Trajectory-based prediction of saccade landing points

Bachelorarbeit am Institut für Neuro- und Bioinformatik
Technisch Naturwissenschaftliche Fakultät
Universität zu Lübeck

von

Max Pagel
Matrikelnummer: 530855

Ausgabe durch:
Prof. Dr. Thomas Martinetz

Betreut von:
Dipl.-Inf. Michael Dorr

15.3.2007

Statement of Originality:

The work presented in this thesis is, to the best of my knowledge and belief, original, except as acknowledged in the text. The material has not been submitted, either in whole or in part, for a degree at this or any other university.

Ich versichere an Eides statt durch meine Unterschrift, dass ich die vorliegende Arbeit selbständig und ohne fremde Hilfe angefertigt und alle Stellen, die ich wörtlich oder annähernd wörtlich aus Veröffentlichungen entnommen habe, als solche kenntlich gemacht habe, mich auch keiner anderen als der angegebenen Literatur oder sonstiger Hilfsmittel bedient habe. Die Arbeit hat in dieser oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen.

Lübeck, March 12, 2007

Contents

1 Abstract

Gaze-contingent displays are sophisticated presentation utilities, that are aware of their user's gaze position. In this way an advanced presentation of the displayed content is possible, as the display can modify the presentation according to the observer's attention, which is equal to the gaze position.
These displays, however, presently suffer from a latency problem, which occurs when reacting on large and fast changes of the gaze position, so-called saccades. As the time needed to update the display may exceed the blindness during a saccade, the observers might be aware of the display change, an often unwanted feature.
This latency problem could be solved by providing a suitable method to predict a saccade's landing point according to it's initial trajectory. In this way the display update can be started before the saccade has ended, and will likely be finished before the observer regains perception.
In the framework of this paper three different approaches to predict the landing point of a saccade will be discussed, including methods of regression and machine learning approaches. One of these approaches provides a suitable prediction method that may be used to predict saccadic landing points at a mean accuracy of 1.6° which is similar to the natural accuracy of saccades themselves.

2 Introduction and motivation

2.1 Short introduction on the process of human vision

The human eye basically runs in three different types of movements while scanning the surrounding area for visual information. These movements, which the observer is unconscious of, are called [Pal99]:

- Fixations
- Saccades
- Smooth pursuits

During the fixation of a certain object, the gaze position remains more or less constant except for slight changes with an amplitude of less than $0.1°$. These vibrating movements are called physiological nystagmus and constantly change the picture projected onto the retina to avoid adaptation of the visual receptors. The observer however is not aware of the physiological nystagmus.
About two to three times per second the gaze position changes from the fixation of one point to another. These rapid ballistic eye movements are called saccades. Being ballistic in this context means that once they have started, the body is not capable of influencing their trajectory or speed any more. A saccade performs a so-called one-shot movement. For this reason the peak velocity of a saccade correlates with the amplitude in the manner, that a larger amplitude is always combined with a higher peak velocity. Figure 1 shows the trajectory of a typical saccade.

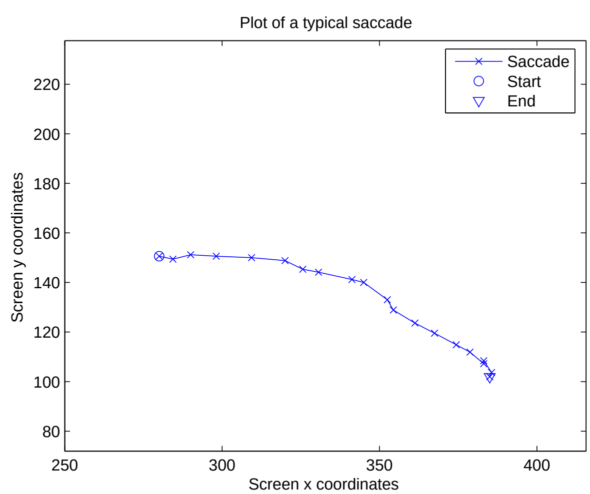

Figure 1: Plot of a typical saccade.

The eyeball is controlled by six different muscles, which are pulling the eyeball in different directions, allowing a very precise control of the eye movements. Saccades are in most cases conducted by the contraction of at least 2 of these muscles. This gives them a slightly curved trajectory, rather than being a straight movement toward their target.
One important fact in the context of this thesis is, that during a saccade, no visual information is processed by the brain, and this means we are effectively blind. This blindness, called saccadic suppression, is not only present during the saccade, but extends beyond it by about 50 ms.
The last type of movement, smooth pursuits, is performed while following a moving object, or to compensate head movements. In this case the eye moves slowly and smoothly to keep the projection of the fixated moving object on the fovea. Smooth pursuit movements especially differ from saccades through their non-ballistic character and their speed. This means that smooth pursuit movements are relying on constant visual feedback from the image, to correct their trajectory.

2.2 Methods of eye tracking

Basically one can separate four different techniques of eye-tracking [DV00]:

- Scleral contact lens/search coil
- Electro-oculography (EOG)
- Photo-oculography (POG) or video-oculography (VOG)
- Video-based combined pupil and corneal reflection

Figure 2: Headmounted camera.

These four methods can be further separated into invasive and non-invasive techniques. The 'scleral contact lens/ search coil' approach, for example, is a heavily invasive method as it makes it necessary to apply a contact lens, containing the search coil, onto the subject's eye, which causes discomfort. The subject's head is then placed into a magnetic field frame, which enables the calculation of the gaze position by measuring the position of the search coil inside the electromagnetic field. For this reason, this method is only useful for scientific applications.
The EOG method is based on the measurement of the skin's potential differences, measured by electrodes, which are attached around the observer's

eyes. It is mostly used to measure horizontal eye movements only. This results from the fact that the electrodes, which are used to measure vertical movements need to be placed above and under the eyes, what is difficult for anatomic reasons. Even though this approach does not cause as much discomfort as the 'scleral contact lens/search coil' method, it is still an invasive method.

POG or VOG include a 'wide variety of techniques based on measurement of distinguishable ocular features' [DV00] i.e. the apparent shape of the pupil, the position of the limbus (the iris-sclera boundary) or the corneal reflection of a direct light source. In most cases the recordings are made by a headmounted camera equipment, as for example displayed in figure 2. All the above methods have in common that they are suitable for eye movement measurements, but seldom provide point of regard measurements.

As this work was motivated by gaze-contingent displays, for which the most important information is the gaze position of the observer, the above eye-tracking methods are not suitable to collect the necessary data. To be able to measure the gaze position either the head must be fixed, so the point of regard can be calculated through the eye-position relative to the head, or one has to record multiple ocular features. The second is done when using the 'Video-based combined pupil and corneal reflection'. By directing a beam of light (usually infrared) at the users eye, the method creates another ocular feature, the reflection of the beam on the cornea. In this way the gaze position is calculated by measuring the position of the cornea reflection relative to the center of the pupil.

The positional difference between cornea reflection and pupil centre changes during pure eye-rotation but remains relatively constant during minor head movements. In this way this method is able to tolerate small head movements and separates eye movements from head movements to calculate the point of regard. Due to this relatively simple approach, these eye-trackers may be set up by using relatively cheap video-cameras and image processing hardware and may even be table-mounted, what makes them minimally invasive.

2.3 Gaze-contingent displays

This work was motivated by the utilisation of gaze-contingent displays which are currently developed at the Institute for Neuro- and Bioinformatics at the University of Lübeck. Gaze-contingent displays are special displays that are linked with an eye-tracker to record the gaze position of their observer. This enables them to adjust the presentation of the content they are displaying according to the gaze position of their observer.

These display adjustments might be, for example, to progressively blur the image with increasing eccentricity outside the current fixated area, or to emphasise certain display regions, to direct the observer's attention. Figure 3 gives an exsample of what a gaze-contingent display might look like.

Figure 3: Example of what a gaze-contingent display might look like. Top: Original image from a video sequences. Bottom: Same image with gaze-contingent temporal filtering applied. The white square at the centre left (below the white sail) indicates the point of regard. Increasing amounts of temporal filtering are visible towards the periphery, i.e. with increasing distance from the square. For example, the two men walking at the left of the image are slightly blurred, and the person who is about to leave the image at the right edge has almost vanished completely.

2.4 The problem

On a gaze-contingent display, the display constantly needs to be updated according to the observer's gaze position. At present, updating the display takes about 30-70 ms. less than 5 ms of this time are produced by the tracker itself, 30 ms come from the graphics update and 0-30 ms are needed to update the screen. Thesedurations are of no importance while the observer performs a fixation, as the gaze-contingent effect normally is not produced just at the exact gaze position. In fact the effect is produced outside the fixated area. In this way the display is not changing significantly during a fixation, as the physiological nystagmus is not pushing the gaze position out of the gaze-contingent effect.

When the observer performs a saccade, however, the system runs into a problem. A saccade performs a large shift of the actual gaze position, out of the presently adjusted area into an area of the screen which has not been adjusted to the gaze position before, meaning that there is no gaze-contingent effect present the moment the gaze reaches its new position.

As mentioned in section 2.1, saccadic suppression extends a saccade for only about 50 ms. So if the display update is not finished 50 ms after the saccade has ended, the observer might become aware of the display change, which behaviour often is not wanted. When looking at figure 3, this would mean that if the subject for example made a saccade from the actual gaze position (white square) to the garbage can on the right side of the image, he/she would see a child that seems to appear from nowhere. Therefor, it would be desirable to start the display update before the saccade has ended, making it necessary to predict the landing point of the saccade according to its initial trajectory.

An ordinary saccade lasts for about 25-50 ms. By enabling a prediction of the landing point, after for example the first 15 ms of the saccade have passed, one would be able to start the display update up to 35 ms earlier. In this way the observer would very unlikely become aware of the display update, making a more user-friendly behaviour of gaze-contingent displays possible. As saccades are not perfectly accurate, but often miss their target for about $1-2°$, a potential prediction method does not need to be perfectly accurate as well.

In the framework of this paper three different approaches to predict the landing point of a saccade will be discussed, including methods of regression and machine learning approaches.

3 Material and methods

3.1 Data source / amount

There were two sets of data available: The first set of data was recorded using a head mounted 250 Hz 'SR Research EyeLink II' eye tracker. In this case 54 subjects had to look at 18 different films of 20 s each, displaying natural scenes in and around Lübeck. The screen used in these recordings was placed at a distance of 46cm to the subjects, showing the films in a resolution of 1280 × 720 pixels, within a visible area of $40.2cm \times 22.7cm$ ($46.9° \times 28.1°$). The outcome of these recordings was a dataset containing 38595 saccades.

The second set was recorded using a 1250 Hz SMI 'IView X Hi-Speed' eye tracker. This eyetracker consists of a tracking column in which the head is fixed, to allow a precise detection of the actual gaze position, free of noise produced by head movements. One single subject looked at black points upon a grey screen, which showed up on a circle around the actual gaze position with a randomly chosen radius of 3 - 20 degrees. The black points were shown in a 900ms interval, leading to the recording of 1041 saccades in total. The screen the subject looked at was set up 65cm away from the subject's face and had a resolution of 800 × 600 pixels, within a visual area of $34cm \times 27cm$ ($29.6° \times 23.1°$).

The saccades within the data recorded by the eye trackers are detected using a two-threshold strategy. If at any time step t the velocity $V(t)$ of the eye movements, averaged over the previous 4 samples, exceeds $138°/s$ the current movement is considered to be a potential saccade. The detector then looks forward and backward in the data to find the time steps $t_{begin}\ and\ t_{end}$ where $V \leq 20°/s$. After having undergone some plausibility checks considering the length and the mean velocity of the potential saccade the data between these two time steps is defined to be a saccade and saved in a record of the form: $Amplitude\ [Timestamp,\ X - position,\ Y - position]^{+}$. Because of the low temporal resolution, only poor results were achieved when using the second dataset. For this reason the research was concentrated on the data recorded at 1250 Hz.

3.2 Preprocessing of the available data

3.2.1 Preprocessing of the 1250 Hz data

It turned out that the detected saccades contained several samples at the beginning and the end, that were recorded from the fixations preceding and following the saccade (see figure 4). Because these fixation points contain no information on the trajectory of the saccade, it was necessary to remove them from the data. To achieve this a second velocity criterion was applied to determine the real start and end points of the saccades. Manual examination of the saccades showed that a good detection of the starting point could be achieved, by defining the saccades starting point as the first point of the first two samples, whose mean velocities were faster than $183.2°/s$. In the same manner the ending point was defined as the first point of the first three points whose mean velocities fell below $137.4°/s$.
Apart from the fixation points, some records consisted of artefacts falsly classified as saccades, or had correction saccades at their ends, which were of no importance for the real saccade as well. These records were, if not already discarded by the velocity criterion, discarded or cut manually. After processing the data this way 999 clean saccades were left from the initial 1041.

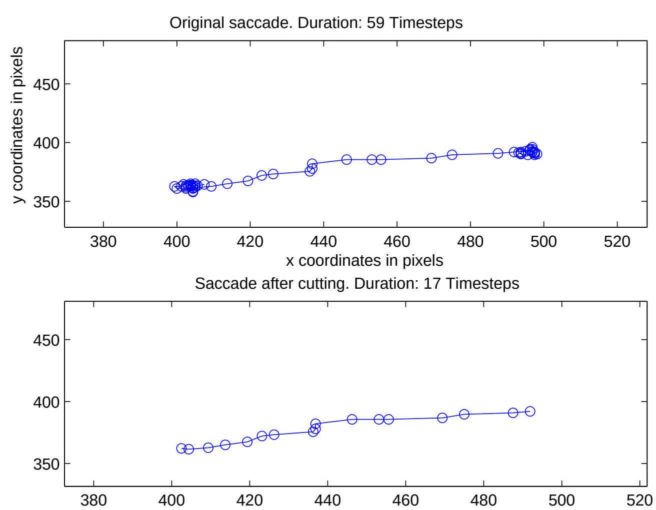

Figure 4: Top: Samples from the preceeding and subsequent fixations are ascribed to the saccade. Bottom: Result of the preprocessing algorithm

3.2.2 Preprocessing of the 250 Hz data

The second data set could not be examined manually because of its size. In this case a different filtering method was used. All 38595 saccades were fitted by a least squares regression, as described in section 3.4, using a polynomial of degree two. In this case, in contrast to the method, described in section 3.4, not only the first n points were used for the regression, but all the available points of the particular saccade. The regression error then was taken as a quality criterion for each saccade such that all saccades whose regression errors exceeded $\frac{1}{5}$ of their over all length were discarded. Figure 5 shows how the value of the quality criterion was determined. Figure 6 gives an example of a bad saccade.

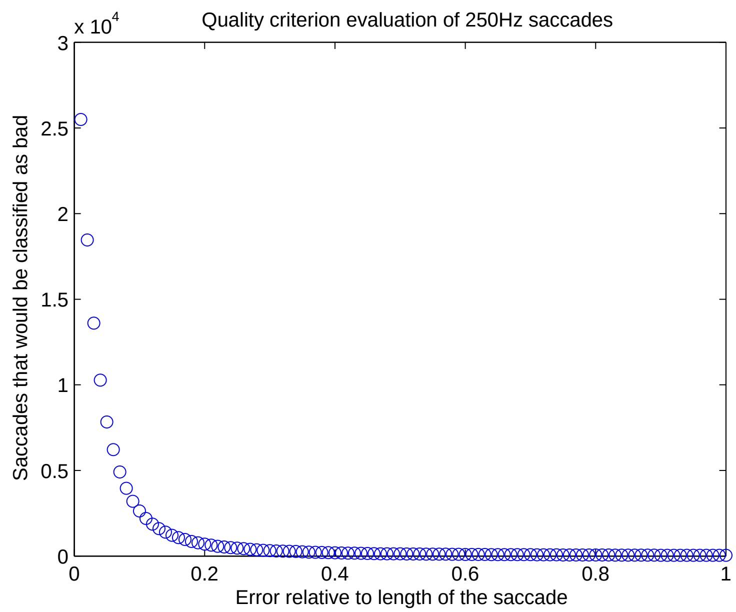

Figure 5: Determine an appropriate value for the quality criterion for the 250 Hz saccades. The figure shows, that a threshold, above which a recorded saccade is classified as bad, smaller than 20% leads to a rapidly increasing amount of sorted out saccades. Therefor 20% of the saccade's length was taken as the relative error from the fit, which is tolerated.

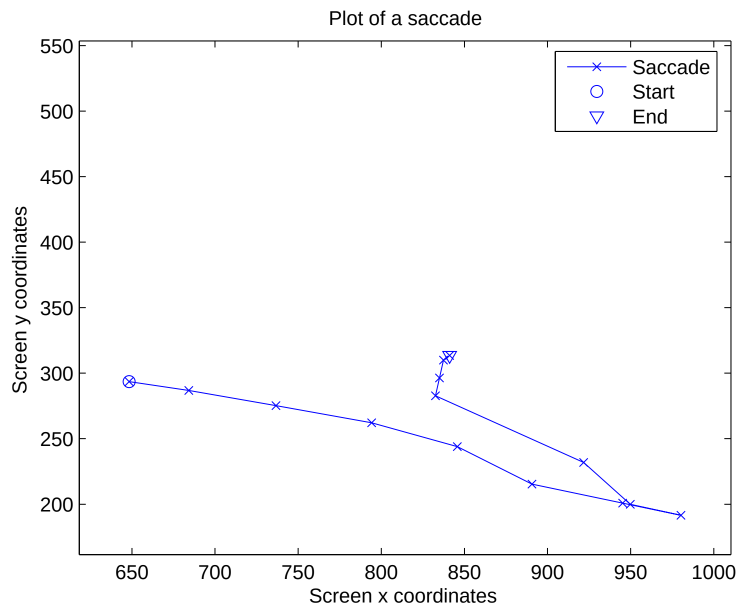

Figure 6: A bad 250 Hz saccade discarded by the automatic filtering method (see section 3.2.2). The almost 180° turn of the saccade is atypical and results in a bad fit result. This record might be the result from a misinterpretation of the detection algorithm, which merged two saccades directly following each other. Another possibility might be that the subject blinked and produced a bad record in this way.

3.3 General approach and rating criterion

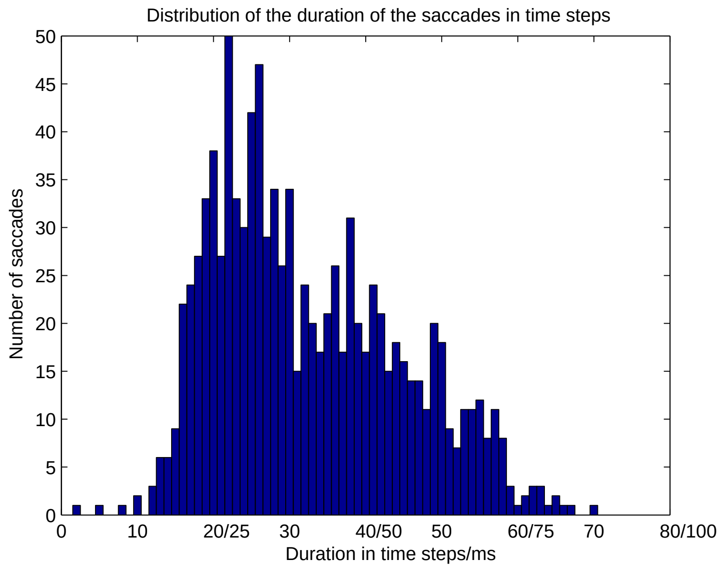

Figure 7: Length distribution in time steps/ms.

Figure 7 shows the distribution of the duration of the saccades in recorded time steps. Obviously about 97% of the saccades exceed 15 samples. This leads to the goal of this work, to predict the landing point of a saccade using only the first 15 samples.
Mathematically a saccade S_i may be viewed as a curve

$$X_i(t_j) \mapsto (x, y) \ where \ t_j \ \epsilon \ T = \{t_1 \ldots \ t_{end}\},$$

where the $X_i(t_j)$ are the known gaze positions from the recordings.
The general approach to predict a saccade S_i is to apply the prediction algorithm on the first 15 gaze points (training samples) of the saccade and try to predict the landing point $X_i(t_{end})$ on the basis of these training samples. The quality of the prediction is rated by the Euclidean distance $E_p(S_i)$ between the predicted landing point $\widehat{X_i}(t_{end})$ and the real landing point $X_i(t_{end})$ of the saccade

$$E_p(S_i) = \|\widehat{X_i}(t_{end}) - X_i(t_{end})\| \tag{1}$$

As a criterion to measure the quality of an algorithm when using n training samples, the mean prediction error over all sacades is compared to the mean of a baseline predictor $E_c(S_i)$ over all saccades. $E_c(S_i)$ is the error which is made if the landing point of the saccade S_i, $X_i(t_{end})$, would be predicted as the last training sample $X_i(t_n)$. This is equal to the case of no prediction at all.

$$E_c(S_i) = \|X_i(t_n) - X_i(t_{end})\| \tag{2}$$

3.4 Using regression for prediction

The first approach was to use a least squares approximation of the saccades to calculate the landing points.
The goal of a least squares fit is to search the polynomial
$P'_r(t) = b_0 + b_1 x + b_2 x^2 + \cdots + b_r x^r$ of degree r which minimizes the sum of the squares of the deviations between the real gaze points $X_i(t_j)$ of a saccade at time t_j, and the modelling polynomial $P'_r(t_j)$.

$$\min_{P'_r \epsilon \Pi_r} \sum_{t \epsilon T} (X_i(t) - P'_r(t))^2 \text{ where } \Pi_r \text{ is the space of polynomials of degree r} \tag{3}$$

Let $X(t)$ be the unknown function, which should be approximated. Assuming that the values $y_i = X(t_j)\ where\ j \,\epsilon\, 1, 2, \ldots, n$ are known, it is possible to do a regression of $X(t)$ by solving a linear system $y = Ab$, where y is of the form:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \tag{4}$$

By using a polynomial of degree 2 for the approximation A would be of the form :

$$\begin{pmatrix} 1 & t_1 & {t_1}^2 \\ 1 & t_2 & {t_2}^2 \\ \vdots & \vdots & \vdots \\ 1 & t_n & {t_n}^2 \end{pmatrix} \tag{5}$$

The resulting values $b_0,\ b_1,\ b_2$ are the coefficients of the modelling polynomial $P'_2(t)$.
As the saccades are of the form $X_i(t_j) \mapsto \{x_j, y_j\}$ they could not be approximated directly, because the standard regression procedure is only applicable on functions of the form $f(x) \mapsto (y)$. Instead the x and y coordinates were approximated separately as functions $f_x(t_j) = x_j$ and $f_y(t_j) = y_j$ and put together afterwards as $\widetilde{X}'_i(t_j) \mapsto \{P'_{rx}(t_j),\ P'_{ry}(t_j)\}$.
The landing point of a saccade is defined as

$$\widetilde{X}'_i(t_{end}) \mapsto \{P'_{rx}(t_{end}),\ P'_{ry}(t_{end})\}.$$

For its calculation t_{end} is needed, which is not known in practical applications and needs to be predicted as well. While testing this approach, the duration of the saccade which should be approximated is known, but in practical

application it is quite difficult to estimate the remaining duration of the saccade after the first n time steps. In the timeframe of this work it was not possible to develop such a method, and therefor a perfect prediction of the length of the saccades was assumed, giving this approach an informational benefit.
Figure 8 shows the 'least squares fit' of a saccade.

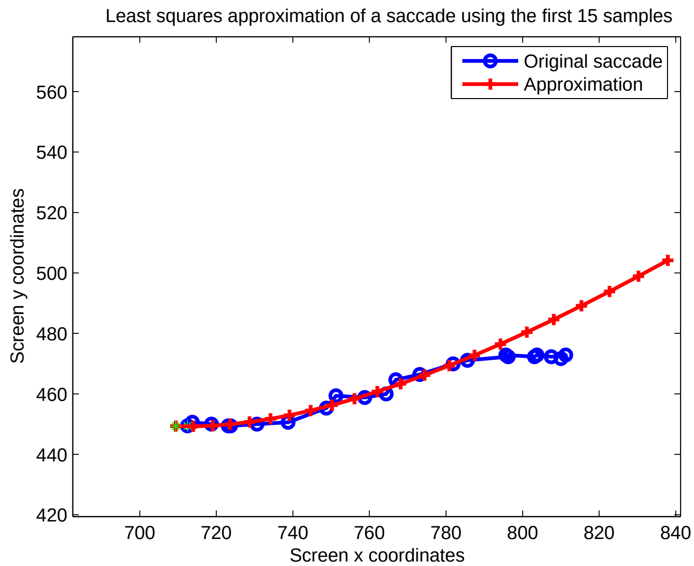

Figure 8: Approximation using least squares fitting.

3.5 Learn from existing saccades

The basic idea of this approach is that saccades that are of the same direction and speed in the beginning might be similar in their landing points as well. To achieve a better comparability, all saccades were shifted to the origin and separated into the four quadrants of the coordinate system, depending on their landing points. The procedure then was to take one saccade S_i^* out of the sample saccades and to find all other available saccades $\widetilde{S}_i$, whose first n points lay within an epsilon hose around the first n points of S^*.

$$\| \sum_{j=1}^{n} (X_i^*(t_j) - \widetilde{X}_i(t_j)) \| \leq \epsilon \tag{6}$$

All saccades that matched this criterion were considered to be similar to S_i^*. As these similar saccades are of different lengths, measured in time steps, the first step was to take the mean of the length of the similar saccades, and then find all the similar saccades that matched the mean length. The saccades found are then taken to approximate S_i^* by computing the mean of them. The resulting averaged saccade is taken as the approximation of S_i^*. The whole process is illustrated in figure 9.

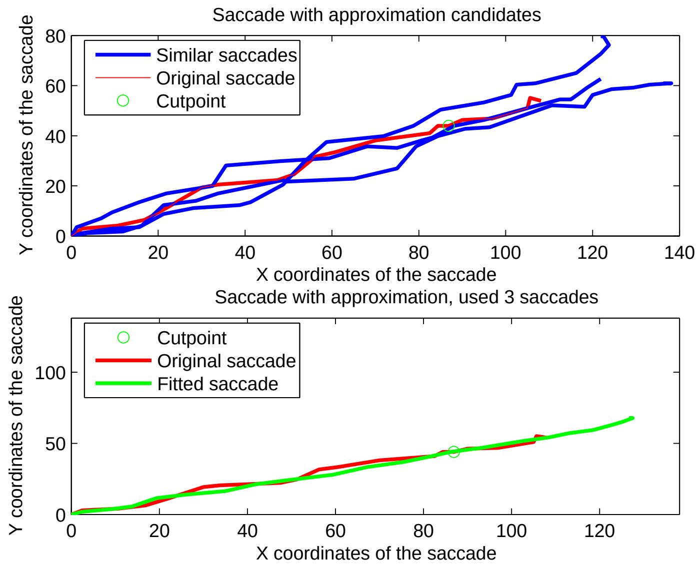

Figure 9: Approximation using existing saccades. The upper part of the figure displays the original saccade together with the similar saccades which matched the mean length. The lower part of the figure displays the original saccade (red) together with the averaged saccade (green), obtained from the 3 similar saccades from the top figure.

3.6 Using the 'machine-learning predictor'

The third approach was to use a modification of a machine learning predictor (ML-predictor) used by [BKBM04] (there called 'non-saccadic predictor') to predict inter saccadic samples. The ML-predictor is a supervised learning approach, designed to predict the gaze point $X_i(t_j) = (x_{t_j}, y_{t_j})^T$ at time step t_j concerning the gaze positions $X_i(t_{j-1}), X_i(t_{j-2}), \ldots, X_i(t_{j-n})$ at time steps $t_{j-1}, t_{j-2}, \ldots, t_{j-n}$.
The goal of this work is to predict the landing point of a saccade using only the first n points. Therefor the ML-predictor is modified to predict the landing point $X_i(t_{end}) = (x_{t_{end}}, y_{t_{end}})^T$ of a saccade instead of the point following the first n known points. The predicted landing point $\widehat{X_i}(t_{end}) = (\widehat{x}_{t_{end}}, \widehat{y}_{t_{end}})^T$ is defined by

$$\widehat{X}_i(t_{end}) = X_i(t_n) + A_{n-1} \times P_{n-1}. \tag{7}$$

$X_i(t_n)$ is the last known position of the saccade that should be approximated. P_{n-1} is a $(n \times 2)$ matrix, the first $n-1$ rows are the distance between $X_i(t_n)$ and the previous points of the form:

$$P_{n-1} = (X_i(t_{n-1}) - X_i(t_n), X_i(t_{n-2}) - X_i(t_n), \ldots, X_i(t_1) - X_i(t_n), Q)^T \tag{8}$$

To detect if the starting position of the saccade has any influence on the trajectory, the last row contains Q, which specifies the starting position of the saccade relative to the center of the screen. Q is of the form $(x_{t_1} - \frac{screenwidth}{2}, y_{t_1} - \frac{screenheight}{2})$. A_{n-1} is the $(1 \times n)$ weight matrix which maps the learned information. A_{n-1} is updated after every prediction, using an incremental learning strategy:

$$A_{new} = A_{old} + \epsilon e P_{n-1}^T \tag{9}$$

$e = X_i(t_{end}) - \widehat{X}_i(t_{end})$ is the prediction error while ϵ represents the learning rate which is estimated to be the optimal learning rate in each iteration, weighted by a parameter α

$$\epsilon = \alpha \frac{eP^T Pe^T}{\|P^T Pe^T\|^2}. \tag{10}$$

The constructed algorithm using a supervised learning strategy works the following way: It picks one saccade out of the 999 samples, tries to predict the landing point, using only the first n points of the saccade and then adjusts A_{n-1} according to the learning strategy 9. This is iterated over the sample saccades as long as $\|A_{new} - A_{old}\| \geq 0.001$. It took about 1556442 iterations to achive this on all 999 saccades of the second data set. Learning on a reduced data set of 100 saccades took about 16000 Iterations.

4 Results

4.1 Using regression for prediction

As mentioned in section 3.4, the regression approach has an informational benefit, as a perfect prediction of the duration of the saccades is assumed. This should be kept in mind during the examination of the results. The results shown in figures 10 and 11 were achieved by using a polynomial of degree 1 for the regression, while the results displayed in figures 12 and 13 were produced using a polynomial of degree 2.

Figure 10 shows the overall mean prediction error as a function of the number of training samples used. Interestingly the error is increasing the more training samples we use. This is an unexpected behaviour, as normally regressions of functions become more exact the more points of the function are known. This abnormal behaviour may result from the fact that the regression is only performed on the first n points, while the remaining trajectory of the saccade is calculated by extrapolating the achieved polynomial.As

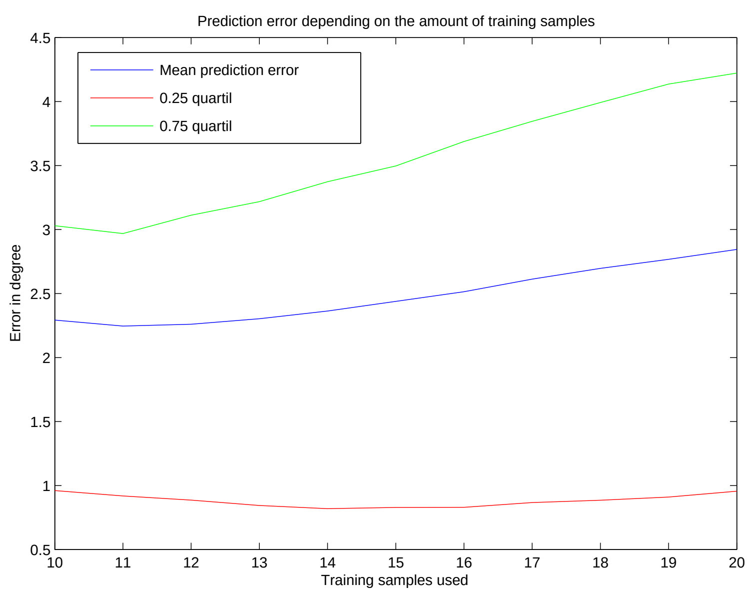

Figure 10: Mean prediction error dependent on how many training samples were used for the regression using a polynomial of degree 1.

mentioned in section 3.3 the goal is to predict a saccade using only the first 15 samples. The results of the regressional approach when using 15 training samples are shown in detail in figure 11. On the one hand the plot shows that the mean prediction error is than that of the baseline predictor E_c most

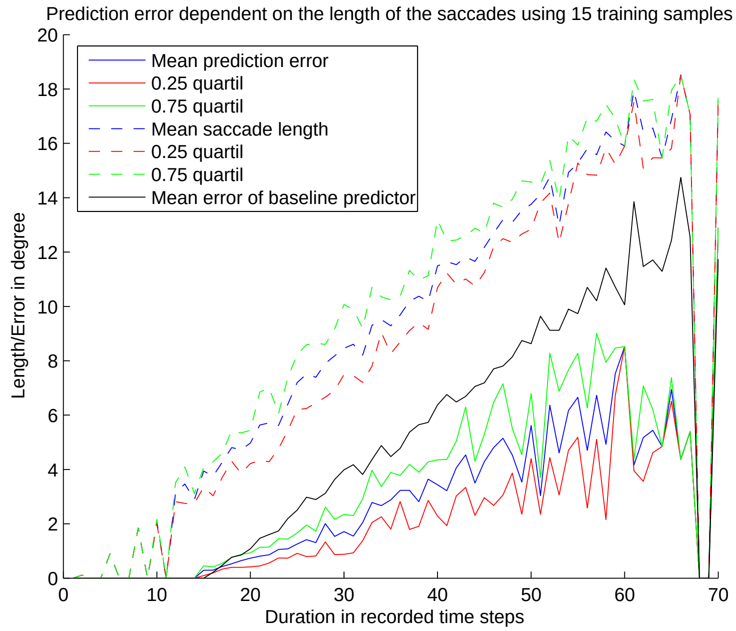

Figure 11: Results of the regression approach using a polynomial of degree 1 and learning on the first 15 recorded gaze positions of the saccades.

of the time. This means that a benefit in comparison to defining the last training sample as the predicted landing point is achieved. On the other hand, one can see that the prediction error is growing linearly as the length of the saccades increases. This leads to poor prediction results when predicting longer saccades, which is a drawback as particularly the long saccades are of interest for the prediction, because the longer the saccade is the more time can be saved by predicting the landing point.

The figures 12 and 13 show the results of the regressional approach when using a polynomial of degree 2. Using a polynomial of degree 2 first seems to suggest itself as most of the time the saccades are a result of the contraction of two different muscles, which pull the eyeball in different directions. Therefor the trajectory of a saccade typically does not follow a straight line, but is slightly curved (see figure 1. The results show something else, though. As can be seen in figure 12, the overall mean error of the prediction is decreasing, the more training samples we use, but it is at all times more than four times bigger than the mean prediction error when using a linear polynomial. This behaviour is displayed more clearly in figure 13. Obviously the prediction error is at all time greater or equal to the baseline predictor, and it is increasing nonlinearly with the length of the saccades.

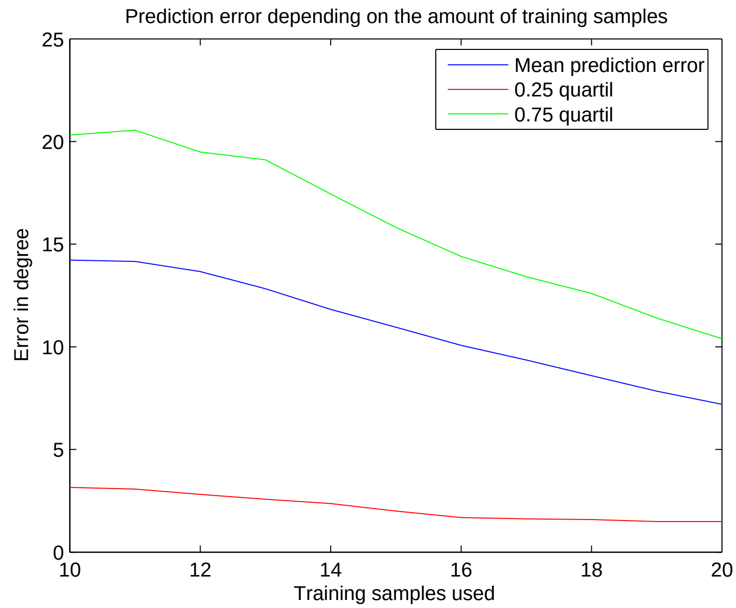

Figure 12: Mean prediction error as a function of how many training samples were used for the regression using a polynomial of degree 2.

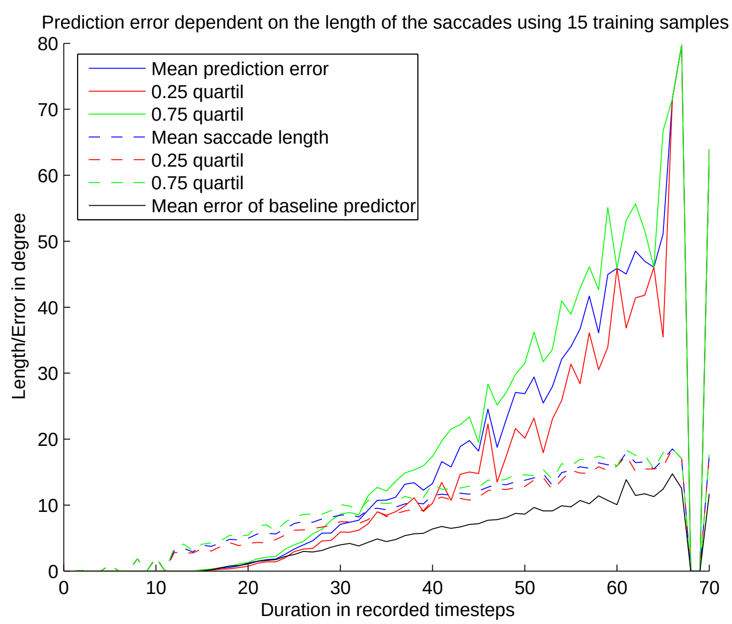

Figure 13: Results of the regression approach using a polynomial of degree 2 and learning on the first 15 recorded gaze positions of the saccades.

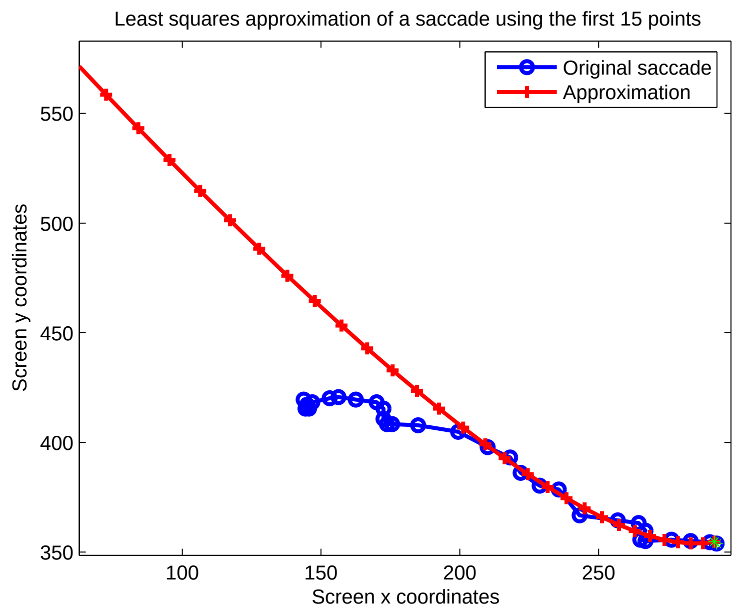

Figure 14: Example of a regression of a saccade producing a very bad result on a long saccade. The fit was performed using a polynomial of degree 2.

4.2 Learn from existing saccades

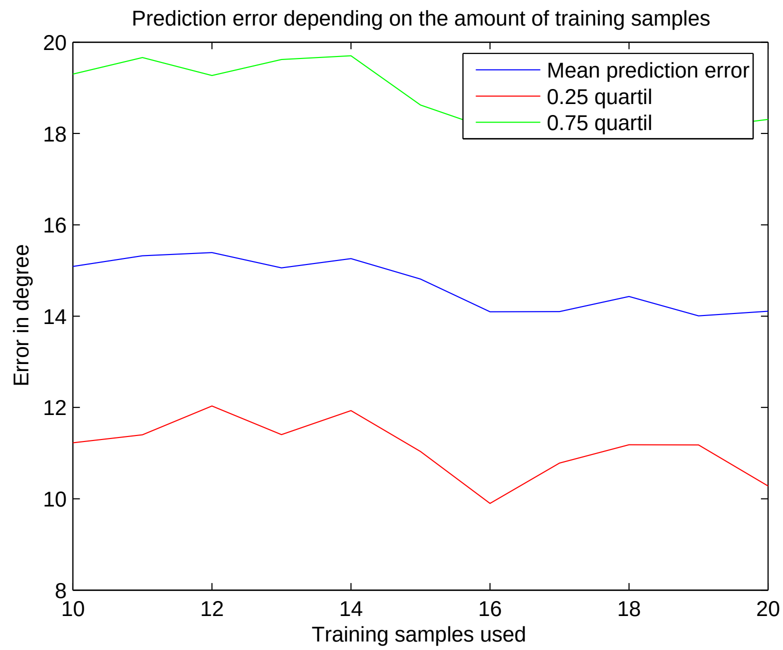

Figure 15: Mean prediction error as a function of how many training samples were used for the data fit.

The results achieved by this approach were overall poor, which is shown in figure 15 displaying the behaviour of the overall mean prediction error according to the amount of training samples. The error lies more or less constantly around 15°, which is in most cases larger than the mean length of the saccades. This can be seen similarly in figure 16 which displays the results in detail when using 15 training samples. The mean prediction error is at all time much bigger than the mean of E_c.
It turned out that this behaviour was caused for the higher part by the relatively small amount of available training saccades. The 999 training saccades that were available were not enough to find a reasonable amount of similar saccades from which a prediction could be averaged. Looking at the prediction procedure in detail revealed, that if any, only one or two saccades were found that matched the similarity criterion for the saccade which should be predicted. The averaged predicted saccade from these few similar saccades had no resemblance with the original saccade in most cases.

4.2.1 Results on 250 Hz data

Applying this approach to the 38595 saccades from the second data set produced similar results, as displayed on figure 17. This time on the one hand, the amount of saccades that matched the similarity criterion was much higher (between 10 to 80) but on the other hand, the temporal resolution of the recorded data was 5 times lower than the resolution of the first data set, which leads to the conclusion that this approach is not working on a temporal resolution of 250 Hz.

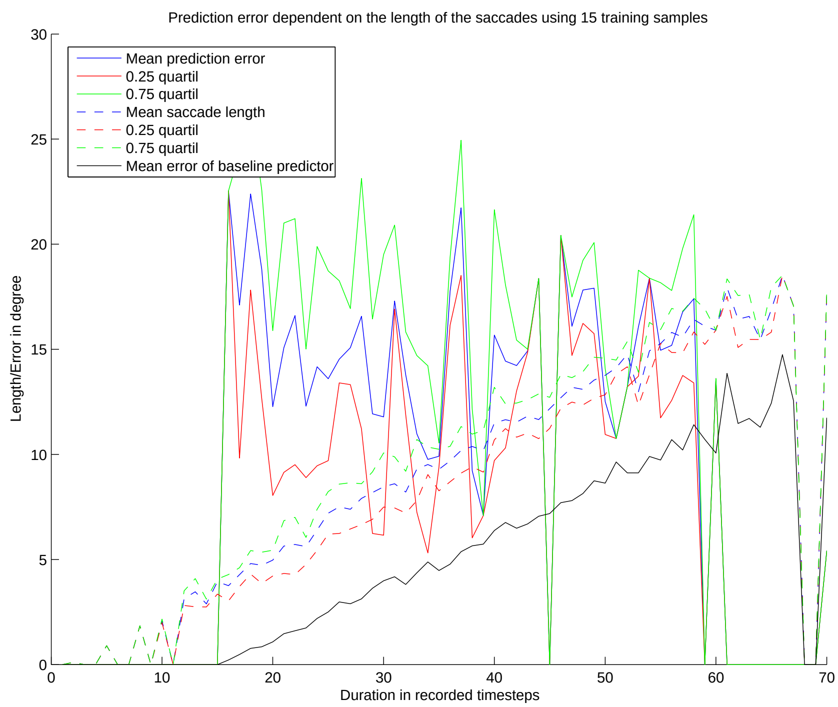

Figure 16: Results of the approach to learn from existing saccades when learning on the first 15 recorded gaze positions of the saccades from the 1250 Hz data set.

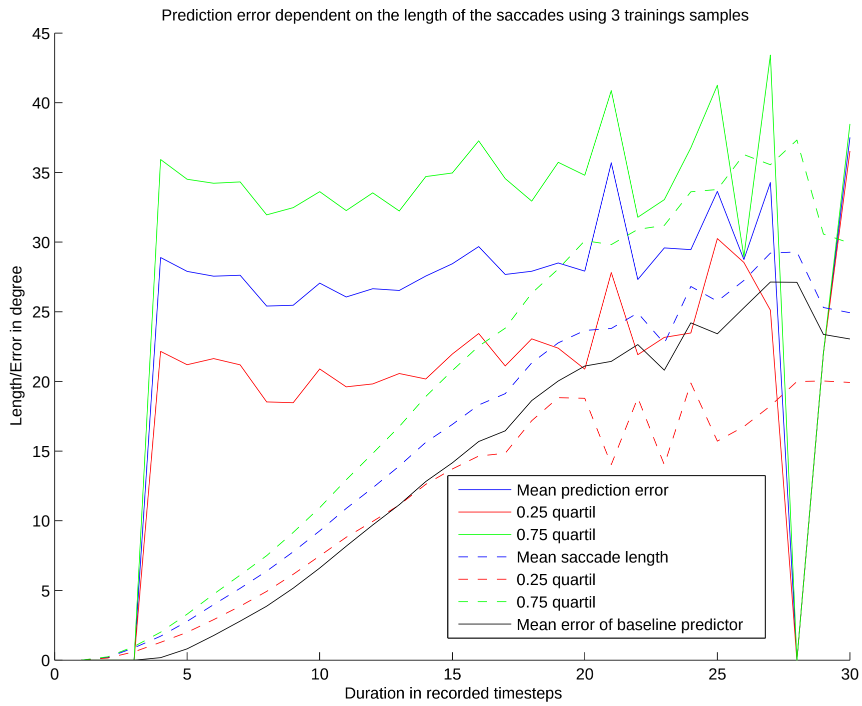

Figure 17: Results of the approach to learn from existing saccades when learning on the first 3 recorded gaze positions of the saccades from the 250 Hz data set.

4.3 Using the 'machine-learning predictor'

The utilisation of the ML-predictor produced the best results of all methods. Figure 18 shows that the overall prediction error is decreasing, the more training samples are used and is residing at about 2° when using 15 training samples.

Figure 19 displays the results in detail when using 15 training samples. The prediction error is smaller than E_c most of the time and lies between 1.5° *and* 4°. Only the prediction of saccades that are longer than 60 time steps produced a higher prediction error. This reults from the small mount of long training saccades, as according to figure 7, only less than 20 saccades are longer than 60 time samples.

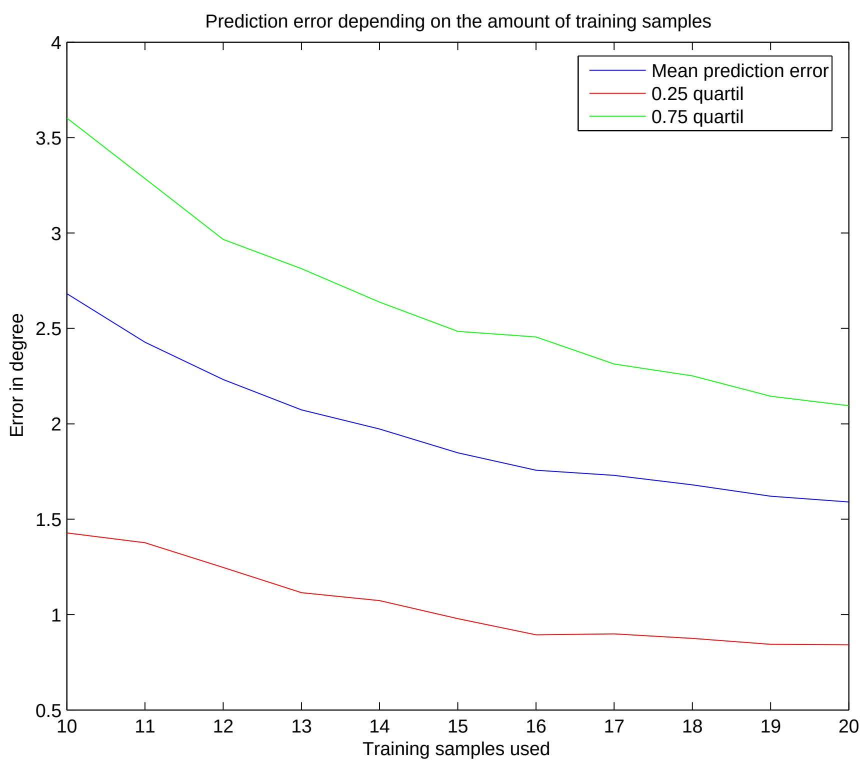

Figure 18: Mean prediction error dependent on how many training samples were used for the learning.

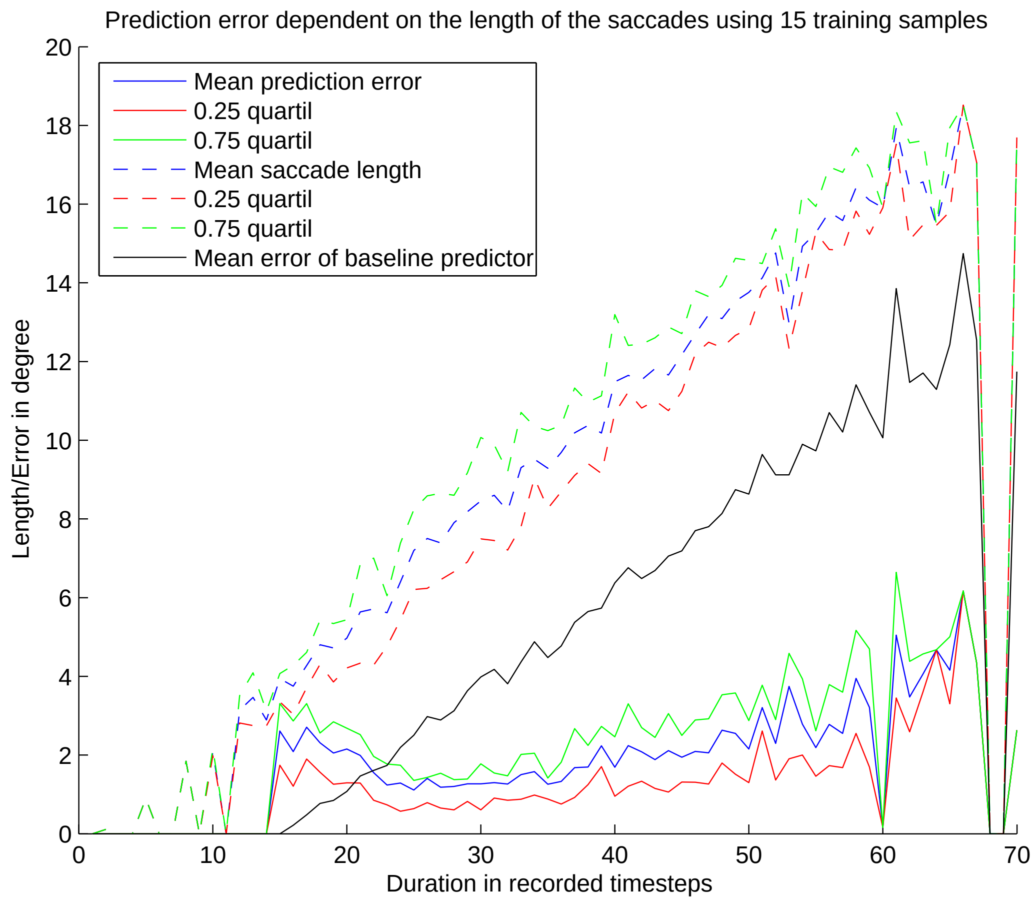

Figure 19: Results of the approach using the ML-predictor when learning on the first 15 recorded gaze positions of the saccades.

4.3.1 Examining the weight matrix

Table 1 displays the resulting weight matrices A for different amounts T_n of training samples. What is of interest here is the fact that the absolute value of the last entry of A which maps the quadrant of the starting point of the saccade is relatively small compared to the other values. This means that the starting point of the saccade only has a little influence on the predictability of the landing point. Apart from this the value preceding the last value is at all time positive and relatively large compared to the other values. This value maps the path of the saccade between the last training sample and the gaze position at the first time step $X_i(t_1) - X_i(t_n)$.
When considering the prediction equation 7 of the ML-predictor it can be seen that negative entries of A are moving the predicted landing point further away from the last training sample $X_i(t_n)$ while positive entries shorten the distance. As the value mentioned before is at all time positive it seems as if it were acting as a brake term for the prediction. The longer the first step of the saccade is, the more needs to be substracted from the predicted landing point, to get a good prediction. Another noteworthly fact is that

T_n	10	11	12	13	14	15	16	17	18	19	20
	-0.5	-0.8	1.1	-0.1	0.8	-0.4	0.4	0.01	0.2	0.2	0.3
	-1.7	-1.7	-0.8	0.2	0.4	0.4	-0.5	0.2	0.01	-0.05	0.2
	-1.2	-0.1	-1.4	0.2	0.04	0.3	0.4	-0.4	0.04	-0.2	-0.1
	-2.4	-0.8	-0.5	-0.8	-0.8	-0.1	-0.1	-0.1	-0.6	-0.2	-0.3
A	-0.3	-1.8	-1.6	-1.2	-0.9	-1.4	-0.5	0.3	-0.3	-0.7	-0.2
	0.6	-0.6	-1.5	-2.1	-1.2	-0.8	-1.0	-0.4	-0.04	-0.4	-0.5
	-1.2	-1.1	-1.1	-1.6	-1.5	-1.0	-1.1	-1.1	-0.3	-0.2	-0.4
	-1.5	-1.2	-0.1	0.3	-0.4	-1.2	-0.5	-1.0	-0.8	-0.3	-0.2
	1.6	-0.4	0.02	-0.2	-0.5	0.05	-0.7	0.01	-0.8	-0.6	-0.3
	-0.3	2.3	-0.3	-0.4	-0.05	-0.4	0.2	-0.8	-0.3	-0.3	-0.4
		-0.2	1.8	-0.8	-0.3	0.1	-0.1	0.1	-0.4	-0.2	-0.3
			-0.2	2.5	0.3	0.2	0.1	0.1	0.3	-0.3	-0.2
				-0.2	1.3	-0.4	0.1	-0.2	0.2	0.2	-0.1
					-0.2	1.3	-0.2	-0.3	0.1	0.3	0.08
						-0.1	0.7	-0.1	-0.1	0.2	0.2
							-0.1	1.1	0.1	-0.02	0.1
								-0.1	0.4	0.1	0.02
									-0.1	0.1	0.1
										-0.07	0.2
											-0.1

Table 1: Averaged learned weight matrices A over 20 iterations of the learning procedure on different amounts of training samples

independent of the amount of training samples, the entries of A are predominantly positive at the beginning and the end of the vector (do not consider the last entry, as it maps the quadrant of the starting position and no component of the trajectory), while the middle of A is dominated by negative entries. This means that the middle of the first n time steps is pushing the predicted point away, while the beginning and the end of the first part of the saccade are working in the opposite direction.

4.3.2 Learn on a reduced data set

One question that arises when using machine learning approaches, is the size of the training data set needed to achieve good results. To examine how the ML-predictor would behave on a reduced data set, smaller training data sets were created by randomly picking a certain amount of saccades out of the initial data set. The learning procedure then was executed 20 times on reduced data sets, each with sizes between 1 and 999 saccades. Afterwards the prediction of all 999 saccades was carried out using the so learned weight matrix A.
Displayed in figure 20 are the results of the learning process on reduced data sets. The figure shows the mean of the mean prediction error over the 20

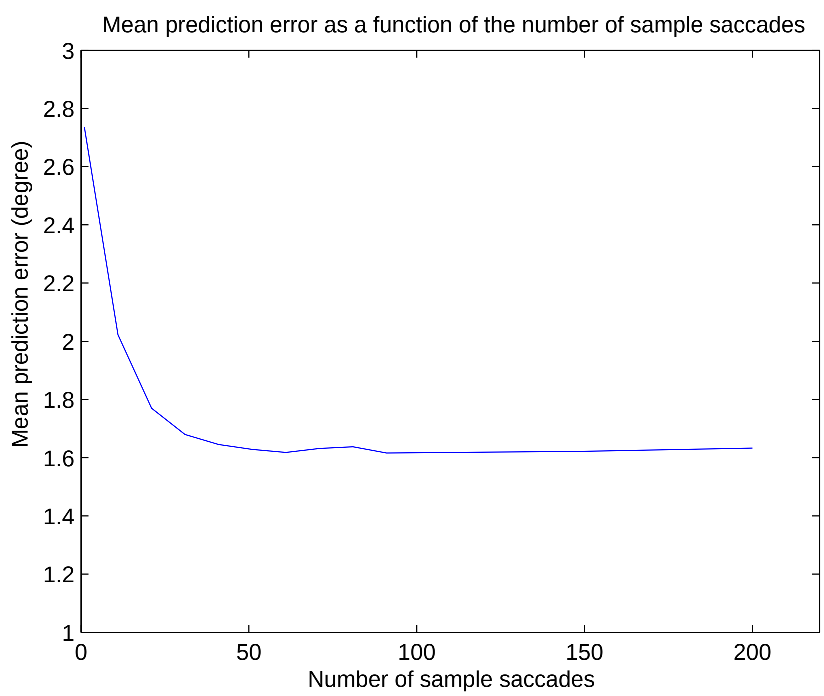

Figure 20: Results of learning on a reduced data set

learning iterations on data sets of different size. It is interesting that learning on only one saccade already allows a prediction with an average accuracy of about $2.7°$. This leads to the conclusion, that the saccades resemble each other more than expected. Another fact that is worth pointing out is that around 100 saccades are sufficient for the learning process. Learning on bigger data sets produces no improvement of the mean prediction error. This leads to a very short learning phase.

5 Discussion

5.1 Optimal method

As already mentioned in section 4.3, the approach using the ML-predictor produced the best results. As saccades are ballistic movements that cannot be influenced in speed and direction after they have been started, they are not perfectly accurate themselves, but normally miss their desired landing point by about one or two degrees. This is normally corrected by small correction saccades that follow directly after the main saccade. As the ML-predictor enables a prediction of the landing points with a mean accuracy of about $1.6°$ (figure 20) it lies close to the aiming accuracy of a saccade itself, making it a plausible prediction method in terms of the error range.
A second important point is the short learning phase, as described in section 4.3.2, which makes it possible to perform the learning procedure online, while the user has already started using the eye tracker. Since the human eye performs 2-3 saccades each second, the collection of 100 suitable training saccades should not exceed 1 minute.
After having learned the weight matrix A, the prediction of one particular saccade is very fast, as it basically consists of a simple matrix vector multiplication, see code listing 1.

5.2 Shortfalls

5.2.1 Regression

The regressional approach has the benefit that it is working without any preparation. Only the saccade that is to be approximated is necessary for the prediction method, which makes it fast and simple, but this fact is not balancing the drawbacks of the prediction results.
The examination of the approach using polynomials of degree 2, 3 ... showed that the growth of the prediction error is increasing in the same order, the degree of the polynomial increases. This leads to the conclusion that the best results are achieved when using straight lines for the approximation of the saccades. The mean prediction error when doing so was about $2.5°$ but the detailed examination showed that the error grows linearly with the length of the saccade, rising up to around $6°$ for saccades exceeding 40 time steps. As particularly long saccades are of interest for the prediction, this is a big drawback.
Another drawback is that additionaly, a prediction of the number of time steps the saccade will last is needed to be able to calculate the approximation of a saccade. While a perfect prediction of the number of time steps the saccades would last was assumed, the results of the prediction were not satisfying. As a second prediction method for the time steps would probably

insert an additional error into the prediction progress, which would probably degrade the result further, the regressional approach is considered to be inadequate.

5.2.2 Learn from existing saccades

The results of this approach were always poor. In case of the second data set this might result from the low temporal resolution of the eye tracking system used for the recording. Because of the lack of data in the first data set, it could not be clarified definitely if this approach would work better if more data recorded at 1250 Hz would be available.
Another significant problem of this approach is that its performance depends on the amount of saccades that need to be compared to find the similar saccades for the prediction. This might be accelerated by separating the available sample saccades into different sectors according to their position in the last trainings sample. In this case only the sector matching the last trainings sample of the saccade that is to be predicted needs to be scanned for similar saccades. This, however, makes some preparation of the sample data set necessary. Nevertheless the performance of predicting a single saccade is inferior to the other two methods.

5.2.3 Learn using the 'machine-learning predictor'

When using the 'ML-predictor' one has to perform the learning procedure, to learn the weight matrix A before being able to predict a saccade. The learning procedure took about 175 seconds on a 1.6 GHz Intel coreDuo CPU machine using Matlab, when learning on all 999 saccades. The time consumed by the learning procedure however is of no big importance, as it needs to be carried out only once and section 4.3.2 showed that learning on a data set of about 100 saccades is sufficient. This allows a much faster learning phase of about 16 seconds on the machine mentioned above. In this context the question arises whether the weight matrix A which was learned on a dataset recorded from only one subject can be applied on the saccades of a different person, or if a calibration on the characteristics of each user is necessary. Such calibration procedures though would not lead to a significant increase of preparation, due to the short learning phase of only about 100 saccades, what makes online adaptation possible. In addition to this, starting the learning procedure with a weight matrix obtained from a previous user, would shorten the learning phase further, as the matrix is initialized with values already in the right range, so that only a fine tuning of the weights is neccessary.

6 Suggestions for future work

The discussion showed that the ML-predictor might be able to improve gaze-contingent displays by achieving a visualisation without latency phenomena becoming aware to the observer. This has to be further investigated in practical applications. Questions that need to be answered in this context are for example, whether the preprocessing of the saccades, which was done partly manually as described in section 3.2, can be automated successfully when operating with recorded saccades online.

Another point of interest would be if the prediction results are of the same quality when working on saccades initiated by natural films or images instead of an artificial procedure as described in section 3.1.

Further research might be conducted on the differences of the weight matrices, resulting from learning on the data recorded from different subjects. This might provide insight into the question whether the saccades are characteristical for every single person, or if they resemble each other independent of the particular subject.

Future research might address the question whether the application of a non-linear predictor instead of the linear predictor used by the ML-predictor, can lead to a further improvement. Such a predictor might be of the form that saccades which have a predicted amplitude larger than a threshold X, would be fed into a second-level predictor. This second level predictor could be specially trained on long saccades, by performing the learning procedure only on training saccades having a larger predicted amplitude than X. In this way a further improvement of the prediction of long saccades might be achieved.

References

[BKBM04] Martin Böhme, Christopher Krause, Erhardt Barth, and Thomas Martinetz. Eye movement predictions enhanced by saccade detection. *n.a.*, 2004.

[DBM04] Michael Dorr, Erhardt Barth, and Thomas Martinetz. Effects of gaze-contingent stimuli on eye movements. Master's thesis, Institut fuer Neuro- und Bioinformatik, Universitaet zu Luebeck, 2004.

[DV00] Andrew Duchowski and Roel Vertegaal. Eye-based interaction in graphical systems: Theory and practice. *SIGGRAPH 2000 Course Notes*, 2000.

[Pal99] Stephen E. Palmer. *Vision Science*. The MIT Press Cambridge Massachusetts, 1999.

7 Appendix

Listing 1: Algorithm executing the ML-prediction

```
function [Anew E] = SaccadePredictor(A,Saccade,alpha,
                                     stepsToUse,verbose)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The SaccadePredictor trys to predict the landing
% point of a Saccade by using
% a learning algorithm with N previous positions
% Input:
% A          (Vector) (1xN-1) The weight matrix
%                     is the last known position
% Saccades   (struct) The Saccade which is to be
%                     learned
% alpha      (scalar) Learning rate weight
% stepsToUse (scalar) The amount of points used for the
%                     prediction
%
% Output:
% Anew     (Vector) (1xN-1) The updated weight matrix
% E        (scalar) The error between predicted and real
%                   ending point
%
% Intern:
% Pos      (matrix) (Nx2) the N prevoiusly known
%                   positions
% Xend     (Vector) The end position of the saccade to
%                   fit (1x2)
% E        (scalar) Prediction error
% Xhat     (vector) (1x2) Predicted landing point
% P        (Matrix) (Nx2) The distance matrix between
%                   the previous points
%                   and the last known point
% epsilon ()        The learning rate
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%The screen geometry data
screenwidth = 800;
screenheight = 600;

%Calculate the distance matrix
Pos = [Saccade.X(1:stepsToUse-1)' ...
       Saccade.Y(1:stepsToUse-1)'; 0 0];
Xend = [Saccade.XEnd Saccade.YEnd];
```

```
N=size(Pos);
N=N(1);
P = zeros(N,2);

%The last known positon
X = [Saccade.X(stepsToUse) Saccade.Y(stepsToUse)];
for j=1:N-1
    P(j,:) = Pos(N-j,:)-X;%-Pos(N-j+1,:);
end

%Identify the starting position relative to the origin
P(N,:) = [Saccade.X(1) - screenwidth/2 ...
        Saccade.Y(1) - screenheight/2];

%Predict the landing point
Xhat = X + A * P;

%Update weight matrix
E = Xend - Xhat;
epsilon = alpha * (E * P' * P * E')/...
        (norm(P' * P * E') ^ 2);

Anew = A + epsilon * E * P';

if verbose==1
    X
    Xend
    Anew
    P
    Xhat
    error = norm(E)
    epsilon
    Achange = epsilon * E * P'
end
E=norm(E);
return
```

List of Figures