# Chapter 14
# Advanced Eye Movement Analysis

Although the information in Chap. 13 is technically sound, accessibility to advanced algorithms and programming languages has increased, making it possible to better illustrate more flexible approaches to fixation and/or saccade (or in general "event") detection. Beyond fixations, new methods have emerged producing greater insight into observed visual behavior than fixations alone could provide.

As suggested in Chaps. 4 and 13, eye movement signals can be approximated by linear filters, both for the proposes of eye movement analysis (this chapter) and synthesis (Chap. 16).

## 14.1 Signal Denoising

Given a raw data stream output by an eye tracker consisting of gaze points $(x_i, y_i, t_i)$, the signal, typically noisy, can be smoothed by an Infinite Impulse Response (IIR) filter such as the Butterworth filter. Treating $x_i$ and $y_i$ independently, smoothing or differentiating (to order $s$) is achieved by convolving $2p + 1$ inputs with filter $h_i^{t,s}$ and $2q + 1$ (previous) outputs $\dot{x}_i$ or $\dot{y}_i$ with filter $g_i^{t,s}$ at midpoint $i$ (Hollos and Hollos 2014):

$$\dot{x}_n^s(t) = 1/(\Delta t^s)\left(\sum_{i=-p}^{p} h_i^{t,s} x_{n-i} - \sum_{i=-q}^{q} g_i^{t,s} \dot{x}_{n-i}\right) \tag{14.1}$$

and similarly for $y_i$ and $\dot{y}_i$, where $n$ and $s$ denote the polynomial fit to the data and its derivative order, respectively (Gorry 1990; Ouzts and Duchowski 2012). In prior work, based on evaluation of calibration data, a 4th order Butterworth filter was found to adequately smooth raw gaze data with sampling and cutoff frequencies of 60 and 6.15 Hz, respectively (Duchowski et al. 2011), but this will vary depending on sampling rate, and possibly other factors, hence fine-tuning of the filter is required.

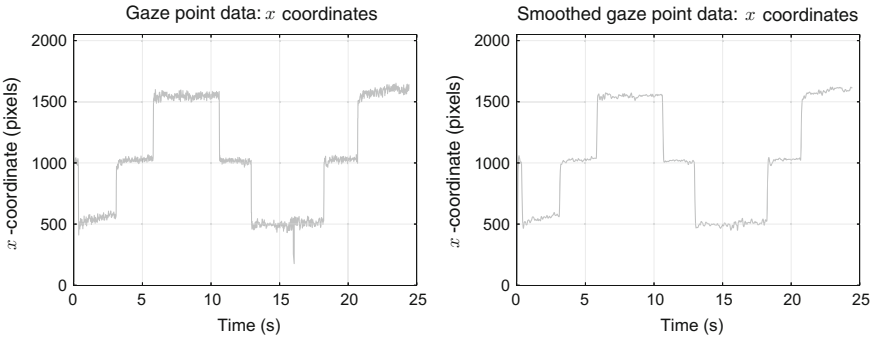


A.T. Duchowski, *Eye Tracking Methodology*, DOI 10.1007/978-3-319-57883-5_14

An example of the effect of the Butterworth filter on the $x$-coordinate of eye movement data is shown in Fig. 14.1, with the 2D eye movement data depicted in Fig. 14.2.

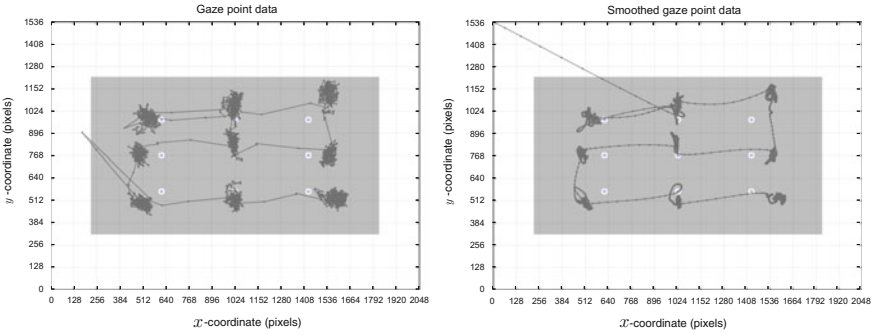
The filter, with proper tuning, is capable of removing very high frequency oscillations in the data. However, when over-tuned, it will tend to smooth (average) the signal which could lead to data loss.

Figure 14.3 shows the effect of over-smoothing the data: clearly what should be two fixation points collapse into one. The reason for this is evident in Figs. 14.1 and 14.2 which show that the saccade between the two gaze points at bottom left



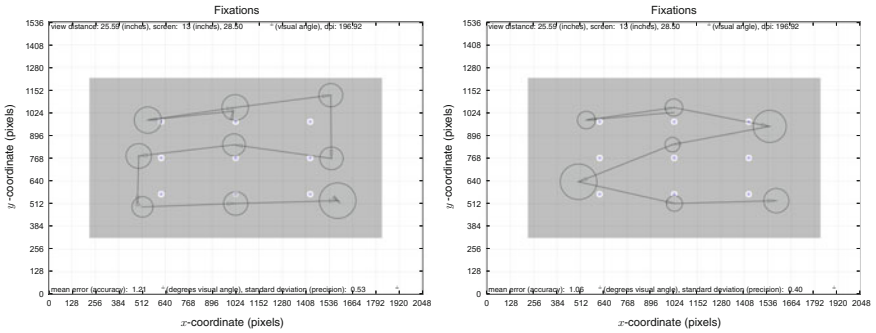

(a) Raw gaze data ($x$-coordinate). (b) Smoothed raw gaze data ($x$-coordinate).

**Fig. 14.1** Example 1D raw gaze $x$-coordinate data captured over a calibration validation grid: **b** Shows the effect of applying the Butterworth filter to the unfiltered data shown in (**a**) Notice in particular the removal of a downward spike at about the 16 s mark




(a) Raw gaze data. (b) Smoothed raw gaze data.

**Fig. 14.2** Example 2D raw gaze data captured over a calibration validation grid: **b** Shows the effect of applying the Butterworth filter to the unfiltered data shown in (**a**). Notice in particular the smoother path curves and reduced data. This effect is especially pleasing in real-time applications as the real-time gaze point appears to move more smoothly to the user without jitter that has been smoothed out by the filter. The long streak of points emanating from upper-left is the result of the Butterworth filter's initially empty history buffer, i.e., the filter needs to build up a history of data points to function properly




(a) Fixations detected without smoothing. (b) Fixations detected following smoothing.

**Fig. 14.3** Example fixation data captured over a calibration validation grid: **b** shows the effect of applying the Butterworth filter prior to velocity-based fixation detection. **a** Shows the effect of not applying the Butterworth filter prior to velocity-based fixation detection. Notice how the bottom-left fixations points get averaged into one fixation

is smoothed out such that the velocity-based filter (see below) used for saccade detection misses the saccade and treats both sets of gaze points as one fixation.

The Butterworth filter is therefore more applicable to real-time applications, producing a smoothly-moving point as feedback for the location of the user's gaze (see, for example, Best and Duchowski (2016) who used just the smoothed gaze point signal in real-time to allow the user to select screen elements, in this work the Butterworth filter was used for real-time smoothing and the Savitzky–Golay filter was used for fixation detection). Without the Butterworth filter, raw gaze data in interactive settings may appear too jerky. For off-line signal analysis, however, the use of a filter with built-in smoothing capabilities, such as the Savitzky–Golay filter (see below) is probably sufficient.

## 14.2 Velocity-Based Saccade Detection

Following Andersson et al. (2010) and Nyström and Holmqvist (2010), a second-order Savitzky–Golay (SG) filter (Savitzky and Golay 1964) can be used to differentiate the (smoothed or unsmoothed, raw) positional gaze signal into its velocity estimate. Using the notation of (14.1) the Savitzky–Golay (SG) filter fits a polynomial curve of order $n$ via least squares minimization prior to calculation of the curve's $s$th derivative (e.g., 1st derivative ($s = 1$) for velocity estimation). Unlike the Butterworth filter and because it lacks a history buffer, the Savitzky–Golay is a Finite Impulse Response (FIR) filter:

$$\dot{x}_n^s(t) = 1/(\Delta t^s)\left(\sum_{i=-p}^{p} h_i^{t,s} x_{n-i}\right) \tag{14.2}$$

which is simply Eq. (14.1) without the $\sum_{i=-q}^{q} g_i^{t,s} \dot{x}_{n-i}$ part which is nothing more than a linear filter applied to the past outputs (history) of the filter.

Following Gorry (1990), the fixations produced in Fig. 14.3 were obtained using an 11-tap (72 ms delay at 150 Hz) SG filter with a threshold of ±30 deg/s to produce fixations. Generally these are the three parameters that need to be tuned for the SG filter: its width (how long the filter is—the longer it is the slow it will work in real-time but the less susceptible it will be to noise), its degree (i.e., order of the polynomial used to fit the curve), and the threshold value. Care must be taken regarding conversion of the data from pixels to degrees visual angle, using the screen dimension, resolution, and distance that the user was away from the display.

Fine-tuning of the filter is best accomplished against some ground truth, such as a calibration grid. If the user is asked to fixate 9 points, then about 9 fixation points are expected. Usually a few more can realistically be expected in practice, due to blinks or untrained users darting their gaze about the screen. In any case, visualization is paramount to filter tuning. Once the filter parameters have been set, they should then be used for all subsequent (e.g., batch) processing.

## 14.3 Microsaccade Detection

Microsaccades can be detected in the raw (unprocessed) stationary eye movement signal, $\mathbf{p}_t = (x(t), y(t))$, using an adapted version of Engbert and Kliegl (2003) algorithm.

The algorithm proceeds in three steps. First, the time series of gaze positions is transformed to velocities via

$$\dot{x}_n = \frac{x_{n+2} + x_{n+1} - x_{n-1} - x_{n-2}}{6\Delta t}, \tag{14.3}$$

which can be done separably for $x(t)$ and $y(t)$. Equation (14.3) represents a moving average of velocities over five data samples. As Engbert and Kliegl note, due to the random orientations of the velocity vectors during fixation, the resulting mean value is effectively zero. Microsaccades, being ballistic movements creating small linear sequences embedded in the rather erratic fixation trajectory can therefore be identified by their velocities, which are seen as "outliers" in velocity space.

Second, velocity thresholds for the detection algorithm are based on the median of the velocity time series to protect the analysis from noise. A multiple of the standard deviation of the velocity distribution is used as the detection threshold (Engbert 2006),

$$\sigma_x = \sqrt{\langle \dot{x}^2 \rangle - \langle \dot{x} \rangle^2}, \quad \sigma_y = \sqrt{\langle \dot{y}^2 \rangle - \langle \dot{y} \rangle^2}$$

where $\langle \cdot \rangle$ denotes the median estimator. Detection thresholds are computed independently for horizontal $\eta_x$ and vertical $\eta_y$ components and separately for each trial, relative to the noise level, i.e., $\eta_x = \lambda \sigma_x$, $\eta_y = \lambda \sigma_y$. Engbert and Kliegl (2003) use

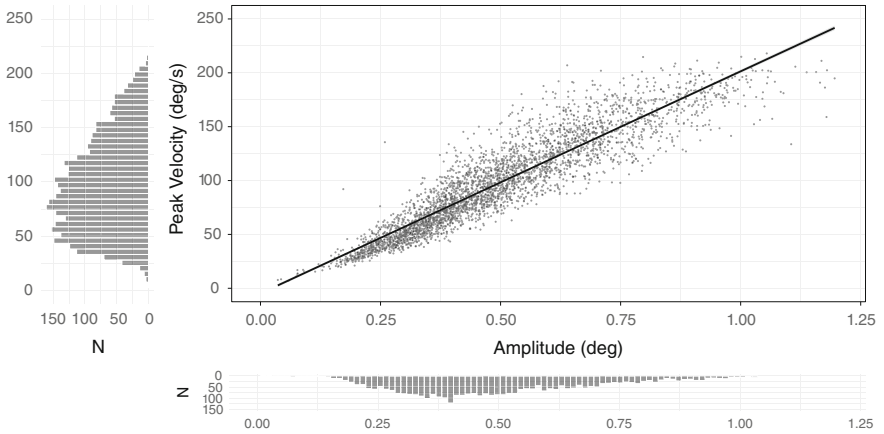



**Fig. 14.4** Plot of microsaccadic peak velocity/amplitude relation from a study where microsaccades were detected

$\lambda = 6$ in their computations[1] and require a minimal microsaccade duration of 6 ms (three data samples at 500 Hz). Engbert (2006) also stipulates a necessary condition for a microsaccade, requiring $\dot{x}$ and $\dot{y}$ fulfill the criterion $(\dot{x}_n/\eta_x)^2 + \left(\dot{y}_n/\eta_y\right)^2 > 1$.

Third, Engbert and Kliegl (2003) focus on binocular microsaccades, defined as microsaccades occurring in left and right eyes with a temporal overlap: if a microsaccade in the right eye starting at time $r_1$ is found that ends at time $r_2$, and a microsaccade in the left eye begins at time $l_1$ and ends at time $l_2$, then the criterion for temporal overlap is implemented by the conditions $r_2 > l_1$ and $r_1 < l_2$.

Oftentimes a good visualization of detected microsaccades is a plot of the peak velocity to amplitude relation (microsaccade main sequence), as shown in Fig. 14.4, similar to the plot given by Siegenthaler et al. (2014) in their work investigating microsaccade response to task difficulty.

## 14.4 Validation: Computing Accuracy, Precision, and Refitting

Following fixation detection, consider computing your own accuracy and precision as well as potentially systematically refitting all data based on a least-squares minimization approach (this was alluded to as "internal calibration" in Chap. 8). These functions depend on capturing gaze data on a calibration type grid where the user is expected to be looking at known points (ground truth). It is a good idea to include a calibration grid as part of the stimulus for validation purposes. Because the eye

[1] Mergenthaler (2009) notes that the choice of $\lambda$ substantially affects the number of detected microsaccades. As $\lambda$ increases, the number of detected microsaccades decreases.

tracking software will require calibration, adding a second calibration image may seem redundant, but providing such a grid image and instructing the user to view the points in a pre-defined pattern will go a long way toward allowing for analysis of precision and accuracy.

Blignaut and Beelders (2012) informally define precision as the compactness of gaze data with respect to a target, i.e., a calibration point, while accuracy refers to the distance between the calibration point and the centroid (mean) of the measurements.

More formally, suppose there are $m$ calibration points,

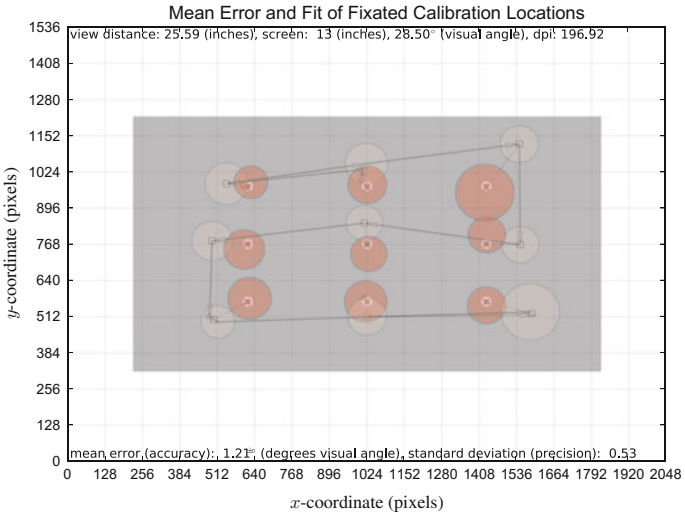
$$\{(s_{1x}, s_{1y}), (s_{2x}, s_{2y}), \ldots, (s_{mx}, s_{my})\}$$

along with $n$ observed data points

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}.$$

Figure 14.5 shows an example—in that particular example, there happen to be as many fixation points as there are calibration points. In general, however, one could expect several fixation points close to the vicinity of each calibration point.

Computation of accuracy relies upon computation of the centroid (mean) of all the fixation points that are in proximity to each calibration point. One way to compute accuracy $A$, is, as given by Johansen et al. (2011):




(a) Accuracy, precision, refitting.

**Fig. 14.5** Example of accuracy, precision and refitting. Calibration points $\{(s_{ix}, s_{iy})\}$ are marked by small $\times$ symbols. Observed fixations $\{(x_i, y_i)\}$ are shown as *faded circles*, with their centroids marked with a $\square$. The *darker shaded circles* are the result of correction via Lagrange's method of least squares (see text)

$$A = \frac{1}{m}\sum_{i=1}^{m}\left[\frac{1}{n}\sum_{j=1}^{n}\|p_{i,j} - s_i\|\right] \tag{14.4}$$

where $\|\cdot\|$ denotes the Euclidean distance, and each $p_{i,j}$ is the $j$th observed point $(x_j, y_j)$ that is in proximity to the $i$th calibration point $s_i = (s_{ix}, s_{iy})$. The inner sum, $1/n\sum_{j=1}^{n}\|p_{i,j} - s_i\|$, simply computes mean of the distances of each point $p_{i,j}$ in proximity to the $i$th calibration point $s_i = (s_{ix}, s_{iy})$. The outer sum, $1/m\sum_{i=1}^{m}[\cdots]$, simply computes the mean of means, hence mean accuracy.

Alternatively, one could compute the centroid of all the points $p_{i,j}$ in proximity to the $i$th calibration point $s_i = (s_{ix}, s_{iy})$, i.e.,

$$\bar{p}_i = \frac{1}{n}\sum_{j=1}^{n} p_{i,j} \tag{14.5}$$

and then compute the mean of means

$$A = \frac{1}{m}\sum_{i=1}^{m}\|\bar{p}_i - s_i\| \tag{14.6}$$

Both approaches are equivalent. Given the computation of accuracy, i.e., the mean distance of the observed points to each calibration point, precision is merely computation of the standard deviation of the distance of the observed points to the corresponding calibration point with respect to the mean distance (accuracy).

The tricky part in the above is finding all points $p_{i,j}$ in proximity to the $i$th calibration point $s_i = (s_{ix}, s_{iy})$. Generally, there will be a small and unchanging number of calibration points $s_i$, e.g., $m = 9$. However, there may be a rather large and unpredictable number of observed data points $p_{i,j}$ about each calibration point. One could iterate through all of the points, compute the distance to each of the $m$ calibration points and then find the one in closest proximity. A faster approach is to set up a $k$d-tree data structure with the $m$ calibration points as the tree nodes. This will produce a small tree that will produce very short query times for finding the nearest neighbor (calibration point) to each of the observed data points. This works quite well in practice and facilitates computation of accuracy and precision.

Beyond accuracy and precision, one could also use the $k$d-tree to compute a second-order correction to all observed points. This results in squashing or stretching (scaling and translating) the point positions to better align them to the calibration points. Correction relies on Lagrange's method of least squares (Lancaster and Šalkauskas 1986, Sect. 2.5).

As used in 9-point calibration described by Morimoto and Mimica (2005), define $(s_{ix}, s_{iy})$ as the $i$th calibration point, and $(\bar{x}_i, \bar{y}_i)$ as centroid of the observed points in closest proximity to the calibration point. The sought second order polynomial is:

$$s_{ix} = a_0 + a_1\bar{x}_i + a_2\bar{y}_i + a_3\bar{x}_i\bar{y}_i + a_4\bar{x}_i^2 + a_5\bar{y}_i^2,$$
$$s_{iy} = b_0 + b_1\bar{x}_i + b_2\bar{y}_i + b_3\bar{x}_i\bar{y}_i + b_4\bar{x}_i^2 + b_5\bar{y}_i^2.$$

The parameters $a_0$–$a_5$ and $b_0$–$b_5$ are the unknowns. Parameters $(a_0, b_0)$ specify translation, $(a_1, a_2, b_1, b_2)$ specify rotation. The rest are higher-order terms that can potentially handle pin-cushion effects and perhaps other deformations.

The above can be reformulated into the general matrix format by writing:

$$\begin{bmatrix} s_{1x} & s_{1y} \\ s_{2x} & s_{2y} \\ \vdots & \vdots \\ s_{nx} & s_{ny} \end{bmatrix} = \begin{bmatrix} 1 & \bar{x}_1 & \bar{y}_1 & \bar{x}_1\bar{y}_1 & \bar{x}_1^2 & \bar{y}_1^2 \\ 1 & \bar{x}_2 & \bar{y}_2 & \bar{x}_2\bar{y}_2 & \bar{x}_2^2 & \bar{y}_2^2 \\ \vdots & \vdots & \vdots & & & \\ 1 & \bar{x}_n & \bar{y}_n & \bar{x}_n\bar{y}_n & \bar{x}_n^2 & \bar{y}_n^2 \end{bmatrix} \begin{bmatrix} a_0 & b_0 \\ a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \\ a_4 & b_4 \\ a_5 & b_5 \end{bmatrix} \tag{14.7}$$

$$\begin{bmatrix} s_{ix} & s_{iy} \end{bmatrix} = \begin{bmatrix} 1 & \bar{x}_i & \bar{y}_i & \bar{x}_i\bar{y}_i & \bar{x}_i^2 & \bar{y}_i^2 \end{bmatrix} \begin{bmatrix} a_0 & b_0 \\ a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \\ a_4 & b_4 \\ a_5 & b_5 \end{bmatrix}, \tag{14.8}$$

or in matrix notation,

$$\mathbf{Y} = \mathbf{X}\hat{\mathbf{B}}.$$

The solution is left-multiplied by $(\mathbf{X}^T\mathbf{X})^{-1}$ to obtain the estimate of $\hat{\mathbf{B}}$:

$$\hat{\mathbf{B}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}.$$

Matrix $\hat{\mathbf{B}}$ is the correction matrix that can now be systematically applied to all gaze data, or perhaps it could be used per individual, providing a type of "personal correction", which is what the eye tracker's calibration is supposed to provide. Using $\hat{\mathbf{B}}$ essentially applies a secondary calibration to the data.

## 14.5 Binocular Eye Movement Analysis: Vergence

When the eyes move through equal angles in opposite directions, *vergence* is produced (Howard 2002). When the visual axes move inwards, the eyes *converge*; when the axes move outwards, they *diverge*. Convergence ensures that the projection of images on the retina of both eyes are in registration with each other, allowing the brain to fuse them together into a single percept, allowing stereoscopic vision of three-dimensional space. Normal binocular vision is primarily characterized by this type

of *fusional* vergence of the disparate retinal images (Shakhnovich 1977). Vergence driven by retinal blur is distinguished as *accommodative* vergence (Büttner-Ennever 1988).

The angle between the visual axes is the *vergence angle*: when fixating at infinity, the vergence angle is zero (the visual axes are parallel). The angle increases when the eyes converge. For symmetrical convergence, the angle of horizontal vergence $\phi$ is related to the interocular distance $a$ and the distance of the point of fixation from a point midway between the eyes $D$ by the expression: $\tan(\phi/2) = a/(2D)$. The change in vergence per unit change in distance is greater at near than at far viewing distances.

Refining the 3D gaze point geometry given in Sect. 7.3, Daugherty et al. (2010) showed how to measure the user's vergence point when viewing stereo displays. As with the 3D gaze in Virtual Reality, measurement of vergence depends on the disparity between the left and right horizontal gaze coordinates, e.g., $x_r - x_l$ given the left and right gaze points, $(x_l, y_l)$, $(x_r, y_r)$ as delivered by current binocular eye trackers.

Of particular interest is the measure of relative vergence, that is, the change in vergence from fixating a point $P$ placed some distance $\Delta d$ behind (or in front of) point $F$, the point at which the visual axes converge at viewing distance $D$. The visual angle between $P$ and $F$ at the nodal point of the left eye is $\phi_l$, signed positive if $P$ is to the right of the fixation point. The same angle for the right eye is $\phi_r$, signed in the same way. The binocular disparity of the images of $F$ is zero, since each image is centered on each eye's visual axis. The angular disparity $\eta$ of the images of $P$ is $\phi_l - \phi_r$. If $\theta_F$ is the binocular subtense of point $F$ and $\theta_P$ is the binocular subtense of point $P$, then $\eta = \phi_l - \phi_r = \theta_P - \theta_F$. Thus, the angular disparity between the images of a pair of objects is the binocular subtense of one object minus the binocular subtense of the other (see Fig. 14.6).

Given the binocular gaze point coordinates reported by the eye tracker, $(x_l, y_l)$ and $(x_r, y_r)$, an estimate of $\eta$ can be derived following calculation of the distance $\Delta d$ between $F$ and $P$, obtained via triangle similarity:

$$\frac{a}{(D + \Delta d)} = \frac{x_r - x_l}{\Delta d} \quad \Rightarrow \quad \Delta d = \frac{(x_r - x_l)D}{a - (x_r - x_l)}. \tag{14.9}$$

For objects in the median plane of the head, $\phi_l = \phi_r$ so the total disparity $\eta$ is $2\phi$ degrees. By elementary geometry, $\phi = \theta_F - \theta_P$ (Howard and Rogers 2002). If the interocular distance is $a$,

$$\tan\frac{\theta_P}{2} = \frac{a}{2(D + \Delta d)} \quad \text{and} \quad \tan\frac{\theta_F}{2} = \frac{a}{2D}.$$

For small angles, the tangent of an angle is equal to the angle in radians. Therefore,

$$\eta = 2\phi \approx \frac{a}{2(D + \Delta d)} - \frac{a}{2D} \quad \text{or} \quad \eta \approx \frac{-a\Delta d}{D^2 + D\Delta d}. \tag{14.10}$$

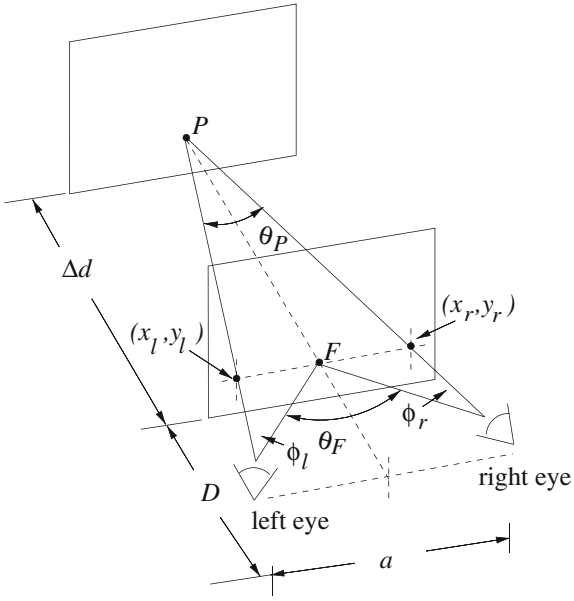



**Fig. 14.6** Binocular disparity of point $P$ with respect to fixation point $F$, at viewing distance $D$ with (assumed) interocular distance $a$ (Howard and Rogers 2002). Given the binocular gaze point coordinates on the image plane $(x_l, y_l)$ and $(x_r, y_r)$ the distance between $F$ and $P$, $\Delta d$, is obtained via triangle similarity. Assuming symmetrical vergence and small disparities, angular disparity $\eta$ is derived

Since for objects within Panum's fusional area $\Delta d$ is usually small by comparison with $D$ we can write

$$\eta \approx \frac{-a\,\Delta d}{D^2}. \tag{14.11}$$

Thus, for symmetrical vergence and small disparities, the disparity between the images of a small object is approximately proportional to the distance in depth of the object from the fixation point.

In essence, $\Delta d$ provides a good estimate of the gaze point's $z$-coordinate, i.e., with gaze disparity computed as $\Delta x = x_r - x_l$, disparity induced gaze depth $z = \Delta d$, relative to the screen position, is obtained via

$$\frac{-z}{D-z} = \frac{\Delta x}{a} \quad \Rightarrow \quad z = \frac{\Delta x\, D}{\Delta x - a}, \tag{14.12}$$

where $z = 0$ denotes gaze depth at the screen plane, with $z$ positive in front of the screen, and negative behind. Note that this derivation is identical to that of (14.9) save for the sign change.

Depending on the stereo display used, the signal can be rather noisy. Wang et al. (2012) showed that 3D calibration improves precision of the gaze depth estimate, and that, for real-time applications, the Butterworth filter adequately smooths the 3D gaze point if it is required, e.g., for pointing and/or selecting in 3D. 3D calibration can be performed via a continuous-type animation of a 3D point, e.g., along a Lissajous-knot path. Such a path specifies a 3D point (e.g., sphere's) position $p(t) = (x(t), y(t), z(t))$

which changes with time $t$ in seconds, according to $p(t) = \mathbf{A}\cos(2\pi\,\mathbf{f}\,t + \phi)$, with component amplitudes $\mathbf{A}$ in cm, frequencies $\mathbf{f}$ in Hertz, and phase angles $\phi$. The following parameters produced useful calibration animations: $\mathbf{A} = (9, 5, 20)$ cm, $\mathbf{f} = (0.101, 0.127, 0.032)$ Hz, $\phi = (0°, -90°, 57°)$. Because the continuous calibration animation in effect produces a very large number of points (e.g., 2400) for least-squares calibration, Wang et al. (2013) showed that it is significantly more accurate for depth estimation than a grid-like calibration based on 27 static calibration points, as suggested by Essig et al. (2006).

One of the more promising applications of 3D gaze estimation was reported by Duchowski et al. (2014), who tested gaze-contingent depth-of-field and found that it significantly reduced visual discomfort associated with 3D displays due to the *accommodation-vergence conflict*. Typical stereo displays fail to simulate accommodative blur, thereby fixing accommodative demand in the presence of depth-variable disparity. The key innovation of the approach was setting the depth-of-field focal plane to gaze depth $z$ directly.

## 14.6 Ambient/Focal Eye Movement Analysis

There is an increasing demand for advanced characterization of eye movements, surpassing traditional categorization of the captured eye gaze sequence $(x_i, y_i, t_i)$ as fixations and saccades into higher-level descriptors of visual behavior. Krejtz et al. (2016) review various approaches of eye movement analysis, including whether or not the user is interacting socially, concentrating on a mental task, engaging in a physical activity, or is inside or outside. They then define $\mathcal{K}$ on a novel parametric scale where positive values on the ordinate indicate *focal* patterns while negative values suggest *ambient* visual scanning. The abscissa serves to indicate time, so that $\mathcal{K}$ acts as a dynamic indicator of fluctuation between ambient/focal visual scanning. The derivation of coefficient $\mathcal{K}$ was based on Unema et al. (2005) original characterization of the two ambient and focal modes of attention but also considered the time course of an individual's eye movement record. Since then, several metrics derived from fixations and saccades have appeared characterizing visual perception along similar patterns, i.e., exploring and inspecting Velichkovsky et al. (2005), skimming and scrutinizing (Lohmeyer and Meboldt 2015), or exploring and exploiting (Peysakhovich 2016).

To compute $\mathcal{K}$, both fixation durations and saccade amplitudes are transformed into a standard score ($z$-score), allowing computation of an ambient/focal attentional coefficient per individual scanpath. The transformation into standard scores represents the distance between the raw score and the mean in units of the standard deviation, allowing for direct mathematical comparison of both measures.

Coefficient $\mathcal{K}$ is calculated for each participant as the mean difference between standardized values ($z$-scores) of each saccade amplitude $(a_{i+1})$ and its preceding $i$th fixation duration $(d_i)$:

$$\mathcal{K}_i = \frac{d_i - \mu_d}{\sigma_d} - \frac{a_{i+1} - \mu_a}{\sigma_a}, \quad \text{such that} \quad \mathcal{K} = \frac{1}{n}\sum_n \mathcal{K}_i, \tag{14.13}$$

where $\mu_d$, $\mu_a$ are the mean fixation duration and saccade amplitude, respectively, and $\sigma_d$, $\sigma_a$ are the fixation duration and saccade amplitude standard deviations, respectively, computed over all $n$ fixations and hence $n$ $\mathcal{K}_i$ coefficients (i.e., over the entire duration of stimuli presentation) (Krejtz et al. 2012).

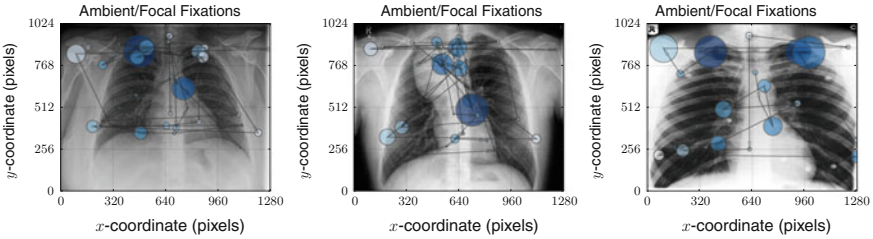
$\mathcal{K}_i > 0$ shows that relatively long fixations were followed by short saccade amplitudes, indicating focal processing. $\mathcal{K}_i < 0$ shows that relatively short fixations were followed by relatively long saccades, suggesting ambient processing. $\mathcal{K}_i = 0$ means that the fixation length and subsequent saccade amplitude are statistically equivalent, suggesting the ambiguous situation of a person exhibiting long saccades preceded by long fixations or vice-versa, exhibiting short fixations followed by short saccades.

$\mathcal{K}$ lends itself well to visualization of both scanpaths and heatmaps (Duchowski and Krejtz 2015, 2017). Normalizing $\mathcal{K}$ over the course of a scanpath will yield a $[0:1]$ range that can be used as an index to a choice of color palettes. For scanpaths, this can produce darker shades of colors for more focal fixations, and lighter shades for ambient fixations.

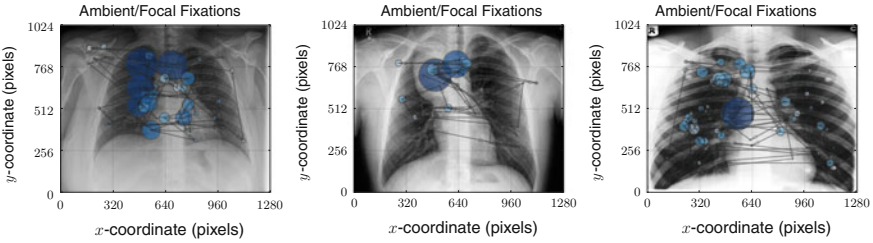
Figure 14.7 shows the use of a sequential color map in blue hues, used to visually distinguish visual inspection of Chest X-Ray (CXR) images as viewed by experts and novices. Radiologists employ a partially endogenous, cognitive visual inspection strategy, related to top-down mechanisms that are based on prior expectations (Mello-Thoms et al. 2002), which in turn are couched in training and experience. In the specific case of CXR reading, this strategy may be typified by the **ABCDEFGHI** mnemonic (Vitak et al. 2012).

The ABCDEFGHI mnemonic guides viewers through a series of checks and assessments to inspect **A**irway, **B**ones, **C**ardiac silhouette, **D**iaphragms, **E**xternal soft tissues, **F**ields of the lungs, **G**astric bubble, **H**ila, and **I**nstrumentation. Figure 14.7 illustrates qualitatively the differences in expert and novice visual strategies: the expert executes the inspection quickly, tending to "check off" the **ABCDEFGHI** elements, not pausing excessively on any particular element. Visualization of $\mathcal{K}$ readily depicts this strategy, especially in the peripheral image regions (e.g., when inspecting bones, diaphragm). Conversely, the novice tends to dwell longer on each of the elements, often revisiting previously examined regions of the film. An "outside-in" ambient-to-focal strategy is thus not as clearly depicted as it is for the expert.

For aggregate gaze visualization, heatmaps provide a depiction of gaze by combining fixations from multiple viewers while sacrificing temporal order information (Duchowski et al. 2012). The heatmap can be thought of as a type of histogram, with accumulation of fixation count recorded at each pixel, but instead of discrete bins of data, each bin is represented by a Gaussian "peak" (or "valley", depending on polarity). Heatmaps are thus also known as Gaussian Mixture Models, or GMMs. The Gaussian functions modeled at each bin (e.g., fixation) results in a smooth height map (Gaussian surface) of relatively weighted pixels. Colorization options vary. One of the more basic is obtained by mapping the height information directly to the alpha channel, resulting in a transparency map. Another popular option of the normalized




(a) Expert reading CXR images: normal (left) and abnormal (middle: ant. mediastinal mass, right: apical pneumothorax).




(b) Novice reading CXR images: normal (left) and abnormal (middle: ant. mediastinal mass, right: apical pneumothorax).

**Fig. 14.7** Example of expert/novice scanpaths over Chest X-ray (CXR) film. CXR images in the middle column feature an anterior mediastinal mass found at about pixel position (635, 768). Images in the right column feature an apical pneumothorax at about pixel position (650, 510). Experts tend to execute the visual inspection task much faster than novices, with novices tending to dwell longer over what they think may be abnormalities. Ambient/focal fixation visualization shows a greater preponderance of experts allocating ambient (lighter) fixations in peripheral image regions. Thanks to Dr. Helena Duchowska (MD, retired) for her help in reading the CXR images and pinpointing the anomalies contained therein

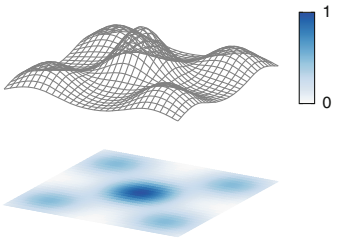
height map uses the pervasive rainbow color palette although sequential or divergent color palettes can also be used to good effect.

The heatmap, or attentional landscape, was introduced by Pomplun et al. (1996), and popularized by Wooding (2002) (both were predated by Nodine et al. (1992) who rendered "hotspots" as bar-graphs). A basic heatmap is generated by accumulating exponentially decaying intensity $I(i, j)$ at pixel coordinates $(i, j)$ relative to a fixation at coordinates $(x, y)$,

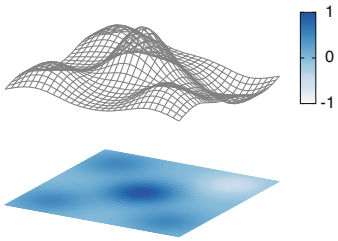
$$I(i, j) = \exp\left(-((x - i)^2 + (y - j)^2)/(2\sigma^2)\right) \tag{14.14}$$

where the exponential decay is modeled by the Gaussian point spread function (PSF). Ambient/focal visualization is facilitated by assigning the sign of $\mathcal{K}_i$ to the direction (polarity) of the Gaussian peak,

$$I(i, j) = \text{sgn}(\mathcal{K}_i) \exp\left(-((x - i)^2 + (y - j)^2)/(2\sigma^2)\right) \tag{14.15}$$




(a) Mixing multiple Gaussian functions.




(b) Negating one Gaussian to produce ambient intensity.

**Fig. 14.8** Mixing Gaussian point spread functions to produce an ambient/focal heatmap. Two hypothetical fixations overlap in the center, with an additional fixation at each corner. Ambient intensity is modeled by negating the Gaussian function (at *upper-right*), producing a valley instead of a peak

Figure 14.8 illustrates the concept. With this construct, it is possible that overlapping fixations at the same location, but with exactly opposite polar magnitudes, would result in a flat surface at that location. The two fixations would effectively neutralize each other. In practice, however, fixations rarely overlap precisely.

## 14.7 Transition Entropy Analysis

Transition entropy, as developed by Krejtz et al. (2015), grew out of a need to statistically compare fixation transitions, particularly when expressed as matrices based on a uniform grid, as presented by Fischer and Peinsipp-Byma (2007). The idea for comparing transitions between fixations can be traced to Ellis and Stark (1986) who likely introduced the concept. Given a uniform grid superimposed over the stimulus, each cell is denoted as the $i$th Area Of Interest, or AOI. The idea for transition entropy requires construction of first-order (fixation) transition matrices, and their transformation into conditional probability matrices for which conditional transition entropy $H_t$ is calculated,

$$H_t = -\sum_{i=1}^{n} p_i \sum_{j=1}^{n} p_{ij} \log_2 p_{ij}, \tag{14.16}$$

where $p_i$ is the simple (observed) probability of viewing the $i$th AOI, $p_{ij}$ is the conditional probability of viewing the $j$th AOI given the previous viewing of the $i$th AOI, and $n$ is the number of AOIs. $H_t$, or *entropy*, provides a measure of statistical dependency in the spatial pattern of fixations represented by the transition matrix, and may be used to compare one matrix to another.

Note that a uniform grid is not a necessity. The transition matrix can be composed from arbitrarily defined AOIs. For example, Ellis and Stark (1986) compared transition matrices of airline pilots viewing a Cockpit Display of Traffic Information or CDTI. The CDTI was fixed with 8 AOIs. Krejtz et al. (2015) provide other examples.

How to interpret the meaning of $H_t$? Weiss et al. (1989) note that in a transition matrix, a small $H_t$ suggests dependencies between the fixation points, whereas a large $H_t$ suggests a random scanning pattern. Stated another way, entropy refers to the "expected surprise" of a given gaze transition. Minimum entropy of 0 suggests no expected surprise, meaning that a gaze transition is always expected to the same $j$th AOI. Maximum entropy, on the other hand, suggests maximum surprise, since transition from source AOI to any destination AOI is equally likely, and hence whichever occurs results in maximum expected surprise. More formally, the term $-p_{ij} \log_2 p_{ij}$ in (14.16) is the transition's contribution to system entropy, modeled by its probability multiplied by its *surprisal* (Hume and Mailhot 2013).

The key difference between Krejtz et al. (2015) approach and that of Ellis and Stark (1986) is that Krejtz et al. consider self-transitions. The use of a grid also makes the transition matrix analysis *content-independent*. The benefit of content-independence is that it allows estimation of transition matrices irrespective of the expected AOIs in the scene. All that is required is knowledge of the dimensions of the screen to establish different grid granularities. This makes the statistical analysis portable among different experimental designs.

Krejtz et al. (2015) compute a transition matrix by setting matrix elements $p_{ij}$ to the number of transitions from the $i$th source AOI to the $j$th destination AOI for each participant. The matrix is then normalized relative to each source AOI (i.e., per row). In practice, it is possible that no transitions from the $i$th AOI are observed. This leads to a zero matrix row sum and division by zero. When this occurs, each of the row entries is set to their uniform transition distribution, namely $p_{ij} = 1/s$, where $s$ is the number of AOIs, thereby modeling an equally likely probability of transitioning to any other AOI given this $i$th source AOI (hence maximum "surprise"). The benefit of this implementation decision is that it leads to the construction of a transition matrix that is regular (specifically a right stochastic matrix), with all entries positive and non-zero, facilitating stationary entropy calculation via Eigen analysis. Note that setting each of the $p_{ij}$ entries to $1/s$ would lead to a uniform matrix with maximum entropy equal to $\log_2 s$ bits per transition. Indeed, maximum entropy is used to normalize the empirical entropy obtained from each transition matrix. That is, statistical comparison of mean entropies per experimental condition is facilitated by computing $H_t$ per individual participant and per condition, then normalizing, $\hat{H}_t = H_t / \log_2 s$. This results in a table of entropies (each entropy computed from an individual's transition matrix) for each of experimental conditions and each of the participants. Analysis of variance (ANOVA) is then used to test for differences in mean (normalized) entropy per condition. An example of entropy analysis is given in Chap. 15.

## 14.8 Spatial Distribution Analysis

Another measure related to spatial distribution of fixation, this time of dispersion rather than transition, is the Nearest Neighbor Index, or NNI, as described by Clark and Evans (1954). Denoted by symbol $\mathcal{R}$, the NNI is based on the "distance from an individual to its nearest neighbor, irrespective of direction." The NNI describes the spatial distribution of points, e.g., fixations, as either ordered ($\mathcal{R} > 1$), random ($\mathcal{R} = 1$), clustered ($\mathcal{R} < 1$), or maximally aggregated, i.e., singular ($\mathcal{R} = 0$). For $n$ points, the NNI, or $\mathcal{R}$, is defined as

$$\mathcal{R} = \frac{2\sqrt{\rho}}{n} \sum_{i}^{n} r_i \tag{14.17}$$

where $r_i$ is the distance from the $i$th (fixation) point to its *nearest* neighbor, and $\rho$ is the density of the observed distribution, i.e., $\rho = n/A$ where $A$ is the observation area (e.g., width $\times$ height in pixels, or perhaps in degrees visual angle, so long as the units match those used in the distance computation). The NNI is fairly straightforward to compute, as the $k$d-tree spatial data structure can be used for fast nearest-neighbor queries.

## 14.9 Summary and Further Reading

The statistical comparison of transition matrix entropies along with the dynamics of the $\mathcal{K}$ coefficient were conceptualized by Dr. Krzysztof Krejtz the SWPS University of Social Sciences and Humanities in Warsaw, Poland. For details on how to implement gaze transition entropy, see Krejtz et al. (2015). For computing the ambient/focal $\mathcal{K}$ coefficient, see Krejtz et al. (2016), and for its visualization, see Duchowski and Krejtz (2017).