velocities 13.98°/s and 15.83°/s, somewhat higher than the average fixation noise of 6–10°/s. As a very simple check that the saccade velocity profile seems valid, only

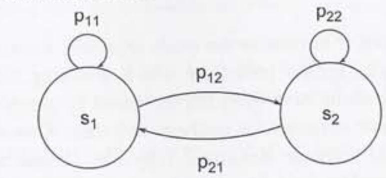


Fig. 5.25 A finite state machine with two states: one representing fixations and one saccades.

saccade candidates that have velocity peaks in the central portion (default are 20%–80% of saccade length) of the saccade are finally accepted. Finally, fixations are detected implicitly as everything that is not saccades, blinks, or undefined events.

Velocity algorithms are better to use with high sampling frequencies, since saccades are short (in the area of 20–50 ms) and you need many samples in each slope (10–25 ms) to calculate velocities correctly in the critical slope ends. A 200 Hz recording gives 2–5 samples per slope (depending on the size of the saccade), which should be considered an absolute minimal sampling frequency for accurate saccade duration calculations (p. 29). If the algorithm is used just for *detecting* the saccade, rather than *measuring* it, then a sampling frequency as low as 50 Hz can be sufficient. For saccadic detection, it is enough to have one velocity sample above the peak threshold, which means two data samples at a large enough spatial distance between them, which you find in 50 Hz data.

The 'Tobii fixation filter' developed by Olsson (2007) is used with low speed data using a 'double-window technique' (similar to the one proposed by Marple-Horvat *et al.*, 1996). First, the algorithm uses two sliding windows on opposite sides of the current velocity sample and finds the average velocity within each such window. These averages are subtracted, and only if the difference exceeds a threshold, a saccade is detected. To prevent two fixations from being identified too closely in space and time, another set of thresholds is used to control for fixation proximity.

An increasingly popular event detection algorithm was presented by Engbert and Kliegl (2003). It was originally developed to detect microsaccades, but works equally well for saccade detection. To detect saccades, the algorithm searches for velocity samples exceeding a threshold, which is calculated based on a median estimation of the eye velocity during a trial. Thus, it adapts the threshold over different trials and participants. Horizontal (v_x) and vertical (v_y) velocity components are treated separately by the algorithm.

Hidden Markov Models (HMM) use a probabilistic model to classify data samples into saccades and fixations based on velocity information. The I-HMM model described by Salvucci and Goldberg (2000) uses two states, S_1 and S_2, each representing the velocity distribution of either fixation samples or saccade samples (see Figure 5.25). Each state is associated with transition probabilities, which estimate the likelihood of the next sample belonging to a fixation or saccade, given the status of the current sample. Typically, consecutive samples have a high probability of belonging to the same type of eye movement, giving small inter-state transition probabilities, $\{p_{12}, p_{21}\}$. The two-state I-HMM model reported by Salvucci and Goldberg (2000) needs eight parameters, which can be estimated from similar sample data. Besides two transition parameters for each state, the model needs to know the observation probabilities in the form of velocity distributions (means and standard deviations). Given the model parameters and a sequence of gaze positions to be classified, dynamic programming such as the Viterbi algorithm can be use to map gaze positions to states (fixation or saccades) in a way that maximizes the probability of a correct assignment according to the model. Finally, neighbouring samples are collapsed into fixation and saccades based on e.g., majority decisions.

Extending the r
to analyse data col
sented data from sm
The I-HMM is,
possible to adapt by
methods, it is more
Algorithms based o

Velocity in comb

Information about
smooth pursuit. Ve
slower than the fa
is therefore used a
acceleration reache
A widely used
oped by SR Resea
systems, but it is
Behrens & Weiss,
heuristic of the di
sumptions built in
described algorith
and movement of
gives the EyeLink
on the recording c
Second, and mayb
can be manipulate
an online algorith
thus needs to mak
sample. Moreover
the chance to corr
In the EyeLink
neous velocity an
robust, detection
their respective t
Link parser detec
or cannot be foun
saccades or blink

5.6 Manu

Manual coding m
is still enough to
instructions and
(1988). While it
option, as in the
recorded from d
advantage of bei
also the weaknes

Extending the two-state approach, Rothkopf and Pelz (2004) proposed a four-state HMM to analyse data collected using head-mounted eye-trackers. The two additional states represented data from smooth pursuits and vestibular ocular reflexes (VOR).

The I-HMM is, according to Salvucci and Goldberg (2000), accurate and robust, and it is possible to adapt by re-estimating the parameters. In comparison with other dispersion-based methods, it is more complex, however, and requires more parameters to be set (or estimated). Algorithms based on HMMs are uncommon and cannot be found in any commercial software.

Velocity in combination with acceleration

Information about eye acceleration is particularity useful when distinguishing saccades from smooth pursuit. Velocity by itself is not sufficient since the slowest saccade velocity can be slower than the fastest smooth pursuit movement (Behrens & Weiss, 1992). Acceleration is therefore used in high speed data for detecting saccade on- and offsets, where the eye acceleration reaches its maximum value.

A widely used velocity- and acceleration-based saccade detection algorithm was developed by SR Research and thus primarily applied to eye-tracking data acquired with EyeLink-systems, but it is not the only algorithm using acceleration data (see Tole & Young, 1981; Behrens & Weiss, 1992). The algorithm developed by SR Research is perhaps the most heuristic of the different algorithms, with several settings and a number of pragmatic assumptions built in. This may stem from its fundamental difference compared to previously described algorithms: it is primarily designed to detect saccades *online* using the position and movement of the eyes, which is (and must be) reflected in its design. This difference gives the EyeLink parser some advantages. First, only the detected events have to be stored on the recording computer, not raw data samples. This saves a substantial amount of space. Second, and maybe most importantly, it allows for gaze-contingent research where stimuli can be manipulated online in synchrony with the eye-movement events. On the other hand, an online algorithm does not have access to all data samples in a trial before detection, and thus needs to make its decisions based only on data recorded before the currently available sample. Moreover, it has only one chance to get this right, whereas a multipass algorithm has the chance to correct initial mistakes.

In the EyeLink algorithm, saccade on- and offsets are detected by comparing the instantaneous velocity and acceleration against user defined thresholds. To make the algorithm more robust, detection is triggered when either velocity or acceleration become higher/lower than their respective thresholds for a predefined number of samples. Besides saccades, the EyeLink parser detects blinks and fixations; blinks are identified when the pupil size is very small, or cannot be found at all in the camera image of the eye; fixations are everything that is not saccades or blinks.

5.6 Manual coding of events

Manual coding means that a person subjectively decides, for example, when the gaze position is still enough to be a fixation or moves fast enough to be a saccade. An example of coding instructions and procedures is given in Harris, Hainline, Abramov, Lemerise, and Camenzuli (1988). While it is very time consuming, manual coding may sometimes be the only real option, as in the case where data is available only as gaze-overlaid videos or when data are recorded from dynamic stimuli. Compared to algorithmic detection, manual coding has the advantage of being able to utilize the powerful pattern matching ability that humans have, but also the weakness of human subjectivity and inconsistency.

To identify fixation durations from video, you need to look through it frame by frame,[19] and notice when the gaze overlay cursor is still and when it moves. Stillness is typically defined as the gaze cursor remaining on the same position (usually an object) in the scene, even if the head is moving, rather than stillness in the head.

Fixation duration coding from gaze-overlaid videos is not very common, and when it is done, the method is rarely published. For instance, Tatler, Gilchrist, and Land (2005) recorded the eye video image onto the gaze-overlaid video, and thus could see the movement in the *eye video* while watching the gaze cursor move when coding for fixations. According to Benjamin Tatler,[20] the movement in the eye video is a much more reliable indicator of fixation and saccade events than the gaze cursor, especially as the scene video looks a bit blurred during saccades and head movements.

Munn *et al*. (2008) tested three coders, with on average 300 hours of coding experience. They were given the instruction to code the gaze-overlaid 30 Hz video for fixation start- and end-frames. They used a 200 ms fixation duration criterion. Coding a 100 second period of data took 80 minutes on average. One data period was from a participant walking around in an office environment, and the other from an animated film. The authors compared the coded fixations to fixations produced by a velocity-based type of algorithm, also taking 30 Hz data, with the addition of an extra fixation duration criterion of 200 ms. Results show that the human coders agreed with each other slightly more often than with the algorithm, but this could be due to the low sampling frequency of the eye-tracker. For head-mounted systems with a higher sampling frequency, we should expect a better performance on the part of the algorithm. In conclusion, Munn *et al*. (2008) found that the algorithm they tested was quite robust in finding fixations (or rather non-saccadic portions of data), and that the algorithm could be used as a preliminary parser that reduces the effort of manual coders to find potential fixations.

Methods for dwell time coding are described on pages 227–229. In dwell time coding, the coded values are the total gaze durations on objects in the scene, irrespective of whether that gaze is composed by fixations, saccades, or smooth pursuit.

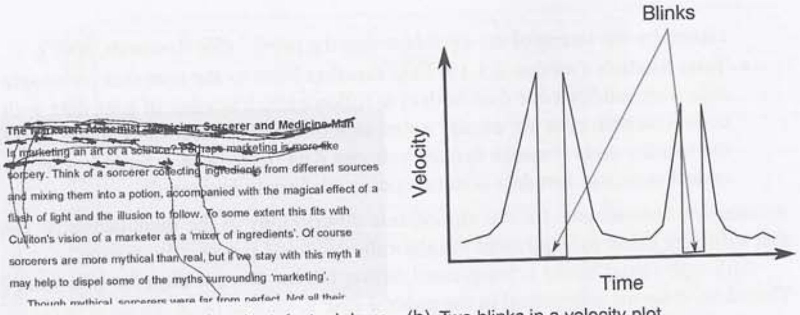
5.7 Blink detection

Blink detection is an important component of an event detection algorithm, both since blinks are related to cognitive functions (p. 410 and Fogarty & Stern, 1989), and because they need to be separated from other types of events such as fixations and saccades. Moreover, blinks are often considered to be artefacts in both eye-tracking and EEG data, and inaccurate detection causes these undesired data to be included in the subsequent analysis. For example, the EyeLink manual advises the user that "it is also useful to eliminate any short (less than 120 millisecond duration) fixations that precede or follow a blink. These may be artificial or be corrupted by the blink" (SR Research, 2007). Together these reasons motivate why most eye-trackers report information about blinks, even though they are not movements of the eye, but rather movements of the eyelid.

There are a number of methods to detect and measure blinks, for example by analysing the eye video directly (Grauman, Betke, Gips, & Bradski, 2001; Moriyama *et al*., 2002), by identifying large vertical EOG potentials (Abel, Troost, & Dell'Osso, 1983), by monitoring markers attached to the upper and lower eyelids (Collewijn, Van Der Steen, & Steinman,

[19] If you record eye movement data at 50 Hz on a video with 25 interlaced frames per second, you can go through the video field by field to see *all* data samples during coding. Not all video viewers allow this, however.

[20] Personal communication, October 21, 2009.



The Marketer: Alchemist, Magician, Sorcerer and Medicine Man
Is marketing an art or a science? Perhaps marketing is more like
sorcery. Think of a sorcerer collecting ingredients from different sources
and mixing them into a potion, accompanied with the magical effect of a
flash of light and the illusion to follow. To some extent this fits with
Culliton's vision of a marketer as a 'mixer of ingredients'. Of course
sorcerers are more mythical than real, but if we stay with this myth it
may help to dispel some of the myths surrounding 'marketing'.
Though mythical sorcerers were far from perfect. Not all their

(a) Two blinks seen as pairs of artefactual downward saccades with intermediate loss of data.

(b) Two blinks in a velocity plot.

Fig. 5.26 Typical appearance of blinks when they appear in data collected with tower-mounted high--speed eye-trackers.

1985), or simply by manual counting of blinks (Taylor *et al.*, 1999; Epelboim & Suppes, 2001). The focus of this section, however, is blink detection with data from modern video-based eye-trackers that output at least time stamped (x, y) coordinates and pupil size.

Blinks are detected as the eyelid descends downward to eventually cover the entire eyeball. On its way down, it covers increasingly more of the pupil, making the calculated pupil centre move downward, and with it the data samples, as though there was a rapid downward saccade. When the eyeball is completely covered, both the pupil and the corneal reflection cannot be tracked any more. Consequently, the eye-tracker reports neither data samples nor pupil size, and instead typically outputs zeros (0). When the eyelid re-opens a corresponding upward-moving saccade-like movement is produced, and actual eye tracking can continue. Figure 5.26 illustrates the typical appearance of (a) data samples and (b) velocity during two blinks recorded with a high-speed tower-mounted eye-tracker.

There are surprisingly few published articles dedicated to blink detection, and the ones that exist can mainly be found in the data analysis sections of the paper. Indeed, most blink detection algorithms consider pupil size and gaze coordinates, often also with a required minimum duration. Bonifacci, Ricciardelli, Lugli, and Pellicano (2008), for example, define blinks as sudden losses of the position signal for more than 96 ms, and Van Orden, Jung, and Makeig (2000) use a pupil diameter threshold in combination with an 83.3 ms minimum duration requirement. Using both pupil and gaze coordinate information, Geng, Ruff, and Driver (2009) required a loss of pupil data for at least 50 ms in combination with eye movement of at least 13° of visual angle. Similarly, Brouwer, Van Ee, and Schwarzbach (2005) defined blinks simply as "those samples during which no eye position was recorded". Karatekin, Marcus, and White (2007) used one of three criteria, defined as followed: "(a) the pupil diameter falling below 1.86 mm or above 5.96 mm, (b) the horizontal or vertical positions of the eye falling outside the limits of the screen, or (c) the diameter of the pupil changing by more than 0.74 mm over 16.7 ms."

Several researchers do not use their own software, but rely on blink detection algorithms implemented by manufacturers. Examples are:

- SMI's analysis software package BeGaze 2.3 detects blinks when "zero data is embedded in 2 saccade events" (SMI, 2007), i.e. exactly what is depicted in Figure 5.26. Note that this makes the algorithms depend critically on how saccades are defined.
- The EyeLink parser describes blink detection as follows in the manual: "Blinks are defined by a period of missing pupil surrounded by a period of artefactual saccade

caused by the sweep of the eyelids across the pupil." (SR Research, 2007).

- Tobii Studio's (version 2.1.12) help function hints to the user that "A couple of gaze data with validity code 4 on both eyes followed by a number of gaze data with validity code 0 on both eyes are usually a sure sign of a blink". According to the help function, the validity code 0 means that all 'relevant data' have been recorded, whereas validity code 4 indicates that data is missing or is "definitely incorrect".

A common denominator for the algorithmic descriptions in the manufacturers' manuals is that sufficient detail to implement the algorithm yourself is typically missing.

Although visual intake is suppressed during blinks, it is not clear for how long (p. 324). Therefore, it seems suboptimal in the general case to choose blink duration thresholds based on what is known about blink suppression.

Data from eyelid trackers show that closing the eyelid is generally faster than opening it (p. 325). This allows us to define at least four duration measures: closing time, closed time, reopening time, and blink duration. The baseline is the position of the eyelid before closing. As we have seen in the eye camera set-up section, eyelids can be more or less droopy in the baseline state, and a droopy eye will have a smaller amplitude during blinks. There are several alternative measures of blinks. For instance, blink closure duration was defined by Morris (1984) as the time from when the lid is half closed and going down, until it is at the same position but heading upward. Lobb and Stern (1986) introduced a whole range of measures, such as time between blink initiation and incursion of the lid on the pupil, duration of lid over the pupil, duration between lid over pupil and full closure, and how long the lid remained closed.

Note that an eye-tracker typically does not measure the movement of the eyelid, and therefore cannot provide detailed information about individual blinks. However, it may give a good idea of how much of the pupil is covered. For instance, closing time could refer to the time it takes the eyelid to traverse the pupil on its way down. Sometimes there is no need to know the precise dynamics of a blink, as long as it can be detected and *blink rate* can be calculated.

5.8 Smooth pursuit detection

Smooth pursuit is the slow motion of the eye as it follows something moving. There is a common belief that smooth pursuit can *only* occur when there is a target to follow, but some studies contradict this. It appears that participants can follow the mental percept of an imagined or offset stimulus for at least a few seconds (Whittaker & Eaholtz, 1982; De'Sperati & Santandrea, 2005). So, for instance, if the participant controls a moving object, the eye can pursue it smoothly even in total darkness (Gauthier & Hofferer, 1976).

Smooth pursuit can be studied in its own right, indirectly revealing properties of the neural systems that underly it. Alternatively, the effect on smooth pursuit of drugs, alcohol, and a variety of disorders can be a reason to use smooth pursuit measures. For instance, Trillenberg, Lencer, and Heide (2004) review a large number of eye-tracking studies with a focus on smooth pursuit, and conclude that "eye movements provide an important tool to measure pharmacological effects in patients and unravel genetic traits in psychiatric disease". Similarly, the large review by O'Driscoll and Callahan (2008) indicates that smooth pursuit impairment is robust in schizophrenia.

Figure 5.27 shows the *x*-coordinate, velocity and acceleration of a person's gaze when trying to follow a pendulum movement—a task that requires smooth pursuit movements. As the figure illustrates, smooth pursuit is characterized by periods with more or less constant velocities accompanied by what is known as 'catch-up saccades'. These are employed when

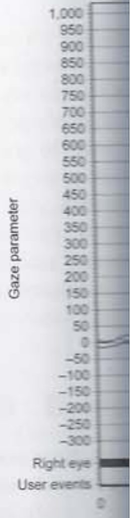


Fig. 5.27 Gaze to
(black) and 'saccad
bottom of the grap

the gaze position
velocity to catch
can interrupt sm
reposition the e
saccades (antici
during smooth p
at least from th
smooth pursuit

In the medic
lation to a movi
participant can
target velocity
the smooth purs
ever, it may be
saccadic gaps to

Applied rese
without head tr
cades. Using th
ation, saccadic
directly mislead
pursuit detectio

As previou
a general 'intak
as the Tobii Fix
as fixations on
images.

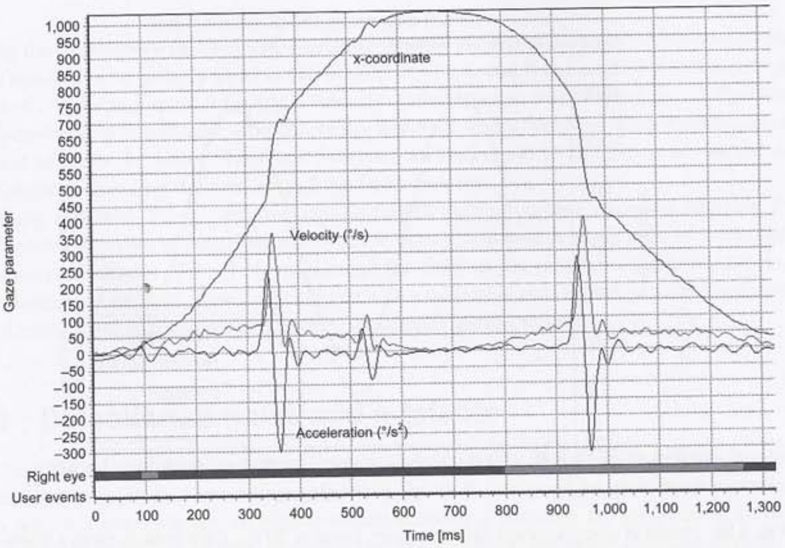


Fig. 5.27 Gaze following a pendulum movement. Recorded with the SMI HiSpeed 1250 Hz. 'Fixations' (black) and 'saccades' (grey) as detected with the SMI velocity algorithm in BeGaze 2.1 are shown at the bottom of the graph.

the gaze position lags behind the target it follows, and therefore temporarily needs to increase velocity to catch up with the target. Besides catch-up saccades, another two types of saccades can interrupt smooth pursuit: *back-up saccades* (in the direction opposite target motion which reposition the eyes on the target when eye position was ahead of the target) and *leading saccades* (anticipatory saccades of any amplitude over 1°). Square-wave jerks can also occur during smooth pursuit (Black, 1984). Fixations interrupt smooth pursuit when it is very slow, at least from the viewpoint of the detection algorithm, which ignores whether or not the smooth pursuit brainware continues to move the eyes.

In the medical research community, data on smooth pursuit is typically calculated in relation to a moving object (dot) that the participant looks at. To estimate how accurately the participant can follow a target dot as it moves, the ratio (*gain*) or distance (*phase*) between target velocity and eye velocity over time can be calculated. There is then no need to *detect* the smooth pursuit first; calculations can be made directly from the raw data stream. However, it may be necessary to remove saccades (including catch-up saccades) and interpolate saccadic gaps to obtain accurate values (Ebisawa, Minamitani, Mori, & Takase, 1988).

Applied researchers use more complex video stimuli, as well as head-mounted eye-trackers without head tracking, and get data where smooth pursuit is mixed with fixations and saccades. Using the current dispersion or saccade velocity algorithms to calculate fixation duration, saccadic amplitude or the percentage of smooth pursuit is of little use here, but can be directly misleading, as we saw earlier in this chapter. For such applications, a general smooth pursuit detection algorithm is needed.

As previously described, some algorithms co-classify fixations and smooth pursuits into a general 'intake' category, like the fixation velocity algorithms used by Itti (2005) as well as the Tobii Fixation Algorithm and the EyeLink parser, which report these combined events as fixations only. Such 'fixations' may not be comparable with fixations recorded from still images.

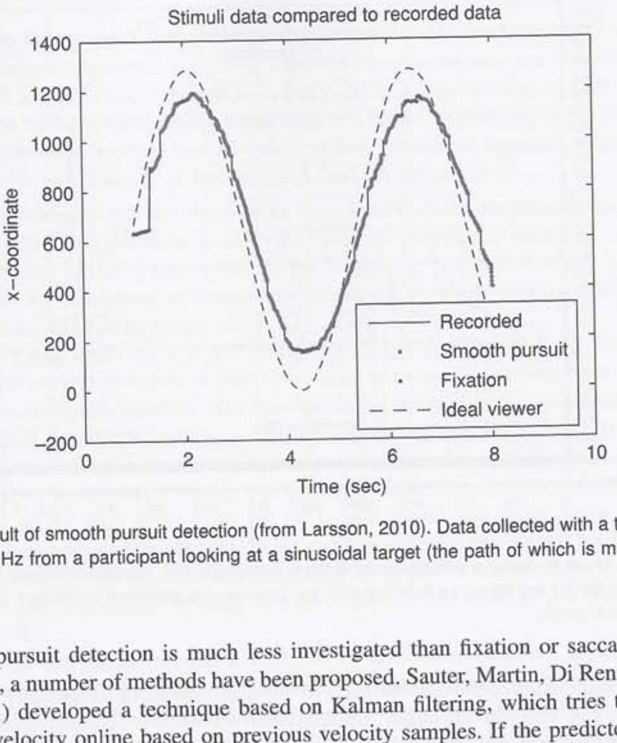


Fig. 5.28 Result of smooth pursuit detection (from Larsson, 2010). Data collected with a tower-mounted system at 500 Hz from a participant looking at a sinusoidal target (the path of which is marked as 'ideal viewer').

Smooth pursuit detection is much less investigated than fixation or saccade detection. Nevertheless, a number of methods have been proposed. Sauter, Martin, Di Renzo, and Vomscheid (1991) developed a technique based on Kalman filtering, which tries to model and predict eye velocity online based on previous velocity samples. If the predicted velocity is significantly different from the observed velocity according to a χ^2-test, a saccade is detected. Otherwise, it is a smooth pursuit movement or a fixation. Komogortsev and Khan (2007) extended this approach by detecting a fixation when the predicted eye velocity was $< 0.5^\circ/\text{s}$ for at least 100 ms. Smooth pursuit was defined as everything that was not a saccade or a fixation, and where the predicted eye velocity did not exceed $140^\circ/\text{s}$.

A different approach was used by Agustin (2010), who in an initial step separated saccades from fixation and smooth pursuit with simple velocity thresholding, and then separated fixations from smooth pursuit by looking at the direction of movement over a limited time window; smooth pursuit was detected only when the standard deviation of the movement directions was below a certain threshold. Since fast smooth pursuit velocity can exceed the velocity of slow saccades, Larsson (2010) used the algorithm by Engbert and Kliegl (2003) adapted to the acceleration domain to separate the quickly accelerating saccades from fixations and smooth pursuit. Smooth pursuit was separated from fixations by statistically testing the uniformity in distribution of sample-to-sample vectors around the unit circle. Only portions of the data where the hypothesis of uniformity was rejected by a Rayleigh test were accepted as smooth pursuit. Finally, a dispersion-based threshold was used to find fixations. Remaining data comprised undefined events. Figure 5.28 shows the result of applying the algorithm by Larsson (2010) to data collected from a participant looking at a sinusoidal target.

Analysing smooth pursuit in monkeys, Ferrera (2000) presented an algorithmic method drawing on eye velocity. Assuming that saccades have already been removed from data, it is implemented as follows. A moving window of 100 ms is applied to velocity samples, for each stop comparing whether the data in the current 100 ms window differs significantly from the data in the previous 100 ms window. Ferrera then uses the calculated significance values to make an ROC (receiver operating characteristic) curve along the eye-velocity curve.

When the ROC curve reaches 0.95, onset of smooth pursuit is assumed to have taken place. This appears to be a fairly precise calculation of onset, and it is based on eye velocity only. Ferrera's major argument for using a velocity- rather than an acceleration-based algorithm is that acceleration is a noisier measure, being a second derivative. Smooth pursuit offset is the inverse of onset. In theory, therefore, Ferrera's method could be used to calculate the offset by simply exchanging the positions of the two windows.

Berg, Boehnke, Marino, Munoz, and Itti (2009) used principal component analysis (PCA) to classify raw eye-movement data within a temporal window. A small ratio of the explained variances (minimum divided by maximum) for each of the two principal components was interpreted as a saccade. Raw data samples with a ratio near zero but with insufficient velocity to be considered as a saccade were taken as smooth pursuit samples.

5.9 Detection of noise and artefacts

There are good arguments to discard imprecise data entirely. But in order to know how much noise there can be in a recording before we delete it, we need to quantify the amount of data which is overly noisy. Detecting which sections of the data are too poor to be used is important, in particular if manufacturers use lower quality eye cameras in the systems they sell.

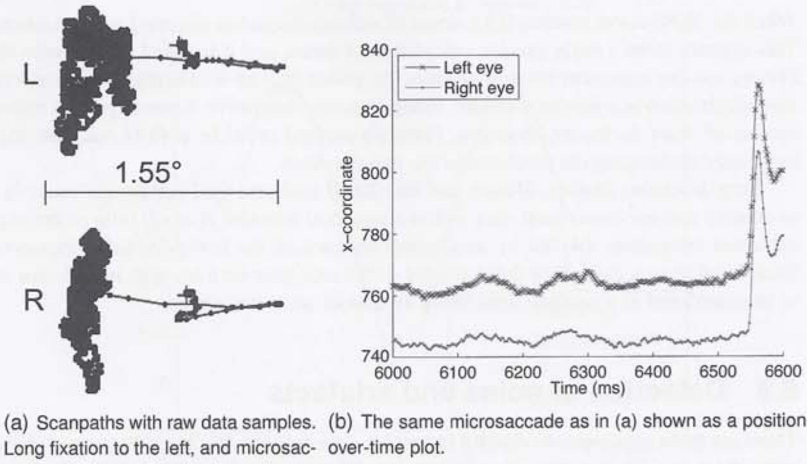
There are many different types of noise detection. First, there is the *system inherent noise* that is related to the precision of your eye-tracker. In high-end systems, this type of noise is low enough to allow the measurements of fixational eye movements, whereas this is not possible for lower end systems. System inherent noise can be hard to separate from real eye movements under normal conditions, and is reduced by filters rather than detected and removed.

Second, there are the high velocity *optic artefacts*—very quick jumps in eye-movement data—that do not originate from actual eye movements, but derive from situations where the eye-tracker temporarily fails to correctly track the pupil and/or the corneal reflection. As a result, movements that violate known eye-movement dynamics can appear in the data. Fortunately, most artefacts with high velocities can be removed by excluding data samples with velocities and accelerations that exceed the theoretical values for how fast human eyes can move; in 1250 Hz data, Nyström and Holmqvist (2010) use upper thresholds of $1000^\circ/\mathrm{s}$ and $100000^\circ/\mathrm{s}^2$ for this purpose.

Data loss, when the eye-tracker cannot report a value of how the eye moves, can also be seen as noise. Typically, lost samples are indicated in the data files, which makes it simple to calculate and report. Depending on your research question, blinks are sometimes included in this category.

Event detection algorithms are typically designed under the assumption that there is relatively little imprecision and few artefacts in the data, and in particular that the level of imprecision is even throughout the recording. In some cases, as in Figure 5.13, precision varies over time. Assuming we have a velocity-based algorithm, there are three options for how to deal with this variation:

1. Let the velocity threshold adapt to the local precision level in the data, as proposed by Tole and Young (1981); Niemenlehto (2009). When precision is high, lower the threshold, when it is low, increase the threshold. This makes the best possible use of data, but since fixations are defined by different criteria due to the varying noise levels, they may not be comparable with each other.



(a) Scanpaths with raw data samples. Long fixation to the left, and microsaccade with immediate backward moving 'overshoot' or 'microglissade'. Left eye above, and right eye below.

(b) The same microsaccade as in (a) shown as a position-over-time plot.

Fig. 5.29 A large microsaccade recorded binocularly at 500 Hz in pupil-only mode, with a video-based tower-mounted eye-tracker and a participant staring at a very small point.

2. Use a fixed, low threshold throughout all data. This will give proper fixations in the high-precision sections of data, but a plethora of unreasonably short fixations and saccades in the low-precision section.
3. Use a fixed, high threshold throughout all data. Noise would be ignored, but also many small saccades.

In summary, noise can be detected as artefacts, data loss, and blinks. The proportion of this type of noise can (and should) be reported in your study. Physiological and system generated noise is more difficult to detect, and is usually reduced by filtering.

5.10 Detection of other events

A number of other patterns occur in eye-tracking data, that could be parsed into events.

Fixational eye movements consist of microsaccades, drifts, and tremor. Of these three types of micro-movements, microsaccades, as illustrated in Figure 5.29, are the most investigated. They can be detected by the algorithm of Engbert and Kliegl (2003). From a ballistic point of view, microsaccades can be regarded as smaller versions of normal saccades, possibly ending with larger overshoots (Møller, Laursen, Tygesen, & Sjølie, 2002).

In short, the algorithm estimates the standard deviation $\sigma_{x,y}$ of the velocity

$$\sigma_{x,y} = \langle v_{x,y}^2 \rangle - \langle v_{x,y} \rangle^2 \tag{5.1}$$

in x- and y-dimensions with a median estimator $\langle \cdot \rangle$ to alleviate the influence of noise. Then, velocity samples above a threshold $\lambda \sigma_{x,y}, \lambda \geq 1$ are potential microsaccade samples. Several researchers use the fact that microsaccades occur in both eyes simultaneously, and therefore only accept those that have a minimum overlap in time between the left and the right eye (Engbert & Mergenthaler, 2006; Otero-Millan, Troncoso, Macknik, Serrano-Pedraza, &

Martinez-Conde, 2008),
size 0.12°–1° with a ve
used 0.05°–1° and > 3
saccades only if their ve

Given the microsac
as periods between the

The term **glissade**
like movement without
glissade was seen as a
fixation. Bahill, Clark,
and are therefore differ

Later, glissades we
be detected as an event
the observed "post-sac
retardation, that make
again. This movement
surement error in the d
which relies on the fo
and corneal reflection
ing that this is the exp
movements very clear
systems from SMI an

For simplicity, Ny
over- and undershoot
terminology used in t
'wobbling' that some
10.28 on page 338),
the entirety of the ne

In current detecti
the EyeLink parser),
Holmqvist (2010) de
threshold within 40
definition, the onset
glissade is terminate
movement peak. Us
24 ms in reading an

The **square-wa**
off the visual target,
tern can be seen in
(0.5–5°) in opposite
cadic latency. Arou
with saccadic overs
inally called 'Gege
first termed square-
physiological and p
can occur at a rate

Research paper
criteria (Abadi & C
for a pair of saccad
cades were detecte

Martinez-Conde, 2008). Hafed and Clark (2002) detected microsaccades as movements of size 0.12°–1° with a velocity $> 8°/s$, whereas Martinez-Conde, Macknik, and Hubel (2000) used 0.05°–1° and $> 3°/s$. Møller *et al.* (2002) accepted fixational movements as microsaccades only if their velocities and accelerations exceeded 5°/s and $2500°/s^2$, respectively.

Given the microsaccades, inter-microsaccadic intervals (IMSIs) can easily be calculated as periods between the microsaccades (Engbert & Mergenthaler, 2006).

The term **glissade** was coined by Weber and Daroff (1972) to describe a slow, drift-like movement without latency that "corrected dysconjugate refixations". In other words, the glissade was seen as a post-saccadic movement intended to realign the eyes before steady fixation. Bahill, Clark, and Stark (1975a) propose that glissadic overshoot last 30–500 ms, and are therefore different from dynamic overshoot, which they argue last only for 10–30 ms.

Later, glissades were hypothesized to be errors in the measurements, not something to be detected as an event in its own right. For instance, Deubel and Bridgeman (1995) explain the observed "post-saccadic movement" as a form of inertia in the eye lens during saccade retardation, that makes it go further than the eyeball tissue, which then pulls then lens back again. This movement of the lens (but not the eye) would cause what is essentially a measurement error in the data, which would be especially grave for the Dual-Purkinje eye-tracker which relies on the fourth Purkinje reflection at the back of the lens. The video-based pupil and corneal reflection systems are much less sensitive to such swinging of the lens, assuming that this is the explanation, but yet the authors of this book see the many large glissadic movements very clearly in our data from all our video-based eye-trackers, including high-end systems from SMI and SR Research.

For simplicity, Nyström and Holmqvist (2010) suggested naming all types of high-velocity over- and undershoots that occur directly after a main saccade glissades, and this is also the terminology used in this book. According to this definition, glissades could be seen as ocular 'wobbling' that sometimes occurs at the end of main saccades (for an example see Figure 10.28 on page 338), and it thus differs from the slow post-saccadic drift that may go on for the entirety of the next fixation.

In current detection algorithms, glissades are mostly assigned either to the fixation (e.g. the EyeLink parser), or to the saccade (for instance Gilchrist & Harvey, 2006). Nyström and Holmqvist (2010) detected glissades separately as movement peaks that exceed a velocity threshold within 40 ms after the offset of a saccade. According to Nyström and Holmqvist's definition, the onset of a glissade equals the offset of the preceding saccade, whereas the glissade is terminated when reaching the first local velocity minimum after the last glissadic movement peak. Using this definition of glissades they found an average glissade duration of 24 ms in reading and scene perception, but glissades between 10–50 ms were common.

The **square-wave jerk** is an involuntary, conjugate saccadic intrusion that takes the eye off the visual target, and then back again (Leigh & Zee, 2006, p. 164). Its typical motion pattern can be seen in Figure 5.30. Square-wave jerks mostly consist of pairs of small saccades (0.5–5°) in opposite directions, separated by a normal or slightly longer (200–400 ms) saccadic latency. Around 10% of square-wave jerks are biphasic, which means that they return with saccadic overshoots, and must be followed by a third, corrective return saccade. Originally called 'Gegenrücke', and sometimes 'saccadic intrusions' (into fixations), they were first termed square-wave jerks by Jung and Komhuber (1964). They can derive from both physiological and pathological sources (Abadi & Gowen, 2004). In normal participants, they can occur at a rate of 0.3 Hz or more (Leigh & Zee, 2006, p. 164).

Research papers used a variety of detection techniques, including amplitude and velocity criteria (Abadi & Gowen, 2004). Fahey *et al.* (2008) used a detection algorithm that searched for a pair of saccades in opposite directions, separated by a 60–900 ms period of stillness. Saccades were detected using a velocity threshold of 10°/s. Moreover, Fahey *et al.* classified the

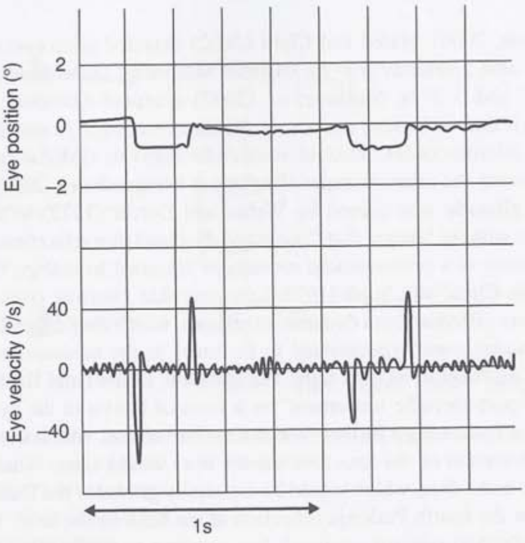


Fig. 5.30 Position and velocity pattern of two square-wave jerks. Recorded from an 11-year old boy fixating a target, using a video-based 120 Hz eye-tracker with magnetic head-tracking. Reprinted from *Pediatric Neurology, 38*(1),Michael S. Salman, James A. Sharpe, Linda Lillakas, and Martin J. Steinbach, Square Wave Jerks in Children and Adolescents, Copyright (2008), with permission from Elsevier.

square-wave jerks into micro (< 0.5°), and macro (0.5–3°) events. Using a similar strategy, Feldon and Langston (1977) detected square-wave jerks as pairs of opposing microsaccades separated in time with at most 750 ms.

If the intrusive movement lacks the intersaccadic interval present in square-wave jerks, and saccades appear 'back-to-back', it is most likely due to *ocular flutter* (horizontal movements shown in Figure 5.31) or *opsoclonus* (movements in all directions) (Leigh & Zee, 2006, p. 165). Unlike square-wave jerks, these types of movements seem to appear mainly in participants with saccadic abnormalities.

Undefined events Some portions of data may simply not be possible to assign to any known category, either because the algorithm is not designed to identify this particular type of event, or because the portion of data does not fit according to any known prototypical pattern. Typically, this is what is left after every known event has been detected, and is sometimes categorized together with noise and artefacts.

Notice that the physical appearance of glissades, small saccades, microsaccades, and the beginning/end of a square-wave jerk can be very similar, and therefore the same algorithm could successfully detect all three types of eye movements. In the end it is up to the researcher to decide which event has been detected, and its functional role. Sometimes terms are used interchangeably; Abadi and Gowen (2004), for example, make no terminological difference between dynamic overshoots (i.e. glissades) and microsaccades, and Hafed and Clark (2002) equate square-wave jerks with a "pair of back-to-back opposing microsaccades".

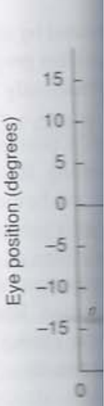


Fig. 5.31 Square-wa
Michael C. Fahey, Phi
Halmagyi, Louise A. C
tin B. Delatycki, Vestib
Copyright (2008) with

5.11 Summ

From this chapter,
rithm should deliv
analysis:

- **Velocity-** an
a filtering th
- **Fixation** ev
ation. Inside
  - **micro**
  - **drifts**
  - **inter-**
- **Saccade** ev
plitude, star
- **Smooth pu**
pursuit even
  - **catch-**
  - **back-**
  - **leadi**
- **Blink** event
- **Glissade** e
- **Square-wa**

None of the
They all have se
but which have p
treat the raw sam
amplitude will be

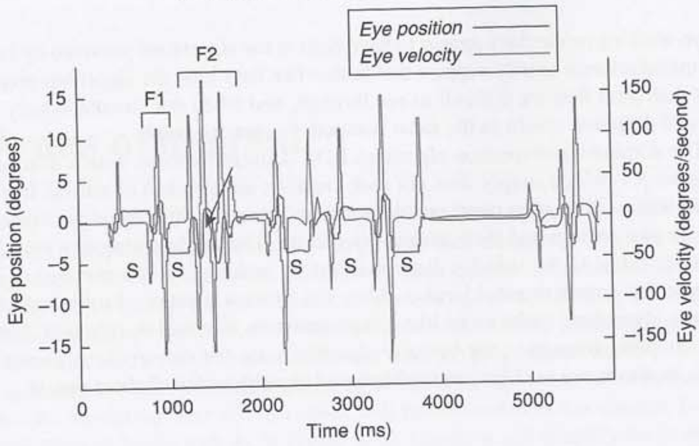


Fig. 5.31 Square-wave jerks (S) mixed with ocular flutter (F1, F2). Reproduced from *Brain, 131*(4), Michael C. Fahey, Phillip D. Cremer, Swee T. Aw, Lynette Millist, Michael J. Todd, Owen B. White, Michael Halmagyi, Louise A. Corben, Veronica Collins, Andrew J. Churchyard, Kim Tan, Lionel Kowal, and Martin B. Delatycki, Vestibular, saccadic and fixation abnormalities in genetically confirmed Friedreich ataxia, Copyright (2008) with permission from Oxford University Press.

5.11 Summary: oculomotor events in eye-movement data

From this chapter, we can conclude that a comprehensive oculomotor event detection algorithm should deliver the following data, correctly calculated, to be used directly or in further analysis:

- **Velocity-** and **acceleration** values calculated from the raw data coordinate samples, via a filtering that does not undermine the detection of any of the events.
- **Fixation** events, each having values for at least position, dispersion, onset, and duration. Inside fixations, we find the fixational eye-movement events, divided into:
  - **microsaccades**, with at least amplitude, duration, and velocity values
  - **drifts**
  - **inter-microsaccadic intervals**
- **Saccade** events, each having values for at least starting position, landing position, amplitude, starting time, duration, peak velocity, and peak acceleration.
- **Smooth pursuit** events, with values for ... well, there is no consensus on how smooth pursuit events should be represented. Within smooth pursuit, we find the event triple:
  - **catch-up saccade**
  - **back-up saccade**
  - **leading saccade**
- **Blink** events, with values for at least *starting time* and *duration.*
- **Glissade** events, with at least the same values as a saccadic event.
- **Square-wave jerk** with at least *amplitude* and *starting position* and *duration.*

None of the algorithms described in this chapter detect and measure all events reliably. They all have settings that are not easy to grasp for the beginning eye-tracking researcher, but which have profound effects on the results produced. Most worryingly, the algorithms treat the raw samples so differently that basic measures such as fixation duration and saccade amplitude will be difficult to compare between algorithms using the same data. The majority of eye-tracking researchers appear to have faith in the algorithms provided by manufacturers, and manufacturers mostly support this faith when they hide the algorithm properties in settings dialogues that are difficult to see through, and when not simultaneously showing raw data and detection results in the same scanpath or velocity graph.

The dominating dispersion algorithm I-DT delivers fixations with a distribution (shown in Figure 5.9) which simply does not look credible, independent of setting. It is imprecise in its calculation of fixation onset and duration, and presents little more, yet bothering the user with the two settings and their unclear interaction. The single setting of a velocity algorithm is easy to relate to the velocity diagrams: just by looking at a few portions of data you can estimate the proper threshold value. Also, the fixation duration distribution of the various velocity algorithms looks more like a representation of a real distribution. For data with a high sampling frequency, the velocity algorithms are the only realistic choice. In the near future, however, we are likely to see improved algorithms for all these events.

6 Area

In the previous
ations, saccade
of interests (A
AOIs define re
did the particip
in the area loo
tected: *dwells*,
segmenting st
into represent
Using AO

- First, w
be done
- In Secti
- We the
transiti
- Five m
using A
matric
several
- Section
namic,
- It is co
of cha
Section
a mini
- Finally
and re

Since A
ware develo
no standard
seven differ
duration' w
are known
and 'Zones'
research. T
pers, and a